EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 151 No. 10
October 2022

Research in Computing Science

Research in Computing Science, Año 21, Volumen 151, No. 10, octubre de 2022, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de octubre de 2022.

Research in Computing Science, year 21, Volume 151, No. 10, October 2022, is published monthly by the Center for Computing Research of IPN.

Volume 151(10)

Advances in Artificial Intelligence

Obdulia Pichardo-Lagunas
Juan Martínez-Miranda
Bella Martínez-Seis
Hiram Calvo-Castro(eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2022

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Shape Retrieval through Polygon Matching

Bryan E. Martínez, Antonio Camarena-Ibarrola, Karina Figueroa

Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Eléctrica,
Mexico

bryaneduardo24@gmail.com,
{antonio.camarena, karina.figueroa}@umich.mx

Abstract. Content-Based Image Retrieval consists in searching for images in large datasets by their shape, it allows for automatic annotation in images, image classification, automatic surveillance, and many other applications. In this work, we model objects as sets of triangles built from keypoints on their border. We use an invariant of these triangles to make them robust to affine transformations such as scaling, rotating, or shearing; this invariant is adapted as a hash for indexing purposes. The experiments show that this method is very effective, achieving 99% accuracy, outperforming state-of-the-art works with the same collection of images.

Keywords: Image retrieval, contour shape, transform invariant.

1 Introduction

Image repositories grow very fast and there is great interest in designing efficient and effective Content-Based Image Retrieval (CBIR) algorithms [19]. CBIR by content is normally accomplished by extracting features from the images such as shapes, textures, and colors.

This work focuses on shape-objects contained in images. Shape is one of the primary low level image features used in CBIR and one of the Contour-Based Descriptors (CBD). Although CBD has been widely used and many researchers have had good results, there are some problems regarding robustness since it is affected by noise and variability.

This is because only part of the shape information, that is contour information, is normally used. Also, in many cases, the shape contour is not available. For some applications, shape content is more important than contour features [22]. For these reasons and also because many images contain not just one but several objects, a perfect retrieval rate using only the shape information has not been possible so far [4].

There are generally two types of shape representation methods in the literature, that is Region-Based and Contour-Based [21]. Both approaches can be subdivided into Structural or Global methods, as shown in Fig. 1. Structural methods are typically used where partial matching is required, while Global methods are used when complete matching is needed. Different methods can be further distinguished between them, some work in space domain and others in some transformed domain [19, 22].

We are interested in both Structural and Global methods, and we use Fourier descriptors to create polygons. Structural methods have several drawbacks [21]: Since

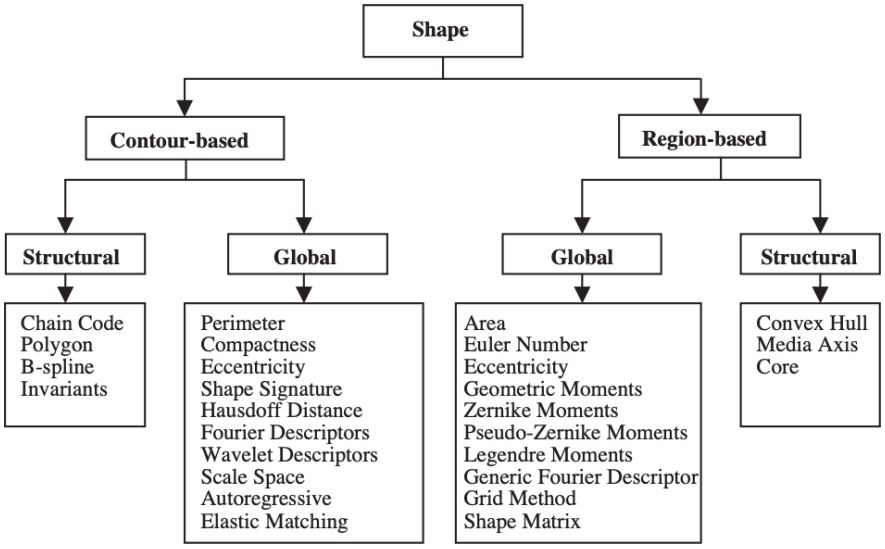


Fig. 1. Classification of shape representation and description techniques [22].

there is no formal definition for an object or shape, the number of primitives required for each shape is unknown. Variations of the object boundary may cause significant variations to local structures. In these cases, global features are more reliable. The shape and its representation is a many-to-one mapping.

Therefore, matching of one or more features does not guarantee full shape matching; Structural methods have higher computational complexity than conventional techniques. The distance or similarity measure used for Shape matching or shape retrieval should be invariant to many distortions including scaling, offset, noise contamination, partial occlusion, shearing, rotation, etc. Techniques for handling these distortions frequently rely either in the representation of the data or in the similarity or distance measure used [6, 12, 13, 16, 17].

Another issue that complicates the problem is the partial occlusion of objects due to the presence or other objects in the image [19]. Rotation invariance is also a challenge, some intended for achieving rotation invariance rely on the representation of the data at the expense of discrimination ability, while others rely on the distance measure used at the expense of efficiency [12].

In order to carry out Shape matching of image objects effectively, contour normalization techniques are widely used, as they save time and space during both feature extraction and similarity matching. The normalized sampled contour points are keypoints that compactly represent shapes. The conventional Shape matching and retrieval applications perform analysis methods of Equal Distance Normalization (EDN) on the shape contours. In addition to EDN, the boundary can also be normalized assuming the enclosed area of the shape captures all the information about the shape [17].

In assessing a method for shape retrieval six factors should be considered: (1) Retrieval accuracy, (2) Compactness of the features, (3) Generality for applications,

(4) Low computation complexity, (5) Robustness, and (6) Hierarchical coarse-to-fine representation [21, 22]. For large databases, sequential searching should be avoided, so shape descriptors should be indexable. Hash tables and M-trees are very used for these activities [6]. Shape recognition and retrieval are complex tasks on non-rigid objects, However, it can effectively be performed using a set of compact descriptors [17].

Shape representation is one of the most challenging aspects of computer vision because they are often more complex than color or texture. Moreover, color and texture can be quantified by a few parameters, unlike shapes that need hundreds of parameters to be represented explicitly [16]. Shape matching and indexing is an essential topic in its own right, and it is a fundamental subroutine in most shape data mining algorithms [12]. In conclusion, Shape is accepted as a stable visual feature for image recognition and retrieval due to its discriminate strength [19, 16, 22].

The rest of this work is divided as follows: First the state-of-the-art is reviewed in Section 2. In Section 3, we give a little description about polygons matching and some basic concepts to generate triangles by using curvature shape. In section 4 we explain our method as applied to the benchmark datasets described in Section 5. Finally, in Section 5 and 6 we show results and conclusions.

2 State of the Art

In early works features extracted from curvature shapes were sequences of values such as curvatures, angles, descriptor values, or polygon coefficients. The Curvature Zero Crossing Contours of the Curvature Scale Space (CSS) was used to represent the shapes of boundary contours of objects by five pairs of integer values. The significant advantage of this method is that it is indexable, and the aspect ratio of the CSS image is used to reduce the search range. Since the matching algorithm has been designed to use global information, it is sensitive to major occlusions, but minor occlusions do not cause problems [1, 16].

Another descriptor, used by many researchers, is the Zernike Moment Descriptor (ZMD), which has many desirable properties, such as rotation invariance, robustness to noise, expression efficiency, fast computation, and multi-level representation for describing the shapes. Kim & Kim showed that ZMD can be used as an adequate global shape descriptor for images in a large database [13]. The experimental results performed in a database of about 6,000 images in terms of exact matching under various transformations and the similarity-based retrieval showed that the proposed shape descriptor is very effective in representing shapes. Zhang and Lu evaluated ZMD and the CSS Descriptor [21].

The idea of getting patterns from curvature shapes led some researchers to use polygons, particularly triangles [4], these polygons represent shape curvature using the spatial positions distributed along the contour, they are quite popular among researchers even thought it focuses in the contour and neglects any information inside the shape [19].

Kumar and Mali remark the importance of selecting good keypoints at contour sampling for shape classification via contour matching [14]. In Boundary-Based Shape matching, Dynamic Time Wrapping (DTW) and Dynamic Space Warping (DSW) have

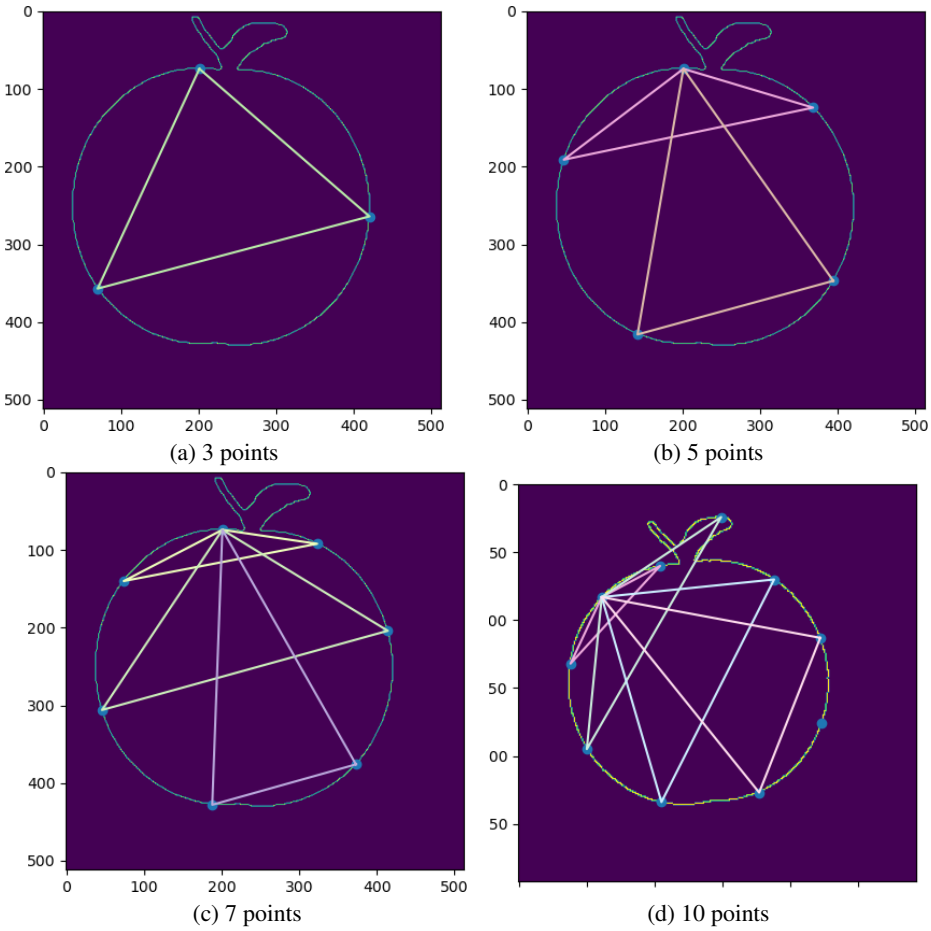


Fig. 2. Representing an apple shape contour with triangles using keypoints.

proved to be useful [3, 19]. Bartolini *et al.* proposed a Fourier-based approach for shape retrieval called WARP [6], they claimed that phase information provides a better accurate description of object boundaries than using only the amplitude of the Fourier coefficients, they use DTW to match images even in the presence of (limited) phase shifts, they also use proximity indices to speed-up the retrieval phase.

Alajlan *et al.* proposed a Shape Retrieval method using Triangle-Area Representation (TAR) for non-rigid shapes with closed contours [4]. The representation uses areas of triangles formed by boundary points to measure convexity at different scales (or lengths of triangle's edges). This representation is effective in capturing both local and global characteristics of a shape, it is invariant to translation, rotation, and scaling.

It is also robust against noise and some partial occlusion. In the matching stage, a Dynamic Space Warping (DSW) algorithm is used to search for the correspondence between the points of two shapes. A distance is computed based on the best alignment between two shape representations. The computational complexity for matching is

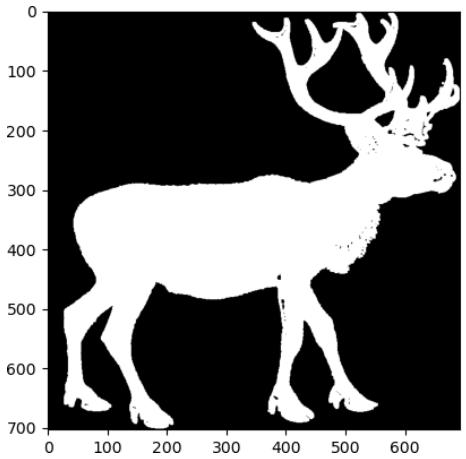


Fig. 3. Deer from the MPEG-7 Core Experiment CE-Shape-1 Dataset.

O(N^2), where N is the number of boundary points. A difficulty associated with DSW is the fact that the starting point of a contour shape is unknown, the same happens when working with rotation angles (i.e. Which angle is the first one?).

Rather than performing an exhaustive search for the correct starting point as in classical approaches, Alajlan proposed Algorithm HopDSW which finds the starting point efficiently [3]. HopDSW operates in a coarse-to-fine manner. The coarse search is global and uses a hopping step to exclude points from the search. Then, the search is refined in the neighborhood of the solution of the coarse search. A criterion for selecting the hopping step parameter is given thus reducing the number of starting point computations. For shape representation, Triangle Area Signature (TAS) is computed from triangles formed with the boundary points.

Paramarthalingam and Thankanadar proposed a procedure for generating normalized contour points from shape silhouettes, this procedure identifies the contour of any object in an image and uses the Object Area Normalization (OAN) method to split the object by its center into regions of equal area. They defined Six descriptors: The Compact Centroid Distance (CCD); the Central Cngle (ANG); the Normalized Points Distance (NPD); the Centroid Distance Ratio (CDR); the Angular Pattern Descriptor (APD); and the Multi-Triangle Area Representation (MTAR). These descriptors conform a feature vector to model the shape of the object [17].

Keogh *et al.* consider rotation to be the hardest distortion in shape matching and indexing. Regular approaches rely on data representation to achieve rotation invariance, they show how to speed up such approaches without loosing accuracy, they make use of existing shape representations and distance measures [12]. Yildirim *et al.* proposed a statistical approach [19], they compute the standard deviation of the angles between the shape centroid and all points around the contour. They quantize angle to integer values, then for each angle they extract three features: The number of contour repetitions; the average distance of the points at that angle to the centroid; and the standard deviation of those distances.

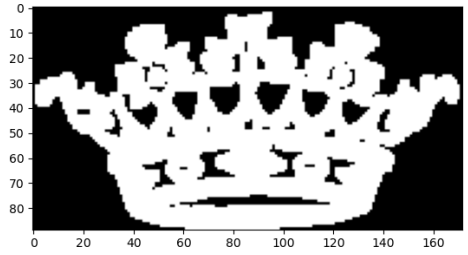


Fig. 4. Crown from the MPEG-7 Core Experiment CE-Shape-1 Dataset.

Kumar and Mali used the center of gravity of the shape of an object as a fixed point, then computed the perpendicular distance from each point on the object contour to the line passing through the fixed point as a geometrical invariant. In the matching stage, they used principal component analysis concerning the moments of the perpendicular distance function, their method is robust to translations and rotations [14].

Xu *et al.* used Partial Shape Matching (PSM) and Dynamic Programing (DP) for retrieval of vertebral boundary shape in X-ray images, their method called corner-guided DP, uses nine landmark boundary points as a multiple open triangle representation. Their method use linear transformations (translation, rotation, and scaling) on a shape to find the best match between two shapes [18].

Arjun and Mirnalinee proposed an iterative algorithm called multi-scale feature integration that use points on the shape curvature, these points are ordered according to their normalized distance to the contour. For feature extraction they use the angular pattern (AP), Binary AP (BAP), and Sequential Backward Selection (SBS) algorithms [5]. Abro *et al.* evaluated some features such as Fourier descriptors, Hierarchical Centroids, Moment-based descriptor and Shape Context Descriptors and showed that fusing several descriptors a better accuracy is achieved. They assessed fusion based on concatenation of features and fusion based on a discriminant correlation analysis achieving an accuracy of 90% [2].

Paramarthalingam and Thankanadar proposed the Object Area Normalization (OAN) method for generating normalized contour points from shapes, they split each object with respect to its centroid into segments of the same size using triangles. From these triangles, they define six contour-based geometric shape features and use them to recognize shapes [17].

Zhang *et al.* proposed an algorithm called shape classification network (SCN) based on convolutional neural networks based on LeNet5 since this basic structure have been used to recognize handwritten numbers achieving good results on MNIST dataset and they claimed it is a similar problem to object shape recognition [20]. Damen *et al.* use *edgelet* constellations for detecting objects in stream video [9]. Edgelets are edge segments and constellations of edgelets are used to characterize shapes even when they are partially occluded.

There are many works on the subject, we described here those with shape retrieval accuracy between 90% and 100%. We found that those works where 100% accuracy was reported, included in their tests just Rotation or Scaling, but not both. In very few works, the authors included shearing in their tests and those who did report an accuracy

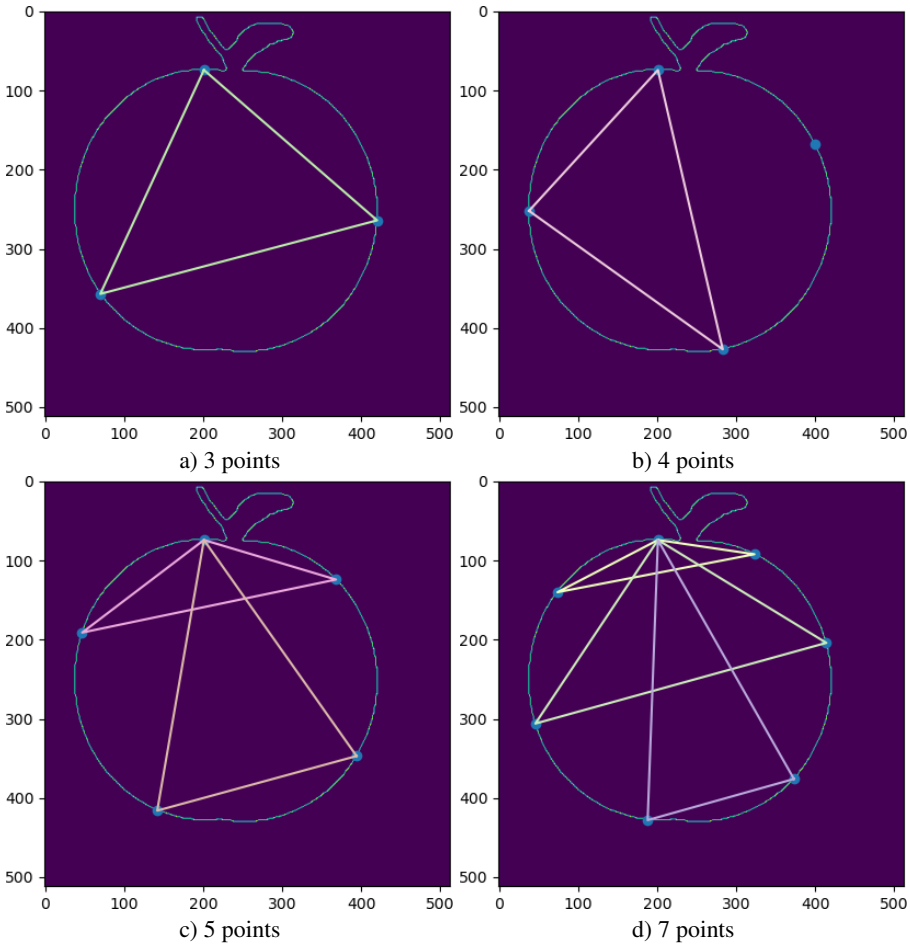


Fig. 5. Representing an apple shape contour with triangles using keypoints.

of about 80%. Those who are unfamiliar with the problem and need basic information should start with [11]. Those familiar with the problem but still need a survey may read [22], for more recent advances in the state-of-the-art may read [19].

3 Theoretical Framework

A generalized polygon is an ordered set of vertices, this notion generalizes the concept of the boundary of a polygonal shape because self-intersections are allowed [10]. Using Polygons to represent Contour shapes, the problem of contour matching is turned into a problem of accomplished polygon matching. For example, in Fig. 5, the contour shape of an apple is used to generate triangles from keypoints, a Start point (or main point) was chosen and you recognize it in Fig. 5 because it is common to all triangles, the number of triangles used to represent the shape is a free parameter and determines the number

Table 1. Results on the Normal Experiment using unmodified images.

Keypoints	Hits/Total	Accuracy
3	1394/1400	0.9957
4	1394/1400	0.9957
5	1394/1400	0.9957
6	1394/1400	0.9957
7	1394/1400	0.9957
8	1394/1400	0.9957
9	1394/1400	0.9957
10	1394/1400	0.9957

of points needed for that matter. In Fig 5 the representation of an apple is shown with 3, 5, 7, and 10 points. Observe that with less than 10 points the stem was not reached.

For the problem of matching polygons, Chavez *et al.* proposed seeing the sequence of vertices that define a polygon as a sequence of complex numbers and so as a small complex signal, then they defined a Fourier descriptor that is invariant under affine transformations of the polygon, including rotation, translation, scaling, and shearing [8]. For that purpose Chavez *et al.* built a collection of complex scalar functions on the space of plane polygons, if two polygons are affine related, the pseudo-hyperbolic distance between their associated values is a constant that depends only on the affine transformation involved, but independent of the polygons.

Point (x, y) in the plane is associated with the corresponding complex number $z = x + jy$, where $j = \sqrt{-1}$. A polygon in the plane, which is an ordered set of points, is then an ordered set of complex numbers, in which the order defines which are the consecutive vertices. Given polygons $Z = (z,1, z_2, z_3, \ldots, z_n) \in \mathbb{C}^n$ and $W = (w_1, w_2, w_3, \ldots, w_n) \in \mathbb{C}^n$, where n is the number of vertices, matching W and Z is the problem of telling if there is an affine transformation f such that $Z = f(W)$ [7, 8, 10].

The approach to polygon matching under affine transformations involves the construction of complex scalar functions $\varphi_\ell : \mathbb{C}^n \to \mathbb{C}, \ell = 1, \cdots, \lfloor (n-1)/2 \rfloor$. Then, finding all the matching polygons in a collection can be achieved very quickly after mapping all the polygons in the collection to complex numbers. It is important to highlight that all similar polygons under affine transformations will be mapped to the same complex number φ_ℓ. This method assumes $n \geq 3$ (we need $n = 3$, because we are using triangles). Function $\varphi_\ell : \mathbb{C}^n \to \mathbb{C} \bigcup \{\infty\}$ is:

$$\varphi_\ell(z_1, z_2, z_3, \cdots, z_n) = \frac{\sum_{k=1}^{n} \lambda^{\ell k} z_k}{\sum_{k=1}^{n} \lambda^{-\ell k} z_k}, \tag{1}$$

where $\lambda = e^{\frac{2\pi j}{n}}$ and ℓ is any integer in $1, \ldots, \lfloor (n-1)/2 \rfloor$.

Table 2. Results on the Extended experiment with modified images.

Keypoints	Scaling (hits/total)	Rotation (hits/total)	Shearing (hits/total)	Average Accuracy
3	1388/1400	**1394/1400**	1394/1400	**0.9942**
4	1382/1400	**1394/1400**	1394/1400	0.9928
5	1397/1400	1379/1400	1394/1400	0.9928
6	1397/1400	1379/1400	1394/1400	0.9928
7	**1399/1400**	1379/1400	1394/1400	0.9933
8	**1399/1400**	1379/1400	1394/1400	0.9933
9	**1399/1400**	1379/1400	1394/1400	0.9933
10	1397/1400	1379/1400	1394/1400	0.9928

4 Description of the Proposal

In this section we will describe our method of contour shape retrieval. We built a proximity index, to do that we process each image in the following way:

1. First, the contour shape in the image has to be determined. We use the canny edge detector and eliminate holes to deal with images such as the crown shown in Fig. 4.
2. We determine the centroid of the contour shape and the nearest point to this centroid that is in the contour. These two keypoints, labeled as 1, and 2 define a line of reference. We translate the image so the centroid corresponds with the origin (0, 0).
3. We split the curvature shape by tracing lines at 120, 90, 72, 60, 51.42, 45, 40 and 36 degrees counter-clockwise with respect to the line of reference between keypoints 1, and 2. The points where these lines intersect with the contour are the keypoints labeled as 3, 4, 5, 6, 7, 8, 9, and 10. When the lines intersect 2 or more points of the contour (think for example of the contour of the Deer shown in Fig. 3), these points can be ordered from its distance to the centroid from the innermost to the outermost. We select as keypoint only the outermost thus favoring bigger triangles.
4. We create triangles starting from keypoint 2, and the keypoints at its left and right. Then add another triangle using always keypoint 2 and the keypoints at its left and right that have not been used until there are no more unused keypoints or until there is just a single keypoint available (we need keypoint 2 and two more to build a triangle).
5. For each triangle built in the previous step we use Equation 1 and compute the magnitude of the complex number that results from that transformation obtaining a single number per triangle.
6. Using the number determined in the previous step, add the triangle as well as its unique identifier to a hash table of size 256. The shape has an entry to the hash table for each triangle built from it.

Table 3. Accuracy obtained of our proposal contrasted with those obtained in similar works that use the same collection of images.

Author	Average Accuracy
SCN	0.7539
BAPmP	0.8797
(SCF + SCF) (DCA)	0.9196
SA-OAN	0.9434
DSW + Global	0.9508
Zernike moment descriptor	0.9588
Our proposal	**0.9942**

In Fig. 5 the importance of a good selection of keypoints is depicted. Using our method for selecting keypoints, the shape may be rotated, and still we select almost the same keypoints, very near as you may observe, this is important if we want the set of triangles to represent the shape.

5 Experiments and Results

For our experiments, we used the MPEG-7 Core Experiment CE-Shape-1 Test Set, which is the most commonly used dataset for the contour shape matching problem, this database consists of 1400 images from 70 classes of natural and artificial objects [15]. Figures 4 and 3 are examples from this collection of images. The set has two parts called A1 and A2, for scaling and rotation respectively. We conducted two experiments on retrieving images by content based on the contour shapes, in both experiments we use all triangles obtained from the query image to search for a match using the hash table.

For the first experiment, which we called the normal experiment, we used the original dataset without modifications. We varied the number of keypoints from 3 to 10, and in all cases we achieved an accuracy of 0.9957 as shown in Table 1. For 1400 queries in 1394 the system identified the shape correctly and failed only in 6. For the other experiment, which we called the extended experiment, the images of the dataset were modified by scaling, rotation, and shearing using the same parameters used by other researchers interested in this problem, these parameters are also used in known datasets such as Kimia99, and ETH-80, they are:

– **Scaling**: .1, .2, .25, .3 and 2.

– **Rotation**: 9, 36, 45, 90 and 150 degrees.

– **Shearing**: -.3, -.2, -.1, 0, .1 and .2.

Modified images were used to retrieve unmodified images. The results of the extended experiment are shown in Table 2. The best accuracies were obtained using 7-9 keypoints for scaling, 3-4 keypoints for rotation and 3-10 keypoints for shearing.

Surprisingly using only 3 keypoints the method works very well, leaving not very much room for improvement when using more keypoints. In Table 3 the achieved

accuracy is contrasted with those obtained from the work of researchers that have used the same collection of images.

6 Conclusions and Future Work

We designed a contour shape matching/retrieval technique that is very robust under rotation due to the way the keypoints that conform the triangles are selected, and is also robust under Scaling and Shearing thanks to the use of invariant obtained with Equation 1. The features extracted are compact since only a few keypoints have to be stored. Extracting these features is a low complexity procedure.

Our method does not require the use of DTW algorithm or any aligning mechanism thanks to the way we select our keypoints, that is, the initial point is always about the same point, this fact is very important since it reduces the computational complexity for comparing features between objects. There are however two drawbacks, the method is sensitive to noise so it works very well when combined with a good noise removal technique. Also, our method works only with non-occluded shapes.

References

1. Abbasi, S., Mokhtarian, F., Kittler, J.: Curvature Scale Space Image in Shape Similarity Retrieval. Multimedia Systems, vol. 7, pp. 467–476 (1999). DOI: 10.1007/s005300050147.
2. Abro, M., Talpur, S., Soomro, N., Brohi, N.: Shape Based Image Retrieval Using Fused Features. EAI Endorsed Transactions on Internet of Things, vol. 5, no. 17 (2019). DOI: 10.4108/eai.31-10-2018.159916.
3. Alajlan, N.: HopDSW: An Approximate Dynamic Space Warping Algorithm for Fast Shape Matching and Retrieval. Journal of King Saud University-Computer and Information Sciences, vol. 23, no. 1, pp. 7–14 (2011). DOI: 10.1016/j.jksuci.2010.01.001.
4. Alajlan, N., El-Rube, I., Kamel, M.S., Freeman, G.: Shape Retrieval Using Triangle-Area Representation and Dynamic Space Warping. Pattern Recognition, vol. 40, no. 7, pp. 1911–1920 (2007). DOI: 10.1016/j.patcog.2006.12.005.
5. Arjun, P., Mirnalinee, T.: An Efficient Image Retrieval System Based on Multi-Scale Shape Features. Journal of Circuits, Systems and Computers, vol. 27, no. 11 (2018). DOI: 10.1142/S0218126618501748.
6. Bartolini, I., Ciaccia, P., Patella, M.: Warp: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 142–147 (2005). DOI: 10.1109/TPAMI.2005.21.
7. Chávez, E., Chávez-Cáliz, A.C., López-López, J.L.: Polygon Matching and Indexing Under Affine Transformations. Computer Vision and Pattern Recognition (cs.CV), pp. 1–9 (2013). DOI: 10.48550/arXiv.1304.4994.
8. Chávez, E., Chávez-Cáliz, A.C., López-López, J.L.: Affine Invariants of Generalized Polygons and Matching Under Affine Transformations. Computational Geometry, vol. 58, pp. 60–69 (2016). DOI: 10.1016/j.comgeo.2016.06.003.
9. Damen, D., Bunnun, P., Calway, A., Mayol-Cuevas, W.: Realtime Learning and Detection of 3D Texture-less Objects: A Scalable Approach. In: British Machine Vision Conference, pp. 1–12 (2012). DOI: 10.5244/C.26.23
10. Hernández, E.A., Alonso, M.A., Chávez, E., Covarrubias, D.H., Conte, R.: Robust Polygon Recognition Method with Similarity Invariants Applied to Star Identification. Advances in Space Research, vol. 59, no. 4, pp. 1095–1111 (2017). DOI: 10.1016/j.asr.2016.11.016.

11. Jain, A.K., Vailaya, A.: Image Retrieval Using Color and Shape. Pattern Recognition, vol. 29, no. 8, pp. 1233–1244 (1996). DOI: 10.1016/0031-3203(95)00160-3.
12. Keogh, E., Wei, L., Xi, X., Vlachos, M., Lee, S.H., Protopapas, P.: Supporting Exact Indexing of Arbitrarily Rotated Shapes and Periodic Time Series Under Euclidean and Warping Distance Measures. The VLDB Journal, vol. 18, no. 3, pp. 611–630 (2009). DOI: 10.1007/s00778-008-0111-4.
13. Kim, W.Y., Kim, Y.S.: A Region-Based Shape Descriptor Using Zernike Moments. Signal Processing: Image Communication, vol. 16, no. 1-2, pp. 95–102 (2000). DOI: 10.1016/S0923-5965(00)00019-9.
14. Kumar, R., Mali, K.: Shape Classification Via Contour Matching Using the Perpendicular Distance Functions. International Journal of Engineering and Applied Physics, vol. 1, no. 2, pp. 192–198 (2021)
15. Latecki, L., Lakamper, R., Eckhardt, T.: Shape Descriptors for Non-Rigid Shapes With a Single Closed Contour. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424–429 (2000). DOI: 10.1109/CVPR.2000.855850.
16. Mokhtarian, F., Abbasi, S., Kittler, J.: Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. Image Databases and Multi-Media Search, pp. 51–58 (1997). DOI: 10.1142/9789812797988_0005.
17. Paramarthalingam, A., Thankanadar, M.: Extraction of Compact Boundary Normalisation Based Geometric Descriptors for Affine Invariant Shape Retrieval. IET Image Processing, vol. 15, no. 5, pp. 1093–1104 (2021). DOI: 10.1049/ipr2.12088.
18. Xu, X., Lee, D.J., Antani, S., Long, L.R.: A Spine X-ray Image Retrieval System Using Partial Shape Matching. IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 1, pp. 100–108 (2008). DOI: 10.1109/TITB.2007.904149.
19. Yildirim, M.E., Ince, O.F., Yucel, B.S., Ince, I.F.: Shape Retrieval Using Angle-Wise Contour Variance. Journal of Electrical Engineering, vol. 72, no. 2, pp. 99–105 (2021). DOI: 10.2478/jee-2021-0013.
20. Zhang, C., Zheng, Y., Guo, B., Li, C., Liao, N.: SCN: A Novel Shape Classification Algorithm Based on Convolutional Neural Network. Symmetry, vol. 13, no. 3, pp. 1–17 (2021). DOI: 10.3390/sym13030499.
21. Zhang, D., Lu, G.: Evaluation of MPEG-7 Shape Descriptors Against Other Shape Descriptors. Multimedia Systems, vol. 9, no. 1, pp. 15–30 (2003). DOI: 10.1007/s00530-002-0075-y.
22. Zhang, D., Lu, G.: Review of Shape Representation and Description Techniques. Pattern Recognition, vol. 37, no. 1, pp. 1–19 (2004). DOI: 10.1016/j.patcog.2003.07.008.

ISSN 1870-4069

A Geometric Strategy for Recognizing Images of Highly Similar Places within Urban Environments

Carlos A. Martínez-Miwa, Mario Castelán, L. Abril Torres-Méndez

Centro de Investigación y de Estudios Avanzados Campus Saltillo,
Robotics and Advanced Manufacturing Group,
Mexico

{carlos.miwa, mario.castelan, abril.torres}@cinvestav.edu.mx

Abstract. The recognition of previously visited places within urban environments is an essential skill for autonomous vehicles, as it may reduce localization errors during their navigation. The search for improvements in detection capabilities within regions where other sensors, such as lasers or GPS (Global Positioning System) do not perform accurately, has contributed to considerable advances in location recognition systems. Some state of the art approaches require a priori knowledge of the environment. However, this is not always useful due to constant changes in the outside world, variations of viewpoint, or the occurrence of similar images captured from different locations. In this work, we propose a methodology to carry out the visual recognition of places with highly similar characteristics, and prone to spatial variations, illumination changes and occlusions. Our recognition strategy is based on image retrieval by means of detector-descriptors pairs, from which the combination GFTT-SIFT (Good Features to Track - Scale Invariant Feature Transform) exhibits the best performance. For results refinement, we use an image similarity threshold based on geometric constraints. Compared to a high-level learning approach the proposed methodology has a greater precision and discrimination power to identify images of similar zones, besides differentiating those belonging to different sites.

Keywords: Place recognition, machine learning, computer vision, feature detection.

1 Introduction

Visual recognition of previously visited places is a fundamental part of our daily lives. The study of how living beings recognize places, taking into account the movement from one place to another, has a long history in neuroscience [3, 11]. Several discoveries in this area have provided a physiological basis for the representation of spatial locations in our brain [22, 24].

As humans, when visiting a place for the first time, we seem to be more attentive to those details that we believe will best represent it, looking for them to be sufficiently distinctive to create a strong association. Hence, by revisiting that location in the future,

even when different conditions exist, selected features may be activated leading to an accurate detection [27].

These concepts find application in a wide variety of research fields. Such is the case of robotics. One fundamental goal of this area is to develop fully functional systems that can operate robustly in the real world. Mobile robots, particularly autonomous ones, must have a deep understanding of their surroundings so that they can be entrusted with highly complicated or risky responsibilities that humans should not take on, for example, preventing natural disasters, space and underwater exploration, or even search and rescue activities. Therefore, visual place recognition (VPR) becomes an extremely important process as it enables robots to reduce uncertainty and location errors during their exploration.

This paper proposes a methodology for the identification of previously visited sites under challenging conditions, mostly based on geometric constraints. In the context of urban environments, the term "challenging" refers to spatial variations, illumination, occlusions and the presence of similar elements with great frequency. We first designed a novel database to depict this sort of settings. We also tested a significant number of local feature detector-descriptor combinations aimed at selecting the one that performed the best for our dataset. Place recognition is determined from a geometrical nature concept that, in spite of being more commonly associated with topics such as visual odometry, generates highly favorable results in the search for previosly seen places. Our method achieves a reliable place recognition, without the requirement of prior training, surpassing the performance of a state of the art algorithm.

This article is organized as follows: relevant work is described in Section 2; the process of gathering the database for testing our method is explained in Section 3; proposed method is briefly depicted in Section 4; experimental results are presented in Section 5; and, finally, conclusions and future research directions are provided in Section 6.

2 Related Work

State of the art related with VPR includes research works such as the displacement of a robot along a previously learned route [14]. The information acquired by means of sensors, i.e. cameras, is first described and then compared with an internal representation, or map of the environment, in order to estimate the probability of data matching an image inside the map. Unfortunately, if a robot intends to act without previous knowledge of its surroundings, this procedure becomes extremely challenging, mainly due to three main factors:

1. Variability in the appearance of the same scene (changing illumination, occlusions, weather conditions).
2. The possibility that a scene viewpoint may not always be the same.
3. Images from different locations looking too similar, effect known as perceptual aliasing.

Other conventional approaches are those based on visual scene detection and description techniques. These can operate with local features (involving a

Fig. 1. The left column exhibits images from the same site at different times of the day. The top left image was captured at 19 h while the bottom left image at 13 h. The second column depicts two frames of locations that were far from each other, but visually similar that they may appear to belong to the same site and the same hour.

detection-description pair), such as scale-invariant feature transformations (SIFT) [13] and Speeded-Up Robust Features (SURF) [2], or, alternatively, resource to whole images without the need for a detection stage [26].

Since feature extraction does not involve a very complex and demanding process, it is not surprising to discover combinations of these methods [18, 20]. Nevertheless, a poor performance of this kind of descriptors has been reported upon varying circumstances, especially those related with illumination changes [8].

In [6] location or object recognition problems are addressed through the Bag of Words (BoW). This involves representing image features in terms of a numeric vector, that can be efficiently compared to other vectors encoding information about a series of words. These words are the names given to the image descriptors. While this approach performs well and is scalable to large amounts of data, its performance and functionality decline when regions with conditions other than those included in the training images are encountered.

Analogous to the BoW model, the "Bag of Relevant Regions" was presented in [15]. This novel method aimed at describing a scene in terms of relevant regions, extracted from a visual attention algorithm. Although this work outperforms well-known approaches such as the Fast Appearance Based-Mapping (FAB-MAP) [5], it outputs a great amount of false negatives. FAB-MAP applies probabilistic calculations, based on the local appearance of a site, for its identification. Perceptual aliasing is tackled not only by considering whether two scenarios are similar in terms of the visual words they have in common, but also that these are sufficiently distinctive.

As a result, if two sites seem similar but their words are frequently observed, FAB-MAP generates a low correspondence probability. FAB-MAP uses a BoW model, with SIFT or SURF features, for image description and computes the dissimilarity of each word during a training phase. Nevertheless, this training causes a computational cost increase.

Authors in [9] suggest a solution for VPR based on a BoW built on a local feature detector-descriptor combination for the purposes of simultaneous localization

Table 1. List of 22 detector-descriptor combinations used to identify previously visited locations at different times of the day. The overall success rate was $52.62 \pm 14.79\%$.

Detector-Descriptor	Success %	Detector-Descriptor	Success %
AVA-ORB	73.33	STAR-SIFT	46.66
AVA-SIFT	73.33	STAR-SURF	44.44
AVA-SURF	73.33	KAZE-KAZE	42.22
GFTT-BRISK	73.33	AKAZE-AKAZE	40
GFTT-ORB	73.33	BRISK-BRISK	40
GFTT-SIFT	73.33	FAST-BRISK	40
GFTT-SURF	73.33	FAST-ORB	40
ORB-ORB	51.11	FAST-SIFT	40
AVA-BRISK	46.66	FAST-SURF	40
STAR-BRISK	46.66	SIFT-SIFT	40
STAR-ORB	46.66	SURF-SURF	40

and mapping (SLAM). The selection of these algorithms aimed at reducing processing time, although no prior evaluation of detector-descriptor combinations was carried out. Moreover, although some of the databases used are spatially dynamic, they do not reflect changes in the hours of the day.

A variety of machine learning methods have also been resorted to. In [23], Histogram of Oriented Gradients (HOG) [7] fetaures and Local Binary Patterns (LBP) [21] are concatenated for visual localization. Then, given an image, a Support Vector Machine (SVM) model identifies the most similar one within a geo-referenced database. Other approaches rely on Convolutional Neural Networks (CNNs) as strong feature extractors for place recognition in changing environments.

Researchers in [4] and [28] performed an analysis of the robustness of different CNN layers against visual appearance and viewpoint modifications across a set of images. It was concluded that intermediate layers exhibit robustness to appearance alterations, while higher level layers perform better facing viewpoint shifts.

Notwithstanding, no mechanism is presented for an automatic selection of the best layer for the task at hand. A dependency on the training database is also evident. Thus, features that generate good results on one dataset, may have little impact against a different one. Further works related to deep learning have emerged recently [1, 10, 19, 31]. Nonetheless, overall the main disadvantage is the need for large amounts of training data and the consequent high computational costs.

3 Dataset Collection

For this work, we collected a new place recognition dataset. Our image collection focuses on depicting urban environments that could reflect highly challenging conditions for a computer vision system. Here, the term challenging alludes to settings that do not contain significant visual information or that are subject to dynamic factors.

The gathering of these images was inspired by [17], where the authors explored how participants recognized, through defiant conditions, different pleaces recorded

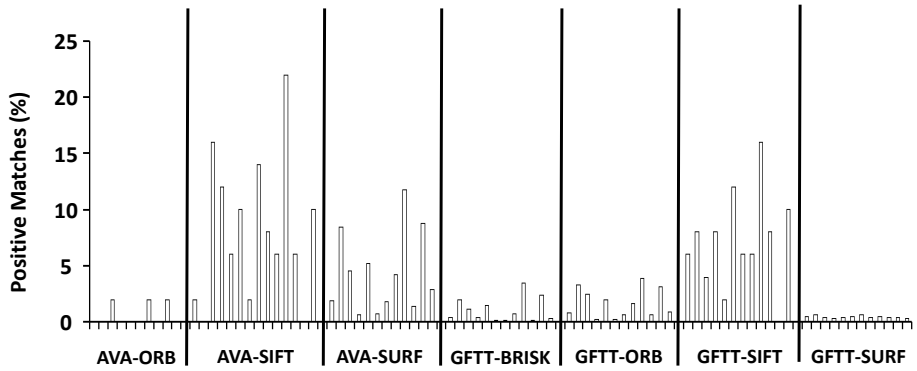


Fig. 2. Match percentages for the 7 best algorithms evaluated on the 12 most challenging images. Note that the strongest performing detector-descriptor pairs are AVA-SIFT and GFTT-SIFT.

along a video sequence describing the navigation of a car. The image collection was gathered at the city of Saltillo, Mexico, driving along a 0.8 km route, at a speed of 30 km/h, through a series of streets which could be identified as belonging to a typical urban neighborhood.

Three sequences compose the dataset, each of which was captured at different hours (07 h, 13 h and 19 h) of the day. For this procedure, a GoPro Hero4 camera mounted on a Chevrolet Cruze vehicle was used.

The data comprises a total of 447 images, 149 per time of the day. The first column of Figure 1 illustrates examples of images representing the same location at 19 h and 13 h. The second column of the figure presents frames from distinct scenarios that share very similar characteristics. The database is challenging, since many of the major image processing problems are addressed, i.e., illumination and spatial variations, occlusions, or the presence of frequent similar elements along navigation.

In addition, the fact that the images were captured under different environmental conditions can result in the detection of mismatches in several elements, for example building colors, plants or even the sky, leading to undesirable detections and confusion.

4 Evaluating Detector-Descriptor Pairs for Geometric-Based VPR

The first need to be fulfilled for geometric-based VPR is to count on a reliable detector-descriptor pair. For this reason, we conducted a thorough evaluation of 22 detector-descriptor couples for identifying whether a reference image was or not included in its related video sequence. Such techniques were designated because of their ease of implementation and access availability. Results are listed in Table 1.

A threshold was applied to every couple in order to determine if the found correspondences were sufficient to establish a positive match between two images. For setting this threshold, 80 % of the maximum number of matched points was selected.

From the analysis of the table, it is possible to appreciate how 7 out of 22 combinations reached the highest success rates, while the worst performance was attributed to other 7 pairs. For this reason, we focused on the 7 highest-rated.

Fig. 3. Example of GFFT-SIFT qualitative results. It is to note how, even at different times of the day and between relatively distant scenes, GFTT-SIFT is capable of detecting enough features so as to determine a positive match among both images.

These 7 best detector-descriptor pairs were then tested on the 12 worst performing images to point out the strongest performance.

AVA-SIFT (Aqua Visual Attention [16] - SIFT) and GFTT-SIFT (Good Features To Track [25] - SIFT) emerged as the most outstanding in terms of the number of matches found, as shown in Figure 2.

Although the number of correspondences is a good indicator for determining the similarity between two scenes, there is a possibility that this parameter carries some uncertainty. The latter refers to the fact that if, for a couple of images, a detector-descriptor generates a number of correspondences lower than the threshold set (80% of the maximum found), these could be enough to state that both scenes depict the same location.

Results of the 7 top detector-descriptor combinations were revisited for the 12 worst performing images, but this time in a qualitative fashion, to verify whether or not they constituted a good match.

In this way, it was possible to identify that, despite not exhibiting the best performance under the previous metric, GFTT-SIFT stood out from the rest. This pairing detected correspondences between images belonging to the same scenario, even if they suffered from a significant spatial offset as illustrated by Figure 3.

Once the detector-descriptor pair was selected, a new metric was defined to better discriminate among images that do and do not represent the same site. We chose this parameter to be based on the epipolar constraint.From this restriction, it is understood that there must exist a transformation $\mathbf{x} \rightarrow \mathbf{l'}$ of a point in one image with its respective epipolar line in a second one. The transformation from points to lines results in a correlation, expressed by the fundamental matrix $\mathbf{F}$ [12]:

$$\mathbf{l'} = \mathbf{Fx}. \tag{1}$$

The fundamental matrix, then, satisfies that for any couple of matching points $\mathbf{x}$ and $\mathbf{x'}$ in two images:

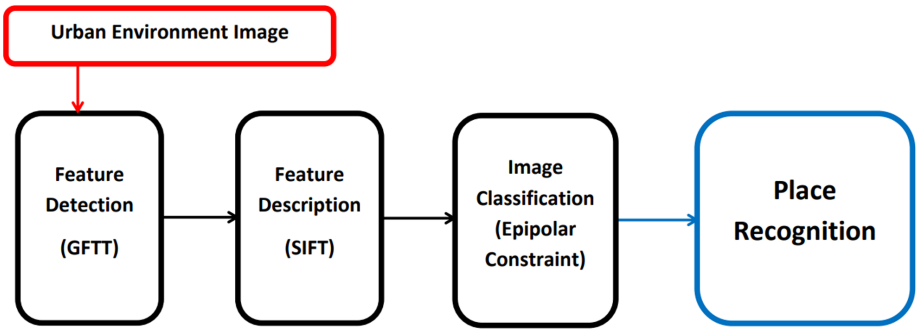


Fig. 4. Methodology proposed for place recognition in challenging environments.

$$\mathbf{x'^{T}Fx} = 0. \tag{2}$$

This condition is true because, if points **x** and **x'** are correspondent, **x'** lies on the epipolar line $\mathbf{l'} = \mathbf{Fx}$ related to point **x**. In other words, $\mathbf{x'^{T}l'} = \mathbf{x'^{T}Fx} = 0$. Besides, if image points satisfy $\mathbf{x'^{T}Fx} = 0$, rays defined by them are coplanar, a necessary criterion to establish correspondence between them.

Taking as a reference equation (2), the proposed metric is introduced: if for two images, the number of matches found by GFTT-SIFT is high enough to generate a fundamental matrix, they will be considered as positive correspondences, that is, coming from the same scenario; otherwise, they will be catalogued as belonging to different locations.

A diagram describing the proposed methodology is shown in Figure 4. As a first step, starting from a given scene to be recognized, the most important features are detected and described in order to locate the best match. For this purpose, a classic detector-descriptor combination such as GFTT-SIFT is adopted.

These correspondences undergo a geometric classification method based on the epipolar constraint. In this procedure, we managed to eliminate all images that lie below a defined threshold, and also managed to categorize the remaining images into four main groups: True and False Positives, and True and False Negatives.

5 Results

The process depicted in Figure 4 was evaluated on the database described in Section 3. The evaluation consisted of an image retrieval task, i.e., each of the 447 images (149 morning $\times$ 149 afternoon $\times$ 149 night) was compared to the rest, aiming at determining whether matches found in each pairing were enough to create a fundamental matrix (i.e., satisfy the epipolar constraint).

By means of these trials, a tool that allows visualization and comparison of the results, known as the similarity matrix, could be constructed. Figure 5a illustrates the similarity matrix at the end of this experimentation.

Each black dot represents a positive match detected in a pair of images, i.e, that they belong to the same site. In order to build a ground truth matrix (Figure 5b), all pairing

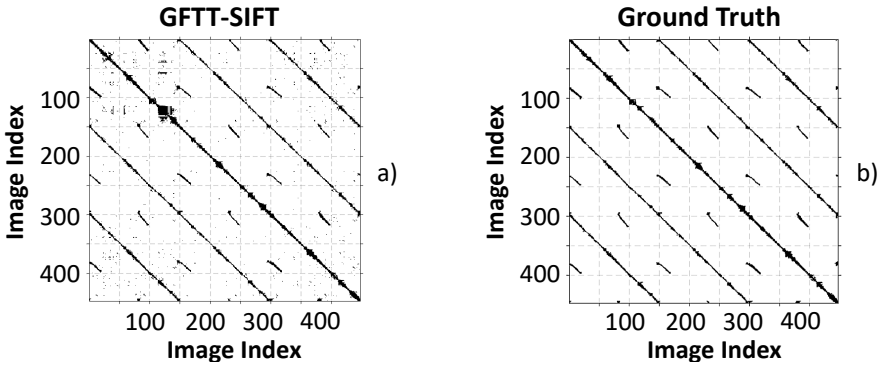


Fig. 5. Similarity matrices. Each point (dark regions) within the matrix represents a correspondence detected in a pair of images from different sequences. On the left side, figure 5a illustrates the results of our method. Figure 5b, on the right, plots the expected result.

images were carefully examined by the main author of this work, who visually decided which pairs of images were positive or negative matches. From Figure 5, it is to note that the similarity matrix derived from our method significantly resembles the ground truth. However, two special cases arise: False Positives and False Negatives. The former allude to additional points found in the figure's white zone, where matches would be assumed to be null.

Further, our results reveal a black box at the upper left side of the matrix. These artifacts, although located on the main diagonal, are made up of dots that should not exist, namely, False Positives.

False Negatives, on the other hand, refer to those images in which, despite depicting the same place, no correspondence between pairs of images was detected. Both cases constitute specific problems in the performance of the proposed method. In order to evaluate the presence of false positives and negatives in the performance or our method, we used a Precision-Recall (PR) curve, shown in Figure 6.

From the curve, it can be clearly perceived that as recall increases, precision decreases, though in a very low proportion. The accuracy achieved is considerably high, obtaining a maximum value of 0.9523, dropping only to 0.8371. The sensitivity factor also produces favorable results. In spite of the minimum value of 0.0519 being quite low, it reaches the recall limit of 1 with a still high precision. Taking these data into account, added to the fact that the area under the PR curve is of great dimension, it is possible to establish that our methodology achieved a strong classification capacity.

For comparative purposes, our dataset was additionally evaluated under a methodology with different *modus operandi*: the fast appearance-based mapping algorithm, or FAB-MAP [5].

FAB-MAP is one of the most popular solutions for VPR based on local image features. This approach turns to probability for the identification of locations and also employs a BoW model built upon appearence-based features, e.g, SIFT and SURF.

Despite representing an important milestone within the state of the art, its performance struggled to obtain favorable results in our database. As evidenced in

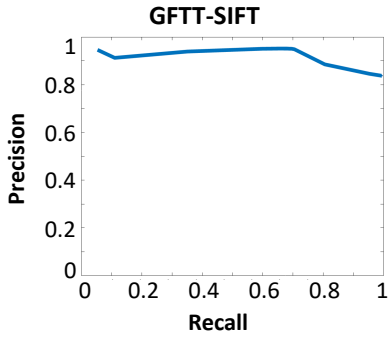


Fig. 6. PR curve produced by GFTT-SIFT detector-descriptor combination. It is noteworthy how, while the recall increases (up to a value of 1), precision decays only to a rate close to 0.85.

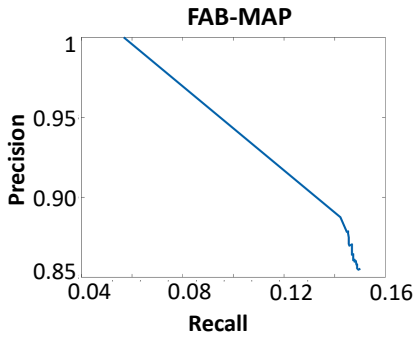


Fig. 7. FAB-MAP Precision-Recall curve. The prevalence of a high accuracy index (0.85) is notable. However, as Precision diminishes, Recall reaches just an index near 0.15.

Figure 7, whereas Precision drops close to 0.85, only a 0.15 Recall is reached. Such score indicates that although this methodology was able to discriminate most of the possible False Positive cases, there were a large number of False Negatives.

The latter is verified through the generated similarity matrix, depicted at the left of Figure 8). From the analysis of Figures 5 through 8 of this section, we can establish that the proposed algorithm, based on a detector-descriptor combination (GFTT-SIFT) under a geometric strategy, outperforms a learning-based method, such as FAB-MAP, for recognizing previously visited places subject to dynamic conditions (spatial, lighting and occlusions).

6 Conclusions and Future Work

In this work we presented a novel database for previously visited places in the context of VPR. The main particularity of these images is the set of challenging conditions for computer vision techniques.

Our video sequences were captured at different times of the day, yielding a combination of spatial and illumination changes, occlusions, similar elements with

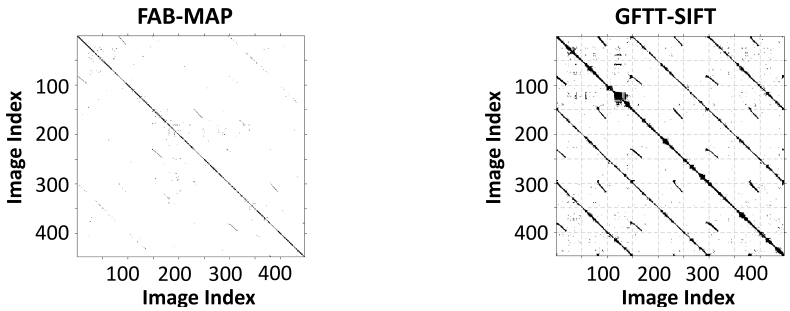


Fig. 8. Similarity matrix generated by FAB-MAP (a) compared to the similarity matrix obtained through our method (b). The presence of a large number of False Negatives can be seen, reinforcing the low recall shown in the PR curve of Figure 7.

high repeatability and even environmental factors that may cause confusion, such as the sky. On the basis of our experiments, we realize that classical computational algorithms, as combinations of detectors and local feature descriptors, generate sufficiently good results in location recognition with greater speed and simplicity and without compromising reliability, in comparison with a state of the art methods that uses BoW.

We conducted an exhaustive search for the best detector-descriptor combination for VPR. The Good Features To Track feature detector, along with the SIFT descriptor, exhibited high robustness in identifying reliable features that corresponded to important regions in the environment even in adverse situations such as changes in lighting due to the different day hours.

The designation of the fundamental matrix as a geometric constraint was a relevant addition for frame classification. Although it is most often applied to tasks such as visual odometry, it proved to be a solid and accurate method for the identification of previously visited sites. Our methodology, by itself, was able to produce highly successful results.

From a Precision-Recall Curve, a high measure of sensitivity (recall) was achieved with a very low decrease in pecision. Finally, these results are supported by a comparison against a learning method considered a milestone in the state of the art: FAB-MAP. The obtained plots exhibit a very low recall for this probabilistic algorithm, as well as a larger drop in precision. Thus, it is demonstrated that our appraoch is able to perform accurate, fast and simple VPR, without the need to rely on large quantities of training data, nor consuming high computational time.

As future work we intend to analyze the incorporation of techniques that provide different perspectives to the geometric ones, for instance, detector-descriptor pairs used for visual attention. In this way, we could address the highly challenging cases that could not be completely solved under the proposed methodology. We also aim to test our methodology under public and commonly used databases related to the VPR problem, for instance [29, 30].

Similarly, we are looking forward to publishing our novel dataset in a public repository so that other researchers can make use of it.

References

1. Atapour-Abarghouei, A., Akcay, S., de La-Garanderie, G.P., Breckon, T.P.: Generative Adversarial Framework for Depth Filling Via Wasserstein Metric, Cosine Transform and Domain Transfer. Pattern Recognition, vol. 91, pp. 232–244 (2019). DOI: 10.1016/j.patcog.2019.02.010.
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359 (2008). DOI: 10.1016/j.cviu.2007.09.014.
3. Brown, J.W.: Neuropsychology of Visual Perception. Psychology Press, vol. 2 (1989)
4. Chen, Z., Lam, O., Jacobson, A., Milford, M.: Convolutional Neural Network-Based Place Recognition. In: Australasian Conference on Robotics and Automation (ACRA) (2014)
5. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. The International Journal of Robotics Research, vol. 27, no. 6, pp. 647–665 (2008). DOI: 10.1177/0278364908090961.
6. Cummins, M., Newman, P.: Appearance-only SLAM at Large Scale with FAB-MAP 2.0. The International Journal of Robotics Research, vol. 30, no. 9, pp. 1100–1123 (2011). DOI: 10.1177/0278364908090961.
7. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893 (2005). DOI: 10.1109/CVPR.2005.177.
8. Furgale, P., Barfoot, T.: Visual Teach and Repeat for Long Range Rover Autonomy. Journal of Field Robotics, vol. 27, no. 5, pp. 534–560 (2010). DOI: 10.1002/rob.20342.
9. Gálvez-López, D., Tardos, J.D.: Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188–1197 (2012). DOI: 10.1109/TRO.2012.2197158.
10. Garg, S., Milford, M.: SeqNet: Learning Descriptors for Sequence-Based Hierarchical Place Recognition. IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4305–4312 (2021). DOI: 10.1109/LRA.2021.3067633.
11. Golledge, R.G.: Do People Understand Spatial Concepts: The Case of First-Order Primitives. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, pp. 1–21 (1992). DOI: 10.1007/3-540-55966-3_1.
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, (2004)
13. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110 (2004). DOI: 10.1023/B:VISI.0000029664.99615.94.
14. Lowry, S., Sünderhauf, N., Newman, P., Leonard, J., Cox, D., Corke, P., Milford, M.: Visual Place Recognition: A Survey. IEEE Transactions on Robotics, vol. 32, no. 1, pp. 1–19 (2016). DOI: 10.1109/TRO.2015.2496823.
15. Maldonado-Ramírez, A., Torres-Méndez, L.A.: A Bag of Relevant Regions for Visual Place Recognition in Challenging Environments. In: 23rd International Conference on Pattern Recognition, pp. 1358–1363 (2016). DOI: 10.1109/ICPR.2016.7899826.
16. Maldonado-Ramírez, A., Torres-Méndez, L.A.: Robotic Visual Tracking of Relevant Cues in Underwater Environments with Poor Visibility Conditions. Journal of Sensors, vol. 2016, pp. 1–16 (2016). DOI: 10.1155/2016/4265042.
17. Martínez-Miwa, C.A., Castelán, M., Torres-Méndez, L.A., Maldonado-Ramírez, A.: Human and Machine Capabilities for Place Recognition: A Comparison Study. In: The Tenth International Conference on Advanced Cognitive Technologies and Applications, pp. 72–77 (2018)

18. Mei, C., Sibley, G., Cummins, M., Newman, P., Reid, I.: A Constant-Time Efficient Stereo SLAM System. In: Proceedings of the 20th British Machine Vision Conference, pp. 1–11 (2009)
19. Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: SeDAR: Reading Floorplans Like a Human-Using Deep Learning to Enable Human-Inspired Localisation. International Journal of Computer Vision, vol. 128, no. 5, pp. 1286–1310 (2020). DOI: 10.1007/s11263-019-01239-4.
20. Newman, P., Chuchill, W.: Experience-Based Navigation for Long-Term Localization. The International Journal of Robotics Research, vol. 32, no. 14, pp. 1645–1661 (2013). DOI: 10.1177/0278364913499193.
21. Ojala, T., Pietikäinen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Featured Distributions. Pattern Recognition, vol. 29, no. 1, pp. 51–59 (1996). DOI: 10.1016/0031-3203(95)00067-4.
22. O'keefe, J., Nadel, L.: The Hippocampus as a Cognitive Map. Behavioral and Brain Sciences, vol. 2, no. 4, pp. 487–494 (1979). DOI: 10.1017/S0140525X00063949.
23. Qiao, Y., Cappelle, C., Ruichek, Y.: Place Recognition Based Visual Localization Using LBP Feature and SVM. In: Mexican International Conference on Artificial Intelligence, vol. 9414. pp. 393–404 (2015). DOI: 10.1007/978-3-319-27101-9_30.
24. Redish, A.D., Touretzky, D.S.: Cognitive Maps Beyond the Hippocampus. Hippocampus, vol. 7, no. 1, pp. 15–35 (1997) DOI: 10.1002/(SICI)1098-1063(1997)7:1 ¡15::AID-HIPO3¿3.0.CO;2-6.
25. Shi, J., Tomasi, C.: Good Features to Track. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994). DOI: 10.1109/CVPR.1994.323794.
26. Siagian, C., Itti, L.: Impact of Neuroscience in Robotic Vision Localization and Navigation. Computational and Cognitive Neuroscience of Vision, Cognitive Science and Technology, pp. 235–276 (2017). DOI: 10.1007/978-981-10-0213-7_11.
27. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley Longman Publishing Co., Inc, (1984)
28. Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., Milford, M.: On the Performance of ConvNet features for Place Recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4297–4304 (2015). DOI: 10.1109/IROS.2015.7353986.
29. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 Place Recognition by View Synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1808–1817 (2015). DOI: 10.1109/CVPR.2015.7298790.
30. Torii, A., Sivic, J., Okutomi, M., Pajdla, T.: Visual Place Recognition with Repetitive Structures, pp. 883–890 (2015). DOI: 10.1109/CVPR.2013.119.
31. Zhang, Y., Bai, Y., Ding, M., Ghanem, B.: Multi-Task Generative Adversarial Network for Detecting Small Objects in the Wild. International Journal of Computer Vision, vol. 128, no. 6, pp. 1810–1828 (2020). DOI: 10.1007/s11263-020-01301-6.

ISSN 1870-4069

Unsupervised Learning Algorithms are Able to Identify Relevant Patterns in the Pollution Data in Mexico City

Victor Lomas-Barrie[1], Tamara Alcántara[2], Sergio Mota[3], Antonio Neme[4]

[1] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Mexico

[2] Universidad Nacional Autónoma de México,
Dirección General de Cómputo y de Tecnologías de Información y Comunicación,
Mexico

[3] Universidad Nacional Autónoma de México,
Postgraduation Program in Computer Science,
Mexico

[4] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Mexico

{victor.lomas, antonio.neme}@iimas.unam.mx, talcantarac@unam.mx

Abstract. Air pollution is a major problem in almost every large city since the affections to human health are numerous, including damage to tissues and an increase in respiratory-related events. Many cities maintain a monitoring system in order to measure the level of several contaminants such as ozone and carbon monoxide that are particularly harmful to humans. By analyzing the temporal dynamics of those pollutants, authorities may decide to increase mobility constraints or activate contingency plans aiming to reduce the pollution levels. The Air quality authority in Mexico maintain a system of over 20 monitoring stations that serves the Metropolitan Area of Mexico City, covering an area of over $300km^2$, and sampling every hour the air for seven pollutants. Based on public data, we applied unsupervised learning algorithms, in particular anomaly detection algorithms, to unveil relevant patterns in data. An anomaly is an observation that does not resemble, under an unknown metric, the vast majority of instances within a dataset. By applying existing anomaly detection algorithms, we identified several observations of pollutant concentrations that differ from the rest of the observations. The existence of anomalies in the air pollution dataset indicates a qualitative change in the pollution dynamics over time, and the adequate identification of anomalies provide specialists with more information about those changes.

Keywords: Air pollution, monitoring system, contaminants, anomaly detection, pollution dynamics.

1 Introduction

The identification of elements that do not resemble the remaining objects from the same collection is a sign of intelligence [5]. An anomaly is an instance that, under certain unknown metric, do not resemble the rest of the elements in the same dataset. Detecting such anomalies is an open task, and several disciplines have dedicated considerable effort to try to solve it. Artificial intelligence has proposed some ideas aiming to identify anomalies by one path or another. In particular, the techniques defined as unsupervised learning have proven to be particularly relevant.

Unsupervised learning is a field in artificial intelligence aiming to learn from data. It is an open task since it is usually unclear what can be learnt from data, and how to fulfil this task is a prolific field. Several aspects can be learnt from data. A rather common aspect to learn is the separation in clusters. A different aspect to learn is whether an observation is anomalous with respect to the rest of the instances within a dataset [28].

The identification of such instances is an open task, and the techniques and approaches that aim to identify them is known as anomaly detection (AD) [21]. Given a dataset, a rather important question to ask is if all observations, instances, data, or any other synonymous term, were generated by the same mechanism. Whatever the process or structure under study, AD algorithms aim to identify a subset of observations that differ, under an usually unknown metric, to the rest of the elements. Anomaly detection aims to identify, within an unlabelled dataset, those instances or vectors that deviate from a common description found in the vast majority of vectors.

There are, in fact, two instances of anomaly detection. The first one is closely related to classification under unbalanced classes. In this scenario, each observation or vector is labelled as either common, normal, or any other synonym or as anomaly. The former is in general much more abundant than the latter, and thus, there is an unbalance in the classes. This scenario is of higher relevance, since in many applications of data science, it is not known before hand what instances constitute anomalies and which ones are common observations.

We are more interested in the second scenario for anomaly detection. In it, the vectors are not labelled and thus, the algorithm has to infer the class of the vectors, or assign an anomaly degree to them, based on undisclosed properties of the data.

Since the properties of the data that are to be taken into consideration for telling apart anomalies from common vectors are not unique, several alternatives exist. Some vectors can be identified as anomalies under certain assumptions, and not under a different set of premises. The working hypothesis is that observations that significantly differ from the common or usual observations are an indication of the presence of an additional mechanism that threads in the usual mechanism.

In this contribution, we face the problem of detecting anomalies in the air pollution levels in Mexico City from 2011 to May 2022. In this context, an anomaly corresponds to a set of measurements of different pollutants that do not resemble the vast majority of the observations.

Anomalies are relevant since they indicate that, besides the obvious errors from faulty equipment or human error, the observed system is affected by an additional mechanism. The dynamics of the atmosphere, although well understood, are far from being completely characterized.

When, in the context of urban pollution, an anomaly is present, it us suspected that changes in the variables that affect the density of pollutants have occurred. The occurrence of such changes is important in order to apply relevant decisions to diminish the use of vehicles and to reduce the activity in certain industrial sectors. The rest of this contribution goes as follows. In section 2 we briefly describe the problem we aim to understand, that is, the air pollution in Mexico City.

We briefly describe the impact in health of some of the measured pollutants. We also describe the monitoring system that allows the existence of massive data. In section 3 we describe the anomaly detection algorithms that are to be applied to the pollution data. We proceed to describe some of the main results in section 4, and we end by offering some conclusions and discussing what we think are some the most prominent aspects of this contribution in section 5.

2 Air Pollution Monitoring in Mexico City

Air pollution has several consequences in human health. It can increase respiratory problems and damage tissues. [9, 15, 19, 24, 26]. Some of the suspended pollutants with the highest impact in human health are:

1. CO. When carbon monoxide is inhaled, it replaces the oxygen in the blood. CO causes damages in vital organs like the brain and heart.
2. NO. Nitric oxide causes irritation in the nose, throat and lungs. In high concentration, NO reduces the oxygen in blood causing headaches and fatigue. A longer exposure may cause pulmonary edema.
3. NO2. Breathing Nitrogen Dioxide can aggravate respiratory diseases and produce asthma.
4. NOX. As well as NO2, NOX can produce asthma and increase risk of respiratory diseases.
5. O3. Ozone produce throat irritation, chest pain, lung inflammation and asthma.
6. PM10. These small particles can infiltrate the lung tissue and get into bloodstream, provoking heart or lung disease.
7. PM2.5. Breathing PM2.5 can damage lung function causing asthma and heart disease.
8. SO2. Sulfure dioxide can cause inflammation of the throat and the lungs. Also can produce asthma.

The Mexico City Atmospheric Monitoring System (SIMAT) is composed by eight automatic equipment and seven manual equipment; and it is divided in four sub-systems:

1. Automatic Atmospheric Monitoring Network (RAMA).
2. Manual Atmospheric Monitoring Network (REDMA).
3. Meteorology and Solar Radiation Network (REDMET).
4. Atmospheric Deposit Network (REDDA).

In addition, a laboratory for the physicochemical analysis of samples (LAA) and a data processing and dissemination center (CICA) are also supporting SIMAT.

In this contribution, we rely on data generated by the RAMA system. SIMAT started operations in the year 2000, and in 2003 it incorporated the measurement of PM2.5 particles; and it is responsible for the permanent measurement of the main air pollutants in Mexico City and its metropolitan area, with more than 40 air quality monitoring stations.

The monitoring carried out in the metropolitan area of the Valley of Mexico covers the 16 delegations of Mexico City, as well as 12 suburban municipalities of the State of Mexico, which are: Acolman, Atizapán de Zaragoza, Chalco, Coacalco de Berriozábal, Ecatepec of Morelos, Naucalpan de Juárez, Nezahualcóyotl, Ocoyoacac, Tepotzotlán, Texcoco, Tlalnepantla de Baz and Tultitlán [6].

An atmospheric monitoring station consists of a stand that contains various equipment intended to measure the concentrations of one or more air pollutants and certain meteorological parameters. Manual stations, normally, after carrying out the sampling of contaminants, the sample is transferred to a laboratory for analysis. Automatic stations are those that are integrated with automatic and continuous measurement equipment. Each monitoring station is classified by its coverage area, following the U.S.

Environmental Protection Agency criteria (micro, local, neighborhood, city or regional), its location (urban or rural) and the predominant source of air contamination. The emissions inventories are defined by the predominant source of air contamination in the area where the monitoring station is located. The main emission sources in an urban development include generally industrial plants of all kinds, vehicles with diesel engines, internal combustion, power plants, incinerators, and heating equipment. The stations are classified into [12]:

1. Mobile or vehicular traffic, when the predominant source of emission is from roads, parking lots and/or vehicle service shops.
2. Area, when the predominant emission source is from services such as restaurants, dry cleaners, wineries, shopping malls, etc.
3. Biogenic, when the predominant emission source is related to streets unpaved, parks or empty lots.
4. Fixed, when the predominant emission source is from an industrial area.

The prediction of pollutants in several cities have been tackled by several artificial intelligence techniques. In [2], authors applied neural networks to detect changes in the ozone concentration in urban areas in Vilnius. Prediction of ozone in a large metropolitan area was performed via machine learning and statistical methods in [20]. A deep learning approach was applied in [3] with the objective of predict the concentration of several pollutants. In [18], neural networks were applied to detect temporal pollution patterns in a large metropolitan area.

3 Anomaly Detection Algorithms

An outlier is an instance or observation that falls off the range of the expected or usual data [10]. The term outlier is usually associated to observations that were obtained by a faulty process, such as errors in measurement, transmission, or human-caused mistakes.

In general, outliers tend to be discarded from datasets since they tend to affect performance metrics, and are considered errors. The term anomaly has been applied to refer to those instances that are different from the rest but that are not considered as errors. More modern on anomalies suggest that they may be an early indicator that some changes in the forces behind the observed phenomena are changing [16], or that a different mechanism is in play [21].

The identification of anomalies is an unsupervised learning task. What the algorithm has to learn is a function that tells apart expected or usual observations from the anomalies within the data. It is an open task since it is not clear neither what that function should be nor what parameters should take.

Traditional statistical techniques have proven valuable to detect outliers. Statistical approaches have offered a deep understanding of air pollution dynamics based on a detailed analysis of air quality data. For example, fig. 1 shows the result of applying two statistical approaches to identify outliers.

The first method is based on the Z-score, which constitutes a distance between the mean of the sample and the observations, weighted by standard deviations. This method identifies as outliers the observations that fall at the extremes in the range.

The second method is median absolute deviation (MAD). In MAD, if the difference between the observation and the mean of the sample is greater than a certain value, expressed in standard deviations, that observation is declared an outlier.

However, the use of statistical methods presents constraints. First, only observations below or above a certain threshold are identified as outliers, which clearly is insufficient to cope with the complexities of real-world phenomena. Second, when the number of dimensions increases, these techniques fall short of being reliable. In third place, the questions that can be answered based on this approach are limited.

From the same data, relying on unsupervised learning algorithms, a different set of questions can be answered. For example, we can ask What is the typical profile of the observations within a certain period for a large group of pollutant, or How different are two groups of observations in terms of their measured pollutant concentration.

In order to try to answer these last two questions, and some other relevant ones, we relied on four anomaly detection algorithms. These four algorithms are of different nature from each other. The four methods make different assumptions in order to compute a metric that is common in the vast majority of the observations, and that is not present in the anomaly set of instances.

3.1 Local Outlier Factor

Several families of anomaly detection algorithms have been created in more than two decades of active research. In particular, those focused on the analysis of nearest neighbors are of particular relevance, since the relative size of the neighborhood are a free parameter and thus, a wide sensitive analysis can be conducted.

Local Outlier Factor, o *LOF* [4], or LOF, is one of the best-known anomaly detection algorithms that take into account the surroundings of each vector in order to compute an anomaly index. Here, a vector v is characterized in terms of its k nearest neighbors.

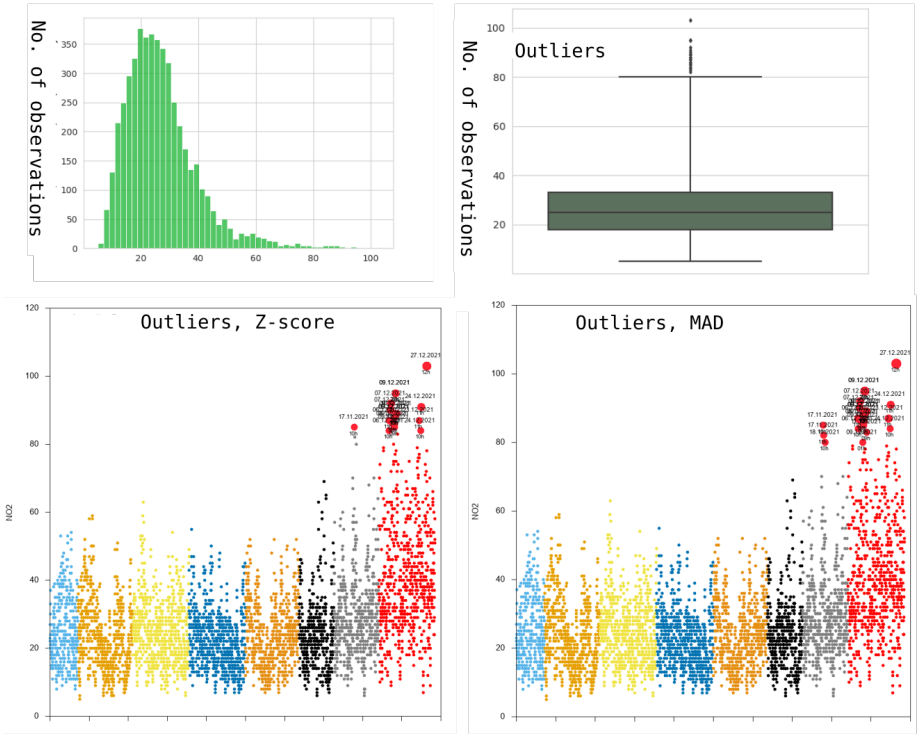


Fig. 1. Identification of outliers based on statistical tools. Top left: Histogram of O_3 concentration at the *Iztapalapa* station for several years. Top right: Boxplot of the same information. Bottom: Time series of the concentration of the pollutant per day. The days that constitute outliers are always in the extremes of the range of values for O_3.

Each of those k neighbors is in turn characterized in terms of its nearest k neighbors. Once the characterizations are concluded, the descriptions obtained from v are compared to those obtained from its k neighbors.

Technically, a vector v is described by a k-distance. k-distance(v) is the distance from v to the $k-$nearest neighbor. The set of neighbors within reach of v based on k-distance(v) is denoted as $N_k(v)$. The reachibility distance from a second vector w and v is given by reachability$-$distance$_k(v,w) = \max($k-distance$(w), d(v,w))$, where d is a distance function. All $k-$neighbours of w will be characterized by the same reachability distance. It should be noted that the reachability distance may be greater than the actual distance. The benefit of this substitution is that it offers more stability for certain distributions.

From the reachability distance, vector v is further described by its local reachability density, defined as:

$$lrd_k(v) = 1/\frac{(\sum_{w \in N_k(v)} \text{reachability} - \text{distance}_k(v,w))}{|N_k(v)|}, \tag{1}$$

$lrd_k(v)$ is a measure of the reachability of vector v from its neighbors. In particular, it is the expected value over all the elements in $N_k(v)$, that is, its $k-$neighbors. From this quantity, the *local outlier factor* or lof is computed:

$$LOF_k(v) = \frac{\sum_{w \in N_k(v)} lrd_k(w)}{|N_k(v)| \times lrd_k(v)}, \tag{2}$$

when $LOF_k(v) > 1$, the local density of v compared to that of its neighbors $N_k(v)$ is lower. On the other hand, if $LOF_k(v) < 1$, it means that vector v presents a higher density of vectors. The former case defines v as an outlier, whereas the latter defines it as an inlier. In this contribution we will refer to both cases as anomalies. The more distant from 1 , the higher the anomaly level.

The control parameter k allows for an increase of the neighborhood and thus. In the extreme case, when k equals the number of elements in the dataset, leads to a global comparison. There is not, however, a formal criteria to identify the correct value of k. As in any other anomaly detection algorithm, if the criteria, in this case defined by the neighborhood size changes, the outcome can also change. This leads to instabilities, but is a problem not tracked in this contribution.

3.2 Isolation Forests

In a high-dimensional feature space, the relative isolation or concentration of a vector offers a path for comparison. Instead of relying on concepts of distance, which are well-known to affect high-dimensional data, the algorithm of isolation forests (*IF*) aims to quantify the anomaly level of each vector based on the effort of isolating it via random decision boundaries [14].

The idea of *IF* is based on exploration of points based on binary trees. In a $N-$dimensional space, an hyperplane of dimension $N - 1$ is needed to create two non-overlapping regions (see fig. 2). For each vector, *IF* randomly selects the dimensions (axis) to create a boundary, and it decides the location of the boundary selecting at random a cut point within the available range.

If the vector of interest is the only within the newly formed region, then the vector is isolated and the number of decision or cuts is linked to the vector. If the vector of interest is not alone in the region, then the algorithm focuses its efforts in that specific region and recursively tries to isolate that vector.

Since *IF* asks binary questions (Is the vector isolated or not?), a binary tree is generated. Graph theory tells us that the number of questions (decisions) needed to identify a node within a binary tree is given by $C(N) = 2 \times H(N-1) - \frac{2(N-1)}{N}$, where N is the number of points in the dataset [22]. Based on $C(N)$, it is possible to compute an anomaly score. If the number of expected trees (decisions) that was needed to isolate vector v is $E(h(v))$, the anomaly level is given by:

$$s(v, N) = 2^{\frac{E(h(v))}{C(N)}}. \tag{3}$$

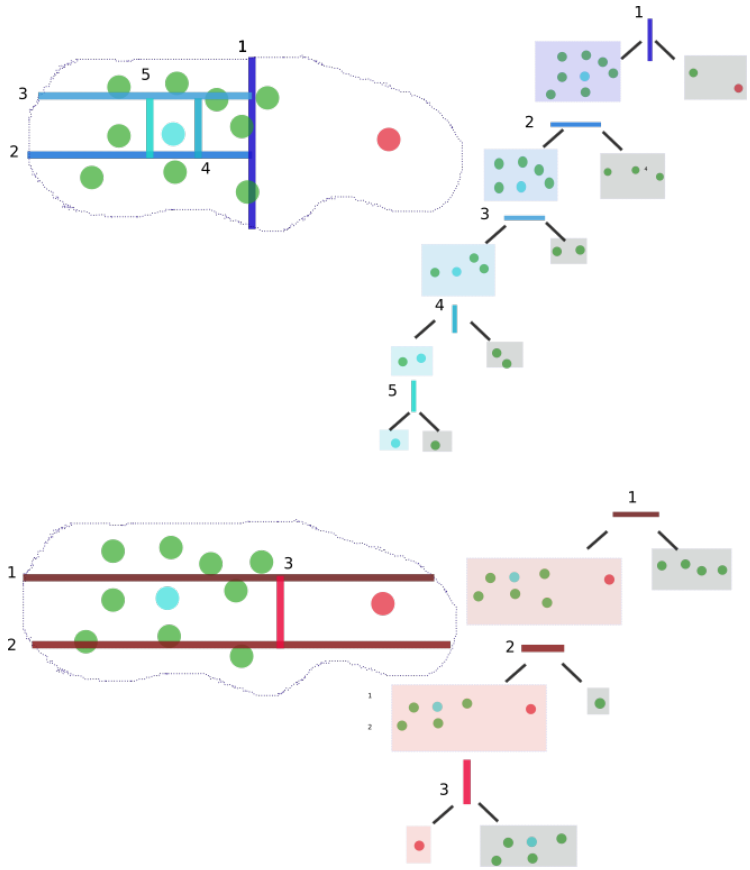


Fig. 2. The difficulty associated to isolating a vector based on random isolation trees is a measure of its anomaly level. The easier a vector is isolated from the rest, the higher its level of anomaly. A vector is isolated when no other vector is contained within the isolated region. The blue vector is harder to isolate than the red one. The process is repeated several times in order to attain a robust measure. The expected number of decisions is taken into account to assign an anomaly level to each vector. Two iterations are shown in the figure.

The closer to 1 is $s(v, N)$, the higher the anomaly level of vector v. The approach followed by *IF* is rather useful since it does not rely on distances, which can be a problem in high-dimensional.

3.3 Support Vector Machines

A support vector machine (*SVM*) can be thought of as an algorithm that maps data into a particularly relevant high-dimensional space. In this mapping space, vectors from two different classes tend to be placed in different regions so that an hyperplane can tell apart the label of the mapped vectors.

SVM constitute an instance of classifiers that map data to a different space so that a linear function can decide the class of the studied vectors [7]. In particular, the hyperplane is placed so that the distance from it to the closest vectors of each class, the support vectors, is maximized. *SVM* map data to the high-dimensional space via a kernel function. This kernel takes as arguments the dot product of the description of each vector. Based on a nice mathematical property derived from Mercer's theorem, the computationally demanding projection to that new space is not explicitly performed.

This *kernel trick* allows the generation of ultra high-dimensional (or infinite-dimensional) spaces in which the decision function can easily classify vectors. The mathematical details of the method, though powerful and highly interesting, are not required its application as anomaly detectors. What is required is the particular method known as one-class support vector machine [25].

In one-class *SVM*, the algorithm is trained with instances of the usual or expected class. The binary function computed by the trained *SVM* will return the same value for all vectors in the training set, which are assumed to belong to the same class, which is the usual or expected class. That value, by convention, is 1. When the trained *SVM* is presented with a vector that does not belong to the same class, that is, which constitutes an anomaly, the decision function returns a -1.

SVM have been successfully applied as anomaly detectors in several contexts. In particular, anomaly detection in time series has proven to be a relevant tool [27]. In [17], SVM are applied to detect anomalies in data obtained by hundreds of sensors in a petroleum facility. The performance of *SVM* is these and many other cases is outstanding.

3.4 Autoencoders

Deep architectures have been successfully applied in several classification tasks [8]. For the anomaly detection problem, in which there is no ground truth about the nature of the observations, an interesting approach comes from the application of autoencoders (*AE*) to detect anomalies in unlabeled data. Trained *AEs* aim to recover the input data at the output layer. The architecture of this type of networks consists of three blocks [11].

The first one is the encoder. In this stage, the usually high-dimensionality of the input space is reduced. This stage constitutes a case of dimensionality reduction, in particular, a non-linear one. The encoder maps input data to a latent space, which constitutes the second block. The latent space is in general of a lower dimension that the input space. It is in this latent space that instances that are anomalies are revealed, since the usual or expected vectors tend to be clustered together, whereas the anomalies tend to form a different cluster [23]. The third block is the decoder, that tries to reconstruct the original or input data from its low-dimensional representation in the latent space.

In an autoencoder, the number of neurons in the input and output layers is the same. In particular, we built an *AE* with two hidden layers, each defined by three neurons. There are several paths to compute anomaly levels in an *AE*. The one we relied on is based on the expected distances in the latent space. By computing a histogram of the expected distances, a decision can be made concerning the cutoff for the discrimination of anomalies and regular or expected vectors. Those vectors with a large distance, compared to the distance shown by the majority, are identified as anomalies.

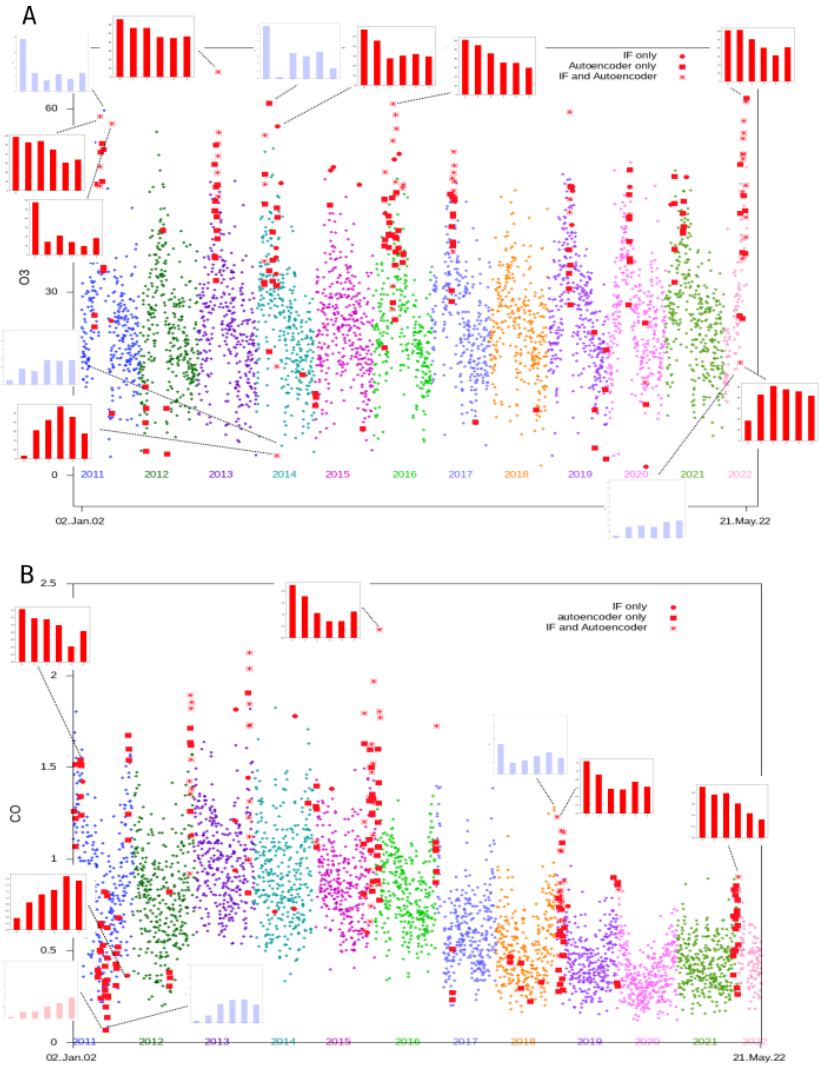


Fig. 3. Anomalies detected in the time series of the average daily concentration of O_3 (A) and CO (B) recorded at the *Tlalnepantla* station. The days detected as anomalies by *IF* only are shown as red filled circles, the days identified as anomalies by the autoencoder only are shown as red squares, and the days identified as anomalies by both methods are shown as stars.

4 Results

From the massive dataset of pollutants, several relevant questions can be answered by relying on machine learning, specifically, in unsupervised learning. The first and most obvious one within this contribution is that of the existence of anomalies. We present in this section some of the results of applying anomaly detection algorithms to the large dataset generated by SIMAT.

Our analysis was conducted focusing on monitoring stations separately in order to disregard the spatial dynamics of air pollution. For each station, we followed two paths of analysis. In the first one, a time series was constructed for each of the monitored pollutants. Instead of detecting changes in consecutive observations, we applied a different approach in order to detect more relevant changes.

For this, a sliding window of size $k = 6$ was applied to the time series in order to embed that point of k coordinates as a point into a $k = 6-$ dimensional space. This embedding is a rather common procedure in anomaly detection of time series [1]. This approach is able to detect relevant patterns in data. Once the time series is embedded as described, the anomaly detection algorithms are applied in the $k-$ dimensional embedding space.

Fig. 3 shows the time series of CO and O_3 for *Tlalnepantla* station from January 1st, 2011 to May 30, 2022. The days detected as anomalies by *IF* or by *AE* are indicated accordingly. It is also shown some of the anomalies as well as some of the expected or usual days. Some days are identified as anomalies by both methods, some others by only one of them, and the majority are not identified as anomalies.

An anomaly in this context is a sequence of six consecutive hours or days, depending on the case, that, in the $k = 6$-dimensional space, does not resemble certain characterizations that are common along the vast majority of the observations. For the IF algorithm, this means, for instance, that the anomalies are rather isolated from the rest of the points in the embedding space since it was easier to isolate it than expected.

For the case of LOF, this means that the sequences detected as anomalies are characterized by neighborhoods that are rather different, in terms of proximity and density, than the neighborhoods of the majority of the vectors. Consecutive observations ,as those observed in 3-A upper left, may differ in nature, that is, one might be an anomaly, and the next one may be an expected observation. Once again, we remind the reader that the algorithm works in the embedding space, not in the time series itself.

In the second approach of anomaly detection, each station is characterized in terms of the concentration of six pollutants: $CO, O_3, NO, NO_2, PM10, SO_2$. That is, for a given station and hour, a point in the six-dimensional space of pollutants is generated. In this approach, anomaly detection algorithms are applied to the points in this six-dimensional space.

Although some sensors suffered occasional problems affecting the records, this does not affect the anomaly detection scheme, since we are not interested in consecutive hours, as is the case for time series analysis. The anomaly detection scheme is applied in the feature space defined by the concentration of the six mentioned pollutants.

Fig. 4 shows the 65,798 recorded hours from January, 2nd 2011 to 5th May 2022 at the *Tlalnepantla* monitoring station. Each hour is linked to a point in the six-dimensional space of pollutant concentration.

It is in that space that the four algorithms are invoked. In 4-A, it is shown, in the y-axis, the ratio of the average pollutant concentration at the corresponding hour and the distance, in the six-dimensional space, from that observation to the next available one. It is also indicated whether the observations at a certain hour were detected as anomalies by one of the four anomaly detection algorithms.

The numbers on top of A indicate the number of instances, per year, that were detected as anomalies by the specified algorithm (label at the right end). The number of usual or expected (non-anomalies) observations is also indicated. The number of observations that were detected as anomalies by one, two, or three anomaly detection algorithms is also displayed.

From fig. 4 it is already available some information that could not be obtained by traditional statistical approaches. As a preamble, 5,822 out of the 65,798 hours were identified as anomalies by at least one algorithm. From the algorithms included in this contribution, SVM is the more stringent one. Only 245 observations were detected as anomalies, and in several years, no anomalies were identified by this method.

Interestingly, though, is that the years 2021 and 2022 are the ones with the highest number of identified anomalies by SVM. This may indicate a change in the dynamics behind the sources of pollution. Indeed, this time frame corresponds to mobility constraints imposed by the government in order to reduce social contact as a policy to reduce *SARS-COV2* contagions. The remaining three methods do not present this change in the number of detected anomalies.

As was already stated in the Introduction, different anomaly detection algorithms make different assumptions in order to identify peculiar or dissonant observations. Fig. 4-B shows a comparison, based on visualization of different categories (a kind of Venn-diagram for several sets) [13], of the intersection among the four anomaly detection algorithms and the non-anomalies in the data. In blue, it is shown that the majority of observations were not detected as anomalies. 8.8% of the observations define the set of anomalies, identified by at least one of the methods.

The autoencoder (AE) is the most sensitive one, as observed by the high number of anomalies detected by it (4,707). *LOF* is the second most sensitive algorithm, as it records 1,181 anomalies. However, the observations detected as anomalies by these two algorithms is rather low, 68 exclusively detected by those two, plus 2 more anomalies detected also by *SVM*. The methods with the highest overlap were *IF* and the autoencoder, with 277 common observations.

In fig. 4-C, it can be seen the expected (typical) observation of the six pollutants detected as usual, or anomalies accordingly to one of the four described algorithms. It is clear the difference between the usual observations (blue) and the anomalies (red). In 4-D, it is shown the distribution of the number of anomalies in the specified hour of the day. The hour with the highest number of anomalies is at 8:00. The reasons of this are still under deeply research, but there is evidence that at this time, the changes in temperature and mobility are rather important factors.

Interestingly, no observation were detected as an anomaly by the four methods. Only 64 four observations were detected as anomalies by at three methods, and the only of such anomalies for 2022 is shown in fig. 4-E. This observation corresponds to March, 22nd. at 9:00.

The points that are anomalous indicate that at a certain hour, the concentration of the six pollutants was rather different to the concentration observed in the majority of points in the six-dimensional space.

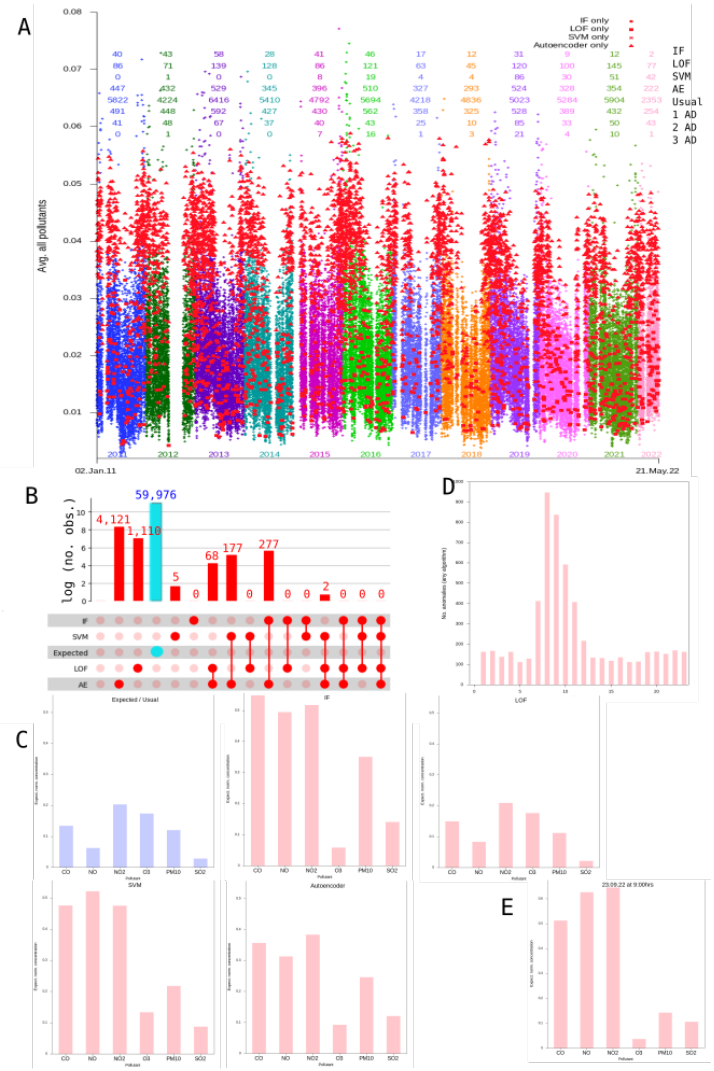


Fig. 4. Anomalies focused on hourly observations at *Tlalnepantla* station, from 02.01.2011 to 21.05.2022. A. Hourly measure of six pollutants. In the y-axis, it is shown the ratio of the average concentration of pollutants at the specified hour and the difference to the set of measurements in the next available hour. It is indicated whether a particular set of observations was identified as an anomaly by any of the four algorithms. It is also shown the number of observations detected by one, two or three algorithms (1AD, 2AD, 3AD). B. UpSet visualization of the intersections among the four anomaly detection algorithms. C. The distribution of anomalies along the 24 hours. The hour with more anomalies in this station was at 8:00. D. The expected concentration (normalized) of the pollutants for each of five cases: usual (expected) observations, detected as anomalies by: *IF, LOF, SVM, AE*. E. The only observation detected as anomaly by three methods during 2022.

5 Discussion and Conclusions

The identification of certain observations that do not resemble the rest of the observations in a dataset is a peculiar, and rather interesting case, of pattern recognition in particular, and of artificial intelligence in general. Although some researchers consider anomaly detection a special case of classification, we stick to a different perspective, in which both tasks are inherently different.

Classification relies on the existence of an assigned label or class to each vector, whereas in anomaly detection, the algorithm has to infer the label for each observation. The label may be either usual or anomalous observation. The second approach for anomaly detection is more complex since the metric to compare observations is unknown and has to be learnt from the existing data. Besides, the criteria to decide whether an observation constitutes an anomaly or not is not unique.

In this contribution, we applied existing anomaly detection algorithms to air pollution data in the Metropolitan Area of Mexico City. The main goal behind our work was to identify non-trivial observations, that is, groups of data from different pollutant sensors, that are rather different to the majority of observations. Those anomalous observations denote special atmospheric circumstances that may indicate transitory changes in the mechanisms and variables that affect the dynamics such as wind, temperature, changes in mobility, among others.

For the case of one station, that of *Tlalnepantla*, the anomaly detection algorithms identified some relevant patterns. For instance, the average concentration of six pollutants of the anomalies detected by the four methods present a wide range. Since in anomaly detection there is no ground truth, it is relevant to capture several possible profiles for some of the possible anomalies.

In particular, we applied Isolation Forests, Local Outlier Factor, support vector machines and autoencoders to the data collected by the Air Quality Authority of The Metropolitan Area of Mexico City. The existing data includes hourly observations of over thirty stations and covering seven different pollutants. We focused our efforts in a subset of the dataset in order to communicate the relevance of applying anomaly detection algorithms to air quality data. To our knowledge, this has not previously been conducted.

Artificial Intelligence tools provide insight into complex phenomena by detecting patterns that otherwise could not be elucidated. In this sense, this contribution describes the use of an instance of unsupervised learning to a complex phenomena, that of air pollution in large metropolitan areas.

Our main conclusion is that the nature of patterns that can be detected by the use of relevant tools is of a subtle nature, and this patterns provide more information to better understand, in this case, the dynamics of air pollution in Mexico City. Several paths are open for future work. For instance, several other anomaly detection algorithms can be applied to the same pollution dataset.

In a more insightful perspective, atmospheric attributes such as pressure, temperature, and humidity may be included in the analysis to gain a broader perspective of the dynamics of pollutants.

Acknowledgments. A.N thanks PAPIIT for the partial support of this research, with grant number IA103921.

References

1. Aguayo, L., Barreto, G.A.: Novelty Detection in Time Series Using Self-Organizing Neural Networks: A Comprehensive Evaluation. Neural Processing Letters, vol. 47, no. 2, pp. 717–744 (2018). DOI: 10.1007/s11063-017-9679-2.
2. Bekesiene, S., Meidute-Kavaliauskiene, I., Vasiliauskiene, V.: Accurate Prediction of Concentration Changes in Ozone as an Air Pollutant by Multiple Linear Regression and Artificial Neural Networks. Mathematics, vol. 9, no. 356, pp. 1–21 (2021). DOI: 10.3390/math9040356.
3. Bekkar, A., Hssina, B., Douzi, S., Douzi, K.: Air-Pollution Prediction in Smart City, Deep Learning Approach. Journal of Big Data, vol. 8, no. 161, pp. 1–21 (2021). DOI: 10.1186/s40537-021-00548-1.
4. Breunig, M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. vol. 29, pp. 93–104 (2000). DOI: 10.1145/335191.335388.
5. Buzsaki, G.: The Brain from Inside out. Oxford University Press, (2019). DOI: 10.1093/oso/9780190905385.001.0001.
6. Comision Ambiental de la Megalopolis: ¿Como se monitorea la calidad del aire en la ZMVM? Gobierno de México (2018)
7. Cristianini, N., Ricci, E.: Support Vector Machines. Encyclopedia of Algorithms, pp. 928–932 (2008). DOI: 10.1007/978-0-387-30162-4_415.
8. Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., Dehmer, M.: An Introductory Review of Deep Learning for Prediction Models with Big Data. Frontiers in Artificial Intelligence, vol. 3, pp. 1–23 (2020). DOI: 10.3389/frai.2020.00004.
9. Harper, A., Croft-Baker, J.: Carbon Monoxide Poisoning: Undetected by Both Patients and Their Doctors. Age and Ageing, vol. 33, no. 2, pp. 105–109 (2004). DOI: 10.1093/ageing/afh038.
10. Hawkins, D.: Identification of Outliers. Monographs on Statistics and Applied Probability, (1980). DOI: 10.1007/978-94-015-3994-4.
11. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. Science, vol. 313, no. 5786, pp. 504–507 (2006). DOI: 10.1126/science.1127647.
12. Instituto Nacional de Ecología y Cambio Climático: Primer catálogo de estaciones de monitoreo atmosférico en México, Instituto Nacional de Ecología (2013)
13. Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., Pfister, H.: UpSet: Visualization of Intersecting Sets. IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 12, pp. 1983–1992 (2014). DOI: 10.1109/TVCG.2014.2346248.
14. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation Forests. In: Eighth IEEE International Conference on Data Mining, pp. 413–422 (2008). DOI: 10.1109/ICDM.2008.17.
15. Luttrell, W.E.: Nitrogen Dioxide. Journal of Chemical Health and Safety, vol. 21, no. 2, pp. 28–30 (2014). DOI: 10.1016/j.jchas.2014.01.008.
16. Markou, M., Singh, S.: Novelty Detection: A Review-Part 1, Statistical Approaches. Signal Processing, vol. 83, no. 12, pp. 2481–2497 (2003). DOI: 10.1016/j.sigpro.2003.07.018.
17. Martí, L., Sanchez-Pi, N., Molina, J.M., Bicharra-García, A.C.: Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. Sensors, vol. 15, no. 2, pp. 2774–2797 (2015). DOI: 10.3390/s150202774.

18. Neme, A., Hernández, L.: Visualizing Patterns in the Air Quality in Mexico City with Self-Organizing Maps. In: Advances in Self-Organizing Maps - 8th International Workshop, vol. 6731, pp. 318–327 (2011). DOI: 10.1007/978-3-642-21566-7_32.
19. NPS: National Park Services Stats (2014)
20. Oufdou, H., Bellanger, L., Bergam, A., Khomsi, K.: Forecasting Daily of Surface Ozone Concentration in the Grand Casablanca Region Using Parametric and Nonparametric Statistical Models. Atmosphere, vol. 12, no. 6, pp. 1–19 (2021). DOI: 10.3390/atmos12060666.
21. Pimentel, M., Clifton, D., Clifton, L., Tarassenko, L.: A Review on Novelty Detection. Signal Processing, vol. 99, pp. 215–249 (2014). DOI: 10.1016/j.sigpro.2013.12.026.
22. Preiss, B.: Data Structures and Algorithms with Object-Oriented Design Patterns in C++. vol. 1 (1999)
23. Sakurada, M., Yairi, T.: Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In: Proceedings of the MLSDA 2nd Workshop on Machine Learning for Sensory Data Analysis, pp. 4–11 (2014). DOI: 10.1145/2689746.2689747.
24. Salladay, S.: Right to Know. Unexpcted Diagnosis . Nursing, vol. 27, no. 11, pp. 22–24 (1997). DOI: 10.1097/00152193-199711000-00013.
25. Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J.: Support Vector Method for Novelty Detection. Advances in Neural Information Processing Systems, pp. 582–588 (2000)
26. United States Environmental Protection Agency: Particulate Matter (PM) Basics (2021)
27. Yokkampon, U., Chumkamon, S., Mowshowitz, A., Fujisawa, R., Hayashi, E.: Anomaly Detection Using Support Vector Machines for Time Series Data. Journal of Robotics, Networking and Artificial Life, vol. 8, no. 1, pp. 41–46 (2021). DOI: 10.2991/jrnal.k.210521.010.
28. Zimek, A., Schubert, E., Kriegel, P.: A Survey on Unsupervised Outlier Detection in High-Dimensional Numerical Data. Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 5, no. 5, pp. 363–387 (2012). DOI: 10.1002/sam.11161.

ISSN 1870-4069

Sequential Frequent Patterns for a DNA Sequence Using Mapping and Mining Techniques

Luis Heriberto García-Islas, Anilu Franco-Arcega, Víctor Ignacio Sobrevilla-Solís, Esteban Rueda-Soriano, Kristell Daniella Franco-Sánchez

Universidad Autónoma del Estado de Hidalgo,
Instituto de Ciencias Básicas e Ingeniería,
Área Académica de Computación y Electrónica,
Mexico

{luishg, afranco, so314246, estebanrs, kristell_franco}@uaeh.edu.mx

Abstract. Biological sequences contain a significant amount of genetic information from living organisms. The analysis of these sequences can provide information that might help biologists better understand them. The discovery of frequent patterns from a specific DNA sequence has become one of the greatest challenges in the application of data mining techniques. This is especially true for those sequences whose length is extensive and/or the number of frequent patterns generated exceeds the time needed to fully reveal themselves. There is a considerable time and effort for obtaining sequential frequent patterns when the methods utilized are based on Apriori algorithms such as GSP or Key-segment. However, these methods can be enhanced and improved. In this paper, we propose a sequence mapping-based algorithm designed to improve the search for contiguous frequent patterns in a single DNA sequence. Our experiment was applied over 11,230 DNA real different sequences with lengths between 118 and 52,255 nucleotides, obtained from a real biological database. This experiment demonstrated a faster algorithm for frequent pattern mining on DNA sequences compared with other related algorithms.

Keywords: Data mining, sequential pattern mining, frequent contiguous patterns, DNA sequences, bioinformatics.

1 Introduction

Large DNA sequences are often composed of several short frequent subsequences that play important functional or structural roles in a living organism [17]. One way to study this is through Frequent Pattern matching [1], which has become a relevant area of research for bioinformatics [4-8, 12, 13, 19-22]. Even though in recent years, several approaches to identify frequent patterns on sequences have been proposed [2, 14], some of them must scan the sequence multiple times to obtain even only the subsequences frequency, which increases the computational effort to read the sequences.

To reduce this complexity, other algorithms perform non-exhaustive searches because they are only able to find fixed-length frequent patterns [9, 15]. However, there is no guarantee that major additional relevant frequent patterns can be found.

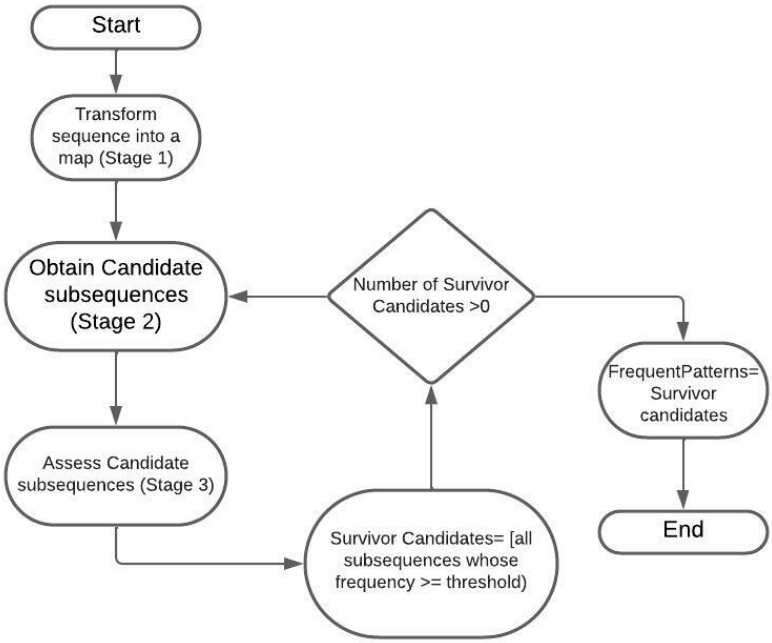


Fig. 1. General approach for proposed frequent pattern algorithm.

Finally, some algorithms can only validate whether a subsequence is a frequent pattern or not [10].

For the algorithms that perform exhaustive searches, there are the classic algorithms such as PrefixSpan, Spade, SPAM, and GSP [2] which perform searches using algorithms such as Naive, KMP or Boyer Moore [11]. New approaches have appeared and demonstrate better performance than classic algorithms, such as Key-segment introduced by Mao [14], which allows the use of a compact data structure to be retained in memory resulting from sequences scanning.

The Mao's algorithm is based on GSP [3] and is used to mine key segments from long DNA sequences. This method uses an exhaustive search to identify frequent patterns, thus the results are more effective than more classic techniques. However, this operation still consumes considerable time to perform its analysis since the sequence must be scanned every time in accordance with the number of occurrences in order to obtain each frequent subsequence.

Despite several researches design to determine frequent patterns, using enhanced algorithms to determine variable length frequent DNA sequence patterns, a significant challenge remains. In this paper, we present an algorithm where the frequent subsequent nucleotide sequences can be identified within a single DNA sequence using a novel mapping technique.

Obtaining these kind of patterns in a single DNA sequence is important because it can establish the basis for discovering more complex behaviors, such as motifs. The most frequent subsequence patterns with variable length will result as output for this

Algorithm 1 Map generation

INPUT: sequence - sequence to be mapped
OUTPUT: map - the obtained map

```
1: function GENERATEMAP(sequence)
2:     for each nucleotide in sequence do
3:         map[char][lastItem] ← [Pos, nextChar]
4:     end for
       return map
5: end function
```

proposed method. The emphasis on finding repeated nucleotide subsequences with different sizes promises to have a significant and positive impact in many fields of genetics and bioinformatics.

The remainder of the paper is organized as follows. Section 2 introduces the proposed algorithm and describes in detail each stage. The experimental results and a comparison with other related algorithms are discussed in section 3. Finally, in section 4 conclusions are presented.

2 Proposed Algorithm

To identify frequent patterns in a single DNA sequence, the proposed algorithm is formed by three stages: sequence mapping, candidate subsequences generation and the assessment for those candidate subsequences. Figure 1 presents these stages. The first stage creates a map from the sequence that can be utilized to perform a fast search to obtain frequencies of subsequences.

Then, the stages that follow will iterate the frequent patterns. In these iterations, the generation of candidates is performed (stage 2), and the process follows by obtaining their number of occurrences (stage 3). During these iterations, all those candidate subsequences whose occurrence numbers are greater or equal to an established threshold will be used as a source to create new candidate subsequences.

Stages 2 and 3 will be iterated until the number of candidates, that fulfill the threshold condition, becomes zero.

2.1 Sequence Mapping

As part of the proposed process, the first step consists of transforming the DNA sequence into a map represented as a table. In this tabular abstraction, the rows represent each different element in the sequence, i.e. a row for each nucleotide (A, C, G, T).

Every row stores a set of pairs formed as following: (Pos, nextChar), where Pos represents the position of the nucleotide within the sequence and nextChar represents the next element. The Algorithm 1 shows the process of creating the map.

The resulting map will be used to obtain the frequency for every possible candidate that will be generated in next stage.

Algorithm 2 Candidate generation

INPUT: CandidateSubsequences - the set of candidates that will be used as base to generate new ones
OUTPUT: NewCandidates - the obtained new candidates

```
1: function GENERATECANDIDATES(CandidateSubsequences)
2:     alphabet ← [A, C, G, T]
3:     for each CandidateSubsequence do
4:         for each letter in alphabet do
5:             NewCandidate ← CandidateSubsequence + letter
6:             if NewCandidate[2:length(NewCandidate)] exists in CandidateSubsequences
   then
7:                 NewCandidates ⇒ append(NewCandidate)
8:             end if
9:         end for
10:    end for
       return NewCandidates
11: end function
```

2.2 Candidate Subsequence Generation

This stage is performed in two steps: the initial stage and the followed by iterative candidate generation. The first occurs right after stage 1 is concluded and requires the initial candidate generation, which consists of creating 2^4 2-length candidates by using the nucleotides alphabet.

The second will be an iterative process. For each iteration i, the generation process uses the n-length survivor candidates of iteration *i-1*, which are obtained in stage 3, in order to create new ones.

For each survivor candidate, this step will generate (i+1)-length possible new candidates by adding all of the chars from the $alphabet$, to each one, i.e. if "AA" is a survivor candidate, four new candidates will be created as {"AAA","AAC","AAG","AAT"}. Then, every possible new candidate will be evaluated by using anti-monotone property of support [18] which is applied to avoid generating unnecessary candidates.

The process of generating subsequence candidates can be observed in Algorithm 2.

2.3 Candidate Subsequence Assessment

Once all new candidates have been created, the next stage will consist of the candidate subsequences assessment. To perform this, a novel approach for obtaining their frequency is proposed. This can be observed in Algorithm 3 where a process is employed to locate all pairs $(pos, nextChar)$ in the map, traveling the row corresponding with every char from candidate subsequence to calculate how many times they appear in the sequence.

This number will represent the number of occurrences, i.e. the frequency for each candidate subsequence. Then, the next step is to create the set of survivor candidates to be used in the next iteration. A candidate survives if $f_{support}(candidate\_sequence) \geq Threshold$. When this set is empty, the algorithm has completed its cycle.

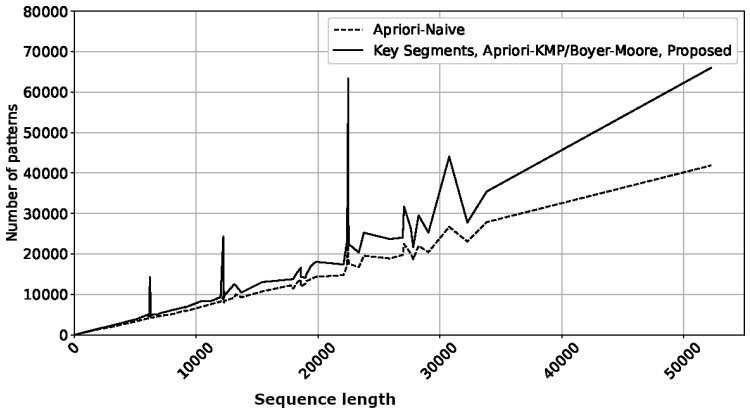


Fig. 2. Sequence length and number of obtained contiguous patterns.

3 Experiments and Performance Evaluation

Some experiments are shown to validate the performance of the proposed algorithm. The algorithm was tested over 11,230 sequences of different lengths between 118 and 52,255 nucleotides from 550 organisms, for example Chikungunya virus, Cactus, Xenopus laevis, among others. These sequences were downloaded from the NCBI repository [16].

We compared our proposal with algorithms based on Apriori using different search methods (Naive, KMP and Boyer-Moore), and with key-segment algorithm. The reason to use these algorithms is because they can be used to obtain frequent patterns contained in only one sequence, unlike other algorithms, such as fp-tree based algorithms that requires a set of sequences to obtain frequent patterns. All of the algorithms were programmed with Python 2.7 and for these experiments it was considered the threshold with a value of 2, because it obtains the whole set of frequent patterns. If the threshold increases his value then the length of the set of frequent patterns will be decreased.

The experimental results show that the proposed algorithm obtained the same amount of patterns than Apriori-KMP, Apriori-Boyer Moore and Key-segments for the 100% of the tested sequences as it can be seen on Figure 2. In particular, Apriori-Naive obtains less frequent patterns than the other algorithms. The reason of this is because it doesn't consider when patterns with same nucleotides appears on contiguous elements, i.e. pattern "AA" on sequence "AAATC", for Naive algorithm, has a frequency of one instead of two.

The efficacy of the proposed algorithm resides in the identification and utilization of a novel method to obtain frequent patterns via a structured mapping search, which consumes less processing time than related algorithms.

The number of patterns for all cases is the same, except for Apriori-Naive but, the major difference is in the processing time needed to obtain them. Table 1 indicates the processing time required by the tested algorithms, our proposed algorithm evidenced the fastest processing times. In the longest sequence tested, there were significant

Algorithm 3 Frequency obtaining

INPUT:

- pattern: whose frequency will be obtained.
- StartPosition: allows the method to find the path of search on a defined position of the pattern.
- positionsArray: enables an positions array whose sequence[pos] are part of the pattern

OUTPUT: length(pos) is the obtained frequency of the candidate to be assessed and pos is the array containing the positions of the substrings corresponding with the assessed pattern

```
1: function GETFREQUENCY(pattern, startPosition, positionsArray)
2:     frequency ← 0
3:     posAnt ← empty        ▷ Represents the array of positions of the previous iteration whose sequence[Pos] are part of the pattern
4:     pos ← empty           ▷ Represents the array of positions on the current iteration whose sequence[pos] are part of the pattern
5:     if length(pos) > 0 then        ▷ this initial iteration allows identify which positions in the selected row fits with the input pattern
6:         currentChar ← pattern[0]
7:         nextChar ← pattern[1]
8:         row ← row of currentChar in map
9:         pos ← row[x][1]∀x[2] = pattern[2]        ▷ Gets all the pairs[pos,nextChar] whose nextChar=pattern[1]
10:        indexPattern ← 2
11:    else        ▷ There is a sub pattern and it will be used as an start to complement
12:        pos ← positionsArray
13:        indexPattern = startPosition − 1
14:    end if
15:    while length(pos) > 0 do
16:        newPos ← empty        ▷ Represents the positions identified whose sequence[pos] are part of the pattern
17:        for each position in pos do
18:            if [position+1,pattern[indexPattern] exists in map[row] then
19:                newPos → append(position)
20:            end if
21:        end for
22:        pos ← newPos
23:        indexPattern ← indexPattern + 1
24:    end while
       return length(pos), pos
25: end function
```

improvements in the processing of 42.7, 46.9, 22.7 and 53.3 times faster compared with Apriori-Naive, Apriori-KMP, Apriori-Boyer Moore and Key Segments, respectively.

The execution time for all the algorithms with the complete set of sequences is exhibited in Figure 3. The average process improvement times of our algorithm were 46.16, 43.75, 24.95 and 39.16 over Apriori-Naive, Apriori-KMP, Apriori-Boyer Moore and Key-segments, respectively.

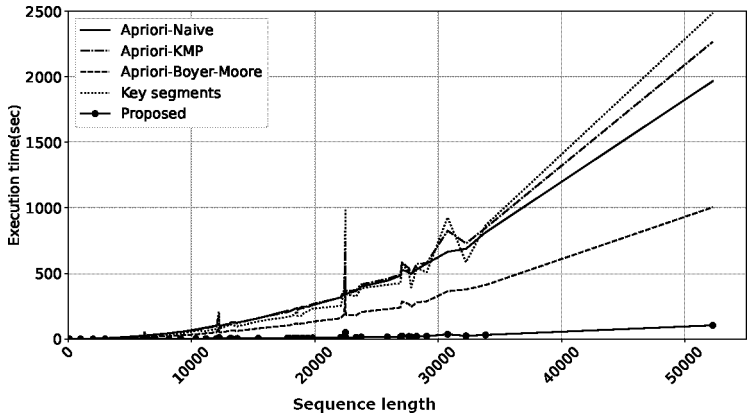


Fig. 3. Processing time to obtain frequent patterns for different length sequences.

As we can see in the previous figure, the maximum improvement for each algorithm was of 59.59, 72.24, 41.34 and 98.53 (on the same order of algorithms). This indicates a significant enhancement when the proposed algorithm is applied. Furthermore, the behavior for the other algorithms presents an irregular increment in execution time for several sequences.

This behavior is related to the number of patterns that a sequence contains combined with its length, as indicated in Table 1. This table only shows a summary of the whole set of experiments performed, there are 14 of 11,230 executions. The summary of Table 1 demonstrates that while larger is the length of the DNA sequence more execution time requires to complete the discovery of patterns.

The proposed algorithm presents a minimum variation of the execution time even when these cases are processed. This means that execution time for our algorithm is not related to the number of patterns but to the sequence length, while the time for the comparison algorithms is associated with both, the length and number of patterns contained in the sequence.

4 Conclusions

In this paper, a novel algorithm to improve frequent pattern generation for a single DNA sequence is proposed. Greater speed and efficacy are obtained through transformation of the sequence by way of a map, and then applying the map in combination with the well known anti-monotone property of support to obtain these frequent patterns.

In addition, a new way to search the number of occurrences for a subsequence using the created map is proposed. Such map is obtained by scanning the entire DNA sequence only once with the application of the algorithm.

Then, it allows the identification of the number of occurrences of a subsequence into a DNA sequence using the improved proposed search, avoiding the need to scan the entire sequence and even the entire subsequence every time it is evaluated. In addition, the use of the anti-monotone property reduces the number of possible frequent subsequence candidates and hence, the number of iterations.

Table 1. Summary of execution time obtained through experimental results with different lengths DNA sequences, in seconds.

Sequence ID	length	Apriori-Naive	Apriori KMP	Apriori-BoyerMoore	Key Segment	Proposed
HL714398	118	0.0159	0.0150	0.0160	0.0160	**0.0016**
NM_007169	1008	1.0160	0.8910	0.5779	0.6099	**0.0460**
NM_053477	2020	4.0900	3.5320	2.2260	2.6570	**0.1719**
NM_001322998	3000	9.2190	8.1139	4.8289	5.7829	**0.3280**
NM_022009	4071	11.4390	0.5160	16.7109	14.4429	**8.6890**
XM_011544263	5103	25.2250	22.9159	13.6380	17.679	**0.6879**
NM_001310478	6037	38.0969	33.4040	18.5870	27.5810	**1.5469**
NM_001102653	7068	48.6870	43.6400	25.8659	32.9810	**1.1870**
NM_002763	8178	65.7319	58.5820	34.5220	45.8090	**1.4220**
NR_133925	9096	80.4949	72.8770	41.7019	54.0269	**1.7500**
NG_029868	10355	101.7990	93.2639	53.5950	75.2279	**2.1099**
NG_013266	22093	488.1619	440.3600	246.3190	365.9060	**9.2349**
NG_011731	30767	941.9620	1100.0530	532.6480	1451.1100	**19.2990**
NG_008111	52255	2735.4070	3007.2460	1456.9200	3413.6280	**64.4760**

Execution time is reduced as well. Experimental results confirm enhanced performance when our algorithm is applied in contrast to others that process only one DNA sequence. A major finding of this study is that the performance of comparative algorithms relies upon both, length and number of patterns obtained from a sequence, while our algorithm does not, regardless of the number of patterns. According to the experiments performance, it can be established that the proposed algorithm is faster than compared algorithms. Furthermore, frequent pattern studies can be enhanced by using this proposed method.

References

1. Aggarwal, C.C.: Data mining: The Textbook. Springer International Publishing, vol. 1, pp. 1–734 (2015). DOI: 10.1007/978-3-319-14142-8.
2. Aggarwal, C.C., Han, J.: Frequent Pattern Mining, Springer Cham (2014). DOI: 10.1007/978-3-319-07821-2_1.
3. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, vol. 1215, pp. 487–499 (1994)
4. Azmi, A.M., Al-Ssulami, A.M.: Discovering Common Recurrent Patterns in Multiple Stringsover Large Alphabets. Pattern Recognition Letters, vol. 54, pp. 75–81 (2015). DOI: 10.1016/j.patrec.2014.12.009.
5. Beernaerts, J., Debever, E., Lenoir, M., Baets, B.D., de-Weghe, N.V.: A Method Based on the Levenshtein Distance Metric for the Comparison of Multiple Movement Patterns Described by Matrix Sequences of Different Length. Expert Systems with Applications, vol. 115, pp. 373–385 (2019). DOI: 10.1016/j.eswa.2018.07.076.
6. Bustio-Martínez, L., Muñoz-Briseño, A., Cumplido, R., Hernández-León, R., Feregrino-Uribe, C.: A Novel Multi-Core Algorithm for Frequent Itemsets Mining in Data Streams. Pattern Recognition Letters, vol. 125, pp. 241–248 (2019). DOI: 10.1016/j.patrec.2019.05.003.
7. Chanda, A.K., Ahmed, C.F., Samiullah, M., Leung, C.K.: A New Framework for Mining Weighted Periodic Patterns in Time Series Databases. Expert Systems with Applications, vol. 79, pp. 207–224 (2017). DOI: 10.1016/j.eswa.2017.02.028.

8. Danger, R., Pla, F., Molina, A., Rosso, P.: Towards a Protein–Protein Interaction Information Extraction System: Recognizing Named Entities. Knowledge-Based Systems, vol. 57, pp. 104–118 (2014). DOI: 10.1016/j.knosys.2013.12.010.
9. Devikarubi, R., Rubi, R.D., Arockiam, L.: IndexedFCP - An Index Based Approach to Identify Frequent Contiguous Patterns (FCP) in Big Data. In: International Conference on Intelligent Computing Applications, pp. 27–31 (2014). DOI: 10.1109/ICICA.2014.15.
10. Gureja, V., Sharma, N., Sharma, A.: An Optimized Tabular Structure Based Pattern Search Over DNA String. In: International Conference on Soft Computing Techniques and Implementations, pp. 72–76 (2015). DOI: 10.1109/ICSCTI.2015.7489540.
11. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, (1997). DOI: 10.1017/CBO9780511574931.
12. Kieu, T., Vo, B., Le, T., Deng, Z.H., Le, B.: Mining Top-k Co-Occurrence Items with Sequential Pattern. Expert Systems with Applications, vol. 85, pp. 123–133 (2017). DOI: 10.1016/j.eswa.2017.05.021.
13. Maestre-Vidal, J., Sotelo-Monge, M.A., García-Villalba, L.J.: A Novel Pattern Recognition System for Detecting Android Malware by Analyzing Suspicious Boot Sequences. Knowledge-Based Systems, vol. 150, pp. 198–217 (2018). DOI: 10.1016/j.knosys.2018.03.018.
14. Mao, G.: An Efficient Mining Algorithm for Key Segment from DNA Sequences. In: 28th Canadian Conference on Electrical and Computer Engineering, pp. 396–399 (2015). DOI: 10.1109/CCECE.2015.7129310.
15. Mutakabbir, K.M., Mahin, S.S., Hasan, M.A.: Mining Frequent Pattern Within a Genetic Sequence Using Unique Pattern Indexing and Mapping Techniques. In: International Conference on Informatics, Electronics and Vision, pp. 1–5 (2014). DOI: 10.1109/ICIEV.2014.6850729.
16. National Center of Biotechnology Information: Home / nucleotide / NCBI (2018)
17. Snustad, D., Simmons, M.: Principles of Genetics. Wiley (2015)
18. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison-Wesley (2005)
19. Tastan-Bishop, O.: Bioinformatics and Data Analysis in Microbiology. Caister Academic Press (2014)
20. Tozammel-Hossain, K.S., Patnaik, D., Laxman, S., Jain, P., Bailey-Kellogg, C., Ramakrishnan, N.: Improved Multiple Sequence Alignments Using Coupled Pattern Mining. IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 10, no. 5, pp. 1098–1112 (2013). DOI: 10.1109/TCBB.2013.36.
21. Wang, Q., Davis, D.N., Ren, J.: Mining Frequent Biological Sequences Based on Bitmap Without Candidate Sequence Generation. Computers in Biology and Medicine, vol. 69, pp. 152–157 (2016). DOI: 10.1016/j.compbiomed.2015.12.016.
22. Zhang, J., Wang, Y., Zhang, C., Shi, Y.: Mining Contiguous Sequential Generators in Biological Sequences. IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 13, no. 5, pp. 855–867 (2016). DOI: 10.1109/TCBB.2015.2495132.

ISSN 1870-4069

Physicochemical Compatible Motifs in Proteins Sequences

Jesús Alberto Correa-Morales[1], Eduardo M. Martin[2], Eunice Esther Ponce de León-Sentí[2], Rogelio Salinas-Gutiérrez[3]

[1] Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Maestría en Ciencias con opciones a la Computación, Mexico

[2] Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Departamento de Ciencias de la Computación, Mexico

[3] Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, Departamento de Estadística, Mexico

al205239@edu.uaa.mx, {emmartin, eponce, rsalinas}@correo.uaa.mx

Abstract. The identification of tridimensional motifs, folding arrangements in protein structure has been a strenuous task. Recently *ab initio* protein structure can be elucidated by computational intelligence algorithms, though it still time consuming and poses new problems. Until enough protein structures have been solved, the identification and classification of tridimensional motifs will remain an arduous task. Therefore, it still important to rely on approaches that are independent of tridimensional information. A methodology that uses only the psychochemical properties of amino acid pairing is here described. This methodology work independently from structural data, maximizing the physicochemical compatibility among amino acid pairs. Therefore, it is far easier to implement, and results can be obtained in a shorter time. This approach returns sequence pairs with high compatibility, which might be part of a protein motif. These can aid in the identification and classification of protein subsequences.

Keywords: Optimization, motifs, proteins, genetic algorithm, psychochemical compatibility.

1 Introduction

Proteins are polymers of amino acids joined by a covalent bond known as peptide bond. Twenty different amino acids are commonly found in proteins, each amino acid has amino and a carboxyl group as well as a side chain name a radical.

The radical of each amino acid vary in structure, size, charge, and hydropathy which confer each amino acid of specific properties. A typical protein has between 150 and up

S	$C(S)$	W	
		w	c
	A:10020000000		
	C:00000000090		
	D:00020001000		
TTFEIPQNVCV	E:00050000000		
LPPAPQVNKCN	F:00100100000	TGVCV	66.67
LPPAPQVNKCN	G:00000005000	VTGVCV	61.11
PNTEVFTDVCV	I:00003200200	TEVVTGVCV	59.26
PNTEVVTGVCV	K:00000000200	VVTGVCV	58.73
PNTEVVTGICV	L:20000020000	PNTEVVTGVCV	58.59
PNTEVVTGICV	N:06000003002	EVVTGVCV	58.33
ANTDIILGVCV	P:42202100000	NTEVV	53.33
SNTDIILGVCV	Q:00000210000		
	S:10000000000		
	T:11600040000		
	V:00004320507		

Fig. 1. This an example of how the search for conserved sequences (**SCS**) algorithm works. From a multiple alignment of sequences S, a consensus matrix is acquired $C(S)$ (for reference only the amino acids that are present in S are shown here), from $C(S)$ a set of conserved sequences W of variable length are order by decreasing c value.

to a few thousand amino acids, the amino acidic composition of a protein will confer it of specific structural and functional properties. Interactions among amino acids from the same protein lead to a specific folding, hence structure and function.

To understand the complexity of protein structure and function, protein's tridimensional structure has been further subdivided, motifs being folding arrangement identifiable as substructures. Therefore, knowing the interprotein interactions among amino acids that conform motifs can provide important information for a given protein [2, 7, 9, 10].

Genetic algorithms have extensive applications in optimizing combinatorial problems [4], the strength of this algorithm reside in its exploration power and capability to escape from a local optimum. The most important element when trying to solve an optimization problem with a genetic algorithm is the modelling of the individual [11], the main points to be considered when dealing with this kind of algorithms is the design of the objective function, the solution's modelling, and the population's conformation. The use of genetic algorithms in the search for motifs has already been explored, though this has been made mainly in DNA sequences working with nucleic acids [5].

The aim of the present is to describe a methodology composed of three stages, which will allow the user to identify pairs of sequence that are mutually compatible in the protein sequence, which might be conforming a tridimensional structural motif.

Algorithm 1 Pseudocode for multiple point mimicry

Input: $selected\_population$
Output: $cross\_population$

$cross\_population$ is initialized as an empty list
for assign $i \leftarrow 1$ every second individual until P **do**
  if CP is greater than a random number between 0 to 100 **then**
    generates four different random numbers between 0 and Lv
    random numbers are saved in $points$
    order the numbers in increasing order
    assign $aux \leftarrow 1$
    while $individual_i$ is equal to $individual_{i+aux}$ **do**
      if aux is greater than or equal to P **then**
        $aux \leftarrow -1$
      increment the value of aux by one
    $child_1$ is made up of segments of $individual_i$, $individual_{i+aux}$, and $individual_i$ defined by the numbers in the even positions of $points$
    $child_2$ is made up of segments of $individual_{i+aux}$, $individual_i$, and $individual_{i+aux}$ defined by the numbers in the odd positions of $points$
    $child_1$ adds to $cross\_population$
    $child_2$ adds to $cross\_population$
return $cross\_population$

2 Methodology

To identify compatible motifs based on amino acid pairing physicochemical properties a three-phase approach is devised, each with a specific objective. The output will be a set of highly compatible pair of sequences an its position in a multiple sequence alignment, these pairs will consist of a highly conserved sequence and a sequence with high homology to an artificially generated sequence.

2.1 First Stage: Search for Conserved Sequences (SCS)

Objective. Identify a set of highly conserved sequences W in a multiple sequence alignment S.
Preprocessing. A set of homologue sequences of a protein of interest in a fasta format file (.faa, .fasta), these will be aligned using Clustal Ω a tool for multiple sequence alignment [6, 8, 13, 14]. A fasta file with all sequences aligned will be acquired and afterwards used as input for the first phase of the methodology.
Input data. File in fasta format (.faa, .fasta).
Processing. The file is read, and the alignment's information is stored in $S_{D\bullet l} = \left\{S^i \middle| i = 1, 2, 3, \ldots, D\right\}$, where D represents the number of sequences, l the length of the sequences, $S^i = \left(S^i_j \middle| j = 1, 2, 3, \cdots, l\right)$ i^{th} protein sequence, and j^{th} the alignment's column. The consensus matrix $C(S)$ stores the count for each amino acid by column for each S_j, $C(S) = \left(C(S)^1, C(S)^2, C(S)^3, \ldots, C(S)^L\right)$ where L represents the total length of the alignment, for $C(S)^L = \left\{(S)^L_b \middle| b \in B\right\}$ B

w_i	wa_i	$\alpha SCM(w_i, wa_i)$	$\beta CCM(w_i, wa_i)$	$\gamma HCM(w_i, wa_i)$	$F(w_i, wa_i)$
TGVCV	SGVGV	15.212	30.423	30.423	76.058
VTGVCV	VSGVGV	15.181	30.363	30.363	75.907
TEVVTGVCV	SKVVSGVGV	15.133	30.266	30.266	75.665
VVTGVCV	VVSGVGV	15.160	30.320	30.320	75.799
PNTEVVTGVCV	PNSKVVSGVGV	15.061	30.123	30.123	75.307
EVVTGVCV	KVVSGVGV	15.176	30.351	30.351	75.878
NTEVV	NSKVV	14.878	29.757	29.757	74.392

Fig. 2. Schematic of the txt file containing the condensed results from the ASG stag. Only the best artificial coupling sequence is shown for each highly conserved sequence.

represents the amino acids with its one letter code, $B = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$.

To obtain highly conserved sequences, the length of the substring is defined as Lv which must comply with the following conditions a minimum length of 10 Lvm, a maximum length of 50 LvM, and a conservation value $c = \sum_{j=n}^{m} C(S)^j$ where $n = L^i$ is any position in the alignment $i = 1, 2, \ldots, L - Lvm$ and $m = n + Lv$. The subsequence's length Lv is assigned automatedly using the following approach. Initially the $C(S)$ matrix is iterated from the first column L until $L - Lvm$ with a window equal to Lvm, all resulting subsequences are sorted decreasingly in an array according to its c value.

A subsequence is chosen orderly from this array adding an extra position to the right of the subsequence according to $C(S)$. The new subsequence must comply to the criterion $\Delta c > (5 + (0.02 * Lv))$ where $\Delta c = c_{Lv+x} - c_{Lv}$ and $x = 1, 2, \ldots, 5$. The algorithm continues adding a new position until the criterion is not met or $Lv = LvM$. Finally, a set of several subsequences of size W with varying Lv length and c values will be obtained, order in a decreasing c value fashion where $W = \{w_i | i = 1, 2, 3, \ldots, k\}$ and $w_i = (w_i^p | p_i = \{Lv\})$ p is the number of positions in the sequence w_i with a variable Lv length. Some of the sequences from the set W will be used as input in Subsection 2.2.

Objective Function. $c = \sum_{i=n}^{m} C(S)^i$ where $n = L^i$ is any position in the alignment $i = 1, 2, \ldots, L - Lvm$ and $m = n + Lv$. Obtain a set of sequences with a conservation value c.
Output data. A txt file that stores W in descending order with its c value (Fig. 1.)

2.2 Second Stage: Artificial Sequence Generation (ASG)

Objective. Generate an artificial sequence for each sequence from a subset of W, to obtain a set WA of artificial sequences maximized by its physicochemical compatibility.
Input Data. W, population size, maximum number of generations, crossover probability, mutation probability, and the weight for the parameter in the objective function physicochemical compatibility.
Processing. From W a subset $SW = (w_i | i = 1, 2, \ldots, s)$ of sequences is retrieved, where s is the number of sequences to be retrieved from W and s is defined by the user. The first s sequences from W are retrieved, which will have the best c values.

Algorithm 2 Pseudocode for biological standard mutation

Input: $cross\_population$
Output: $mutated\_population$

$mutated\_population \leftarrow cross\_population$
can_muta generates random number between 1 and P equal to the number of individual to be mutated
these random numbers are saved in ind
for each x contained in ind **do**
    generate a random number between 1 and Lv
    save the random number in $genome$
    Generate a random number between 0 and 1
    Save the random number in $type$
    if $type$ is in the range [0, 0.6) **then**
        mutation by substitution
    if $type$ is in the range [0.6, 0.8) **then**
        mutation by insertion
    if $type$ is in the range [0.8, 1] **then**
        mutation by deletion
    the mutated individual is deposited in $mutated\_population$
return $mutated\_population$

An artificial will be generated for each sequence in SW, this generation of artificial sequences is done by a genetic algorithm (GA) [4]. The GA must be initialized with the following parameters population size P, maximum number of generations G, crossover probability CP, mutation probability MP, and the weight for the parameter in the objective function physicochemical compatibility (PCC).

The GA will repeat the following stages selection, crossover, mutation, passing, and new generation, until the given number of generations previously assigned has been reached. The objective function to maximize the values of these three physicochemical properties with the previously assign weights. The values used to evaluate a given amino acid pair compatibility come from three different matrices: size (SCM), charge (CCM), and hydropathicity (HCM), these are symmetrical matrices that give a compatibility value to each amino acid pair [1].

For each sequence w_i in SW the GA starts with a population of size P of artificial sequences of length $AL = Lv_i$ where i correspond to i in w_i, the sequence composition is generated randomly from B adding a single amino acid until AL length is reached. A set of size P sequences known as WA is acquired, each sequence in WA will be evaluated by the objective function. The selection method can be done by several option such as tournament, roulette wheel, elitism, truncation, or stochastic universal sampling, any one can be used though roulette is chosen by default.

Once the individuals have been selected the crossing process is initiated, also here several option for the crossing can be chosen. The default crossing method is multiple point mimicry, though multiple point, single point, and uniform crossing methods can be selected. The multiple point mimicry method allows a pair of individuals to mimic the composition of the other one, to create two different children with a mixed composition (Algorithm 1).

wa_i	L_i	SP_i
SGVGV	5	17.778
VSGVGV	4	22.222
SKVVSGVGV	1	14.815
VVSGVGV	3	19.048
PNSKVVSGVGV	0	21.212
KVVSGVGV	2	16.667
NSKVV	1	28.889

Fig. 3. Example of the results of the SSA phase that will be stored in a txt.

For the mutation method biological standard mutation is the default option, uniform and standard methods can also be chosen. In the biological standard mutation method, an amino acid mutation propensity matrix is used, this matrix is built considering all single nucleotide insertion, deletion, and substitution that each amino acid coding codon can experiment for each of its three positions. Therefore, each amino acid can only be replaced by an amino acid whose propensity is higher than zero.

Hence, restricting the sequence to a strict evolutionary path that is biologically sound (Algorithm 2). Once this stage is over each individual will be tested with the objective function allowing to make a passing in which all parent sequences will be replace by its offspring, besides the best individual from the parental sequences will be kept if there is no offspring sequence with a better value.

These will breed a new generation of solutions each time the cycle is repeated and on the final generation a population of artificial coupling sequences $WA_f = (wa_i|1, 2, \ldots, P)$ with high compatibility values will be produced for each of the highly conserved sequences evaluated w_i. This set WA_f is saved as txt file and the best wa_i from the set is chosen as result and input for 2.3.

Objective Function. $PCC = max\left(\sum_{j=1}^{AL} F\left(w_i^{Lv_i}, wa_i^{AL}\right)\right)$ where F is an aggregation function of the form $F(a,b) = \alpha SCM(a,b) + \beta CCM(a,b) + \gamma HCM(a,b)$ where $\alpha, \beta, \gamma \in [0,1]$, $\alpha + \beta + \gamma = 1$, SCM, CCM, and HCM are compatibility matrices for each pair of amino acids [1].
Output Data. A txt file containing the following information per column w_i, wa_i, $\alpha SCM(w_i, wa_i)$, $\beta CCM(w_i, wa_i)$, $\gamma HCM(w_i, wa_i)$, and $F(w_i, wa_i)$ (Fig 2).

2.3 Third Phase: Search for Sequences in the Alignment (SSA)

Objective. Locate the column in S where there is a greater similarity with an wa_i.
Input Data. The consensus matrix $C(S)$ that is generated in 2.1 and each wa_i in the txt output file from 2.2.
Processing. The algorithm starts to iterate over $C(S)$ from 1 to $L - AL_i$ in segments of size AL_i for each wa_i looking for the greatest similarity between a segment of $C(S)$ and the wa_i sequence. The position in the alignment L_i is deposited in the variable $max\_homology$, every time a better similarity value is encountered the value of L_i is replaced in the variable $max\_homology$. Once the iteration over all the length L of

Table 1. Input parameters for ASG.

Parameter	**Value**
conservation threshold	80
Population size	100
Maximum number of generations	5000
Crossover probability	80 %
Mutation probability	10 %
α	0.2
β	0.4
γ	0.4

$C(S)$ has finished the following information is saved to a txt file wa_i, L_i, and SP_i that is the value given by the objective function called similarity percentage (SP).

The latter is done for each wa_i, at the end the txt file will contain the best SP_i values and L_i position for each wa_i.

Objective Function. $SP = \max \sum_{j=1}^{AL_i} F2\left(wa_i^j, C(S)^j\right)$ where $F2\left(wa_i^j, C(S)^j\right) = \left\{C(S)_b^j \middle| b = wa_i^j\right\}$

Output Data. A txt file with three columns, in the first one is the artificial sequence, in the second the column's position where the highest similarity percentage is found and in the third the artificial sequence (Fig. 3).

3 Implementation of the Methodology

For the implementation it is first necessary to have a set of proteins of interest, in this case the dataset cliques_066_batch_clique0.faa is used. This dataset is part of a previous work done by our group; it consists of 66 protein homologues clustered by a Bidirectional Best Hits methodology (BBH) [3]. The preprocessing of the data to obtain a multiple sequence alignment is done with Clustal Ω [12].

For the first stage the parameters are the ones described in 2.1, in the second stage it is necessary to set the parameters (Table 1), while the third stage runs as default. Only 9 sequences from the search for conserved sequences algorithm observed the threshold defined. Therefore, the genetic algorithm returned 9 highly compatible sequences, which are searched for similarity segments in the multiple sequence alignment from the dataset cliques_066_batch_clique0.faa (Table 2).

4 Conclusions and Future Work

There is certainty in that the methodology works correctly since the experiments are run several times and the same artificial sequences are found. These artificial sequences had a percentage of compatibility greater than 70%. The weights of α, β, and γ influence greatly the performance of the psychochemical compatibility objective function since each of the parameters from the objective function are mutually exclusive.

Table 2. Results for each stage.

SCS			ASG		SSA	
w_i	Lv	c_i	wa_i	PCC_i	L_i	SP_i
KWPWYVWLLI	10	84.706	DAPAGVAVVI	72.993	1773	31.176
NECVKSQSSRYGFCG	15	83.922	NKGVDSKSSDGGVGG	75.188	1747	36.176
QVDRLITGRLAAL	13	83.823	KVKDVISGDVAAV	75.307	1518	28.959
ECVKSQSSRYGFCGN	15	82.843	KGVDSKSSDGGVGGN	75.188	1748	35.098
YIKWPWYVWLL	11	82.62	GIDAPAGVAVV	72.812	1771	28.877
KVNECVKSQSSR	12	82.23	DVNKGVDSKSSD	75.431	1745	35.049
IEDLLFDKVVT	11	81.684	IKKVVVKDVVS	75.691	1516	26.203
DRLITGRLAALNAFV	15	81.274	KDVISGDVAAVNAVV	75.327	1711	35.196
IKWPWYVWLLI	11	81.15	IDAPAGVAVVI	73.01	1704	29.278

Finding artificial sequence with high compatibility and its subsequent similarity to segments of a protein homologue family can allows us to identify regions of interprotein interaction that might be important for protein function or structure.

The time it takes for the algorithm to run through all stages and return the final output is considerable low, in the order of just a few minutes. Moreover, each stage of the algorithm can be used separately if needed or applied to a different methodology with a separate goal.

4.1 Future Work

Improve the performance of each of the stages to reduce the time taken for the methodology to runs as an all. Explore different options of metaheuristic in the search for highly compatible artificial sequences. Generate a multi-objective paradigm, where each compatibility matrix is separate objective, thus obtaining a set of non-dominated solutions instead of just one with the current aggregation function.

Aknowledgments. The first author is very grateful for the financial support given by the National Council of Science and Technology of México (CONACYT), which allows him to do his postgraduate studies at the Universidad Autónoma de Aguascalientes (UAA) and the authors also acknowledge the support for the PIINF22-3 project, given by UAA, Mexico.

References

1. Biro, J.C.: Amino Acid Size, Charge, Hydropathy Indices and Matrices for Protein Structure Analysis. Theoretical Biology and Medical Modelling, vol. 3, pp. 1–12 (2006). DOI: 10.1186/1742-4682-3-15.
2. Feduchi, E., Blasco, I., Romero, C., Yáñez, E.: Bioquimica conceptos esenciales. Médica Panamericana (2010)
3. Galvis-Motoa, S.I., de Leon, E.P., Marin, E.M., Cuellar-Garrido, D.: Acquisition and Preprocessing of Proteomic Data for Bidirectional Best Hits Methodology: A Study Case in the Coronaviridae Family. Research in Computing Science, vol. 150, no. 9, pp. 1–12 (2021)
4. Haldurai, L., Madhubala, T., Rajalakshmi, R.: A Study on Genetic Algorithm and its Applications. International Journal of Computer Sciences and Engineering, vol. 4, no. 10, pp. 139–143 (2016)

5. Huo, H., Zhao, Z., Stojkovic, V., Liu, L.: Optimizing Genetic Algorithm for Motif Discovery. Mathematical and Computer Modelling, vol. 52, no. 11–12, pp. 2011–2020 (2010). DOI: 10.1016/J.MCM.2010.06.003.
6. Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., Mcgettigan, P.A., Mcwilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J., Higgins, D.G., Bateman, A.: Clustal W and clustal X version 2.0. Bioinformatics, vol. 23, no. 21, pp. 2947–2948 (2007). DOI: 10.1093/bioinformatics/btm404.
7. Mckee, T., MacKee, J.R.: Bioquímica : Las bases moleculares de la vida. Mc Graw Hill, Access Medicina (2014)
8. Mora-Gutiérrez, R.A., Ramírez-Rodríguez, J., Elizondo-Cortés, M.: Heurística para solucionar el problema de alineamiento múltiple de secuencias. Revista de Matemática: Teoría y Aplicaciones, vol 18, no. 1, pp. 121–136 (2011). DOI: 10.15517/rmta.v18i1.2118.
9. Murray, R.K., Bender, D.A., Botham, K.M., Kennelly, P.J., Rodwell, V.W., Anthony, W.P.: Harper Bioquimica Ilustrada. Mc Graw Hill, (2013)
10. Nelson, D.L., Cox, M.M.: Principles of Biochemistry. Lehninger (2013)
11. Rothlauf, F.: Representations for Evolutionary Algorithms. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. Association for Computing Machinery, pp. 526–546 (2020). DOI: 10.1145/3377929.3389872.
12. Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J.D., Higgins, D.G.: Fast, Scalable Generation of High-Quality Protein Multiple Sequence Alignments Using Clustal Omega. Molecular Systems Biology, vol. 7, pp. 1–6 (2011). DOI: 10.1038/MSB.2011.75.
13. Zambrano-Vega, C., Cárdenas-Zea, M., Aguirre-Pérez, R.: A Multi-Objective Approach to the Optimization of Multiple Sequence Alignment (MSA). Latin American Journal of Computing, vol. 3, no. 1, pp. 43–51 (2016). DOI: 10.5281/zenodo.5748527.
14. Zambrano-Vega, C., Nebro-Urbaneja, A., Aldana-Montes, J.F.: Metaheurísticas de optimización multiobjetivo aplicadas a la inferencia filogenética y al alineamiento múltiple de secuencias. Universidad de Málaga (2017)

ISSN 1870-4069

A Comparison of Tiny-Nerf Versus Spatial Representations for 3D Reconstruction

Saulo Abraham Gante, Juan Irving Vasquez,
Marco Antonio Valencia, Mauricio Olguín-Carbajal

Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Mexico

sganted1500@ipn.mx

Abstract. Neural rendering has emerged as a powerful paradigm for synthesizing images, offering many benefits over classical rendering by using neural networks to reconstruct surfaces, represent shapes, and synthesize novel views, either for objects or scenes. In this neural rendering, the environment is encoded into a neural network. We believe that these new representations can be used to codify the scene for a mobile robot. Therefore, in this work, we perform a comparison between a trending neural rendering, called tiny-NeRF, and other volume representations that are commonly used as maps in robotics, such as voxel maps, point clouds, and triangular meshes. The target is to know the advantages and disadvantages of neural representations in the robotics context. The comparison is made in terms of spatial complexity and processing time to obtain a model. Experiments show that tiny-NeRF requires three times less memory space compared to other representations. In terms of processing time, tiny-NeRF takes about six times more to compute the model.

Keywords: Neural rendering, NeRF, 3D reconstruction, mapping.

1 Introduction

The recent and continuous advances in neural rendering have shown numerous applications and became a new field of study in the graphics community. Some of these efforts are in the implicit functions which represent shapes in three dimensions (3D) [3, 15]. The tool for creating the neural representations is a multi-layer perceptron (MLP). This MLP works as a general implicit function approximator. On the other hand, there are plenty of methods to reconstruct surfaces, and represent shapes and volumes in 3D space. Some examples are meshes [2], point clouds [1], voxel maps [7], and octrees [4]. The latter offers diverse capabilities to reconstruct or create 3D models and those have diverse applications in robotics and artificial vision [18, 10, 17, 8].

In this paper, we perform a comparison between neural representations, point clouds, meshes, and voxel maps in terms of memory space and processing time required to obtain a model. The main objective is to show, clarify or put some important considerations for future works related to object reconstruction.

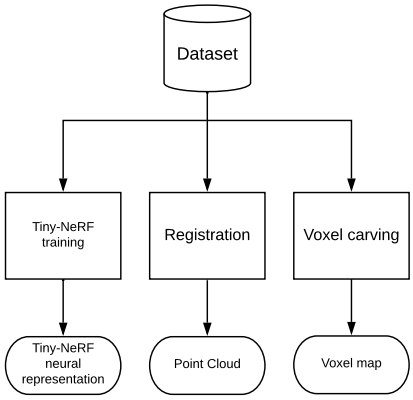


Fig. 1. General diagram of the experiments. Given a dataset which contains positions in 3D space and images, the data required is extracted with a simulator, as a result the three representations to be compared are obtained.

The MLP employed follows the architecture proposed in tiny-NeRF by the authors of Neural Radiance Fields (NeRF) [6]. We design a grid search-based experimentation. For the tiny-NeRF, the independent variables are i) learning rate ii) encoding functions and iii) seed; the resulting grid search has 36 experimental units. In addition, the experiment employs the same capturing positions to create the neural model, point cloud, and voxel map. The experiment shows that neural representations require 3 times less memory to store a model but on the other hand they takes about 6 times more to compute a model concerning the other representations.

The rest of the paper is structured as follows. Section 2 introduces the required concepts. Section 3 presents the related work and advances of volume, surface and neural representations. In Section 4, we present the methodology used to perform the experiments carry out in Section 5, where results are also reported. Finally, the conclusions and the future work are given in Section 6.

2 Preliminaries

In this section, we define certain concepts required to understand the topics tackled in this paper. A commonly used data representation is the point cloud, that is a representation of body shapes and is made by points mapped in the 3D space which are usually produced by sensors or scanners.

The point cloud could be processed in order to create more accurate representations. Meshes are representations of shapes formed by a set of nodes and connections between them, one of the advantages is that it has a range of different resolutions which means that it could be as accurate as wanted but the more resolution those have the more computation it needs to complete the representation. Another 3D shape representation is the voxel, which is a cube of unitary distance, and the union of a set creates a voxel map representation.

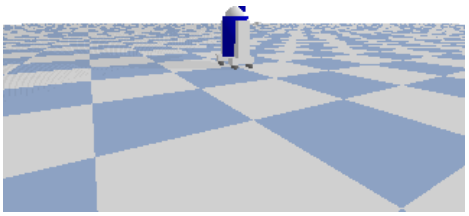


Fig. 2. World setup. An object is placed in the simulated world.

The voxel map represents shapes of objects or it is possible to represent the volume/solid by using a voxel carving technique. Those representations are commonly used to reconstruct shapes, objects and maps.

According to [16] there is no definition for neural rendering and suggests a definition for Neural Rendering as: "Deep image or video generation approaches that enable explicit or implicit control of scene properties such as illumination, camera parameters, pose, geometry, appearance, and semantic structure."

A recent proposed neural rendering technique is NeRF [6], it became one of the most popular and extensively used to render objects in 3D space due to its capabilities to create novel views in the reconstructed scene.

NeRF [6] is an approach for creating novel view synthesis, it uses a set of input views to optimize a continuous volumetric scene function, as a result, this optimization produces a novel view of a complex scene. Its input is a 5D vector function, which contains the 3D space location (x,y,z) and 2D viewing direction (θ, ϕ) and the output is an emitted color: Red, Green, Blue (r,g,b) and volume density (σ). NeRF uses the concept of encoding functions where the purpose of these functions is to map the input into a higher dimensional space where the MLP can more easily approximate higher frequency functions.

To generate a NeRF from a specific viewpoint, first, a set of rays are marched through the scene, the data generated is fed into the neural network and produce a set of RGBσ values then the data is structured into a 2D image.

3 Related Work

The 3D reconstruction of spaces and objects is not a new research topic and has many approaches which reconstruct scenes employing different techniques, the accuracy of the volumetric representations relies on the resolution employed to map, and the more resolution is wanted the more computation is needed which means more time is needed to achieve a good result.

Volume representations in 3D space have many methods to represent synthetically objects, like meshes [2], point clouds [1], voxel maps [7], and octree [4]. Those offer diverse capabilities to reconstruct or create 3D models and those have diverse applications in robotics and artificial vision [8, 10, 17, 18]. Despite the popularity that they have, resolution of the representations is one of it cons. Also, the memory consumption between them is variable and usually requires memory in the order of Megabytes (MB).

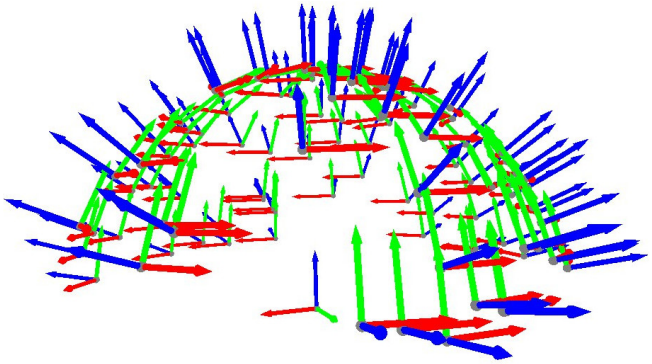


Fig. 3. Capturing poses. In this figure it is shown the 106 poses used, the frame of reference is the described by OpenGL [5] for synthetic cameras.

Neural rendering has gained popularity since it employs a multi-layer perceptron (MLP) to achieve these tasks [11, 12]. In [3, 15] are presented different techniques to represent shapes and volumes in 3D space.

The proposed methods concentrate its effort in creating those representations and compare it with state-of-the-art. On the other hand, in [14] they propose an approach for volume compression and compare it with voxel maps. A Simultaneous Localization And Mapping system is proposed by [13], they compare it with truncated signed distance function (TSDF) method, both approaches do a comparison in terms of memory consumption, and stand out a good performance.

We believe that time taken in the process of the reconstructions is an important variable to take in consideration, and that the consulted approaches do not report those differences in terms of time.

4 Methodology

We want to experiment with neural representations, exploring the advantages that those have over existent representations used for 3D space reconstruction. We do a comparison between neural representations proposed by the authors of NeRF [6], and three different spatial representations used to model objects, such as meshes, voxel maps, and point clouds.

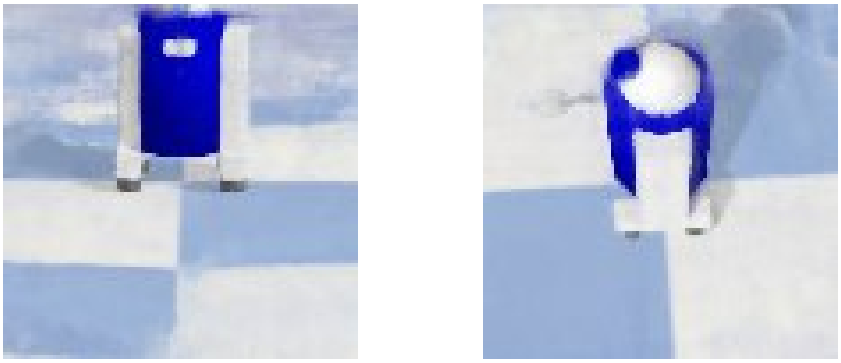
We use a dataset that contains 106 (see Fig. 3) pairs of sensor poses, and using those poses, we extract the required data in order to create the proposed representations (Figure 1). The main reason to use one dataset is that we want to give the algorithms the same point of view for a fair comparison in terms of data that could be extracted given the poses in the data set.

4.1 Dataset

Given that we propose a comparison with synthetic data, we use a simulator to render images of an object (simulating a camera inside the simulated world). See Figure 2.

(a) Frontal point of view. (b) Lateral point of view.

Fig. 4. Images of the object of study extracted using Pybullet.



(a) Frontal point of view. (b) Lateral point of view.

Fig. 5. Neural representations created with the Tiny-NeRF.

Then, a world is needed to set up, configure it with a ground, and place the mesh of the object of interest in it. From the synthetic datasets used in NeRF [6], we apply transformation matrices as in Equation (1):

$$T = \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix}, \tag{1}$$

where R indicates the rotation matrix, whose values represent the rotations over the three axes, and the p indicates the position vector, whose values contain the position of a body in a 3D space (x, y, z). Please see Figure 3.

Having those positions in 3D space the required datasets are extracted, that is RGB images, ray casting points, and depth data. All in order to create the proposed reconstructions.

Table 1. Grid search. ID express an identification number, the variable values employed for each experiment with Tiny-NeRF and the results expressed in terms of Loss and PSNR.

ID	Factor 1	Factor 2	Factor 3	Metric 1	Metric 2
	Seed	Learning Rate	Coding functions	Loss	PSNR (dB)
1	2057	5×10^{-3}	6	0.5463	2.6257
2	2057	5×10^{-3}	9	0.0030	25.2288
3	2057	5×10^{-3}	10	0.0493	13.0715
4	2057	5×10^{-3}	12	0.5463	2.6257
5	2057	5×10^{-4}	6	0.5463	2.6257
6	2057	5×10^{-4}	**9**	**0.0025**	**26.0206**
7	2057	5×10^{-4}	10	0.5463	2.6257
8	2057	5×10^{-4}	12	0.5463	2.6257
9	2057	5×10^{-5}	6	0.5463	2.6257
10	2057	5×10^{-5}	9	0.0026	25.8503
11	2057	5×10^{-5}	10	0.5463	2.6257
12	2057	5×10^{-5}	12	0.5463	2.6257
13	7461	5×10^{-3}	6	0.0921	10.3574
14	7461	5×10^{-3}	9	0.0032	24.9485
15	7461	5×10^{-3}	10	0.5463	2.6257
16	7461	5×10^{-3}	12	0.5463	2.6257
17	7461	5×10^{-4}	6	0.5463	2.6257
18	7461	5×10^{-4}	9	0.0026	25.8503
19	7461	5×10^{-4}	10	0.5463	2.6257
20	7461	5×10^{-4}	12	0.5463	2.6257
21	7461	5×10^{-5}	6	0.5463	2.6257
22	7461	5×10^{-5}	**9**	**0.0025**	**26.0206**
23	7461	5×10^{-5}	10	0.5463	2.6257
24	7461	5×10^{-5}	12	0.5463	2.6257
25	5680	5×10^{-3}	6	0.5463	2.6257
26	5680	5×10^{-3}	9	0.0032	24.9485
27	5680	5×10^{-3}	10	0.5463	2.6257
28	5680	5×10^{-3}	12	0.0033	24.8149
29	5680	5×10^{-4}	6	0.5463	2.6257
30	5680	5×10^{-4}	9	0.0027	25.6864
31	5680	5×10^{-4}	10	0.5463	2.6257
32	5680	5×10^{-4}	**12**	**0.0024**	**26.1979**
33	5680	5×10^{-5}	6	0.5463	2.6257
34	5680	5×10^{-5}	9	0.0027	25.6864
35	5680	5×10^{-5}	10	0.5463	2.6257
36	5680	5×10^{-5}	12	0.0027	25.6864

4.2 Point Cloud and Voxel Map

Open3D library [21] allows us to visualize objects and create representations. For the point cloud, it is created by the use of ray-tracing which emits synthetic rays in simulation when those touches or intersect with a surface return a value, having this is possible to calculate in $\mathbb{R}^3$ and map them into points in space, creating a point cloud.

Table 2. Average of the results in grid search.

LR	Functions	Loss	PSNR (dB)
5×10^{-3}	6	0.3949	4.0351
5×10^{-3}	9	0.0031	25.0863
5×10^{-3}	10	0.3806	4.1953
5×10^{-3}	12	0.3653	4.3735
5×10^{-4}	6	0.5463	2.6256
5×10^{-4}	**9**	**0.0026**	**25.8502**
5×10^{-4}	10	0.5463	2.6256
5×10^{-4}	12	0.365	4.3770
5×10^{-5}	6	0.5463	2.6256
5×10^{-5}	**9**	**0.0026**	**25.8502**
5×10^{-5}	10	0.5463	2.6256
5×10^{-5}	12	0.3651	4.3758

This process is repeated every capture, then the resulting points are concatenated and filtered to reduce possible noise created by captures. We create a voxel model using the technique of *voxel carving*, using a pinhole camera and homogeneous transformation matrix is possible to create a voxel dense given the resulting images and a silhouette to employ a *carve silhouette* method provided by Open3D, resulting in a voxel model.

4.3 Tiny-NeRF

As explained above, NeRF [6] receives as input a set of data that express location and viewing direction where the output is an emitted color and a volume density.

Tiny-NeRF is a simplified version of NeRF, which is an MLP conformed by 6 fully-connected ReLU layers each with 256 filter size, one fully-connected ReLU layer with a filter size of 64 then an output layer that expresses the emitted RGBσ at a certain position with a four filter size layer. The process starts by getting rays according to the pose, then the returned rays become useful to map 3D points which are going to be fed into Tiny-NeRF input, the output of the model is used to compute opacities and RGB data, finally the weights are calculated and the process is repeated.

5 Experiments

We evaluate the Tiny-NeRF describing a grid search where certain variables are changed over the experiments. Using the same position captures we perform reconstruction with voxels and a point cloud. All the data was synthetic and obtained using Open3D.

Our experiments run in Python and the libraries employed are Pytorch [9] for the MLP or neural representations, Pybullet and Open3D [21].The hardware employed for those experiments is the CPU/GPU provided by Colab which allows us to use a Graphic card: Tesla P100-PCIE-16GB with 16GB of GPU-RAM, 25.46 GB of RAM, and 166.83 GB in Hard disc drive.

(a) Simulation captures.

(b) Neural representations.

Fig. 6. Qualitative results using Tiny-NeRF. We extracted different poses and visualizations.

5.1 Tiny-NeRF Training

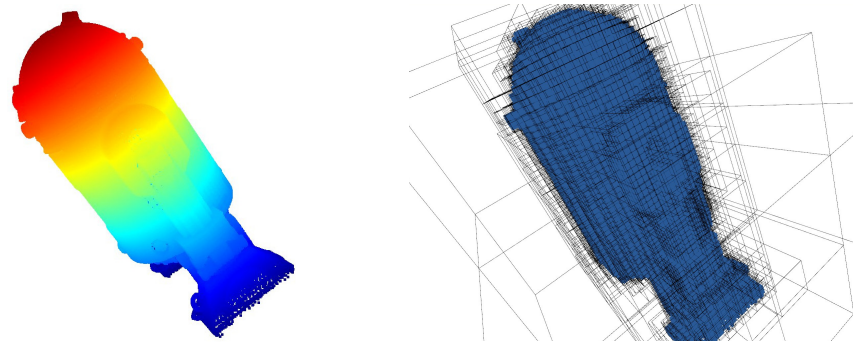
Tiny-NeRF is a simplified version of NeRF, which is an MLP conformed by six fully-connected ReLU layers each with a 256 filter size, one fully-connected layer with a filter size of 64 then an output that expresses the RGBσ values. The grid search proposed to vary over three variables and the values are:

- **Seed**: 2057, 5680 and 7461,
- **Learning rate**: $5x10^{-3}$, $5x10^{-3}$ and $5x10^{-3}$,
- **Encoding functions** : 6, 9, 10 and 12.

For the Neural Networks (NN) training a commonly used metric is Loss since it evaluates how bad predicts on an example, the *Peak Signal-to-Noise Ratio* (PSNR) is used to measure the ratio between a signal and the noise which affects the representation of this signal; in this case, the PSNR is used to measure how well the Tiny-NeRF does a representation compared to the original images. On the other hand, to measure time the unit employed is seconds (s) and to measure space in memory we utilize MB.

To obtain the data set we employed Pybullet simulator which let us set simulated worlds, set objects in it (Figure 2) and create pinhole cameras to extract or create synthetic images, among other things. Once the object is set in the world, it is possible to create a synthetic camera given its position, target position, field of view (FOV), near and far plane distance, weight and height of the image. The positions are given by the data capturing positions, the FOV is $17.70°$, the weight and height are equal to 100. Resulting in images like the ones in Figure 4.

The experiments with the Tiny-NeRF are iterated for five thousand epochs each and the variables are modified at each experiment, the number of experiments is 36. The results of this experiment are shown in Table 1. The entire experiment took about 65,100 seconds which means that approximately every experiment took 1,808.33 seconds to be completed. To summarize the information in Table 1, the average was calculated (Table 2) in order to easily extract which parameters perform better results with the MLP.



(a) Point cloud reconstruction. (b) Voxel reconstruction.

Fig. 7. Volumetric reconstructions.

Analysis of experiments showed that nine coding functions help the Tiny-NeRF to accurately (Figure 5) create a neural representation of the object, and the learning rate helped to achieve good performance in fewer epochs. Additionally, the neural representations took **1,808.33 seconds** to complete an experiment, and the memory space to store a representation is **1.5 MB**.

Additionally to PSNR, we perform evaluations over two more metrics SSIM and LPIPS [19, 20] which are commonly used to measure distances over images, looking for a measure of how well the Tiny-NeRF is rendering views. Comparing images like the ones in Fig. 6 the metrics proposed gave as a result **0.8481** and **0.0565**, respectively. Those results affirm that the representations are good in quality but it could improved.

5.2 Comparison of Tiny-NeRF Versus Spatial Representations

Once Tiny-NeRF has been trained and the representations were created, we compared the time taken to do a representation. To measure the time, it was printed every time a process started and finished the difference between those shows the time taken. The space in memory is measured by the file space in memory that is required to store the representations.

The data employed to reconstruct was obtained by capturing in the positions of the data set, mentioned above, once the captures are done the process of data was done employing Open3D [21].

The point cloud (Figure 7 (a)) was obtained by mapping the points resultant of a ray-tracing operation into XYZ or 3D space, those points are concatenated and finally filtrated to avoid noise in the reconstruction. The experiment took about **2 seconds** and the memory space needed is **12 MB**.

The resultant voxel map (Figure 7 (b)) was created with the voxel carving method which not only reconstructs the surface of an object, it creates a voxel map that is a cube of certain dimensions and according to the visualized data, the algorithm carves the shape into it, creating a solid voxel representation. The experiment took about **166 seconds** and the memory space needed is **21.2 MB**.

The performed experiments results showed that implicit or neural representation requires at least 3 times less memory compared with other representations (Table 3).

Table 3. Comparative of memory size.

Representation	Size (MB)
Meshes	4.5
Point cloud	12.0
Voxelization	21.2
Implicit representation	**1.5**

Table 4. Comparative of time taken to perform a representation.

Representation	Time (s)
Meshes	28800
Point cloud	**2**
Voxelization	166
Implicit representation	1008

In terms of time to process a representation, point clouds and voxel maps build the representations in about 6 times less time than the implicit representations (Table 4).

6 Conclusions and Future Work

This paper tackles the trend research topic, neural rendering, which has many advances in graphics generation. We compare a simplified version of NeRF with different volume representations commonly used in robotics and vision reconstruction tasks, all compared in terms of memory space and time to build representations.

First, we experimented with Tiny-NeRF that computes the colors over a certain position with a viewing direction; the experiments were conducted by a grid search looking to perform good representations of an object. In addition, we perform a reconstruction using voxels and point clouds. The comparison, in terms of memory space and time, shows that the Tiny-NeRF architecture (MLP) requires less memory but takes more time to build a representation.

On the other hand, this experiment showed that the neural representation relies on the fine-tuning of the variables implied in the training of the MLP. We believe that the results of the experiments can offer some relevant information or considerations to take when a reconstruction task is needed. In a future work we will experiment with more objects and with a mobile manipulator robot.

References

1. Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A., Silva, C.: State of the Art in Surface Reconstruction from Point Clouds. In: Eurographics - State of the Art Reports, pp. 161–185 (2014). DOI: 10.2312/egst.20141040.
2. Delingette, H.: General Object Reconstruction Based on Simplex Meshes. International Journal of Computer Vision, vol. 32, pp. 111–146 (1999). DOI: 10.1023/A:1008157432188.
3. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local Deep Implicit Functions for 3D Shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4857–4866 (2020). DOI: 10.1109/CVPR42600.2020.00491.

4. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. Autonomous Robots, vol. 34, pp. 189–206 (2013). DOI: 10.1007/s10514-012-9321-0.
5. Learn OpenGL: Camera https://learnopengl.com/Getting-started/Camera (2018)
6. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: European Conference on Computer Vision, vol. 12346, pp. 405–421 (2020). DOI: 10.1007/978-3-030-58452-8_24.
7. Muglikar, M., Zhang, Z., Scaramuzza, D.: Voxel Map for Visual SLAM. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4181–4187 (2020). DOI: 10.1109/ICRA40945.2020.9197357.
8. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262 (2017). DOI: 10.1109/TRO.2017.2705103.
9. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., De-Vito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., et al.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 8026–8037 (2019). DOI: 10.48550/arXiv.1912.01703.
10. Respall, V.M., Devitt, D., Fedorenko, R., Klimchik, A.: Fast Sampling-Based Next-Best-View Exploration Algorithm for a MAV. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 89–95 (2021). DOI: 10.1109/ICRA48506.2021.9562107.
11. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization, pp. 2304–2314 (2019). DOI: 10.48550/arXiv.1905.05172.
12. Sitzmann, V., Martel, J.N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions. In: 34th Conference on Neural Information Processing System, pp. 1–35 (2020). DOI: 10.48550/arXiv.2006.09661.
13. Sucar, E., Liu, S., Ortiz, J., Davison, A.: iMAP: Implicit Mapping and Positioning in Real-Time. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 6229–6238 (2021). DOI: 10.48550/arXiv.2103.12352.
14. Tang, D., Singh, S., Chou, P.A., Häne, C., Dou, M., Fanello, S.R., Taylor, J., Davidson, P.L., Guleryuz, O.G., Zhang, Y., Izadi, S., Tagliasacchi, A., Bouaziz, S., Keskin, C.: Deep Implicit Volume Compression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1293–1303 (2020). DOI: 10.1109/CVPR42600.2020.00137.
15. Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., Zhang, L.: Sign-Agnostic CONet: Learning Implicit Surface Reconstructions by Sign-Agnostic Optimization of Convolutional Occupancy Networks. In: International Conference on Computer Vision, pp. 1–16 (2021)
16. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J.M., Nießner, M., Pandey, R., Fanello, S.R., Wetzstein, G., Zhu, J.Y., Theobalt, C., Agrawala, M., Shechtman, E., Goldman, D.B., Zollhöfer, M.: State of the Art on Neural Rendering. Computer Graphics Forum, vol. 39, no. 2, pp. 701–727 (2020). DOI: 10.1111/cgf.14022.
17. Uyanik, C., Secil, S., Ozkan, M., Dutagaci, H., Turgut, K., Parlaktuna, O.: SPGS: A New Method for Autonomous 3D Reconstruction of Unknown Objects by an Industrial Robot. In: Annual Conference Towards Autonomous Robotic Systems, vol. 10965, pp. 15–27 (2018). DOI: 10.1007/978-3-319-96728-8_2.
18. Vasquez-Gomez, J.I., Sucar, L.E., Murrieta-Cid, R.: View Planning for 3D Object Reconstruction with a Mobile Manipulator Robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4227–4233 (2014). DOI: 10.1109/IROS.2014.6943158.

19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612 (2004). DOI: 10.1109/TIP.2003.819861.
20. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018). DOI: 10.1109/CVPR.2018.00068.
21. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A Modern Library for 3D Data Processing. Computer Vision and Pattern Recognition, pp. 1–6 (2018). DOI: 10.48550/arXiv.1801.09847.

ISSN 1870-4069

Detection of Tomato Ripening Stages Using Yolov3-Tiny

Gerardo Antonio Alvarez-Hernandez[1], Juan Carlos Olguin[1], Juan Irving Vasquez[1], Abril Valeria Uriarte[1], Maria Claudia Villicaña-Torres[3,2]

[1] Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Mexico

[2] Consejo Nacional de Ciencia y Tecnología,
Ciudad de México,
Mexico

[3] Centro de Investigación en Alimentación y Desarrollo A.C.,
Sinaloa

{jvasquezg, auriartea}@ipn.mx, maria.villicana@ciad.mx

Abstract. One of the most important agricultural products in Mexico is the tomato (*Solanum lycopersicum*), which occupies the 4th place national most produced product . Therefore, it is necessary to improve its production, building automatic detection system that detect, classify an keep tacks of the fruits is one way to archieve it. So, in this paper, we address the design of a computer vision system to detect tomatoes at different ripening stages. To solve the problem, we use a neural network-based model for tomato classification and detection. Specifically, we use the YOLOv3-tiny model because it is one of the lightest current deep neural networks. To train it, we perform two grid searches testing several combinations of hyperparameters. Our experiments showed an f1-score of 90.0% in the localization and classification of ripening stages in a custom dataset.

Keywords: Tomato detection, yolo tiny v3, deep learning, precision agriculture.

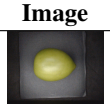
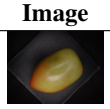
1 Introduction

One of the most important agricultural products in Mexico is the tomato (*Solanum lycopersicum*), which occupies the 4th place national most produced product. Likewise, it occupies the 9th place globally in the production of this product, with the amount of 3,370,877 tons in 2020 [1]. In the production of tomatoes, shape and color have a considerable influence on the expectations of their commercial value.

In terms of shape, it is desirable to be round, globular, or oval, depending on the type and in terms of color, it has a uniform color ranging from orange to deep red, with no green shoulders. Its appearance should be smooth and with small scars corresponding to the floral tip and peduncle, in addition, it is firm to the touch, that is, it is not soft or easily deformed due to over maturity [2].

With this in mind, it is necessary to monitor the ripening stages of tomatoes during the production process, because whether it is not done correctly, it could have a poor

Table 1. Examples of dataset.

Image	Label	Image	Label	Image	Label
	Green		Striped		Salmon
	Orange		Red-orange		Red

quality tomatoes causing losses. This task is carried out by visual inspection with people trained to identify: the degree of ripening and fruit sizes, however, these methods are subjective.

Several methodologies have been proposed for the creation of systems that help in this task, such as Vishal et al. [3] where the authors propose a monitoring model using an Arduino, with color characteristics L*a*b to discern the ripening stage of the tomato. Anna et al. [4] propose instead of extracting the color, to use tomato florescence based on the reflection and emission of heat that the fruit has, with the drawback that the tomato must be in a certain temperature range to perform a correct classification.

On the other hand, Kejin et al. [5] use these two characteristics, color and fluorescence using a colorimeter, with the drawback that the detection is only effective when the tomatoes are at a temperature of 25 degrees Celsius. As can be seen, these solutions focus on finding the decision pattern through the expert's knowledge, another way of solving is with the use of machine learning models so that the decision patterns are chosen automatically without the need of an expert such as the one of Aranda-Sanchez et al. [6] which uses the colorimeter to obtain the data and through the use of a Bayesian classifier determines the degree of ripeness of the tomato.

Seeing that these systems focus only on the classification of tomatoes in their ripening stages, there are other systems that, in addition to classifying them, also locate them, commonly called detection systems, such as Nuttakarm et al. [7], the authors propose to use a R-CNN (region-based convolutional neural network) to locate the tomato, then the images are segmented using a k-means algorithm with 2 centroids $k = 2$: area of interest (fruit) and uninterested region. The segmented areas of interest are converted to Hue-Saturation-Value (HSV) color space to extract characteristics, that are then used with a Support Vector Machine (SVM) to classify the ripening stage of the tomato.

Dasom et al. [8] propose to use the R-CNN , but as a multiclass detector in which the classes are the different ripening stages, being able to detect the ripening stages with occlusions, but at the expense of a large computational cost when using this detection model.

In this work, we propose to solve the problem of tomato ripening stage detection to be used in an automatic tomato monitoring system. Therefore, we propose to solve this problem by means of a vision system implementing an existing detection model. This model is based on a one-step convolutional network with multi-class detection YOLOtiny-V3 [9].

This network was chosen because it is is one of the lightest current deep neural network architectures. We used a database of 3,000 images of tomatoes labeled with

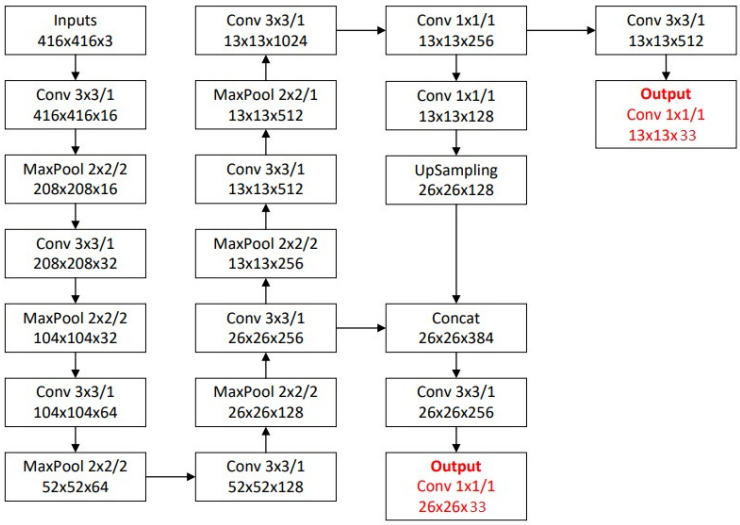


Fig. 1. Block structure of the YOLOv3-tiny architecture used for the detection of this case study.

the class name, corresponding to their ripening stage, and a bounding box enclosing the tomato [19]. In addition, we set up an experimental design with various combinations of its hyperparameters to obtain 90.0% in f1-score.

The rest of the paper is organized as follows. Section 2 presents some related work on the detection and classification of tomato maturity stages. Section 3 describes the components that compose our proposed solution. Section 4 describes the experiments performed, as well as the analysis of results, and Section 5 presents the conclusions and suggestions for future work.

2 Related Work

This section reviews the various research approaches that have currently been adopted to solve the problem of detecting and classifying ripening stages of tomato, as well as other fruits.

Starting with ripeness sorting, Dasom et al. [8] propose a model of tomato ripeness monitoring inside a hydroponic greenhouse. They use the fastest R-CNN for the process of detecting the tomato. Once the fruit region is detected, the k-means algorithm is used to separate the background from the fruit region, then it is converted from RGB to HSV color space extracting only the hue channel, to classify the ripeness using 6 color shades.

On the other hand, Nashwa et al. [10] use Principal Component Analysis (PCA) to extract the characteristics of tomatoes in their different ripening stages in order to be delivered to a SVM to perform the classification, a Linear Discriminant Analysis (LDA) was also tested, but best classification score was obtained by the one-against-one multi-class SVMs system.

Another work like the one by De Luna et al. [11] propose to locate and detect ripening stages of tomatoes, using two pre-trained convolutional models, the R-CNN

Table 2. Exploratory grid search.

ID	Factor 1	Factor 2
1	1×10^{-3}	Adam
2	1×10^{-3}	SGD
3	1×10^{-5}	Adam
4	1×10^{-5}	SGD

and SDD (Single Shot Detector) with better performance of SDD to detect flower and fruit. For maturity classification, 3 basic classifiers were used: artificial neural network (ANN), k-nearest neighbors (K-NN) and SVM, the latter having the highest classification accuracy. Lui et al. [12] proposed an improved DenseNet model to detect tomato ripeness, to improve it, they proposed a structured sparse operation by splitting the convolution kernel into multiple groups, obtaining a computational reduction of 18% of the original network. It can detect tomatoes in a complex background and different sizes.

Rangarajan et al. [13] classify tomato leaf images with 6 different diseases. Using 2 different deep learning architectures AlexNet and VGG16net, a transfer learning approach, where Alexnet obtained the highest classification performance with a 97.99% accuracy. Hong et al. [14] test 5 deep networks Resnet50, Xception, MobileNet, ShuffleNet, Desnet121_Xception to detect tomato plant diseases, to decrease the computational time they used transfer learning and data augmentation. Desnet121_Xception obtained the best performance with 97.10% accuracy.

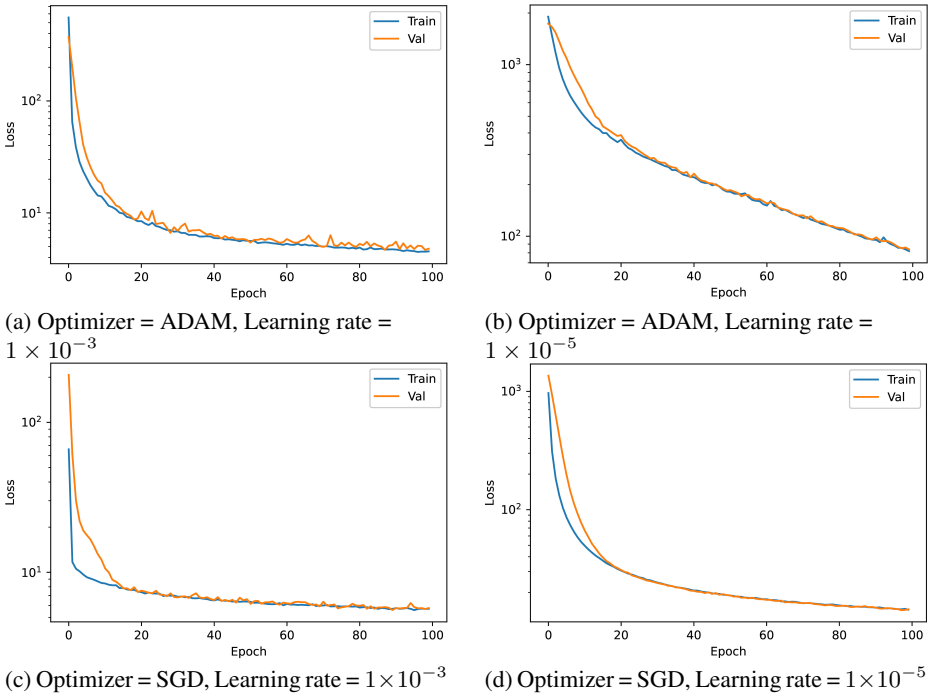
There are several approaches that have been used for fruit detection, such as Liu et al. [15] where the authors used a gradient-oriented descriptor (HOG) to train an SVM to detect tomatoes, followed by an FCR (false color removal) to eliminate false positives and an NMS (Non-Maximal Suppressor) to eliminate repetitions. Zhang et al. [16] implementing a convolutional network (CNN), they focused on testing different data augmentation methods.

These were: rotation, scaling, Gaussian noise, salt noise, pepper noise, as well as combinations. In addition, the method uses t-Distributed Stochastic Neighborhood embedding (t-SNE) to remove poor-quality images to obtain a better dataset. Rotational enhancement, scaling and salt noise methods performed better, reaching an accuracy of 91.9%.

Y. Mu et al. [17] built a model that detects green tomatoes regardless of whether they are occluded or at different stages of ripening using an R-CNN coupled with Resnet 101 using COCO data for transfer learning.

They obtained an average accuracy of 87.83% taking into account that the tomatoes were in their natural environment compared to others where they are in controlled conditions. Although with a high computational cost, Lui et al. [18] modified the YOLO v3 model for tomato recognition called YOLO-Tomato. it incorporates a DenseNet architecture to reuse image features, and replaces the traditional rectangular box with a circular one.

They applied data augmentation with scaling and cropping operations in order to increase performance. This model is able to detect tomatoes, taken with various light intensities, achieving a prediction rate of 94.58%.



(a) Optimizer = ADAM, Learning rate = 1×10^{-3}

(b) Optimizer = ADAM, Learning rate = 1×10^{-5}

(c) Optimizer = SGD, Learning rate = 1×10^{-3}

(d) Optimizer = SGD, Learning rate = 1×10^{-5}

Fig. 2. Graphics of training performance on the exploratory grid search.

To summarize this review, our proposal detects the ripening stages using a single model, compared to the work of Dasom et al. [8] and De Luna et al. [11] whose approaches were to separate the detection by occupying two models, one for locating and one for classifying. Furthermore, the works of Nashwa et al. [10] and Rangarajan et al. [13] only classify ripening stages and do not detect them. And in the case of Luis et al. [12, 18], and Hong et al. [14] only detects only the tomato class in contrast to ours which solves a more particular problem which is to detect each of the tomato ripening stages.

3 Multi-Class Detection of Ripening Stages

To achieve the detection of tomato ripeness, we propose a vision system that uses a general-purpose deep learning architecture to perform multi-class detection. This architecture is the YOLOv3-tiny which is a small version of Yolo v3 [9]. Given that our problem has not been reported previously, we train the network architecture using a custom dataset. To train the network, we design two grid searches, these searches allow us to find adequate network parameters.

We use a previous dataset [19], we labeled it and then we trained the network with various parameters. In the end, we validate it using detection thresholds to calculate accuracy, precision, recall, f1-score, and IOU.

Table 3. Table of performance metrics obtained by the exploratory grid search.

Experimets		**Performance thresholds**		**Orthogonal Parameters**		**Metrics**					**Label**
Optimizer	Learning rate	Score	IOU	parameters	Batch size	Accuray	Precision	Recall	F1-score	Average IOU	Not detected
ADAM	1×10^{-3}	0.7	0.5	100	64	**0.95**	**0.93**	**0.95**	**0.93**	**0.86**	**117**
ADAM	1×10^{-5}	0.7	0.5	100	64	-	-	-	-	-	600
SGD	1×10^{-3}	0.7	0.5	100	64	0.91	0.71	0.89	0.74	0.76	531
SGD	1×10^{-5}	0.7	0.5	100	64	-	-	-	-	-	600

3.1 Dataset

The dataset consists of 3,000 images of tomatoes, fruit of *Solanum Lycopersicum*, with a resolution of 800 $\times$ 600 pixels, in JPG format and RGB channels. The original dataset was published in [19]. In that work, the dataset contains only the classes of each of the images (no bounding boxes). The tomato images were taken in controlled conditions, varying the position of the tomatoes to have a diversity of view of the object.

The 6 labels handled in the dataset are: Orange (500 of this class), Striped (500 of this class), Red (500 of this class), Red-Orange (500 of this class), Salmon (500 of this class) and Green (500 of this class) (see Table 1). Therefore, the database is balanced.

For this work, we have manually labeled the bounding boxes enclosing the tomato found in each image, as well as the label indicating the ripening class of the tomato. This labeling was done by using the "labelImag" tool [20]. The images in the dataset are re-scaled to fit the input size required by Yolov3-tiny, which is 416 x 416 pixels.

3.2 Deep Learning Architecture YOLOv3-Tiny

The deep learning algorithm used belongs to the You Only Look Once (YOLO) family. This family of algorithms exploits the use of convolutional neural networks for object detection. They are one of the fastest object detection algorithms available and are a good choice for real-time detection without compromising accuracy.

YOLO is an object detection architecture proposed by Joseph Redmon et al. in 2015 [21] and improved with the second version in 2016 (YOLOv2) [22] and with it was third version in 2018 (YOLOv3) [23]. The YOLOv3 structure consists of six types of layers, defined as net, convolutional, max-pooling, yolo, route, and upsample. The net layer configures the parameters of the entire network.

The convolution layer has three modules: convolutional operation, batch normalization, and activation function. In addition, the convolution operation includes feature map conversion and general matrix multiplication.

Feature learning is done through convolutional layers, which have a similar structure to ResNet, in YOLOv3 they are called "Residual Blocks" and are used for feature learning. Residual blocks consist of several convolutional layers and jump connections.

One feature of YOLOv3 is that it performs detection at three different scales. The scales are 13×13, 26×26, and 52×52. The depth of model is calculated with the following operation:

$$depth_{yolo} = (5 + n_{clases}) \times n_{anchors}, \tag{1}$$

Table 4. Exploitation grid search.

ID	Factor 1
1	1×10^{-3}
2	1×10^{-4}
3	1×10^{-5}

$$depth_{yolo} = (5 + 6) \times 3, \tag{2}$$

$$depth_{yolo} = 33. \tag{3}$$

In this case, there are 6 maturity classes and 3 anchors for each scale, which results in a depth of 33 in each yolo output. The YOLOv3 architecture, considering as input images the images of this work, whose dimension is $416 \times 416 \times 3$ (where the three represents the RGB channels).

There is another version of YOLO, which has a reduced depth of the convolutional layers and it is called YOLO v3-Tiny. It was also proposed by Pranav Adarsh [9].

A characteristic of this architecture is that it has only convolutional layers for feature extraction, which causes that the execution speed to increase significantly.

YOLO v3-Tiny uses a max-pooling layer and thus reduces the image in the convolution layer. It predicts using a three-dimensional tensor containing objectivity score, bounding box, and class predictions on two different scales.

It divides an image into $S \times S$ grid cells. For final detections, ignore the bounding boxes for which the objectivity score is not the best using the non-maximum suppression method (NMS).

The bounding box prediction occurs at two feature map scales, which are 13×13 and 26×26. The YOLOv3-tiny architecture for input images with dimension $416 \times 416 \times 3$ is shown in Fig. 1.

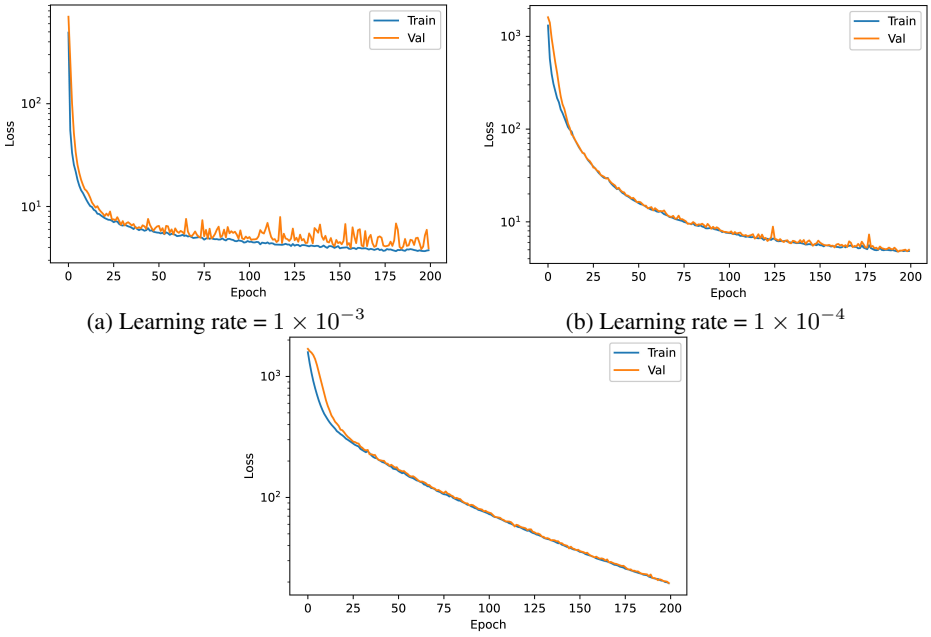
4 Experiments

In this work, two different grid searches were performed to find the most promising result, each of these grids will be detailed in this section, as well as the results obtained and their analysis. The performance of each experiment was evaluated using the following performance measures: accuracy, precision, recall, f1-score [24] and IOU [25].

4.1 Design of Experiment

Two grid searches were used, one is the exploratory grid search where several factors were tested to see their behavior and to investigate in the search for the most suitable ones. With this, it was observed which factors were the best performers.

From there, another grid search was created to make a more specific search with the factors chosen by the exploratory grid search. Images were divided into 2 batches 2,400 images for training and 600 images for validattion. The training was performed using Google Colab platform with Python 3, Keras library version 2.1 and Tensorflow library version 1.15.



(a) Learning rate = 1×10^{-3}

(b) Learning rate = 1×10^{-4}

(c) Learning rate = 1×10^{-5}

Fig. 3. Graphics of training performance on the exploitation grid search.

To evaluate the classification performance, two parameters were used as thresholds, a score of 0.7 and IOU of 0.5, that will be used to filter the pure results of the model, those that did not meet the results provided by the model were not considered as detected. The accuracy, precision, recall, and f1-score metrics were calculated using a function made by us in python. In the calculation of the metrics the total number of labels was considered as the number of labels that were recognized of the images in which a tomato was detected and not the total number of images in the test set.

4.2 Exploratory Grid Search

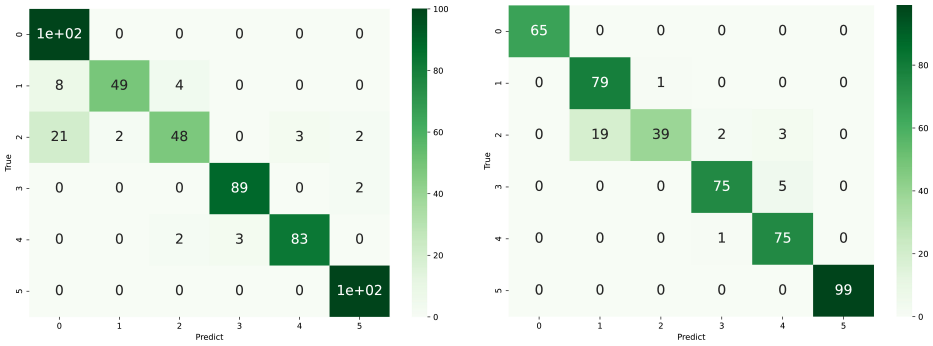
For the exploratory grid search, two hyperparameters were used as factors: the optimizer and the learning rate. The batch size of 64 and the number of epochs of 100 were kept as orthogonal factors. Since the computational cost was the reason for choosing 100 epochs to keep the training time within a reasonable time margin. The grid search is shown in table 2.

The training behavior of this exploratory grid search can be seen in Figure 2. It is observed that the loss curve when using the 1×10^{-3} learning rate, regardless of which optimizer is used, decays faster compared to the 1×10^{-5} rate.

With this first experiment (see table 3) the best performing was the ADAM optimizer with the learning rate of 1×10^{-3} obtaining a score of 0.93 in the f1-score. On the other hand, where the learning rate was 1×10^{-5} it was not possible be calculated, since the found weights were not enough to reach the thresholds established for performance

Table 5. Table of performance metrics obtained by the exploitation grid search.

Experimets	**Performance thresholds**		**Orthogonal Paremeter**			**Metrics**					**Label**
Learning rate	Score	IOU	Optimizer	Epochs	Batch size	Accuray	Precision	Recall	F1-score	Average IOU	Not detected
1×10^{-3}	0.7	0.5	ADAM	200	64	0.91	0.92	0.90	0.90	**0.87**	**84**
1×10^{-4}	0.7	0.5	ADAM	200	64	**0.93**	**0.94**	**0.92**	**0.92**	0.87	137
1×10^{-5}	0.7	0.5	ADAM	200	64	-	-	-	-	-	531



(a) (b)

Fig. 4. Confusion matrix of the two experiments, (a) confusion matrix for 1×10^{-3} experiment and (b) confusion matrix for 1×10^{-4} experiment. The number 0 represents the red class, 1 the red-orange, 2 orange, 3 striped, 4 salmon and 5 green.

measurement. The conclusion of this grid is that it is convenient to use a learning rate of 1×10^{-3} as it was the one with the best results. In the case of the ADAM optimizer, it is convenient to use it, since it was the one that had the best performance. With this, we perform an exploit grid search now keeping the Adam optimizer fixed using a series of learning rate values.

4.3 Exploitation Grid Search

For the exploitation grid search, only the learning rate was considered as a factor. The batch size of 64 and the number of epochs of 200 were kept as orthogonal factors. As in this grid, the number of experiments is smaller, therefore, it was decided to increase the number of epochs to 200 so that the training takes enough time to converge. And from the conclusions of the exploratory grid search, the ADAM optimizer will be kept fixed for being the one with the best performance.

The grid search is shown in the table 4. The training behavior of this exploitation grid search can be seen in Figure 3. The table of metrics obtained from the exploitation grid search is shown in table 5.

4.4 Analysis of Results

The results obtained in the grid search of exploitation (see table 5) the best metrics was the one with the learning rate of 1×10^{-4}, but with less amount of detected images.

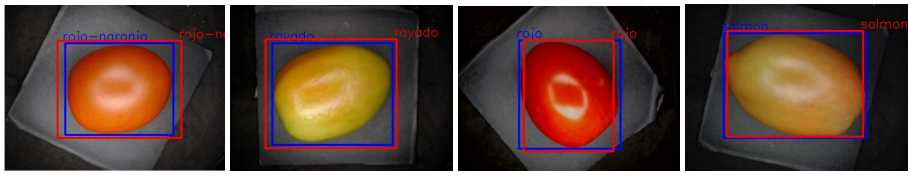


Fig. 5. Detection result with parameters: ADAM optimizer and learning rate 1×10^{-3}.

While, the 1×10^{-3} rate had slightly lower performance, it achieved the highest number of detected images. In addition, the average IOU in both models are identical, which indicate a similar behavior on average over all classes.

In the case of the rate of 1×10^{-5} due to the low number of images detected, we did not obtain its metrics. For a deeper analysis of the behavior of these two experiments, we present their confusion matrices (see figure 4). In these matrices, the labels were coded to numbers for better visualization.

With these matrices it is observed that the experiment with learning rate of 1×10^{-4} detects well the classes 0 and 5, on the other hand, the one with rate of 1×10^{-3} detects more images of that class but performs more classification errors. The experiment with rate of 1×10^{-3} detects more images of class 3 and 4 compared to the other one, but this one presents more errors when classifying these classes. Class 2 in both experiment had the highest error rate. So for us, the best experiment was the 1×10^{-5} rate experiment because it detects more images. Figure 5 shows an example of the detection results obtained by the best experiment.

5 Conclusion and Future Work

A tomato ripening stage detection system was constructed, obtaining an accuracy of 91.0% at the best score. The detection performance of the ripening stages was 90.0% in the f1 score. The relevance of our configuration is that the location of the tomatoes is very close to true, this is observed in the average IOU which means that when the tomato is detected the model is 87% sure of its location in the image.

The point of improvement for this model is to make it capable of detecting tomatoes in their natural habitat so that it can be used by a harvesting robot, as it is currently only capable of detecting them under very controlled conditions. In future work, a data augmentation optimization stage will be introduced to the system so that the model will be able to detect tomatoes in varying circumstances.

References

1. Servicios de Información Agroalimentaria y Pesquera (SIAP): Panorama agroalimentario. Agricultura, Secretaría de Agricultura y Desarrollo Rural, https://nube.siap.gob.mx/gobmx_publicaciones_siap (2021)
2. Ruíz-Martínez, J., Vicente, A.A., Montañéz-Saenz, J.C., Rodríguez-Herrera, R., Aguilar-González, C.N.: Un tesoro perecedero en México: El tomate, tecnologías para prolongar su vida de anaquel. Investigación y Ciencia: de la Universidad Autónoma de Aguascalientes, vol. 20, no. 54, pp. 57–63 (2012). DOI: 10.33064/iycuaa2012544087.

3. Thengane, V.G., Gawande, M.B.: Arduino Based Tomato Ripening Stages Monitoring System. International Journal of Innovative Research in Science, Engineering and Technology, vol. 7, no. 1, pp. 530–536, https://www.ijirset.com/upload/2018/january/13_4_Arduino.PDF (2018)
4. Hoffmann, A.M., Noga, G., Hunsche, M.: Fluorescence Indices for Monitoring the Ripening of Tomatoes in Pre-and Postharvest Phases. Scientia Horticulturae, vol. 191, pp. 74–81 (2015). DOI: 10.1016/j.scienta.2015.05.001.
5. Konagaya, K., Al-Riza, D.F., Nie, S., Yoneda, M., Hirata, T., Takahashi, N., Kuramoto, M., Ogawa, Y., Suzuki, T., Kondo, N.: Monitoring Mature Tomato (Red Stage) Quality During Storage Using Ultraviolet-Induced Visible Fluorescence Image. Postharvest Biology and Technology, vol. 160, no. 111031 (2020). DOI: 10.1016/j.postharvbio.2019.111031.
6. Aranda-Sanchez, J.I., Baltazar, A., González-Aguilar, G.: Implementation of a Bayesian Classifier Using Repeated Measurements for Discrimination of Tomato Fruit Ripening Stages. Biosystems Engineering, vol. 120, no. 3, pp. 274–284 (2009). DOI: 10.1016/j.biosystemseng.2008.12.005.
7. Kitpo, N., Kugai, Y., Inoue, M., Yokemura, T., Satomura, S.: Internet of Things for Greenhouse Monitoring System Using Deep Learning and bot Notification Services. In: IEEE International Conference on Consumer Electronics (ICCE), pp. 1–4 (2019). DOI: 10.1109/ICCE.2019.8661999.
8. Seo, D., Cho, B.H., Kim, K.C.: Development of Monitoring Robot System for Tomato Fruits in Hydroponic Greenhouses. Agronomy, vol. 11, no. 11, pp. 1–14 (2021). DOI: 10.3390/agronomy11112211.
9. Adarsh, P., Rathi, P., Kumar, M.: YOLO v3-Tiny: Object Detection and Recognition Using one Stage Improved Model. In: 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 687–694 (2020). DOI: 10.1109/ICACCS48705.2020.9074315.
10. El-Bendary, N., El-Hariri, E., Hassanien, A.E., Badr, A.: Using Machine Learning Techniques for Evaluating Tomato Ripeness. Expert Systems with Applications, vol. 42, no. 4, pp. 1892–1905 (2015). DOI: 10.1016/j.eswa.2014.09.057.
11. Luna, R., Dadios, E., Bandala, A., Vicerra, R.: Tomato Growth Stage Monitoring for Smart Farm Using Deep Transfer Learning with Machine Learning-Based Maturity Grading. AGRIVITA, Journal of Agricultural Science, vol. 42, no. 1, pp. 24–36 (2020). DOI: 10.17503/agrivita.v42i1.2499.
12. Liu, J., Pi, J., Xia, L.: A Novel and High Precision Tomato Maturity Recognition Algorithm Based on Multi-Level Deep Residual Network. Multimedia Tools and Applications, vol. 79, no. 13, pp. 9403–9417 (2020). DOI: 10.1007/s11042-019-7648-7.
13. Rangarajan, A.K., Purushothaman, R., Ramesh, A.: Tomato Crop Disease Classification Using Pre-Trained Deep Learning Algorithm. Procedia Computer Science, vol. 133, pp. 1040–1047 (2018). DOI: 10.1016/j.procs.2018.07.070.
14. Hong, H., Lin, J., Huang, F.: Tomato Disease Detection and Classification by Deep Learning. In: International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), pp. 25–29 (2020). DOI: 10.1109/ICBAIE49996.2020.00012.
15. Liu, G., Mao, S., Kim, J.H.: A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. Sensors, vol. 19, no. 9 (2019). DOI: 10.3390/s19092023.
16. Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., Wang, M.: Deep Learning Based Improved Classification System for Designing Tomato Harvesting Robot. IEEE Access, vol. 6, pp. 67940–67950 (2018). DOI: 10.1109/ACCESS.2018.2879324.
17. Mu, Y., Chen, T.S., Ninomiya, S., Guo, W.: Intact Detection of Highly Occluded Immature Tomatoes on Plants Using Deep Learning Techniques. Sensors, vol. 20, no. 10 (2020). DOI: 10.3390/s20102984.

18. Liu, G., Nouaze, J.C., Touko-Mbouembe, P.L., Kim, J.H.: Yolo-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. Sensors, vol. 20, no. 7, pp. 1–20 (2020). DOI: 10.3390/s20072145.
19. Martinez-Guevara, A.: Desarrollo e implementacion de un sistema inteligente para clasificacion de tomates (Solanum Lycopersicum). Universidad Autónoma de Chapingo (2021)
20. GitHub, Inc.: HumanSignal, https://github.com/heartexlabs/labelImg (2018)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016). DOI:10.1109/CVPR.2016.91.
22. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6517–6525 (2017). DOI: 10.1109/CVPR.2017.690
23. Farhadi, A., Redmon, J.: YOLOv3: An Incremental Improvement. Computer Vision and Pattern Recognition, pp. 1–6 (2018). DOI: 10.48550/arXiv.1804.02767.
24. Grandini, M., Bagli, E., Visani, G.: Metrics for Multi-Class Classification: An Overview. Machine Learning, pp. 1–17 (2020). DOI: 10.48550/arXiv.2008.05756.
25. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 658–666 (2019). DOI: 10.1109/CVPR.2019.00075.

ISSN 1870-4069

Determination of Hazardous Asteroids Using Machine Learning

Luis Vicarte, Juan Salas, Alfredo Murillo, Hiram Ponce

Universidad Panamericana,
Facultad de Ingeniería,
Mexico

{0199800, 0183674, 0252446, hponce}@up.edu.mx

Abstract. Potentially Hazardous Asteroid, or PHA, are near-Earth asteroids with a minimum orbital intersection distance of 0.05 AU or less with the Earth. This distance is roughly one-twentieth of the Earth-Sun mean distance, and it is thought to be the biggest conceivable orbital perturbation within a 100-year time frame that might result in a collision. This work proposes to build a machine learning model that can be used to provide a preventive forecast that confirms which asteroid are harmful to the Earth and which are not. After an analysis of a public dataset in asteroid and comparing four machine learning models, we found that the XGBoost model performs with 98% of accuracy. We use this model to find the most relevant features in the dataset, and we validated this information with the literature.

Keywords: Asteroids, artificial intelligence, support vector machines, XGBoost, classification

1 Introduction

Potentially Hazardous Asteroid, or PHA, are near-Earth asteroids with a minimum orbital intersection distance of 0.05 AU or less with the Earth. This distance is roughly one-twentieth of the Earth-Sun mean distance, and it is thought to be the biggest conceivable orbital perturbation within a 100-year time frame that might result in a collision [1].

PHAs account for around 20 percent of near-Earth asteroids. Asteroids are thought to have a chance of colliding with Earth, causing damage ranging from minor local destruction to mass extinction [2].

The fall of rock or iron asteroids greater than 50 m in diameter occurs every hundred years on average, causing local disasters and tidal waves. Asteroids greater than a kilometer create global disasters every few hundred thousand years. In the latter instance, the debris from the collision spreads across the Earth's atmosphere, causing acid rain, partial sunlight interruption, and massive flames created by high-temperature particles that fall to the Earth following the collision [3].

We consider these are scientifically isolated phenomena with significant significance for our world. As part of our study, we propose to identify those that are particularly hazardous to the globe and develop a machine learning model that can reliably

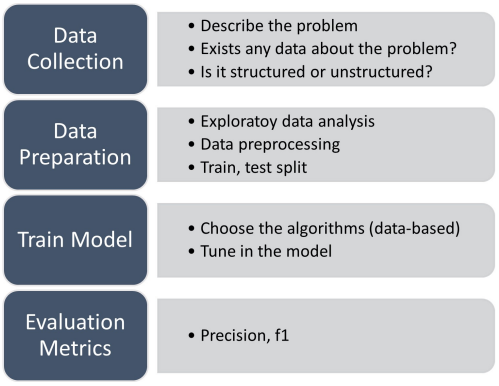


Fig. 1. Machine Learning Workflow.

anticipate them. According to new data from the Jet Propulsion Laboratory, Earth is surrounded by about 28,000 asteroids of all forms and sizes.

However, no evidence exists that any are now in the process of colliding with our planet [4]. This work proposes to build a machine learning model that can be used to provide a preventive forecast that confirms which asteroids are harmful to the Earth and which are not.

The rest of the paper is organized as follows. Section 2 summarizes the related work using machine learning for asteroids identification. Section 3 describes the methodology of the work, including the exploratory analysis over the dataset used, the data prepatation, the machine learning models and the experimentation. Section 4 shows the experimental results and discussion. Finally, Section 5 concludes the paper.

2 Related Work

Machine learning algorithms have recently been utilized to identify members of asteroid families and identify resonant argument pictures of asteroids in three-body resonances, among other uses, in the field of asteroid identification [5]. In comparison, machine learning applications to Solar System bodies, a broader topic that encompasses imaging and spectrophotometry of tiny bodies, have previously been classed as in progress [5].

The application of machine learning leads to the identification of new celestial objects or features, and research groups and procedures are more established. Machine learning is commonly used to study asteroid dynamics, although it is still in its infancy, with smaller groups and fewer articles yielding results [6].

Large observational studies of asteroids, like as those carried out at the Vera C. Rubin Observatory, will yield crucial data sets on their physical and orbital features [7]. ML applications are now being developed for clustering, picture recognition, and anomaly detection, among other things, and are predicted to be quite useful [8].

In contrast, our study focuses primarily on a mathematical and computational approach rather than astrophysics or space. Many solutions to these difficulties include imaging and visualization challenges, which our project does not consider at all and is exclusively dependent on machine learning model execution.

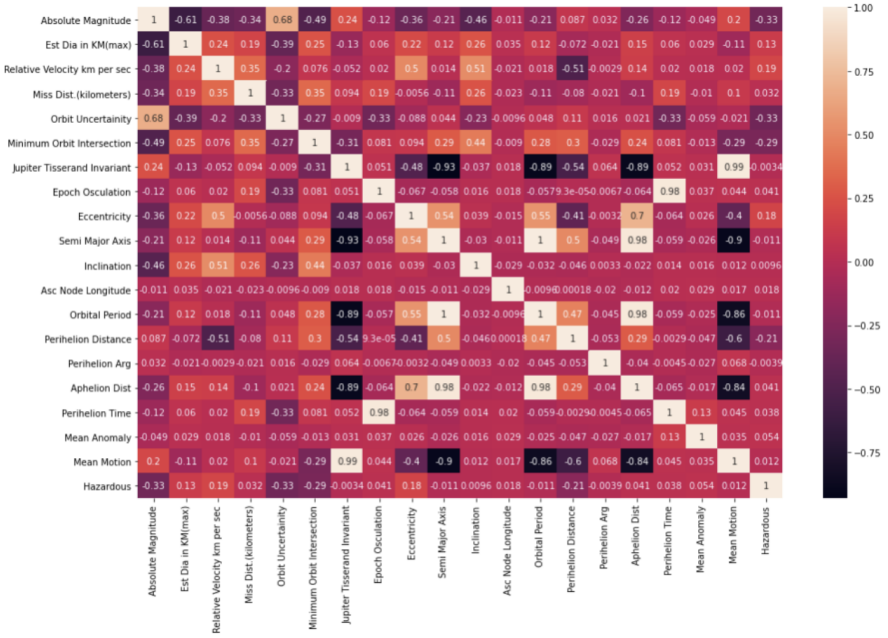


Fig. 2. Correlation Analysis.

3 Methodology

We adopt the classical workflow of machine learning, as depicted in Fig. 1. It comprises four main steps: data collection, data preparation, train model, and evaluation model.

3.1 Data Collection

The data is about Asteroids and is provided by NEOWS (Near-Earth Object Web Service[1]). This collection contains 4,687 rows and 41 columns.

In summary, our variables include the asteroid's name, diameter, proximity to the Earth, inclination, and distance from the Earth's orbit, among others. The variables are mostly integers, floats, and strings.

3.2 Data Preparation

The exploratory data analysis began with a review of null or zero values, which were replaced by their mean to be used in the model; it was also reviewed if the number of nulls was not very significant, eliminating these values to eliminate any possibility of noise and that they did not affect the total number of classes and asteroids; and it was also reviewed if the number of nulls was not very significant, eliminating these values to eliminate any possibility of noise and that they did not affect the total number of classes and aster.

[1] http://neo.jpl.nasa.gov

Table 1. Evaluation of the models.

Model	Accuracy	Precision	Recall	F1-score
LR	0.95	0.89	0.91	0.90
SVM	0.96	0.90	0.90	0.90
XGBoost	1.0	0.99	1.0	0.99
MLP	0.95	0.90	0.91	0.91

Following that, we conducted a correlation analysis of variables, as shown in Fig. 2, discovering that the mean movement is connected to Jupiter's invariant; oscillation is related to time; and the period of orbit is related to distance, among others.

We deleted the data columns for the variables that were not important after selecting the most and least important variables, which were: Unnamed; Neo Reference ID; Est Day in Miles, min, max; Orbiting distance; Relative velocity in kilometers per hour.

The "Hazardous" variable was then label encoded such that the model 0 or 1 informs us if it is harmful to the earth or not. The preceding is critical in ensuring that the model functions properly and that the data is normalized.

3.3 Machine Learning Models

We split the data into 80% for training and 20% for testing. We consider four machine learning models for our proposal:

- *Logistic regression (LR)*– it is a supervised learning model that allows classification based on the transformation of numerical values into categories (or classes) using the sigmoidal function. The training of a logistic regression uses the error function called logistic loss. It penalizes wrong predictions rather than reward correct predictions [9].
- *Support Vector Machines (SVM)*– Support vector machine (SVM) is a computer algorithm that learns by example to assign labels to objects!. For instance, an SVM can learn to recognize fraudulent credit card activity by examining hundreds or thousands of fraudulent and nonfraudulent credit card activity reports. Alternatively, an SVM can learn to recognize handwritten digits by examining a large collection of scanned images of handwritten zeroes, ones and soporth. SVMs have also been successfully applied to an increasingly wide variety of biological applications [10].
- *XGBoost*–xgboost is short for eXtreme Gradient Boosting package. It is an efficient and scalable implementation of gradient boosting framework. The package includes efficient linear model solver and tree learning algorithm. It supports various objective functions, including regression, classification and ranking [11].
- *Multilayer Perceptron (MLP)*–The perceptron algorithm is due to Rosenblatt in the late 1950s. The perceptron, a simple computing engine which has been dubbed a 'linear machine' is best related to supervised classification [12].

3.4 Experimentation

We use the following metrics to compare the performance of the models: accuracy (1), precision (2), recall (3), and F1-score (4); where, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively:

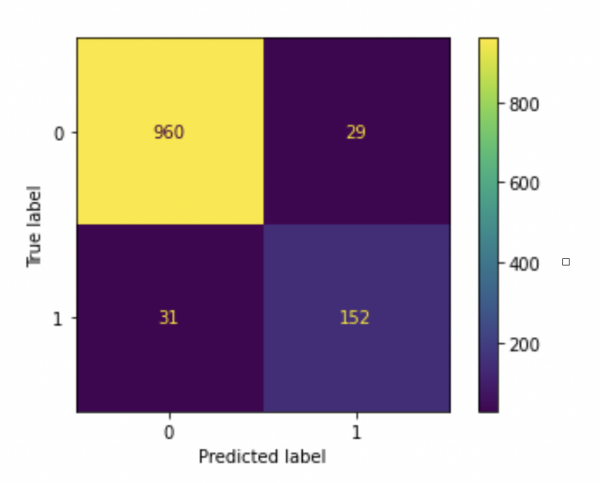


Fig. 3. Confusion matrix of the XGBoost model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{4}$$

4 Experimental Results

We conduct a 5-fold cross validation in each of the models, and Table 1 summarizes the results of the models in terms of accuracy, precision, recall, and F1-score.

As shown in the results, the XGBoost and the multilayered Perceptron models produced the greatest outcomes. In Fig. 3, we show the confusion matrix obtained by the predictions of the best model found so far.

Moreover, we found the most relevant variables in the data set using the XGBoost model, especifically its feature importance option, as shown in Fig. 4. They include the following:

- Absolute magnitude: the magnitude that a star would appear to have if it were located at a standard distance of 10 parsecs.
- Minimum orbit intersection: measure used in astronomy to assess potential close approaches and collision risks between astronomical objects.
- Inclination: measures the tilt of an object's orbit around a celestial body.
- Aphelion distance: the point in the orbit of an object where it is farthest from the Sun.

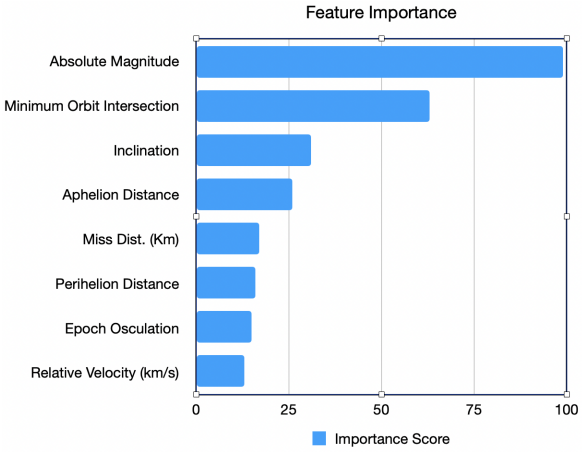


Fig. 4. Analysis of feature importance.

- Miss distance: the maximum distance at which the explosion of an artifact head can be expected to seriously damage its target.
- Perihelion distance: the point in orbit where an object is nearest to the sun.
- Epoch osculation: the instant of time at which the position and velocity vectors are specified.
- Relative velocity km per sec: the velocity of an object B in the rest frame of another object A.

From an astronomical standpoint, two primary characteristics are examined for an asteroid to be deemed dangerous, among which the following stand out: absolute magnitude and minimum orbit intersection, among many other considerations [13]. We may conclude from the above that our primary variables make sense in the real world.

5 Conclusions

In this work, we proposed to build a machine learning model that can be used to provide a preventive forecast that confirms which asteroids are harmful to the Earth and which are not. To do so, we identified the most important variables, ran tests with various classification models, and finally chose the best ones that produced high efficiency in this problem. As a result, we have results ranging from 94% to 100%.

For future work, we consider it is critical to continue gathering data on asteroids, feeding the model, and testing alternative algorithms in order to develop a more accurate method of detecting harmful asteroids for the planet in future research.

References

1. NASA Jet Propulsion Laboratory: NASA System Predicts Impact of Small Asteroid. Asteroids and Comets, https://www.jpl.nasa.gov/news/nasa-system-predicts-impact-of-small-asteroid/ (2022)

2. Bland, P.A., Artemieva, N.A.: The Rate of Small Impacts on Earth. Meteoritics and Planetary Science, vol. 41, no. 4, pp. 607–631 (2006). DOI: 10.1111/j.1945-5100.2006.tb00485.x.
3. NASA Jet Propulsion Laboratory: NASA's Next-Generation Asteroid Impact Monitoring System Goes Online. Asteroids and Comets, https://www.jpl.nasa.gov/news/nasas-next-generation-asteroid-impact-monitoring-system-goes-online (2021)
4. Carruba, V., Aljbaae, S., Domingos, R.C., Lucchini, A., Furlaneto, P.: Machine Learning Classification of New Asteroid Families Members. Monthly Notices of the Royal Astronomical Society, vol. 496, no. 1, pp. 540–549 (2020). DOI: 10.1093/mnras/staa1463.
5. Carruba, V., Aljbaae, S., Domingos, R.C., Barletta, W.: Artificial Neural Network Classification of Asteroids in the M1: 2 Mean-Motion Resonance with Mars. Monthly Notices of the Royal Astronomical Society, vol. 504, no. 1, pp. 692–700 (2021). DOI: 10.1093/mnras/stab914.
6. Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., Estévez, P.A., Sánchez-Sáez, P., Arredondo, J., Bauer, F.E., Carrasco-Davis, R., Catelan, M., Elorrieta, F., Eyheramendy, S., Huijse, P., Pignata, G., Reyes, E., Reyes, I., Rodríguez-Mancini, D., Ruz-Mieres, D., Valenzuela, C., Álvarez-Maldonado, I., Astorga, N., et al.: The Automatic Learning for the Rapid Classification of Events (ALeRCE) Alert Broker. The Astronomical Journal, vol. 161, no. 5 (2021). DOI: 10.3847/1538-3881/abe9bc.
7. Goldstein, M., Uchida, S.: A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PloS One, vol. 11, no. 4 (2016). DOI: 10.1371/journal.pone.0152173.
8. Wright, R.E.: Logistic Regression. Reading and Understanding Multivariate Statistics, pp. 217–244 (1995)
9. Noble, W.S.: What is a Support Vector Machine? Nature Biotechnology, vol. 24, no. 12, pp. 1565–1567 (2006). DOI: 10.1038/nbt1206-1565.
10. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K.: Xgboost: eXtreme Gradient Boosting. R package version 0.4-2, vol. 4, no. 1, pp. 1–4 (2015)
11. Murtagh, F.: Multilayer Perceptrons for Classification and Regression. Neurocomputing, vol. 2, no. 5-6, pp. 183–197 (1991) DOI: 10.1016/0925-2312(91)90023-5.
12. Shustov, B.M., Shugarov, A.S., Naroenkov, S.A., Prokhorov, M.E.: Astronomical Aspects of Cosmic Threats: New Problems and Approaches to Asteroid—Comet Hazard Following the Chelyabinsk Event of February 15, 2013. Astronomy Reports, vol. 59, no. 10, pp. 983–996 (2015). DOI: 10.1134/S1063772915100066.

ISSN 1870-4069

A Novel Machine Learning Method Applied to the Forecast of a Stock Market Index

Enrique González-Nuñez, Luis A. Trejo

Tecnologico de Monterrey,
Escuela de Ingeniería y Ciencias,
Mexico

A00457801@itesm.mx, ltrejo@tec.mx

Abstract. The present paper recalls the aim of the ongoing work of applying the Artificial Organic Networks (AON) machine learning framework, to develop a new algorithm capable of generating a prediction for a stock market index, based on the Index Tracking Problem (ITP); thus, a conception of a new AON arrangement is needed. Pursuing this main goal, a first approach toward the definition of a new topology is presented, stating some general ideas along with the following discussion. Finally, we offered some preliminary results related to these main notions, considering the employment of a multiple non-linear regressive (MNLR) model, to build an AON structure; the relative error obtained along the experiments was of the order of $1x10^{-2}$.

Keywords: Machine learning, nature-inspired, metaheuristic, stock market index, forecast.

1 Introduction

The stock market prediction or Index Tracking Problem (ITP), is a complex process affected by dynamic and non-linear factors across time, and yet an important research topic in financial area with big challenges [2–4, 6–8, 21–23, 25]. The ITP is a trading strategy based on the buy-and-hold of assets, that uses an index tracker to reproduce the performance of a stock market index or any other security found in the capital markets; the behavior of the performance is reproduced by developing models capable of yielding a forecast.

1.1 Related Work

Previously, in [5] has been reported the aim of applying the Artificial Organic Networks (AON) metaheuristic machine learning framework, to develop a new efficient algorithm, able of generating a short-term market trend forecast; the complexity of the problem was narrowed down by two main constraints:

1 The forecast would be done using the historic prices of the concerning index rate and at least two additional macroeconomic variables (MEVs).
2 The MEVs would be selected based on their correlated coefficient (CC) to the index being analyzed.

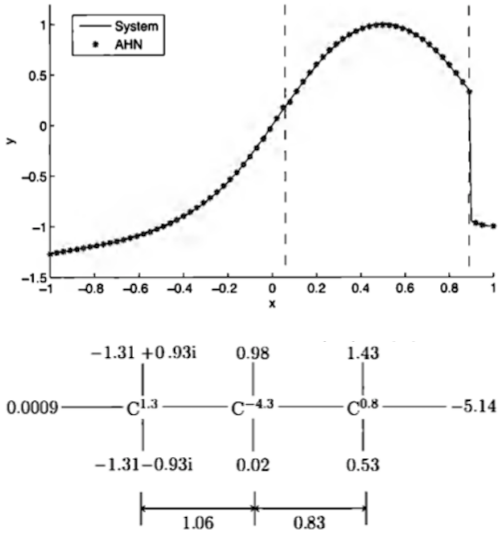


Fig. 1. Example [19, 20] of the AHN process as a learning method, where an AON structure is built through the algorithm identified as f. Along the process, a structure of an organic compound is produced by segmenting a dataset (target function f) and fitting a second-degree or third-degree polynomic term for each section.

2 Methodology

Following the notions and main characteristics of AON as a machine learning class [19–20], a new algorithm is to be defined. Within its characteristics, AON requires to use of a topological configuration for its implementation; one main limitation of this method is that it only has been implemented through one existing topology, Artificial Hydrocarbon Networks (AHN), which have shown improvements in predictive power and interpretability in contrast with other well-known machine learning models, comparatively to neural networks and random forest, but has the disadvantage of being very time-consuming and is not able to deal big data since the model uses stochastic gradient descent (SGD).

AHN as a chemically bio-inspired algorithm performs an optimization of a cost-energy function in two levels to build organic compound structures:

1 It uses least-squares regression (LSR) to define the structure of molecules.
2 It uses SGD to optimize the position of the molecules in the feature space.

2.1 Conception of a New Topology

Since AHN is the only existing arrangement for the AON framework, the postulation of the new algorithm adept to perform the stated objectives, is going to be based on the conception of a different topology to provide better capability to deal with big data and reduce time consumption, to avoid losing predictive power as one of the two main characteristics of the original AHN topology mentioned above.

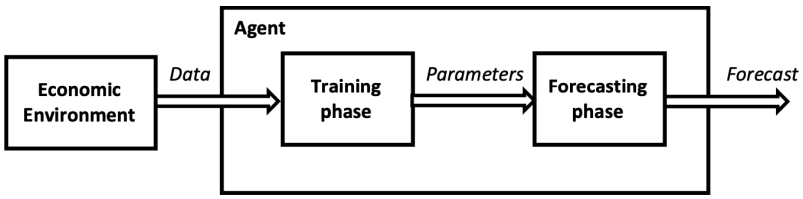


Fig. 2. Diagram of the proposed agent organized in two phases; this new topology would be capable of performing a forecast for a stock market index, using at least two additional MEVs chosen based on their CC to the index being analyzed. Along the process, a dataset will be received and segmented by the training phase, for each section a curve would be fitted using MNLR to compute the structure of an organic compound. The parameters output of the training phase are the values to build the AON structure that models a given system (function f).

Because the research presented here is still ongoing, only a general approach to a future topology is presented as a follow-up of the work introduced in [5], being aware that further experiments are being conducted to formalize the details of the new algorithm.

In the simplest notion, as explained in the literature [19–20], the algorithm that builds the structure of an AON as a learning method is identified as f, and through the AHN topology produces a structure of an organic compound by segmenting the dataset of the information received and fitting a second-degree or third-degree polynomic term for each section utilizing LSR (see Figure 1).

Consequently, the general approach of a different topology is mainly driven -up to now- by the next main ideas:

1 To define a new AON arrangement based on a functional group different from the hydrocarbons.
2 The new organic structure will fit the information of each segment by using a multiple non-linear regressive (MNLR) model [5], using three variables: the historic prices of the index, and a minimum of two additional MEVs chosen depending on their CC to the index rate.
3 As a significant feature to characterize a new AON arrangement, it will substitute the type of curve that is fitted when a polynomic term is computed for each segment of the dataset that is being modeled.

3 Exploratory Analysis & Preliminary Results

In this work, the main concepts of AON are recovered to postulate a new topology through the implementation of an agent [5] that receives external data from the economic environment; the agent will be formed in two phases (see Figure 2):

1 The training phase: will find the parameters to build the structure of an AON.
2 The forecasting phase: will estimate a prediction of a stock market, using the AON structure.

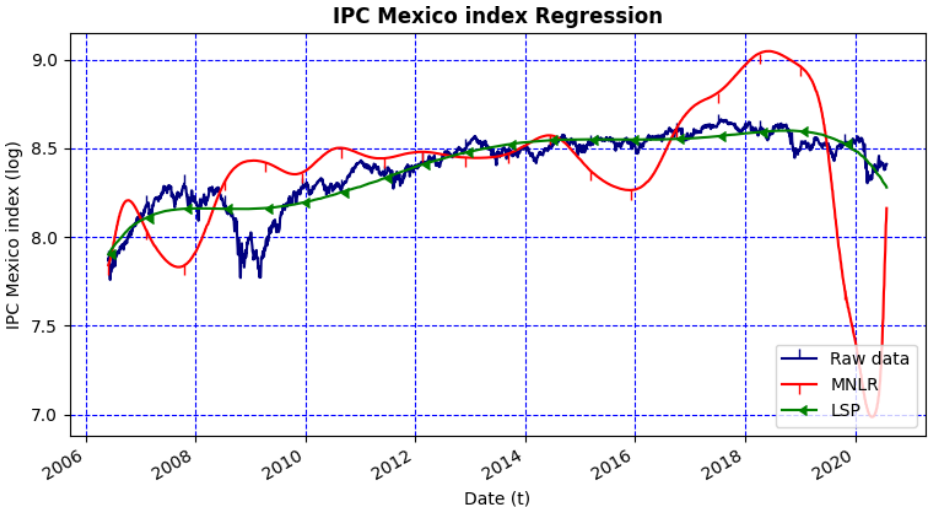


Fig. 3. Curves for the IPC Mexico using the original data (blue line), a MNLR model that replaces the second-degree term with a SIN term (red line), and the LSP model (green line).

In Python, some preliminary experiments have been carried out [5] and still are being performed, to evaluate the viability and effectiveness of conceptualizing a new AON arrangement by applying an MNLR method to build an organic compound structure, while substituting the type of curve that is fitted when a polynomic term is computed for each segment along the process. These backtest experiments have been done using the SciPy [24], and Scikit-learn [9] libraries.

The tests have been carried out using existing data from Mexico comprehending the period from 1/6/2006 to 30/7/2020, the cognition behind the size of data selected is to assure that at least one short economic cycle is used [5].

The dataset includes the following variables: the daily reported (labor days) IPC Mexico stock market index, the quarterly reported gross domestic product (GDP), the daily reported (labor days) MXN-USD foreign exchange rate (FX), the monthly reported consumer price index (CPI), the monthly risk-free rate (RFR), the monthly unemployment rate (UR), the monthly reported current account to GDP rate (BOP), and the monthly reported Investment rate (GFCF). The IPC data was retrieved from Yahoo Finance, the FX was acquired from the USA's Federal Reserve Board, and the rest of the variables were obtained from the OECD.

It is relevant to explain that -up to now- the experiments have been done randomly splitting the values of the data into subsets of 80% for training and the rest for testing, as commonly happens in machine learning, since the approach is to find good solutions by reducing the computing time, in contrast of the classic statistical model like ARIMA where the data is split based on the DateTime [1, 25].

Despite being an unusual practice for time series, good performance has been obtained previously with this approach [5]. Also, it would be noticed that the tests include an approximation using a least-squares polynomial regression (LSP) based only on the historic data of the IPC Mexico, this was for baseline purposes.

Table 1. Mean, SD, MAD, Max, Min, and Range of the relative error from the MNLR model using a SIN term.

Relative Error (MNLR)					
Mean	**SD**	**MAD**	**Max**	**Min**	**Range**
0.029072	0.031475	0.021225	0.167772	0.000013	0.167758

Table 2. RSS, SSR, TSS, and R-square of the MNLR model using a SIN term. It can be observed that SSR is larger than SSE, following one of the criteria of a good regression model.

Relative Error (MNLR)			
RSS	**SSR**	**TSS**	**R-square**
367.413282	418.723554	786.136836	0.532634

For each experimentation, a MNLR model of the IPC was computed employing the previously defined seven MEVs (FX, GDP, CPI, RFR, UR, BOP, and GFCF); the data was preprocessed in three steps:

1 The MEVs were treated as "continuous signals", so for each input, an independent approximation was done.
2 The data was standardized by removing the mean.
3 The dimensionality of the data was reduced using principal component analysis (PCA), this was done using three components.

Furthermore, is important to remark here that being an initial approach, the dataset was not -yet- segmented as the AON framework would formally do. The experiments considered most relevant will be described now in the subsequent sections.

3.1 Experiment 1: Establishing a MNLR Model Using SIN Terms

Exploring the possibility of changing the type of curve that is fitted by a molecule for each segment, when an AON structure is computed, the MNLR model for the IPC Mexico was estimated by replacing the second-degree polynomic terms with a sinusoidal (SIN) term; Figure 3 shows the graph of the obtained curve using this approach. As expected, in the graph discrepancies can be appreciated across the time (t) between the original curve of the IPC Mexico and the computed MNLR model.

Nonetheless, as stated above, the dataset was not -yet- segmented as the AON framework would formally do, considering this is a significant factor for the discrepancies. It is considered that in the future these divergences will be reduced once the data is segmented; moreover, in the graph can be observed along (t) that in some intervals (e.g., around the years 2013-2014) the approximated model behavior is very much the same as the trend of the original data.

Despite the discrepancies, the estimation provided satisfactory results based on the the relative error ε_t [5]; in this regard, Table 1 shows the mean, the standard deviation (SD), the mean absolute deviation (MAD), the maximum (Max) value, the minimum (Min) value, and the range of the obtained relative error. Figure 4 shows the relative error of the MNLR model using a SIN term.

The performance of the model was also measured by computing the Residual Sum of Squares (RSS), the Sum of Squares Regression (SSR), the Total Sum of Squares (TSS), and the coefficient of determination R-square; the results are reported in Table 2.

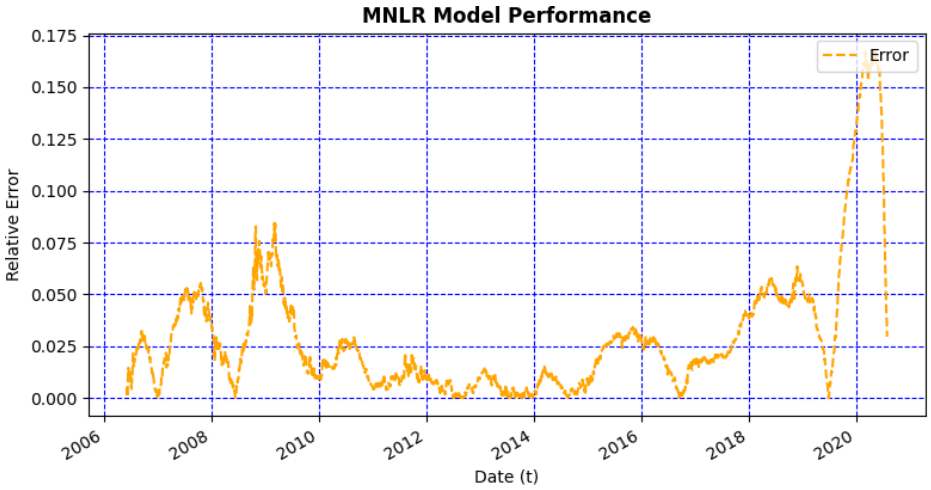


Fig. 4. Relative error for the estimated IPC using the MNLR model and replacing the second-degree term with a SIN term.

Table 3. Mean, SD, MAD, Max, Min, and Range of the relative error from the MNLR model using an EXP term.

Relative Error (MNLR)					
Mean	**SD**	**MAD**	**Max**	**Min**	**Range**
0.016755	0.014252	0.011017	0.063554	0.00002	0.063534

3.2 Experiment 2: Establishing a MNLR Model Using EXP Terms

Again, to test the possibility of changing the type of curve that is fitted by a molecule for each segment, when an AON structure is computed, in experiment two the MNLR model was assessed by replacing the second-degree polynomic terms with an exponential (EXP) term; Figure 5 shows the graph of the curve attained using this approach. Once more, in the graph are appreciated discrepancies across (t) between the original curve and the computed MNLR model.

As before, the data was not -yet- segmented as the AON framework would formally do, considering this as an important factor for the discrepancies. As well, it is considered that in the future these divergences will also be reduced once the data is segmented; likewise, in the graph can be observed along (t) that in some intervals (e.g., around the years 2012-2013, and 2020) the approximated model behavior is very much the same as the trend of the original data.

Once more, after estimating the model, the relative error was computed; however, in this case, the performance increased in comparison to the results acquired in the first experiment. Table 3 shows the mean, the SD, the MAD, the Max value, the Min Value, and the range achieved by the relative error in this case. Figure 6 shows the relative error of the MNLR model using an EXP term.

Subsequently, the performance of the model was also measured by computing the RSS, the SSR, the TSS, and the R-square; the results are reported in Table 4.

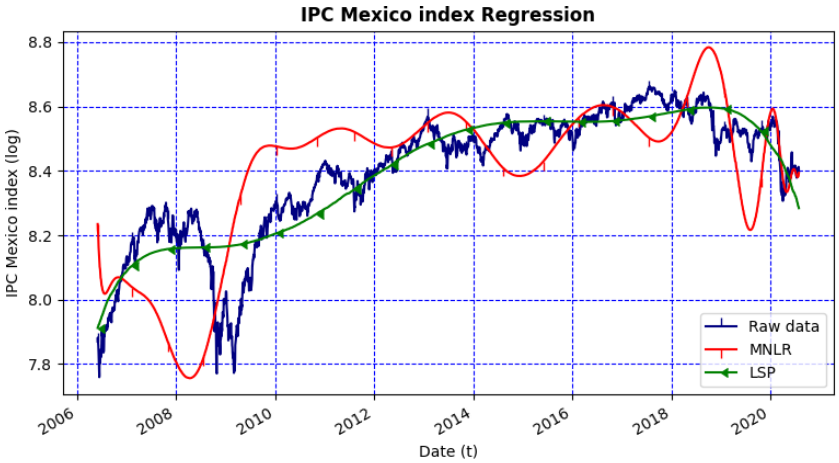


Fig. 5. Curves for the IPC Mexico using the original data (blue line), a MNLR model that replaces the second-degree term with an EXP term (red line), and the LSP model (green line).

Table 4. RSS, SSR, TSS, and R-square of the MNLR model using an EXP term. It can be observed that SSR is larger than SSE, following one of the criteria of a good regression model.

Relative Error (MNLR)			
RSS	**SSR**	**TSS**	**R-square**
93.568285	168.641547	262.209833	0.643155

4 Conclusions & Future Work

Through this paper is evoke the objective of producing a new algorithm based on the AON machine learning class, to yield a short-term stock market trend forecast, using at least two additional MEVs.

As stated, the AON compelling concepts are observed to design a new AON topology; in this respect, contemplating that the AHN topology produces a structure by segmenting the dataset, and fitting a curve to each section, the present state of the ongoing research has been focused on undertaking experiments to identify different type of math expressions for substituting the second-degree and third-degree polynomic terms employed by the AHN algorithm.

As a first approach, sinusoidal and exponential terms have been used in the MNLR model to replicate the behavior of the IPC Mexico. Through the first approximations exemplified here, disparities have been observed between the acquired results and the raw data; however, it has been remarked that the procedure applied still has not segmented the data as the AON framework do.

In addition to the last statement, the obtained results from the experiments have provided a relative error of the order of $1x10^{-2}$; regarding the last remarks, is being pondered that is plausible to define in further experiments a new AON arrangement, by using the MNLR model and substituting the type of math expression to fit different curves of the segments along (t) of the function f.

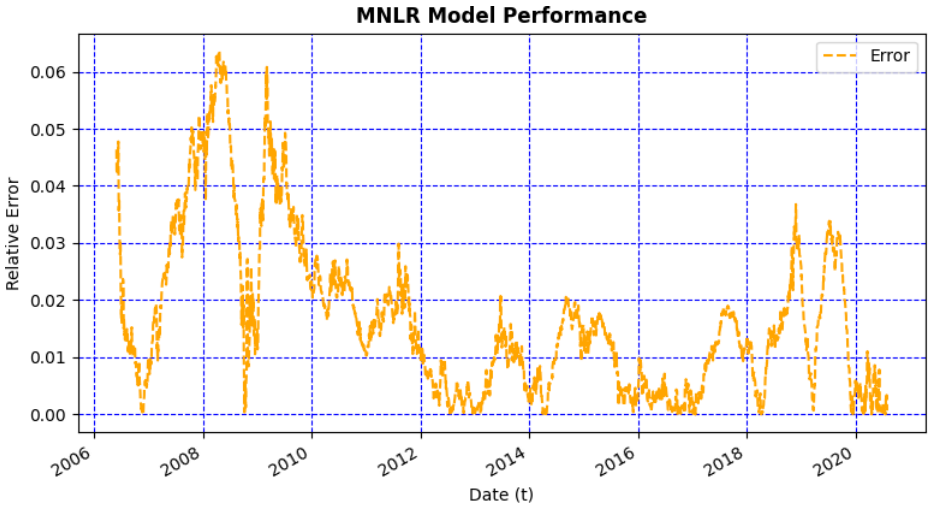


Fig. 6. Relative error for the estimated IPC using the MNLR model and replacing the second-degree term with an EXP term.

To improve the performance and reduce the discrepancies of the results obtained by now, the next tasks considered for future work include: i) segmenting the data as the AON framework formally does while producing a structure of an organic compound, ii) doing further experiments considering different kinds of math expressions that are used to characterize curves, iii) the procedure of splitting the data into train and test subsets set up on sequence (based upon the DateTime).

Aknowledgments. Supported by Tecnológico de Monterrey and CONACyT.

References

1. Aronson, L.: ARIMA Modeling and Train/Test Split. Learn (2020)
2. Chacón, H., Kesici, E., Najafirad, P.: Improving Financial Time Series Prediction Accuracy Using Ensemble Empirical Mode Decomposition and Recurrent Neural Networks. In: IEEE Access, vol. 8, pp. 117133–117145 (2020). DOI: 10.1109/ACCESS.2020.2996981.
3. Elliott, G., Granger, C., Timmermann, A.G.: Handbook of Economic Forecasting. North-Holland, vol. 1 (2013)
4. Focardi, S.M., Fabozzi, F.J.: The Mathematics of Financial Modeling and Investment Management. The Frank J Fabozzi Series, Wiley Finance (2004)
5. González, E., Trejo, L.: Artificial Organic Networks Approach Applied to the Index Tracking Problem. In: Mexican International Conference on Artificial Intelligence, pp. 23–43 (2021). DOI: 10.1007/978-3-030-89817-5 2.
6. Hou, X., Wang, K., Zhang, J., Wei, Z.: An Enriched Time-Series Forecasting Framework for Long-Short Portfolio Strategy. In: IEEE Access, vol. 8, pp. 31992–32002 (2020). DOI: 10.1109/ACCESS.2020.2973037.
7. Ordóñez, J.M.: Predicción del comportamiento de los mercados bursátiles usando redes neuronales. Technical report, Depto. Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, pp. 1–129 (2017)

8. Ortiz, F., Cabrera-Llanos, A.I., López-Herrera, F.: Pronóstico de los índices accionarios DAX y S&P 500 con redes neuronales diferenciales. Contaduría y administración, vol. 58, no. 3, pp. 203–225 (2013). DOI: 10.1016/S0186-1042(13)71227-0.
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Matthieu-Perrot, E., Duchesnay: Scikit-learn: Machine learning in Python (2017)
10. Ponce, H., Acevedo, M.: Design and Equilibrium Control of a Force-Balanced One-Leg Mechanism. In: Advances in Computational Intelligence (MICAI), pp. 276–290 (2018). DOI: 10.1007/978-3-030-04497-8_2.
11. Ponce, H., Acevedo, M., Morales-Olvera, E., Martínez-Villaseñor, L., Díaz-Ramos, G., Mayorga-Acosta, C.: Modeling and Control Balance Design for a New Bio-Inspired Four-Legged Robot. In: Advances in Soft Computing (MICAI), vol. 11835, pp. 728–739 (2019). DOI: 10.1007/978-3-030-33749-0_58.
12. Ponce, H., Campos-Souza, P.V., Junio-Guimarães, A., González-Mora, G.: Stochastic Parallel Extreme Artificial Hydrocarbon Networks: An Implementation for Fast and Robust Supervised Machine Learning in High-Dimensional Data. Engineering Applications of Artificial Intelligence, vol. 89 (2020). DOI: 10.1016/j.engappai.2019.103427.
13. Ponce, H., González-Mora, G., Morales-Olvera, E., Souza, P.: Development of Fast and Reliable Nature-Inspired Computing for Supervised Learning in High-Dimensional Data. In: Nature Inspired Computing for Data Science, vol. 871, pp. 109–138 (2019). DOI: 10.1007/978-3-030-33820-6_5.
14. Ponce, H., Martinez-Villaseñor, M.L.: Interpretability of Artificial Hydrocarbon Networks for Breast Cancer Classification. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 3535–3542 (2017). DOI: 10.1109/IJCNN.2017.7966301.
15. Ponce, H., Martínez-Villaseñor, M.L.: Versatility of Artificial Hydrocarbon Networks for Supervised Learning. In: Advances in Soft Computing (MICAI), pp. 3–16 (2018). DOI: 10.1007/978-3-030-02837-4_1.
16. Ponce, H., Martínez-Villaseñor, M.L., Miralles-Pechuán, L.: A Novel Wearable Sensor-Based Human Activity Recognition Approach Using Artificial Hydrocarbon Networks. Sensors, vol. 16, no. 7 (2016). DOI: 10.3390/s16071033.
17. Ponce, H., Miralles-Pechúan, L., Martínez-Villaseñor, M.L.: Artificial Hydrocarbon Networks for Online Sales Prediction. In: Advances in Artificial Intelligence and Its Applications (MICAI), pp. 498–508 (2015). DOI: 10.1007/978-3-319-27101-9 38.
18. Ponce, P., Ponce, H., Molina, A.: Doubly Fed Induction Generator (DFIG) Wind Turbine Controlled by Artificial Organic Networks. Soft Computing, vol. 22, pp. 2867–2879 (2018). DOI: 10.1007/s00500-017-2537-3.
19. Ponce-Espinosa, H., Ponce-Cruz, P., Molina, A.: Artificial Organic Networks: Artificial Intelligence Based on Carbon Networks. Springer, vol. 521 (2014). DOI: 10.1007/978-3-319-02472-1.
20. Ponce-Espinosa, H.E.: A New Supervised Learning Algorithm Inspired on Chemical Organic Compounds. Ph.D. Thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey (2013)
21. Sheta, A.F., Ahmed, S., Faris, H.: Evolving Stock Market Prediction Models Using Multi-Gene Symbolic Regression Genetic Programming. Artificial Intelligence and Machine Learning (AIML), pp. 11–20 (2015)
22. Soler-Dominguez, A., Juan, A.A., Kizys, R.: A Survey on Financial Applications of Metaheuristics. ACM Computing Surveys, vol. 50, no. 1, pp. 1–23 (2017). DOI: 10.1145/3054133.
23. Stoean, C., Paja, W., Stoean, R., Sandita, A.: Deep Architectures for Long-Term Stock Price Prediction with a Heuristic-Based Strategy for Trading Simulations. PLoS ONE, vol. 14, no. 10 (2019). DOI: 10.1371/journal.pone.0223593.

24. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van-Der–Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R., Jones, E., Kern, R., Larson, E., Carey, C.J., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, vol. 17, pp. 261–272 (2020). DOI: 10.1038/s41592-019-0686-2.
25. Zheng, X., Chen, B.M.: Stock Market Modeling and Forecasting: A System Adaptation Approach. Springer, (2013). DOI: 10.1007/978-1-4471-5155-5.

ISSN 1870-4069

Design of a Soft Emotion Sensor for Food Recommendation Using Deep Learning

Alberto Espinosa-Juárez, Marco A. Moreno-Armendáriz

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciencias Cognitivas Computacionales,
Mexico

{aespinosaj2021, mam_armendariz}@cic.ipn.mx

Abstract. The analysis of emotions and moods in people allows us to know, among many things, tastes, which allows us to offer personalized products or services. This article shows a way of detecting and classifying people's moods using selfie photographs for subsequent use in restaurant dish recommenders. For this solution we propose to make use of two neural networks; the first one will be used for the detection of Action Units (AU) present in the works of Paul Ekman, and a second neural network to classify the correct mood once the AUs have been previously detected and classified.

Keywords: Emotions, facial, classification, neural networks.

1 Introduction

The emotions are present in us as humans at all times and is often the reason for choices in different areas of our lives, from whether or not to go out with a friend to whether or not to see a movie.

Paul Ekman, the pioneering psychologist in the study of human emotions and expressions, mentions in his numerous investigations the importance of studying them to understand part of human behavior and, in his study Universal and cultural differences in facial expression of emotion [6], he classifies for the first time the seven facial expressions associated with universal emotions: anger, disgust, fright, surprise, happiness, sadness, and contempt.

Although recent research, such as that published in Nature [5] shows that there can be up to 16 facial expressions, this is because they take into account the culture and the confusion or association that can be made of an expression to a particular emotion.

For a better reading of facial expressions,published The Facial Action Coding System (FACS)[8] and an update in 2002. FACS is a system that aims to measure all visually distinguishable facial movements and while this system has dozens of applications, mood analysis and states of mind is one of them. FACS is based on single action units (AU) distinguishable by a number (code). Each AU corresponds to a visually distinguishable facial activity. It also has a collection of head and eye movements and positions.

Fig. 1. Images present in the training set of FER2013.

FACS describes all visually distinguishable facial activity based on single action units and several categories of head and eye positions and movements. Each AU has a numerical code (the designation of which is quite arbitrary). Ekman and Friesen [7] first coined the term microexpressions, defining them as those that show a hidden emotion and that can last half a second or less, so, for this work, macro expressions will be used, these being longer lasting (between half a second and four seconds).

Paul Ekman [3] classifies facial expressions associated with emotions; however, in this article, we work with moods for the following reasons: One, conceptual clarity is a basis of science, and two, emotion biases behavior, whereas mood biases cognition.

The relationship between emotion and food is complex. Being happy, stressed or sad can make us choose a certain type of food, additionally, nowadays, compared to previous decades, we have available in restaurants a wider variety of dishes that we can choose from and this can also influence how we feel; there is bidirectionality between what we eat and part of our mood and also in the food we choose depending on our mood, for example, eating foods high in glycemic index (parameter present in carbohydrate foods that classify them taking into account the speed at which they are digested, absorbed and metabolized affecting blood glucose and insulin levels) is possibly a causal effect of depression [13].

Computationally, intelligent recommenders have become an important part of the industry, because based on information provided by the user, such as tastes, searches, weather and even location, it is possible to recommend a product or service. We are looking to create a food recommender that takes into account as a main parameter, the emotion of people through artificial vision. We will make use of convolutional neural networks, which, according to the state of the art, have proven to be highly efficient in image classification problems. However, the scope of this paper is the classification of Action Units (AUs) to human emotions.

2 State of the Art

In 2020, the book *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)* [12] was published, in which different applications of FACS are discussed.

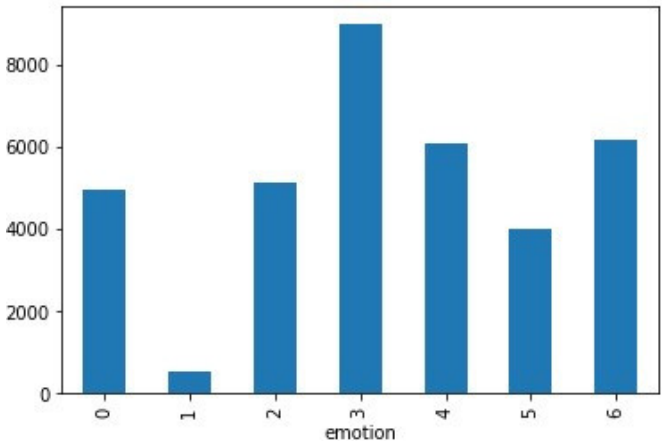


Fig. 2. Number of images per class in the FER2013 training set.

This book is divided into chapters and although there are many works dedicated in the book to emotion recognition and classification, in the chapter *The Next Generation of Automatic Facial Expression Measurement* a project on face detection and classification of facial expressions using support vector machines is described, achieving an accuracy of up to 93% accuracy in all the categories to be classified.

Li and Deng conducted a study on FER (*Facial Expression Recognition*) by analyzing the most popular datasets in this field: JAFFE, FER2013, SFEW 2.0, TFD, MMI and CK+. Each dataset was analyzed using different methods, such as the use of convolutional neural networks, recurrent neural networks, and MSV (support vector machines). In conclusion, they mention that the use of Deep Learning techniques (especially the use of convolutional neural networks) gives better results than the use of other types of neural networks.

However, despite finding congruence between what is learned by CNNs (Convolutional Neural Networks) and FACS Action Units, it is shown that they are unable to capture powerful convolutional features.

In *Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network* [11] the use of convolutional networks in attention for facial expression classification is proposed, achieving accuracies of 70% for FER2013, 99.3% for FERG, 92.8% for JAFFE and 98% for CK+.

Agrawal and Mittal [1] perform a study about how kernel size and several filters affect a convolutional neural network architecture proposed by the authors for facial expression classification, showing as conclusions a significant affectation between kernel size and several filters with the accuracy of the neural network using the FER2013 dataset as a basis, adding that they are of great help because they are not only simple in their architecture but also unique in terms of hyperparameter selection in the layers of the network.

In *Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy* [10], they achieve accuracies of up to 97% on CK+ and JAFFE datasets using a new face cropping and rotation method.

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Fig. 3. FER2013 dataset architecture.

Their method consists of generating new data to compensate for the imbalance in the datasets by randomly rotating and flipping the images with important conditions, such as rotating an image such that both eyes of the face are aligned in a straight line, cropping the faces, and spanning only from the edge of the eyebrows to the bottom edge of the chin.

Gogic and Ahlberg [9] propose a new facial expression classification method based on a trainable feature extraction process that uses sets of decision trees producing sparse binary feature vectors (LBF) and a shallow neural network. Such a method, according to its authors, is ideal for facial expression classification in resource-constrained environments such as mobile and embedded platforms.

2.1 Difference between this Proposal Concerning the State of the Art Presented Above

Currently, the use of FACS for facial expression analysis is rarely used and the preprocessing of the data often lacks measures for low and very high contrasts, as well as for varying light levels during the day.

We propose to use FACS to make a more complete dataset and in the case of machine vision, to set key points of interest in muscles using action units; we plan a modification of the datasets by including photographs taken in different locations (not only in the studio), as well as at different times of the day.

To balance the categories to be classified, we suggest the generation of random data using techniques seen in the state of the art, such as flips and flips of photographs looking for eye leveling. In the case of the neural network, we will continue to use convolutional neural networks because of their high efficiency in images and video, as demonstrated in the state of the art.

3 Solution Development

3.1 Description of the Solution to the Problem

The first step is the choice of the data sets to be used. The most used so far is called FER2013. This dataset contains 28,709 48x48 pixel grayscale images for use in training, 3,589 images for validation and 3,589 images for testing.

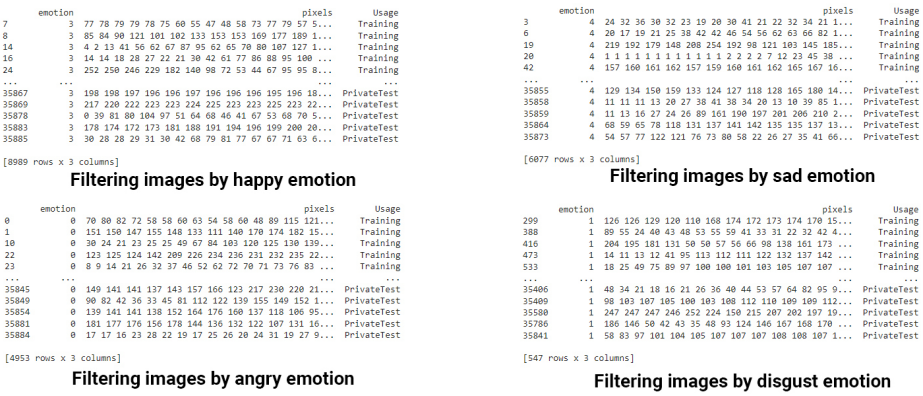


Fig. 4. Filtering of emotions in dataset FER2013.

The dataset has labels for universal emotions, represented as 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. However, when exploring the amount of data per class within FER2013, as seen in Figure 2, class 1 is found to be unbalanced. The reason for the unbalance is the small amount that the class has, which although a neural network model can be trained in that way, it would cause the training in that class to be null.

The architecture of FER2013 is based on 3 columns:

1. *Emotion*, corresponds to the type of emotion.
2. *Pixels*, containing a list of the number of values that make up the image (values from 0 to 255).
3. *Usage*, Usage, which identifies the recommended data type to use: *Training*, *Private* (Validation) and *Public* (Testing).

To make use of the pixels that form the images it was necessary to manipulate the data set. The original data type found in the column containing the pixels is of string type, so it was decided to transform a 1-dimensional tensor to achieve a more accurate and simple manipulation of the data. First, a conversion to list data type was made and consequently, the list was converted to a NumPy array (which allows us to represent collections of data of the same type in several dimensions and belongs to the NumPy numeric calculation library of Python), which allowed us to perform a reshape; for this reshape a size of 48x48 was chosen, which corresponds to the size of the image in the dataset indications (48x48 pixels) and only one channel (because the images are in black and white).

In the final part of the treatment, we chose to convert this NumPy array into a tensor (collection of n-dimensional vectors) of dimension 1, thus preparing the data for training and also allowing us to easily visualize it using the matplotlib library (Python library that allows us to make data graphics).

As the objective of this classifier of facial expressions corresponding to moods is to recommend a food dish in a restaurant, not all facial expressions are helpful. For this reason, we have decided to eliminate some facial expressions, modifying the dataset to focus on the ones we are interested in.

Fig. 5. Sample of the *Happy* category in FER2013.

Fig. 6. Sample of the *Sad* category in FER2013.

To achieve this, it was necessary to manipulate the original dataset using the pandas and NumPy Python libraries. With pandas, we filtered and eliminated those rows that correspond to facial expressions that are not of interest to us, and with NumPy we manipulated the list of pixels that correspond to each image to convert it into a NumPy array, to form a tensor that will be used to train the neural network and at the same time, show examples such as those presented below for each kind of facial expression that corresponds to a emotion. The final dataset was organized by the presence of only the following facial expressions:

– *Happy.*
– *Sad.*
– *Angry.*
– *Disgust.*

Fig. 7. Sample of the *Disgust* category in FER2013.

Fig. 8. Sample of the *Angry* category in FER2013.

Figures 5, 6, 7 and 8 represent examples found within the FER2013 dataset. We have recreated, thanks to the pixels provided by the same dataset a representative image.

It has been decided to leave only the categories of facial expressions mentioned above because as shown in some research, such as Salari-Moghaddam et al. [13] and AlAmmar et al. [2] and the correspondence between mood and food is between happy, sad and depressive moods.

To better encompass such mood states, the facial expressions of *Angry* and *Disgust* will become part of the sad or depressive mood state according to our next selection progress. Subsequently, manipulation of the dataset will be made, making modifications to the data it contains, that is to say, making a preprocessing.

During this preprocessing, we will try to use images with different shades of contrast and brightness, as well as with different backgrounds and in different locations, to have adequate variation and not fall with examples in the "ideal state" where all photographs used to comply with perfect shades of illumination.

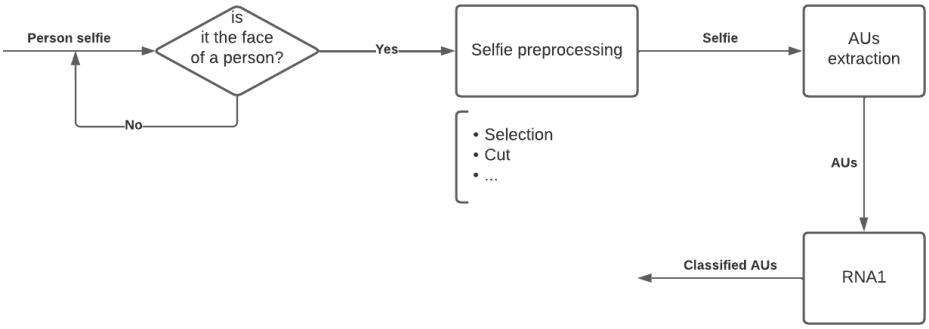


Fig. 9. General operation of the AU classifier.

In addition, we will try to generate new images to compensate for the unbalance of the dataset classes. To make sure that we correctly label the images we will use, we will make use of FACS, a system that will allow us to know, according to the AUs, to which facial expression each photograph corresponds. For this purpose, a scheme has been created using AUs that allows us to correctly classify the facial expressions.

Of all the main AUs [4], those that are not used for this work have been purged. The result is as follows:

0. Neutral Face,
6. Cheek Raiser: The orbicularis oculi muscle is lifted (below where dark circles usually appear),
4. Brow Lowerer: Wrinkling the glabella area (area between the two eyebrows) and home of the procerus muscle,
9. Nose Wrinkler: Actuating the levator muscle of the upper lip and the wing of the nose,
12. Nasolabial Deepener: Lifting the zygomaticus major muscle (located in the cheek); causes the smile,
15. Lip Corner Depressor: Actuating the depressor muscle of the angle of the mouth (located below the final corner of the lips),
16. Lower Lip Depressor: Movement in charge of the depressor labii inferioris muscle (located in the lower left middle part of the mouth).

For emotions, the AUs necessary to achieve them are already defined. In the case of the present work, the AUs will be used:

– Happyness: 6+12 (UA 6 plus UA 12),
– Sadness: 1+4+15 (UA 1 plus UA 4 plus UA 15),
– Disgust: 9+15+16 (UA 9 plus UA 15 plus UA 16),
– Neutral: 0.

Once all the images are labeled in the new dataset, the necessary features of the face are extracted based on FACS, and the image is cropped taking into account only the area of interest. With these data, a first neural network will learn to classify these AUs.

Once we have these individual AU classifications, we enter a second neural network, although this time, it will only give us a facial expression classification according to the previously detected AUs. In this way we make sure to find a facial expression more efficiently, guided by the FACS and implementing our solution.

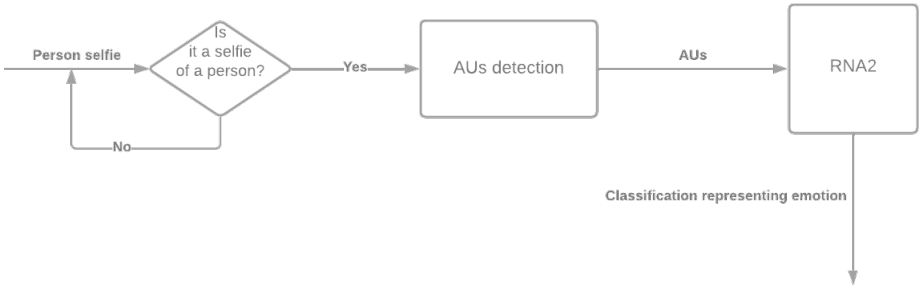


Fig. 10. General operation of the emotion classifier.

3.2 Scientific Novelty

Add to the state of the art a new way to analyze facial expressions that correspond to moods in images using FACS and deep neural networks.

3.3 Evaluation

The way to evaluate the correct performance of this facial expression classifier is based on the outputs of our neural network against different parameters, for example, making use of a k-fold cross-check, which allows us to give an estimate of the performance of the neural network based on unseen data, dividing the training set into subsets and then training all but one of them independently. This process is repeated until all subsets have been separated from the training iteration and the result is averaged across all models created.

4 Conclusions

The state of the art and theoretical research on saucer recommenders allows us to prove the feasibility of the work. The analysis of moods for the recommendation of products and services is an area of great interest that will allow us to make more appropriate recommendations to people, and the use of deep neural networks for the analysis of facial expressions that correspond to moods can be of help in the investigation of these because as Paul Ekman mentioned in his research and various psychologists today, moods lead us to choose, do and think differently than we would do depending on that state.

The future work is to find the ideal characteristics for the dataset, work on the formation of new data for this dataset that contemplate the modifications previously mentioned, work on an artificial vision to select the areas of interest in the faces and design the architecture of the convolutional neural network that will allow us to make the correct classification of the data.

Acknowledgments. This work has been possible thanks to the support of the Mexican government through the FORDECYT-PRONACES program of Consejo Nacional de Ciencia y Tecnología (CONACYT) under grant APN2017 – 5241; the SIP-IPN research grants SIP 2083, SIP 20220533; and IPN-COFAA and IPN-EDI.

References

1. Agrawal, A., Mittal, N.: Using CNN for Facial Expression Recognition: A Study of the Effects of Kernel Size and Number of Filters on Accuracy. The Visual Computer, vol. 36, no. 2, pp. 405–412 (2020). DOI: 10.1007/s00371-019-01630-9.
2. AlAmmar, W.A., Albeesh, F.H., Khattab, R.Y.: Food and Mood: The Corresponsive Effect. Current Nutrition Reports, vol. 9, no. 3, pp. 296–308 (2020). DOI: 10.1007/s13668-020-00331-3.
3. Beedie, C., Terry, P., Lane, A.: Distinctions Between Emotion and Mood. Cognition & Emotion, vol. 19, no. 6, pp. 847–878 (2005). DOI: 10.1080/02699930541000057.
4. Cohn, J.F., Ambadar, Z., Ekman, P.: Observer-Based Measurement of Facial Expression with the Facial Action Coding System. The Handbook of Emotion Elicitation and Assessment, vol. 1, no. 3, pp. 203–221 (2007). DOI: 10.1093/oso/9780195169157.003.0014.
5. Cowen, A.S., Keltner, D., Schroff, F., Jou, B., Adam, H., Prasad, G.: Sixteen Facial Expressions Occur in Similar Contexts Worldwide. Nature, vol. 589, no. 7841, pp. 251–257 (2021). DOI: 10.1038/s41586-020-3037-7.
6. Eckman, P.: Universal and Cultural Differences in Facial Expression of Emotion. In: Nebraska Symposium on Motivation, vol. 19, pp. 207–284, https://www.paulekman.com/wp-content/uploads/2013/07/Universals-And-Cultural-Differences-In-Facial-Expressions-Of.pdf (1972)
7. Ekman, P., Friesen, W.V.: Nonverbal Behavior and Psychopathology. In: Friedman, R.J., Katz, M.M. (eds), The Psychology of Depression: Contemporary Theory and Research, pp. 203–232, https://www.paulekman.com/wp-content/uploads/2013/07/Nonverbal-Behavior-And-Psychopathology.pdf (1974)
8. Ekman, P., Friesen, W.V.: Facial Action Coding System. Environmental Psychology & Nonverbal Behavior, (1978). DOI: 10.1037/t27734-000.
9. Gogic, M., Manhart, M., Pandzic, I.S., Ahlberg, J.: Fast Facial Expression Recognition Using Local Binary Features and Shallow Neural Networks. The Visual Computer, vol. 36, no. 1, pp. 97–112 (2020). DOI: 10.1007/s00371-018-1585-8.
10. Li, K., Jin, Y., Akram, M.W., Han, R., Chen, J.: Facial Expression Recognition with Convolutional Neural Networks Via a New Face Cropping and Rotation Strategy. The Visual Computer, vol. 36, no. 2, pp. 391–404 (2020). DOI: 10.1007/s00371-019-01627-4.
11. Minaee, S., Minaei, M., Abdolrashidi, A.: Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network. Sensors, vol. 21, no. 9, pp. 3046 (2021). DOI: 10.3390/s21093046.
12. Rosenberg, E.L., Ekman, P.: What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS). Oxford Academic, (2020). DOI: 10.1093/acprof:oso/9780195179644.001.0001.
13. Salari-Moghaddam, A., Saneei, P., Larijani, B., Esmaillzadeh, A.: Glycemic Index, Glycemic Load, and Depression: A Systematic Review and Meta-Analysis. European Journal of Clinical Nutrition, vol. 73, no. 3, pp. 356–365 (2019). DOI: 10.1038/s41430-018-0258-z.

ISSN 1870-4069

Video Surveillance of Beehives Using Computer Vision and IoT

Apolinar Velarde-Martínez, Gilberto Gonzalez-Rodríguez, Maria Teresa Ibarra-Rodríguez, Braulio Orozco-Cortéz

Tecnológico Nacional de México/Instituto Tecnológico El Llano Aguascalientes, Mexico

{apolinar.vm, gilberto.gr, maria.ir}@llano.tecnm.mx, 19900130@llano.tecnm

Abstract. Different animal species beneficial to the environment and human beings face imminent extinction due to different types of problems. The preservation of these species for the survival of humans and ecology is necessary. To preserve species, it is necessary to identify diseases and pests, through surveillance and observation by man. The honey bee (apis mellifera) is an endangered specie, and its care and preservation are already necessary. The Llano Aguascalientes region, Mexico faces problems of the extinction of this specie. This research work presents and describes a video surveillance system for the observation and care of hives using the Internet of Things and Computer Vision. The hardware prototype, the power supply system, the communication system between the apiary and the developed software system, as well as the developed software system, of this research work are described. Experiments were carried out in real environments for each part of the project and the software system which uses computer vision for the recognition of the bee in different positions is evaluated. The results obtained are described and the feasibility of this project which includes artificial vision and Internet of Things, as a support tool for the preservation of the apis mellifera species, is demonstrated. This research will allow in the near future the development of research projects applied to industry 4.0 for the conservation of this species.

Keywords: Apiary, beehive, cluster, computer vision, video surveillance.

1 Introduction

Environmental problems such as global warming caused by the destruction of flora, the disappearance of fauna, greenhouse gases, pollution and others that the planet currently faces, have been the reason for the issuance of government laws, and environmental organizations, to cite some references [1-4].

Currently, efforts have been directed towards the preservation of flora and fauna as measures to reduce the effects of environmental destruction. The preservation of beneficial fauna for humans, such as the honey bee, is carried out by the governments of different countries and organizations around the world by means of the implementation of special programs for its preservation.

The honey bee is a specie whose constant activity allows the pollination of different plants during the flowering process, and is cultivated for the production of food [5, 6, 8, 9] such as honey and propolis, as well as for support in the flowering of cultivable plants [14] such as corn [10], alfalfa [7] and beans [11].

Protecting and propagating the honey bee generates heated debates in the scientific environment because it is a species manageable by man; debates are generated due to competition with local wild pollinators [12], which it displaces from the spaces where is cultivated, but its preservation and care has become necessary.

Recent research shows bee populations have decreased considerably, due to different factors and diseases, such as the phenomenon of the disappearance of bee colonies [2, 15], the problem of the Varroa parasite which produces the Varroasis disease in bees [14, 16-18] and others.

Knowing the current problems of bees in the world, and considering the region of the Llano Aguascalientes, México as a producer of food from beekeeping and where crops depend largely on pollination by bees, field research, observations and interviews with beekeepers in this region were conducted; the objective of these investigations works is to know the diseases produced by the parasites which attack bees and the manual solutions applied to find these diseases; the results showed the existence of attacks by parasites and destructive insects of the hive that inhabit the spaces where the bee is cultivated; also the use of methods with the sacrifice of bees for the observation of parasites, such as those described in [13].

According to the above, it was found there is no system with computer technologies to automate hive inspection processes to avoid parasite infestation, nor a system to detect temperature and humidity levels in the hive, to avoid the appearance of bacteria which destroy bee brood queen.

In this research work, a real video surveillance environment with Internet of Things and computer vision for the preservation of the honey bee, in the region of El Llano Aguascalientes, México, is proposed. This real video surveillance environment is made up of a set of systems. The first system is related to the supply of normal electrical energy and solar electrical energy. A second system is the communications system between the apiary and the data processing center. The third system is the software which manages all the hardware devices, installed in the Liebres InTELigentes server cluster.

1.1 Justifications

This field research and the development of a real environment for video surveillance of the honey bee in the region Llano Aguascalientes, Mexico, is justified for the following reasons:

- This region is a producer of food from the cultivation of the honey bee, for which a set of surveys and interviews were carried out with beekeepers in the region to learn about and document the parasitic diseases of bees, and how the disease detection on bees is carried out manually.
- To propose an automated surveillance method to prevent the attack of parasites on honey bees, and protect bee colonies in this region because they are cultivated with the purpose of pollinating crops such as alfalfa, beans and sorghum.

– This research proposes and develops a solution made up of a hardware structure controlled by a software system, also a system to data acquisition and images which are transmitted to a Data Processing Center (DPC) made up of a cluster of servers for subsequent processing, and proposes development a system for mobile devices for end users (beekeepers).
– The prototype can be highly beneficial to stop and reduce a pandemic in a bee colony, making a timely diagnosis and preventing the hives from being in danger of dying due to parasitization.
– The Varroa parasite can be detected through image analysis; this parasite is highly destructive in bee colonies, which it attacks from birth to adulthood, a system with IoT technology and computer vision can support beekeepers to identify hives with bees infected by this parasite.

This work is composed of the following sections. The related works section describes some papers related to this research paper. The Basic Definitions section displays a list of definitions used in this work. Structure of the System is developed in section 4. Experimentations with the system in section 5 are described. The conclusions obtained so far are presented in section 6. Works currently being developed and will be presented in future papers are mentioned in section 7.

2 Related Works

There are several projects with video surveillance of bees for different purposes. In order not to extend this section, some research works are mentioned.

The bee surveillance project described in [21] sought information of the prevalence of honey bee colony losses. [22] presents a simple, non-invasive, system for pollen bearing honey bee detection in surveillance video obtained at the entrance of a hive; classifies into two classes: pollen-bearing honey bees and honey bees that do not carry pollen load; classification is performed using the nearest mean classifier, with a simple descriptor consisting of color variance and eccentricity features.

Inexpensive and modular system, to allow beekeepers to remotely observe the progress of their hives without opening them, is presented in [23]; this system sends sensor data acquired from beehives to a server for further analysis, obtains video and audio recordings which are used in some image and signal processing analysis.

In [24] video surveillance to monitor bee acitivity within male and female pumpkin flowers in 2011 and 2012 across a pollination window of 0600–1200 h was used. In [22] the importance of the use of video surveillance systems to understand and improve the health of bees, as well as the detection of parasites in hives, is highlighted.

According to the works mentioned, video surveillance has been adopted has a technology for the care and preservation of bees. A system which uses industry 4.0 technology is presented in this paper.

3 Basic Definitions

This section defines a set of terms related to the research described in this work. Knowledge of these terms enables the reader to understand the following sections.

- Bee. Hymenoptera insect Apis mellifera [3].
- Hive or brood chamber. Lodging of a colony or family of bees [3].
- Apiary. According to [13] an apiary is a space where the hives are located.
- Cluster LITEL. To define the cluster, let me use some previous definitions, then, this cluster is a set of loosely coupled, autonomous processing nodes [25]; each node may consist of a tightly coupled multiprocessor system [26].
- Data Processing Center (DPC). It is a cluster of servers with a high speed network, which works with free software, Linux operating system, C++, Apache2, PHP, HTML, OpenCV and Postgres [27].
- The system communication of multipoint point. An installation of a local computer network to allow communication between the DPC and the apiary; this system covers a distance of 800 meters, which complies with the regulation of the distances between the apiary and the passage of people or animals.
- Data and image acquisition device in the apiary (DADi). A device constructed with an Arduino board, an image acquisition camera, an electrical system operated with solar energy, and a Raspberri Pi device. The Raspeberri Pi (RasPi) device works as the Central Processing Unit and makes the set of devices work and allows communication between the DPC and the apiary. An electrical system for the acquisition of energy, consisting of a solar panel connected to a battery and an eliminator; this system provides power to the devices described in the following paragraphs installed in the hive. The Arduino Nano device is equipped with temperature, humidity, and motion sensors; the camera allows the acquisition of images of the hive entrance; a Raspberri Pi (RasPi) device is responsible for connecting to the Arduino and the camera, as well as for transmitting the information to the DPC.
- Liebre1 Software Agent. Software developed with the C++ programming language, the free distribution OpenCV (Open Computer Vision) image processing tool, Arduino language, the Postgres database management system and the web programming environment: PHP and HTML; the host operating system is the freely distributed Ubuntu Server.

4 Structure of the System

The general structure of the video surveillance system (prototype) using the Internet of Things and Computer Vision, is composed of:

- Brood chamber (hive). Meets the environmental requirements for use with live honey bees.
- Electric power supply system for the apiary antenna and the brood chamber.
- Communication system between apiary, data processing center and the end user.
- Software system using open source tools and operates in the server cluster.

In the following paragraphs, each of the parts is explained in detail for the reader's understanding.

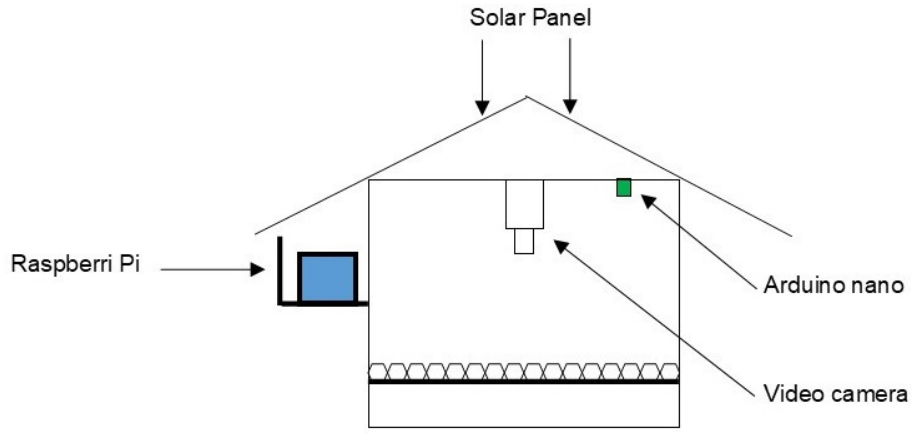


Fig. 1. General scheme of the hive.

4.1 Brood Chamber (Hive)

This part of the project consists of a brood chamber with the environmental requirements to be used with live honey bees, and contains the following special attachments for data and image acquisition:

1. Two solar panels placed on the top of the hive, built crosswise for the acquisition of solar energy which feeds the Arduino devices, Raspberry Pi, video camera and humidity and temperature sensors.
2. A Raspberri Pi device, for sending and receiving information between the apiary and the DPC.
3. A video camera for the acquisition of images in the hive entrance.
4. An arduino nano to connect the humidity and temperature sensors of the hive.
5. A wireless antenna to communicate the hive with the communication antenna router.

This brood chamber is located at a distance of 800 meters from the university campus, to avoid contact between people and honey bees. It works 12 hours a day to transmit information to the DPC. To carry out the tests, 10 bee racks were installed inside the chamber with approximately 3,000 bees, according to the beekeeper's estimates. Figure 1 shows the general scheme of the hive.

4.2 Electrical Power Supply System

The electrical power supply system consists of two types of power: the power supply from the electrical network to enable functionality of the antennas installed in the apiary, and the solar energy system installed in the hives-prototype (brood chamber).

For the supply of electrical energy in the brood chamber, the system was designed as follows: two solar panels connected to a deep-cycle battery to guarantee the supply to the installed hardware. The energy consumption is 20 watts per hour; the solar panels manage an energy of 120 watts per hour for a cycle of 12 hours, which guarantees devices will maintain their full functionality.

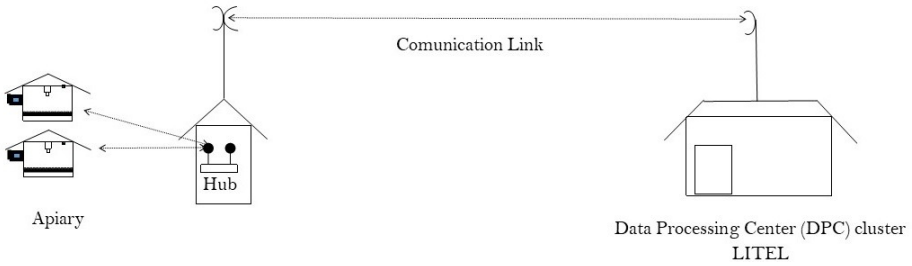


Fig. 2. The communication system between the CPD and apiary.

4.3 Communication System

The communication model is constituted by the system communication of multipoint point, which allows the DPC to communicate with the apiary permanently. The system works for a period of 12 hours during the day; all data is collected in the Liebre1 software agent system and is stored on high-speed devices for later processing.

The acquisition of data such as images, video and hive parameters from the apiary to the DPC is performed every minute, namely, the DADi sends images to the DPC for processing as shown in figure 2. This system was installed exclusively to carry out this research project, so it is not used for the transmission of any other type of information.

4.4 Software System

The Liebre1 software agent system is responsible for processing the data and images collected from the hives. Once the temperature and humidity parameters are collected and stored in a database, the images are sent to an subdirectory on the host server.

To carry out the video surveillance, a web page is provided for the beekeeper; this page shows the temperature and humidity data, as well as the images extracted from the hive. In addition to the above, in the DPC a preprocessing of the images is carried out to detect intrusions in the hive. This detection is done with image processing, which is explained in the paragraphs below.

Image Processing. The images are acquired from the apiary, specifically from the hives where the DADi is installed and they are sent to the DPC where each hive is identified by the registry provided by by the government office Agricultura [28]. The objective of this part of the project is to develop a sufficiently robust technique for contour detection; this technique must be tested with different images of the possible positions of the bees in the entrance, as well as considering the lighting with sunny days, cloudy days and rainy weather.

The image is acquired with a Raspberri Pi device, every minute; in the DPC, the image is processed with the following steps: crop the image, gray scale generation, frequency histogram creation, image recognition and contour generation [29-31]. On next paragraph every step is describen in reduced form:

a) Crop the image. This process segments the image into partitions; this partitions will be used on following processing phases.

Fig. 3. Phases of image processing.

b) Gray scale generation. The grayscale of an image is generally used as a preprocessing step of the image to prepare it for more complex image processing.
c) Creation of the frequency histogram. histogram of the image is created.
d) Shape recognition of the bee with convolutional neural networks (CNN).

Figure 3 shows the phases of image processing. The above steps are capable of issuing an alert in case of the following detections: the insect detected by the system is not a bee in the entrance (there is an intruder), or the insect detected by the system is a bee in the entrance. The proposed system runs on the Liebres InTELigentes server cluster (LITEL cluster), considered as the DPC. The cluster is made up of 3 high-end servers, with a high-speed network. The basic characteristics of the LITEL cluster are:

a) DELL Power Edge server, R330.
b) DELL Power Edge server, R540.
c) HP Proliant DL320E Gen 8 v2 server.
d) CISCO SG350-28 switch with 24 Gigabit ethernet ports.

All the hardware of the LITEL cluster is concentrated in a communications rack, located 800 meters from the central apiary of the campus.

5 Experimentations

The experiments of the video surveillance system have been carried out in the apiary of the Instituto Tecnológico del Llano, Aguascalientes, Mexico with 2 prototype hives, which are registered in in Agriculture Government Office and installed at the regulatory distance from the campus. The experiments was carried out with the three developed systems explained before, namely, the electrical power supply system, the communication system and the software system. The experiments described in this research work are carried out in controlled scenarios with the calibration of the camera.

The prototype is currently being developed in a real scenario. By space reasons, only the experimentations with the software system are described. To carry out the experiments with the data processing system, generation of the image database was built; this database contains 400 images of bees in different positions; figure 4 shows the bee in different positions based on the quadrants. For each quadrant 100 images were generated. For all images the background remains the same. In subsequent experiments, the background will be changed.

For this research work, a single kind of bee was used, Italian honey bee; Italian honey bee lives in the region of the Llano Aguascalientes, México. It is clear that, by changing the bee class, the image database must be changed or the number of images must be increased.

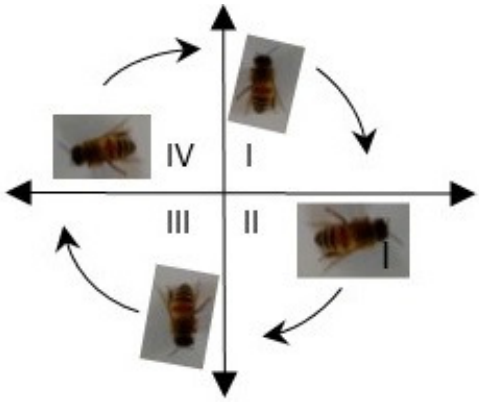


Fig. 4. Positions of the bee in the entrance.

Image database allows the recognition of the bee, placed in different positions; the recognition of the bee in different positions is necessary because when the bee leaves the hive, it can do so in different ways and the camera can acquire images of the bee every minute, without considering the position that the bee can take. In paragraphs below the software testing and experiments are going to explain.

Software testing. For the recognition of the bees in the hive entrance, the following experiments were considered, using solar lighting and in a scenario without solar lighting (cloudy); these scenarios were considered because the prototype will be installed in a real environment. Also different loads of images was considered. Only first three experiments with loads will be explained. Similar results was obtained with subsecuent experiments.

100 presentation images to the system (one by one). 50 images were acquired when there was sunlight on the stage, and 50 images when the weather was cloudy. The percentage of recognition was 56% and 61% respectively. In these experiments we have found that the system fails to acquire (generate) the thoracic part of the lower legs; it's an essential part for shape recognition. Figure 5 shows a loss of the thoracic part of the bee.

150 presentation images to the system (one by one). The number of images presented to the system was increased to determine the recognition percentage and detect false positives. For this phase of experimentation, a new filter was added to the image processing; this filter seeks to make the contours more visible. The results obtained show a higher recognition percentage. In the case of images with sun, a 69% assertiveness of the system was obtained and for images with a cloudy scenario (there was no sunlight), a recognition of 70% was obtained.

300 presentation images to the system. In an iteration of 300 cycles the system sought to recognize the bee. The recognition percentages remain lower than in the two previous experiments. By adding a new filter to image processing, the number of images containing more sunlight was increased, resulting in lower recognition percentages. The results obtained are 51% recognition with images containing direct sunlight on the hive. 53% recognition by avoiding direct sunlight.

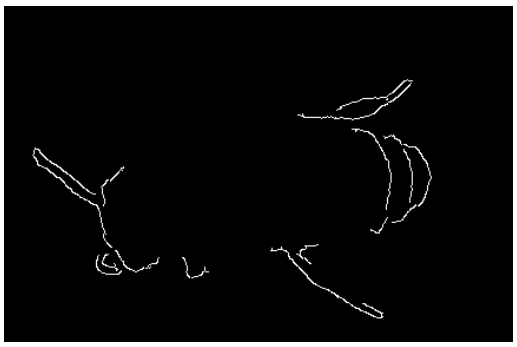


Fig. 5. Image showing the loss of the thoracic part of the bee.

Figure 6 shows the percentages of recognition of the three experiments explained, and others with different amounts of images.

The experiments show the brightness of the image is decisive in the recognition of the shape of the bee. It has been found in these experiments, the introduced images to the system with brightness of sunlight are likely to not be recognized, while images with low lighting may have a higher probability of being recognized. Although controlled environments are not an optimal system test, they do provide information for verification of system functionality. Other experimentation pending is the acquisition of images with rain.

NOTE. All the experimentations carried out in this research work were carried out with bees had already perished in the hive; no insect was sacrificed under any circumstances. All experiments are in compliance with the Law for the Protection of Animals for the State of Aguascalientes, Mexico, of Principle I and those emanate from it: Every animal has the right to live and be respected.

6 Conclusions

This research work presents the architecture of a system based on the Internet of Things and Computer Vision, for the preservation of the honey bee in the region of the Llano Aguascalientes, México. Experiments carried out for the recognition of the bee in the hive entrance are presented and described.

Very interested issues related to illumination in the scenarios where experiments was executed, was found. The thoracic part of the honey bee is an important part in the recognition phase; different filters was used to enhance the contours and let to the software system determine the presence or not presence of the bee in the entrance of hive. Altough our results are not so good so far due to the factors of movements of the bee, we are continue to improve both.

The electrical design and the data transmission speed tests between the apiary and the data processing center, due to space issues, were not presented. With this research applied to beekeeping, it is possible to demonstrate, the technologies used are feasible to prevent parasite invasions in hives and help in the preservation and care of the honey bee.

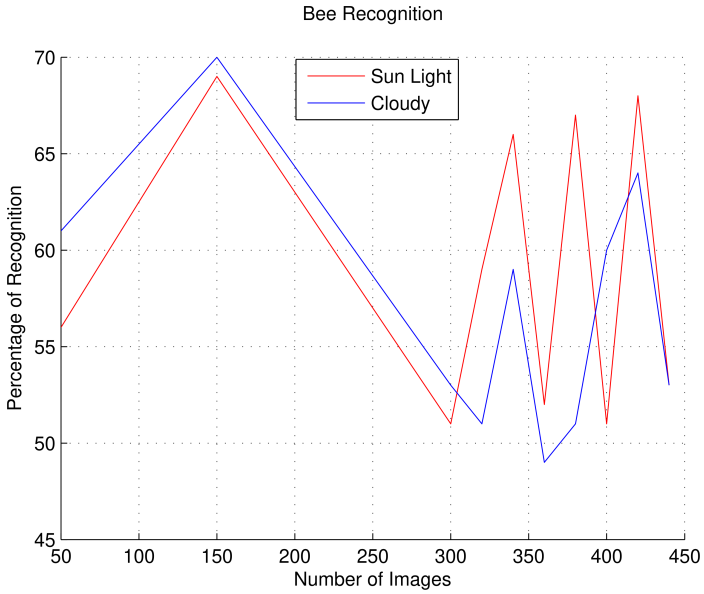


Fig. 6. The recognition percentages of the bees in each of the three experiments carried out and using different amounts of images.

7 Future Works

The future works of this research have been proposed in three important developments. First, the acquisition of real-time video of the hive, which will be sent to the DPC and placed on a web page to allow the beekeeper to view the video. Second, it is the recognition of the thoracic structure of the honey bee for the early detection of the Varroa parasite using Computer Vision.

Third, design of a system in software and hardware for the transmission of the audio from the hive to the data processing center, to retransmit the audio to the mobile devices through the web page, because according to the explanation of the experts, the movement (behavior) generated by the bees allows detecting the current status of the hive in terms of unwanted visitors.

Aknowledgments. This project is supported by IDSCEA Instituto para el Desarrollo de la Sociedad del Conocimiento del Estado de Aguascalientes, México. Special Thanks to Instituto Tecnológico el Llano Aguascalientes, México.

References

1. LXV Legislatura del Estado de Aguascalientes: Iniciativa con carácter de decreto de la nueva "Ley de protección y fomento apícola del estado de Aguascalientes". Diputado Cuauhtemoc Escobedo Tejada (2022)

2. Marcelino, J., Braese, C., Christmon, K., Evans, J. D., Gilligan, T., Giray, T., Nearman, A., Niño, E. L., Rose, R., Sheppard, W. S., vanEngelsdorp, D., Ellis, James, D.: The Movement of Western Honey Bees (Apis mellifera L.) Among U.S. States and Territories: History, Benefits, Risks, and Mitigation Strategies. Frontiers in Ecology and Evolution, vol. 10 (2022). DOI: 10.3389/fevo.2022.850600.
3. H. Congreso del Estado de Guanajuato: Desarrollo apícola para el estado de Guanajuato. LXIV Legislatura Publicada: P.O. Num. 235, Segunda Parte (2019)
4. Regulation (EC) no 1107/2009 of the European Parliament and of the Council of 21 October 2009 Concerning the Placing of Plant Protection Products on The Market and Repealing Council Directives 79/117/EEC and 91/414/EEC. Oficial Journal of the European Union, pp. 309/1–309/50 (2009)
5. Ullah, A., Shahzad, M. F., Iqbal, J., Baloch, M. S.: Nutritional Effects of Supplementary Diets on Brood Development, Biological Activities and Honey production of Apis mellifera L. Saudi Journal of Biological Sciences, vol. 28, no. 12, pp. 6861–6868 (2021). DOI: 10.1016/ j.sjbs.2021.07.067.
6. Machado-De-Melo, A. A., Almeida-Muradian, L. B., Teresa-Sancho, M., Pascual-Maté, A.: Composition and Properties of Apis Mellifera Honey: A Review. Journal of Apicultural Research, vol. 57, no. 1, pp. 5–37 (2017). DOI: 10.1080/00218839.2017.1338444.
7. Hagler, J. R., Mueller, S., Teuber, L. R., Machtley, S. A., Van-Deynze, A.: Foraging Range of Honey Bees, Apis mellifera, in Alfalfa Seed Production Fields. Journal of Insect Science, vol. 11, no. 1 (2011). DOI: 10.1673/031.011.14401.
8. Soares, S., Grazina, L., Mafra, I., Costa, J., Pinto, M. A., Duc, H. P., Oliveira, M. B., Amaral, J. S.: Novel Diagnostic Tools for Asian (Apis cerana) and European (Apis mellifera) Honey Authentication. Food Research International, vol. 105, pp. 686–693 (2018). DOI: 10.1016/ j.foodres.2017.11.081.
9. Almeida-Muradian, L. B., Barth, O. M., Dietemann, V., Eyer, M., Freitas, A., Martel, A. C., Marcazzan, G. L., Marchese, C. M., Mucignat-Caretta, C., Pascual-Maté, A., Reybroeck, W., Sancho, M. T., Gasparotto-Sattler, J. A.: Standard Methods for Apis Mellifera Honey research. Journal of Apicultural Research, vol. 59, no. 3, pp. 1–62 (2020). DOI: 10.1080/ 00218839.2020.1738135.
10. Smart, M. D., Otto, C. R., Carlson, B. L., Roth, C. L.: The Influence of Spatiotemporally Decoupled Land Use on Honey Bee Colony Health and Pollination Service Delivery. Environmental Research Letters, vol. 13, no. 8 (2018). DOI: 10.1088/1748- 9326/aad4eb.
11. Garratt, M. P., Coston, D. J., Truslove, C. L., Lappage, M. G., Polce, C., Dean, R., Biesmeijer, J. C., Potts, S. G.: The Identity of Crop Pollinators Helps Target Conservation for Improved Ecosystem Services. Biological Conservation, vol. 169, pp. 128–135 (2014). DOI: 10.1016/j.biocon.2013.11.001.
12. Russo, L.: Positive and Negative Impacts of Non-Native Bee Species Around the World. Insects, vol. 7, no. 4 (2016). DOI: 10.3390/insects7040069.
13. Food and Agriculture Organization of the United Nations: TECA - Technologies and Practices for Small Agricultural Producers. https://www.fao.org/teca/es/technologies/8663
14. Traynor, K. S., Rennich, K., Forsgren, E., Rose, R., Pettis, J., Kunkel, G., Madella, S., Evans, J., Lopez, D., vanEngelsdrop, D.: Multiyear Survey Targeting Disease Incidence in US Honey Bees. Apidologie, Springer Verlag, vol. 47, no. 3, pp.325–347 (2016). DOI: 10.1007/s13592-016-0431-0.
15. VanEngelsdorp, D., Traynor, K. S., Andree, M., Lichtenberg, E. M., Chen, Y., Saegerman, C., Cox-Foster, D. L.: Colony Collapse Disorder (CCD) and Bee Age Impact Honey Bee Pathophysiology. PLoS ONE, vol. 12, no. 7 (2017). DOI: 10.1371/journal.pone.0179535.
16. Traynor, K., Mondet, F., Miranda, J., Techer, M., Kowallik, V., Oddie, M., Chantawannakul, P., McAfee, A.: Varroa Destructor: A Complex Parasite, Crippling Honeybees Worldwide. Preprints (2020). DOI: 10.20944/preprints202002.0374.v1.

17. Higes, M., Martín-Hernández, R., Hernández-Rodríguez, C. S., González-Cabrera, J.: Assessing the Resistance to Acaricides in Varroa Destructor from Several Spanish Locations. Parasitology Research, vol. 119, no. 11, pp. 3595–3601 (2020). DOI: 10.1007/s00436-020-06879-x.
18. Schäfer, M. O.: Varroosis of Honey Bees (infestation of honey bees with Varroa spp.): Chapter 3.2.7. Manual of Diagnostic Tests and Vaccines for Terrestrial Animals 2021: OIE Terrestrial Manual 2021, OIE - Office International des Epizooties (2021)
19. Martelli, A.: An Application of Heuristic Search Methods to Edge and Contour Detection. Communications of the ACM, vol. 19, no. 2, pp. 73–83 (1976). DOI: 10.1145/359997.360004.
20. Lee, K., Steinhauer, N., Travis, D. A., Meixner, M. D., Deen, J., vanEngelsdorp, D.: Honey bee surveillance: A Tool for Understanding and Improving Honey Bee Health. Current Opinion in Insect Science, vol. 10, pp. 37–44 (2015). DOI: 10.1016/j.cois.2015.04.009.
21. Hendrikx, P., Chauzat, M. P., Debin, M., Neuman, P., Fries, I., Ritter, W., Brown, M., Mutinelli, F., Le-Conte, Y., Gregorc, A.: Bee Mortality and Bee Surveillance in Europe. SCIENTIFIC REPORT submitted to EFSA, vol. 6, no. 9 (2009). DOI: 10.2903/sp.efsa.2009.EN-27.
22. Babic, Z., Pilipovic, R., Risojevic, V., Mirjanic, G.: Pollen Bearing Honey Bee Detection in Hive Entrance Video Recorded by Remote Embedded System for Pollination Monitoring. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 51–57 (2016). DOI: 10.5194/isprsannals-III-7-51-2016.
23. Tashakkori, R., Hernandez, N. P., Ghadiri, A., Ratzloff, A. P., Crawford, M. B.: A Honeybee Hive Monitoring System: From Surveillance Cameras to Raspberry Pis. SoutheastCon 2017, pp. 1–7 (2017). DOI: 10.1109/SECON.2017.7925367.
24. Phillips, B. W., Gardiner, M. M.: Use of Video Surveillance to Measure the Influences of Habitat Management and Landscape Composition on Pollinator Visitation and Pollen Deposition in Pumpkin (Cucurbita pepo) agroecosystems. PeerJ, vol. 3, pp. e1342 (2015). DOI: 10.7717/peerj.1342.
25. Tannenbaum, A., Van-Steen, M.: Distributed systems: Principles and paradigms. Prentice Hall (2007)
26. Nehmer, J., Haban, D., Mattern, F., Wybranietz, D., Rombach, H. D.: Key Concepts of the INCAS Multi Computer Project. IEEE Transactions on Software Engineering, vol. SE-13, no. 8, pp. 913–923 (1987). DOI: 10.1109/TSE.1987.233510.
27. Velarde-Martinez, A.: Implement of a high-performance Computing System For Parallel Processing of Scientific Applications and the Teaching of Multicore and Parallel Programming. In: International Conference on Innovation, Documentation and Education, pp. 203–213 (2018). DOI: 10.4995/INN2018.2018.8908.
28. Secretaría de Agricultura y Desarrollo Rural: fortalecen méxico y belice acciones de cooperación en el sector agroalimentario. Agricultura https://www.gob.mx/agricultura/aguascalientes. (2017)
29. Liu, Z., Wang, H., Zhou, T., Shen, Z., Kang, B., Shelhamer, E., Darrell, T.: Exploring Simple and Transferable Recognition-aware Image Processing. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3032–3046 (2022). DOI: 10.1109/TPAMI.2022.3183243.
30. Yang, Z., Li, Y., Yang, J., Luo, J.: Action Recognition with Spatio–temporal Visual Attention on Skeleton Image Sequences. IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 8, pp. 2405–2415 (2019). DOI: 10.1109/TCSVT.2018.2864148.
31. Haralick, R. M.: Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts. Pattern Recognition, vol. 5, no. 4, pp. 391–403 (1973). DOI: 10.1016/0031-3203(73)90029-0.

ISSN 1870-4069

Linguistic Elements Selection from a BCI Matrix Using Intelligent Computing

Marco Antonio Cardoso-Moreno, Francisco Hiram Calvo-Castro, Cornelio Yáñez-Márquez

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{mcardosom2021, hcalvo, cyanez}@cic.ipn.mx

Abstract. Brain-computer interfaces (BCI) are systems primarily developed to help people with a certain level of motor disability, achieve new channels of communication and interaction with their environment that depend only on each individual's brain activity, which is translated, by means of an algorithm, to the user's desired commands and actions. In other words, the main objective of BCI systems is to provide users with the ability to control computers or machines without requiring any limb movement, but only by their own brain activity instead. There are many ways in which brain activity is recorded; electroencephalography (EEG) being the technique most widely used. One of the most common BCI systems is the P300 Speller BCI, which bases its functioning on the occurence of event-related potentials (ERP); specifically P300 ERP, which appear in the EEG readings in the form of positive deflections around 300 ms after an unexpected stimulus has occured. The goal of spellers is to select and write the desired character from an array of letters presented to the user in the form of a matrix; the characters in this matrix are randomly flashed in order to activate the P300 potentials in the EEG readings. This paper analyzes the current state of the art regarding the algorithmic aspect of the BCI P300 spellers; we also present a first approach for signal preprocessing, intended to deliver feature vectors suitable for target character classification.

Keywords: BCI, EEG, ERP, P300.

1 Introduction

Brain-computer interfaces (BCI) are systems whose main purpose is to provide the user with new communication channels that are independent from those typically used by the brain, namely muscles and the nervous system [6, 3, 15-17].

BCI systems work by reading and interpreting brain activity from the user in order to translate it into commands [3, 8, 13]; data is gathered by electroencephalography (EEG) [4, 8, 13], which detects electrical charges resulting from brain activity by positioning electrodes on the user's scalp [13].

Although some other methods are available for brain activity data gathering, namely: functional magnetic resonance imaging (fMRI), near-infrared spectroscopy

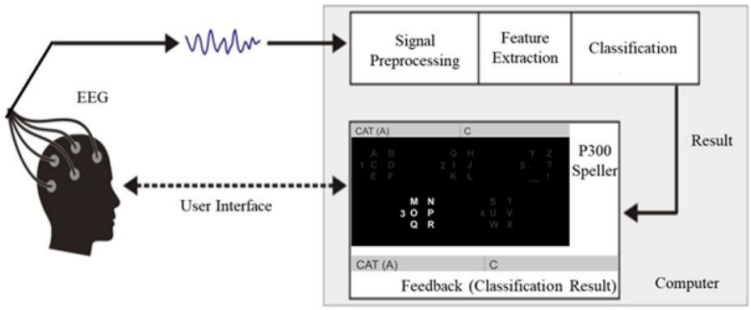


Fig. 1. Schematic of a P300 BCI speller system [13].

(NIRS) and magnetoencephalography (MEG) [**?**], EEG is widely prefered due to its noninvasiveness [6,13]

In general, a BCI consists of input and output channels, as well as an algorithm; recorded brain activity is used as input, whereas the user's desired commands correspond to the output. Between these channels we found the algorithm, which, in common terms, translates input data into output commands; since it works directly with brain activity, the algorithm must be capable of adapting to the dynamic features of this organ [17].

This work focuses on the BCI systems known as spellers, whose main objective is to provide the user with the ability of writing only by their brain activity; these systems are between the most studied ones, being the P300 BCI Speller the most widely adopted [3]. This BCI detects P300 ERP in the EEG readings, which appear in the form of a positive deflection in the signal, in the lapse of 250 ms to 750 ms after the user has been exposed to certain stimulus [18]; it bases its functioning on the oddball paradigm, which states that the less common the stimulus (unpredictable for the user), the higher the P300 intensity signal [2, 8, 15, 18]

Figure 1 shows an schematic of how, in general, a P300 BCI system is structured. The main components are: the visual interface, where the user will select the desired characters from; the EEG system needed to record brain activity followed by a signal acquisition phase; next there are a signal preprocessing stage, a feature extraction phase and a Machine Learning (ML) algorithm for the task of character classification and prediction; lastly, the algorithm's output is passed to the system in order for it to write the desired character.

The common approach to these BCI systems consists on facing the user with a 6×6 matrix, where each cell corresponds to an individual character: 26 letters, 9 digits and one blank space symbol, for a total of 36 cells. Randomly, a row or column is selected, and its light intensity is increased for a brief lapse; this procedure is carried on until all columns and rows have been intensified (Figure 2).

Therefore, for every character the user focuses their attention on, two out of twelve stimuli will contain such symbol, allowing for the isolation of particular P300 ERP corresponding to a given character in the matrix [2, 7].

Although widely adopted, there still are some aspects of the P300 BCI that need improvement. There is the need to find the right balance between the accuracy of the

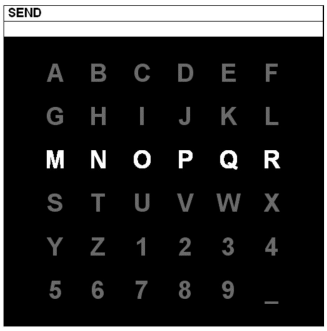


Fig. 2. Matrix presented to the user. The light intensity of the third row, from top to bottom, is increased in order to generate a P300 ERP [7].

character prediction algorithm and the information transfer rate (ITR); the algorithm's performance may vary drastically from subject to subject, and even between sessions for a given user; and lastly, there are people who do not meet the necessary ERP to use these systems [18], phenomenon generally refered to as "BCI illiteracy" or "BCI inefficiency" [5].

Our focus is, mainly, on the algorithmic aspect, specifically on character classification, recognition and prediction, as well as feature vector dimensionality reduction and signal processing. The dataset we selected to work with is the Wadsworth BCI Dataset (P300 Evoked Potentials) [7]; here, the user was faced with a matrix as the one presented in Figure 2.

2 State of the Art

In general, a P300 Speller has the following structure: a signal preprocessing phase, where one or several filters are applied to the EEG reading signals in order to increase the signal-to-noise ratio (SNR); a dimensionality reduction phase, where feature selection and feature extraction algorithms are applied to the feature vectors obtained after the preprocessing stage; and a classification phase, where a machine learning model is applied on the reduced vectors in order to classify the P300 signals [3, 4].

The classification problem consists on predicting whether a given feature vector in a character epoch belongs to one of 2 classes: target character (a P300 signal is present) or non-target character (no P300 signal is present) [10].

As these spellers are, in general, complex systems, there are two general approaches to trying to improve the system performance: the BCI user interface, where the P300 potentials are evoked, which includes the structure and layout of the characters in the screen, their morphology and the physical stimulation applied to the user; and the algorithm, which includes the preprocessing, feature extraction and selection, and classification phases [3].

There have been some efforts in improving the user interface presented to the user by, for example, imitating the nine-key keyboard layout used on mobile phones where users

Table 1. Commonly used techniques for each stage of a P300 BCI speller system.

BCI stage	Operation
Preprocessing	CAR filters
	Butterworth filters
	Chebyshev filters
Feature Selection and Extraction	ICA
	PCA
	Fast Fourier Transform
	Wavelet Transform
	Autoregressive model
	Cohen's class distribution
	Subsampling
Character Classification	Bayesian classifiers
	LDA
	SVM
	Artificial Neural Networks
	Random Forest

selected the initial characters from a 3×3 matrix, and then word suggestions were made from a dictionary [1]; also, there have been changes in the structure interface, from a 2D structure to a 3D one [11, 12].

The morphology of the symbols is another area of interest, it has been observed that changing a letter for a human face as part of the visual stimuli increases the speller's performance when compared to other, more traditional, approaches [5, 9]. Changing the typical one-color light intensification in the matrix for a mixed colors interface has also shown an increase in accuracy of approximately 16% [10]. Other approaches involve other kind of stimuli, such as auditory or a combination of both visual and auditory stimuli, as presented in [10].

Regarding the data preprocessing stages, the most common techniques used are: temporal filters, spatial filters, frequency filters, time-frequency filters and time-spatial filters [3]. For example, [10] used a common average reference (CAR) spatial filter after data standarization; [12] applied a bandpass Butterworth filter followed by spatial filters, [16] also used a Butterworth bandpass filter followed by a normalizarion procedure, whereas [13] made use of a Butterworth filter only; [4] performed a fourth order Chebyshev type I filter on the EEG signals, while [8] used an eighth order Chebyshev filter.

After the EEG data have been preprocessed, it is necessary to perform some feature selection and feature extraction tasks; here, the most common approach is to perform independent component analysis (ICA) and principal component analysis (PCA), as well as, althoug less commonly used, the Fast Fourier Transform, Wavelet Transform and autoregresive model [3].

[4] made use of the Cohen's class distribution, in conjunction with a fourth order Chebyshev filter to reduce the feature vectors dimension on their dataset. On the other hand, [8] applied a PCA to remove the less important features from their dataset.

[9, 13] used similar approaches to dimensionality reduction of their feature vectors with sub sampling, by selecting one sample for every 4 values, and 40 out of the 400 obtained samples, respectively. [18] made use of the stepwise linear discriminant

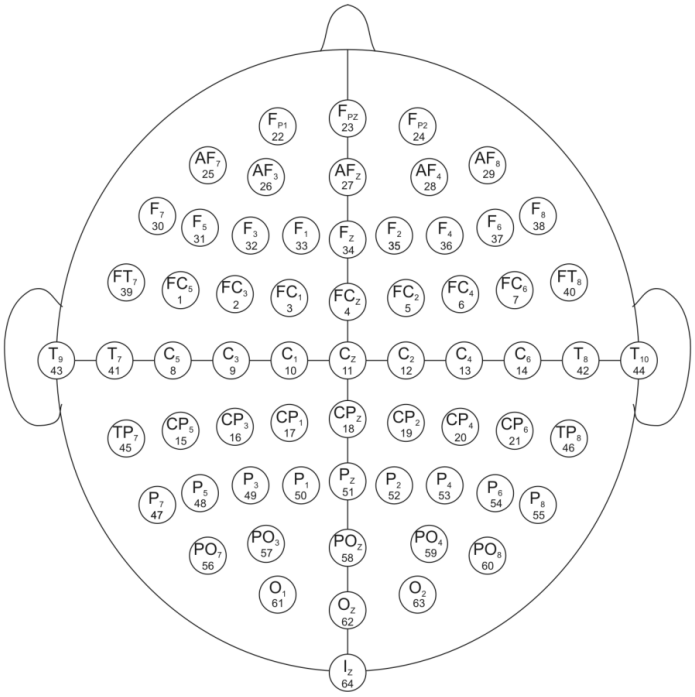


Fig. 3. Electrode layout and index designation [7].

analysis (SWLDA) to reduce the feature space by removing and adding features to it, depending on their contribution to the classification task, after the corresponding downsampling of the preprocessed EEG data.

The next step is to perform classification tasks over the resulting feature vectors, for which a machine learning algorithm is required, among the most widely adopted algorithms we found: Bayesian classifiers, linear discriminant analysis (LDA), support vector machines and artificial neural networks [3].

[2] used an LDA classifier, with a split validation of 10 symbols used for training and 40 symbols used for testing, since, according to the authors, cross-validation techniques are not a good fit for BCI applications due to the amount of data available and overfitting concerns.

LDA is also used by [5] with a validation split, where the first iteration of each subject was used as the training data and the following 4 runs were used as the testing set; performance was measured by defining the accuracy as the number vectors correctly classified as target characters divided by the total ammount of target characters; [9], as well as [12] made use of a Bayesian linear discrimant analysis (BLDA); finally, [13] performed a Stepwise linear discriminant analysis.

Support vector machines classifiers are also widely applied, [8] proposed an Ensemble of Weighted SVM (EWSVM), which was then compared against an Ensemble Weighted LDA (EWLDA) and combinations of SVM and LDA (linear

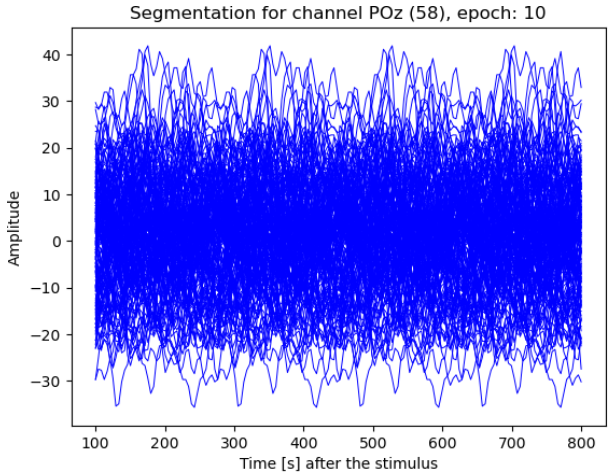


Fig. 4. Segmentation on channel POz for epoch 10.

kernels were used for all SVM implementations), the results show that the performance of EWSVM is better than that of EWLDA; [10] made a comparison between a linear SVM and an LDA, both with and without PCA feature extraction, where the LSVM without PCA was the model with best results, followed by LSVM with PCA.

[16] proposed different models, including SVM, Random Forest and Extreme Gradient Boosting (XgBoost), these models were applied to the dataset after oversampling techniques to overcome the imbalanced classes inherent to the speller; the results showed that SVM was the best algorithm with an accuracy of 94.2%, followed by XGBoost with an accuracy of 94.19% and finally, the random forest with a performance accuracy of 94.16%.

Artificial Neural Networks are another approach to character classification in P300 BCI spellers, [4] proposed the use of a Deep Neural Network (DNN), they used Stacked autoencoders to further reduce the dimensionality of the feature vectors from 651 features to 10, these 10 final features were passed to a Softmax classifier; in this study a duplication of the feature vectors labeled as target characters is carried on in order to overcome the imbalance issues. Table 1 presents commonly used techniques in each P300 BCI speller stage.

2.1 Difference of this Proposal with Respect to the State of the Art

It is observed that most of the solutions proposed in the state of the art for character classification and prediction involve certain assumptions, for example, many of the classifiers are linear, namely LDA and support vector machines with the use of a linear kernel, which means that it is assumed that the preprocessed EEG data is linearly separable.

It is important to further study up to what extent this is a valid assumption, since EEG readings have some non-linear features which might not be filtered out in the preprocessing and feature selection stages.

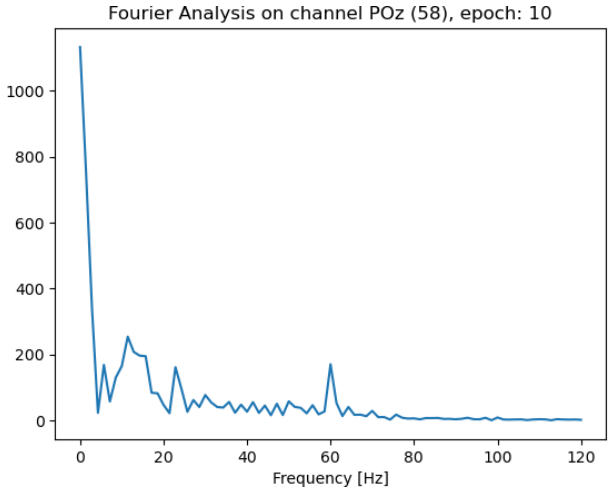


Fig. 5. Fourier analysis on Channel POz for epoch 10.

In future study, different classification algorithms will be tested in order to determine a new approach to character recognition in P300 BCI spellers, specifically: Multi-layer Perceptron, Long Short-Term Memory (LSTM) Neural Networks and a new paradigm, called Minimalist Machine Learning [19].

3 Project Development

3.1 The Dataset

As previously stated, the selected dataset for this work is the Wadsworth BCI Dataset (P300 Evoked Potentials) [7]. The dataset consists of four **.mat* files containing information on two test subjects (A and B) results, one for training and one for testing; i.e., there is a training and a testing file pair for each subject; training files have data regarding 85 character epochs, whereas testing files contain data of 100 character epochs.

Each character epoch consists of the following steps: a 2.5 s periods where the matrix was displayed with equal intensity for every row and column; then each row and column was randonmly intensified for 100 ms for a total of 12 intensifications, this process was repeated 15 times per epoch, resulting in 15 blocks of 12 intensifications (complete rows and columns), for a total of 180 intensifications; after each intensification, the matrix went back to an equal intensity state for 75 ms.

Signals were collected with a 64 electrode setup, whose layout, as well as index designation for this particular dataset, are shown in Figure 3; the matrix arrangement of characters is depicted in Figure 2. Data was gathered at a sample rate of 240 Hz.

3.2 Data Preprocessing

Signal data is presented as a three dimensional array, where the first dimension corresponds to character epochs, the second dimension corresponds to different time

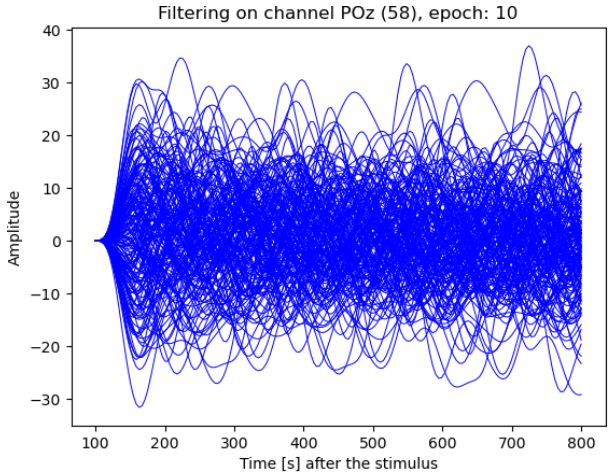


Fig. 6. Filtering on Channel POz for epoch 10.

steps along the epoch and the third dimension corresponds to each channel in the EEG hardware (electrodes). All values and results discussed in this section correspond to training data for subject A and EEG channel POz, located in the parieto-occipital region of the brain, since the occipital lobe is responsible for visual perception [14].

The first stage in data preprocessing is segmentation, i.e., breaking each character epoch into segments where the stimulus has been applied, to this purpose, for each channel, we break the whole epoch into 700 ms windows starting at exactly 100 ms after, and up to 800 ms, after the stimulus was applied; since this is the period where P300 ERP peaks show in EEG readings. This results, for each channel, in new two dimensional arrays where the first dimension corresponds to each intensification (180 in total), whereas the second contains the segment signals for 700 ms (168 samples in total). Figure 4 shows the results after the segmentation process.

From Figure 4, it is clear that EEG readings carry a lot of noise, therefore, it is imperative to apply a filter in order to reduce the ammount of noise present in the signal while keeping valuable data. With a Fourier analysis (Figure 5) we see that the frequencies at which more information was read are low frequencies, which is expected since P300 ERP are low frequency phenomena [6]. Based on these results, a filtering procedure was performed with a 5th degree Butterworth bandpass filter, with low and high cut frequencies of 0.1 and 15 Hz, respectively. After this Butterworth filter is applied, we end up with a smoother signal (Figure 6).

Once data have been filtered, a summing process is performed in order to unite all intensifications for each column or row; here an average operation was carried on, which led to a new two dimensional array of dimensions 12×168, where the first dimension corresponds to the 12 rows and columns in the character matrix, and the 168 values of the second dimension are the result of averaging those values from the 15 repetitions.

When performing a sum over the 15 repetitions of a particular stimulus intensification, it is expected that noise values will cancel each other, since these readings are considered to be completely random values, while the signal of interest

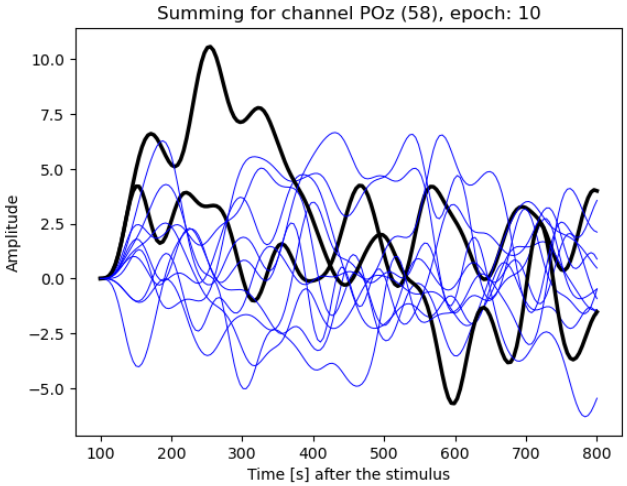


Fig. 7. Resulting signals after summing on channel POz for epoch 10. The thickest signals show the row and column containing the target character.

will be present in each repetition, thus, amplifying the corresponding reading values; the arithmetic mean is computed in order to return the signal amplitude to its initial range of values.

In Figure 7, we show the resulting average values for each row and column in the matrix, here, two signals are depicted as thicker lines, while the rest of the signals appear as thinner series, in the background; the thicker signals correspond to the row and column that contained the target character of the tenth epoch for subject A.

The nex step in the preprocessing stage is to downsample the arrays in order to reduce data dimensionality without losing valuable information; downsampling was carried on via a decimation procedure with a downsampling factor $q = 5$, meaning that every fifth signal value is preserved while the previous four are thrown away thus, reducing the signal dimensionality by a factor of five. This reduction makes each vector for a row or column, to reduce its number of elements from 168 to 34.

When comparing Figures 7 and 8, we see that the average shape of the signals is conserved, hence, downsampling is a valid approach for dimensionality reduction when working with EEG reading signals, since the general structure of the problem is preserved while reducing the number of attributes the model will deal with, demanding less computational resources and providing results in less time; aspects of primary importance in BCI systems.

Then, intersections of each row and column are computed by averaging both signals. In Figure 9, all indexes are shown as represented in the dataset so, for example, the average of column 1 and row 7 represents the signal read for character A, since it is the character at the intersection of such row and column; the average of column 5 and row 10 corresponds to character W, and so on.

This operation results in a new array of dimensions 36×34, the first dimension corresponds to each one of the characters in the matrix, while the second dimension contains the average values of the intersection for a given character. Figure 10 shows

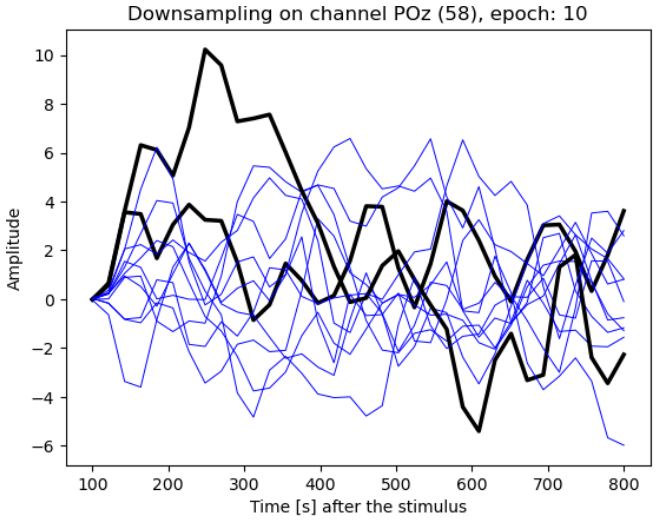


Fig. 8. Downsampling on Channel POz for epoch 10. The thickest signals show the row and column containing the target character.

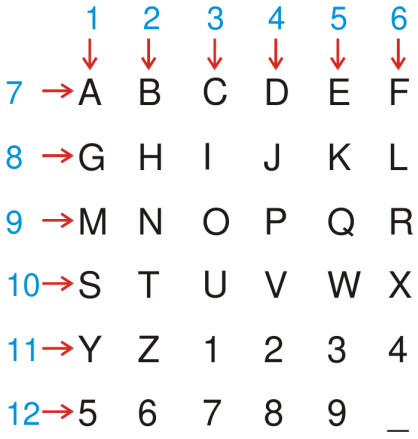


Fig. 9. Indexes of rows and columns in the dataset [7].

the resulting signal values, averaged for each character in the matrix, with the given target character appearing as the thickest signal, for epoch 10.

For this case, the goal of any ML classifier is to perform as best as possible, by trying to assign the thicker signal to the target character class, while assigning the rest of the signals to the non-target character class.

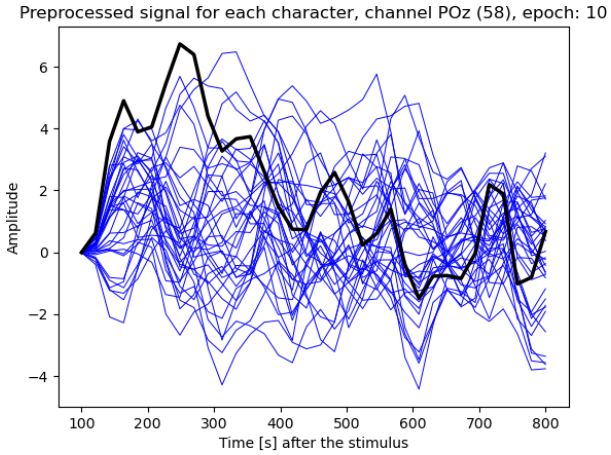


Fig. 10. Signal averaged values for each character in the epoch. The thickest signal corresponds to the target character of the epoch.

4 Conclusions

In this work we presented a summary of the state of the art for P300 Speller BCI, making emphasis on the techniques most widely used in every stage of the system, as well as pointing out some of the deficiencies still present, namely: BCI illiteracy and ML algorithms performance, from a computational perspective.

We also presented the signal preprocessing stage for the given dataset. Although results were only shown for one particular example, i.e., one subject, one epoch and one electrode, this procedure, utilizing the same set of parameters, applies for the whole dataset nevertheless.

It is important to emphazise the validity of the approach presented. The several plots shown indicate that even by reducing the feature vectors dimensionality by five, the main characteristics of the signals are preserved, which should allow for a classifier to perform without issues, within its own limitations.

This aspect results of great importance since two models to be tested in future work are neural network architectures, known for requiring longer times of training when compared to other ML paradigms.

References

1. Akram, F., Han, S. M., Kim, T. S.: An Efficient Word Typing P300-BCI System Using a Modified T9 Interface and Random Forest Classifier. Computers in Biology and Medicine, vol. 56, pp. 30–36 (2015). DOI: 10.1016/j.compbiomed.2014.10.021.
2. Artzi, N. S., Shriki, O.: An Analysis of the Accuracy of the P300 BCI. Brain-Computer Interfaces, vol. 5, no. 4, pp. 112–120 (2018). DOI: 10.1080/2326263X.2018.1552357.
3. Fang, T., Song, Z., Niu, L., Le, S., Zhang, Y., Zhang, X., Zhan, G., Wang, S., Li, H., Lin, Y., Jia, J., Zhang, L., Kang, X.: Recent Advances of P300 Speller Paradigms and Algorithms. In: 2021 9th International Winter Conference on Brain-Computer Interface (BCI), pp. 1–6 (2021). DOI: 10.1109/BCI51272.2021.9385369.

4. Ghazikhani, H., Rouhani, M.: A Deep Neural Network Classifier for P300 BCI Speller Based on Cohen's Class time-frequency Distribution. Turkish Journal of Electrical Engineering and Computer Sciences, vol. 29, no. 2, pp. 1226–1240 (2021). DOI: 10.3906/elk-2005-201.
5. Guger, C., Ortner, R., Dimov, S., Allison, B.: A Comparison of Face Speller Approaches for P300 BCIs. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004809–004812 (2016). DOI: 10.1109/SMC.2016.7844989.
6. Guger, C., Ortner, R., Dimov, S., Allison, B.: A Comparison of Face Speller Approaches for P300 BCIs. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004809–004812 (2016). DOI: 10.1109/SMC.2016.7844989.
7. Krusienski, D., Schalk, G.: Documentation Wadsworth BCI dataset (P300 evoked potentials). https://www.bbci.de/competition/iii/desc_II.pdf. (2004)
8. Kundu, S., Ari, S.: P300 Detection with Brain–computer Interface Application using PCA and Ensemble of Weighted SVMs. IETE Journal of Research, vol. 64, no. 3, pp. 406–414 (2018). DOI: 10.1080/03772063.2017.1355271
9. Lu, Z., Gao, N., Zhou, W., Yang, J., Wu, J., Li, Q.: A Comparison of Facial P300-speller Paradigm Based on Famous Face and the Familiar Face. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6 (2019). DOI: 10.1109/CISP-BMEI48845.2019.8965892.
10. Meshriky, M. R., Eldawlatly, S., Aly, G. M.: An Intermixed Color Paradigm for P300 Spellers: A Comparison With Gray-scale Spellers. In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems, pp. 242–247 (2017). DOI: 10.1109/CBMS.2017.123.
11. Noorzadeh, S., Rivet, B., Jutten, C.: Beyond 2D for brain-computer interfaces: Two 3D Extensions of the P300-speller. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5899–5903 (2014). DOI: 10.1109/ICASSP.2014.6854735.
12. Noorzadeh, S., Rivet, B., Jutten, C.: 3-D interface for the P300 speller BCI. IEEE Transactions on Human-Machine Systems, vol. 50, no. 6, pp. 604–612 (2020). DOI: 10.1109/THMS.2020.3016079.
13. Oralhan, Z.: A New Paradigm for Region-based P300 Speller in Brain Computer Interface. IEEE Access, vol. 7, pp. 106618–106627 (2019). DOI: 10.1109/ACCESS.2019.2933049.
14. Rathi, K., Gomathi, V.: Human Vision Reconstruction Using Brain Activity Profiles. In: 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1–5 (2018). DOI: 10.1109/CCAA.2018.8777542.
15. Rezeika, A., Benda, M., Stawicki, P., Gembler, F., Saboor, A., Volosyak, I.: Brain–computer Interface Spellers: A review. Brain Sciences, vol. 8, no. 4 (2018). DOI: 10.3390/brainsci8040057.
16. Sarraf, J., Vaibhaw, Pattnaik, P. K.: A Study of Classification Techniques on P300 Speller Dataset. Materials Today: Proceedings, vol. 80, pp. 2047–2050 (2021). DOI: 10.1016/j.matpr. 2021.06.110.
17. Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., Vaughan, T. M.: Brain-computer Interface Technology: a Review of the First International Meeting. IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pp. 164–173 (2000). DOI: 10.1109/TRE.2000.847807.
18. Won, K., Kwon, M., Jang, S., Ahn, M., Jun, S. C.: P300 Speller Performance Predictor Based on RSVP Multi-feature. Frontiers in Human Neuroscience, vol. 13 (2019). DOI: 10.3389/fnhum.2019.00261.
19. Yáñez-Márquez, C.: Toward the Bleaching of the Black Boxes: Minimalist machine learning. IT Professional, vol. 22, no. 4, pp. 51–56 (2020). DOI: 10.1109/MITP.2020.2994188.

ISSN 1870-4069

Path Planning Method for Navigation and Exploration with Drones Using the 3D-RRT Algorithm

Camilo Espinosa-Martinez[1], Lina Maria Aguilar-Lobo[1], Oscar J. Suarez[2], Ulises Davalos-Guzman[1], Gilberto Ochoa-Ruiz[3], Fabian Castaño[4], Alberto Ochoa-Zezzatti[5]

[1] Universidad Autónoma de Guadalajara, Mexico

[2] Universidad de Pamplona, Colombia

[3] Tecnologico de Monterrey, Mexico

[4] Pontificia Universidad Javeriana, Colombia

[5] Universidad Autónoma de Ciudad Juárez, Mexico

{camilo.espinosa, lina.aguilar, ulises.davalos}@edu.uag.mx, oscar.suarez@unipamplona.edu.co, gilberto.ochoa@tec.mx, fabian.castano@javerianacali.edu.co, alberto.ochoa@uacj.mx

Abstract. In the past few years, drone development has opened doors to new application areas giving us the chance to carry sensors onto more complex environments, such as cities and forests, in which there are proximity conditions of considerable size that could be categorized as highly dangerous. In this context, the path planning method is commonly used in robotics applications to find a valid sequence to move the drone to a target point. This method aims to find the shortest path length and obtain a safe trajectory avoiding collisions with obstacles. To achieve this objective, we present a novel path planning method for navigation and exploration with drones based on a 3D version of the RRT algorithm. The proposed algorithm developed in Python have two principal contributions, first are used a box model to encapsulate obstacles to avoid collisions and a dynamic range bias for the sampling, and, a giving orientation technique is employed on the exploration steps to reduce the number of computational operations and processing time when is obtained a valid path. Simulations results are performed to validate the algorithm using different scenarios and 3D obstacles randomly located. The results illustrated that the 3D-RRT algorithm finds a valid path avoiding obstacles with benefits on computational cost and better processing time.

Keywords: Path planning, 3D-RRT algorithm, navigation, drones applications.

1 Introduction

Path planning algorithm is a computational method consisting of generating a valid output path free of obstacles from a start point to a target point. Several path planning algorithms have been proposed in the literature [1-4], and in [5, 6] these algorithms were classified into three main categories:

- **Classical Methods**: They can find a solution or ensure a missing solution. However, these methods typically entail very costly and complex computational workloads. Therefore, they are not practical solutions for real environments [7].
- **Sampling-Based Methods**: They run sampling of configurations in space to model obstacles and possible paths. Different techniques have been implemented in past years, and the most relevant is Rapidly-Exploring Random Trees (RRT)[7].
- **Optimization Methods**: They try to solve the problem as a numeric optimization problem, this kind of algorithm start with a set of trajectories (it can be free of obstacles or not), then try to use optimization to find a solution for a valid path free of obstacles and optimal cost function, i.e., length, steps, time, among others. However, some of those cost functions usually have many local minimum [7].

Sampling-based methods are widely used because of their effectiveness and low computational cost on high dimensional spaces [8-11]. These methods use a representative configuration space and build a collision-free road map connecting points sampled from the obstacle-free space. One of the most well-known sample-based methods is the RRT algorithm proposed by [10].

Most RRT algorithm implementations have focused on path planning for holonomic systems in the 2D world, where successful solutions end in commercial products such as automatic vacuum, robots, and sweepers robots [12-15]. However, 3D point array processing is an essential task working in the 3D world to computer obtain a better description. Moving in a 3D space includes a challenge to move through space with obstacles at different heights, and the vehicle could set a path starting at low levels to go up and continue trajectories to reach the target suddenly. The novelty of the present paper is summarized as follows:

1. A path planning method for navigation and exploration with drones based on a 3D version of the RRT algorithm, which uses box encapsulation as a modeling tool to define obstacles in the 3D space. The advantage of this encapsulation is saving computing power and simplifying obstacle management to select the best trajectory avoiding obstacles from the start point to the target point in a 3D surface.

2. The implementation and validation of the proposed algorithm are performed using Phyton.

The rest of the document is organized as follows: In section 2, related works of path planing methods are presented. In Section 3 the classical RRT algorithm is provided.

Section 4 presents our model of the 3D-RRT algorithm, which is the main contribution of this paper. Simulation results are reported in Section 5, illustrating different scenarios. On the other hand, the discussion is presented in Section 6. Finally, conclusions and future works are drawn in Section 7.

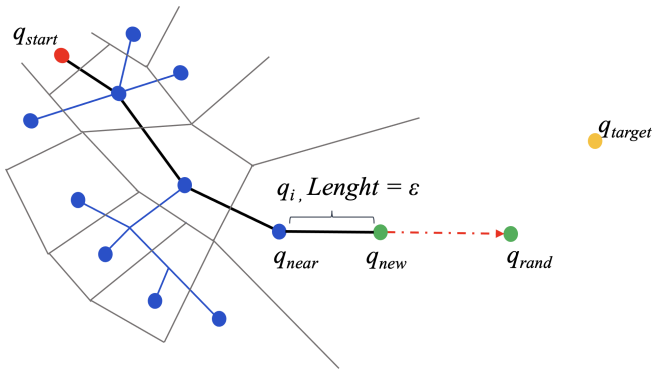


Fig. 1. RRT explore space to find a trajectory P from q_{start} to q_{target} composed by segments q_i, using random points and length ϵ.

2 Related Works

Recently, in the literature, several works related to methods for the path planing problem has been reported. In [18] is presented a path planing system for ground robots in 3D environments using point clouds as input using a 3D range sensors, presenting advantages in efficiency computation time and obtained a safety and effective path. The main reference here is use of special geometries to speed up the computational operations and reduce the subset of sampling space used by RRT algorithms.

In reference [19] is presented an improves RRT-connect algorithm for path planning for urban low altitude UAV with an improves RRT-connect algorithm, using a optimization to search step length, parent node selection and branch orientation to reduce the path length and algorithm time. However this technique is reduced to qualify new branches compared to an angle ideal for UAV. Our proposal is covering a wide range of solution, using searching direction focused around a vector in direction of target, not new point, this speed up the global search, not only new branches.

The work presented in [20] proposes a hybrid algorithm for path planing in complex offshore areas, using an improve of the particle swarm optimization (PSO) for global path planing and Artificial Potential Field (APF) to solve the local minimum problem. In [21] is presented three novel versions of the RRT algorithm with metaheuristics algorithms to solve 3D path planning problem in autonomous UAVs, where are employed the advantages of the two methods. These new hybrid models try to find solutions close to the optima, avoiding obstacles with a efficient execution time and space. However, the metaheuristic-based algorithms are disadvantaged as they demand a predetermined knowledge of intermediate stations.

Another work [22] proposes an improved version of B-RRT, named BPIB-RRT*, that employed a greedy connect a heuristic for the connection of two-directional trees. However, it is still not very successful in exploration. On the other hand, although it is recommended as a 3D path planning method, it is possible to use it mainly in 2D environments. This restriction is mainly valid to all B-RRT versions, and the results shown an execution time higher that our model.

Algorithm 1: RRT

```
1  Input: Initial Configuration with q_start and q_target located in space C ;
   Result: Path valid from q_start point to q_target point
2  τ ← INITTREE ( q_start)
3  while ¬ STOPCRITERIA do
4  |   q_rand ← RANDCONF()
5  |   q_new ← EXTEND (τ , q_rand )
6  |   if q_new = q_target then
7  |   |   return Path
8  |   else
9  |   |   return EMPTY
10 |   end
11 end
```

There are other approaches in the state of art that include dynamic considerations using point-mass model for cluttered environments in specific [23] propose a solution of 3 level to plan and find a feasible trajectory, in the third level use a modified RRT (SST) in a reduced sampling space, the current RRT 3D proposal could give and improvement in computational time for the third level even reducing more the searching volume.

3 RRT algorithm

Taken from [7], the path planing method considers a configuration C in $\mathbf{R^d}$ space where $C \subseteq \mathbf{R^d}$, then C contains all possible configurations in the space and C_{free} contains the set of configurations free of obstacles.

In this context, the state $q \in \mathbf{R^d}$ is the point in the configuration space that indicates the position and direction in the space C. Considering a trajectory P as a series of N configurations joining points q_i linked by $N-1$ segments, where each segment is a direct line segment from q_i to $q_i + 1$ represented by $(q_i\ ,\ q_{i+1})$, this trajectory P is presented in equation (1) as follows:

$$P = \cup_{i=1}^{N-1}(q_i, q_{i+1}). \tag{1}$$

Based on equation (1), two points are selected (start point (q_{start}) and a target point (q_{target})). The purpose of RRT algorithm is to find a valid trajectory P (showed as black line in Figure 1), where all segments $q_i \in C_{\text{free}}$.

To achieve this purpose, RRT algorithm will build a tree with line segments found exploring the space of configurations C using q_{rand} points $\in C$ as can be seen in Figure 1.

Algorithm 1 shows the classical RRT proposed in [10]. This algorithm builds a tree of line segments that explores multiple regions of space C using randomly based functions to generate q_i configurations until they meet q_{target} with a good trajectory free of obstacles. The RRT algorithm receives the space C which contains q_{start} and q_{target} and returns a valid path P.

The algorithm initializes the search tree with root in q_{start} (line 1). Then a loop is used to explore over the space, searching the path free of obstacles. Line 2 shows the

Algorithm 2: EXTEND - RRT

1 **INPUT:** τ, q_{rand}
  Result: q_{new}
2 $q_{\text{near}} \leftarrow \text{NEAREST}(\ \tau\ ,\ q_{\text{rand}}\ )$;
3 $q_{\text{new}} \leftarrow q_{\text{near}} + (q_{\text{near}} + \frac{|\ q_{\text{rand}_i} - q_{\text{near}_i}\ |}{\epsilon})$
4 **if** $\text{VALIDSEGMENT}(q_{near}, q_{new})$ **then**
5 | ADDSEGMENT
6 | **return** q_{new}
7 **else**
8 | **return** EMPTY
9 **end**

use of STOPCRITERIA function to stop the loop in one of two possible scenarios; the first one q_{target} has been met from q_{start}, and the second one is a maximum number of iterations in order to prevent infinite loop. Line 3 select a q_{rand} from valid space C, as possible next step in the trajectory. This value is used as a parameter for the EXTEND function. As result a q_{new} is added to tree τ (line 4).

If q_{new} is equal to q_{target}, the complete path has been founded (line 6 and line 7); otherwise, the process continues running until STOPCRITERIA function stops the algorithm. The second scenario in STOPCRITERIA function uses a maximum number of iterations as a parameter; it is usually set to a specific number of iterations (100, 300, 500, 1000, and others) if the algorithm does not find a valid trajectory under this maximum number of iteration returns an empty tree.

The EXTEND function details (line 4 in Algorithm 1) are show in Algorithm 2, where it finds the nearest point q_i to q_{rand} in τ (line 1), then a line segment from q_{near} to q_{rand} is calculated using a factor ϵ in direction of q_{target} (line 2) see Figure 1.

In Algorithm 2, VALIDSEGMENT function determines whether the line segment is hitting an obstacle in the possible trajectory (line 3). If the function finds an obstacle, an empty segment is returned; otherwise, the segment is added to τTREE. Some implementations of the RRT algorithm use a cost function to evaluate the quality of the segments, such as the length and energy consumption on a robot's trajectory; those functions can be used by VALIDSEGMENT function to determine possible segments free of obstacles meet cost function criteria to be added to τTREE.

4 Proposed 3D-RRT Algorithm

The RRT algorithms proposed by [10,17] use a randomized data structure for path planning and the main goal is to find a continuous path from initial point (q_{start}) to target point (q_{target}), where q_{start} and $q_{\text{target}} \in C$ that is the valid configurations space. All the paths found by the traditional RRT algorithm follow the equation (1), and obstacles are modeled as polygonal structures in the 2D space, then C is presented as a possible set of rectangle paths.

The path planning in 3D scenarios need to solve the problem of find a path from start point to target point using the 3D space, not only a 2D solution implemented in a

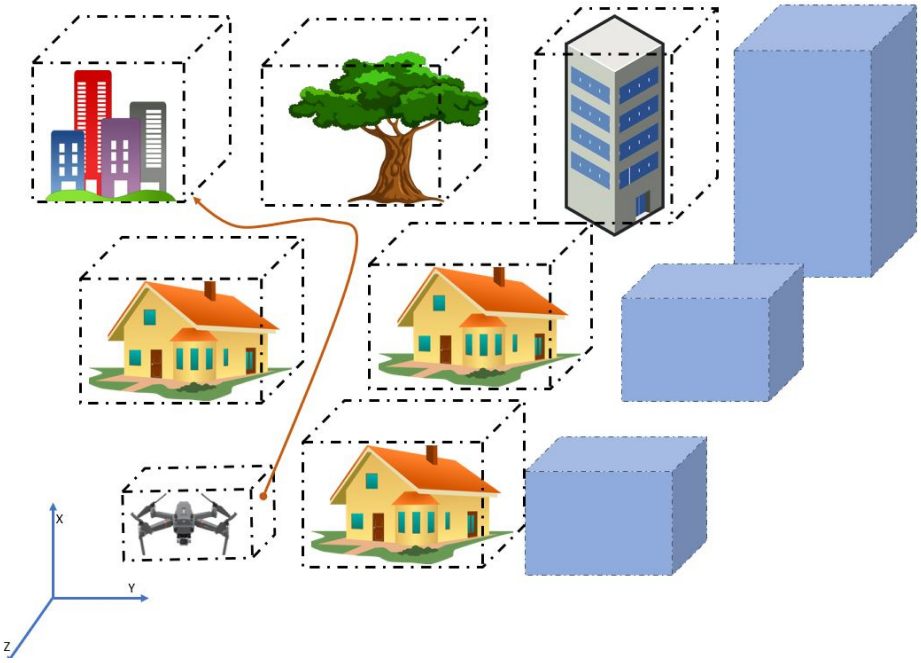


Fig. 2. Box model for obstacles.

specific Z coordinate (that allows robot's movement over the obstacles). That means to produce a real 3D solution path in the space C. In implementations for real world, the systems should be efficient in time, number of steps and iterations.

For example, a robot using 3D planning algorithm should consider the number of iterations and time to find a path as factors that consume computational power and energy. Moreover, in implementations of navigation with drones, the battery became a crucial resource, therefore, an algorithm that help to generate energy saving will be preferred over other implementations.

This paper proposes a novel path planning method for the navigation and exploration with drones based on the 3D version of the RRT algorithm using box models to encapsulate obstacles and the cost function of 3D vectorial distance as vector distance.

Our model is a solution that have the two factors became essential in the path planning method that are, the time and the iterations numbers to find a path free obstacles. Using box models to encapsulate obstacles, the cost function of 3D vectorial distance as vector distance and focus box of possible random points following vector distance to reduce time on the calculus of τTREE.

Our algorithm uses the information of the 3D space from q_{start} point to q_{target} point with various obstacles modeled as a series of rectangular boxes that can be used to set safe trajectories avoiding collisions even with not regular geometry or cubic style obstacles.

The box encapsulation allows different model obstacles from the real world with boxes to simplify finding a valid path in complex scenarios with multiple obstacles

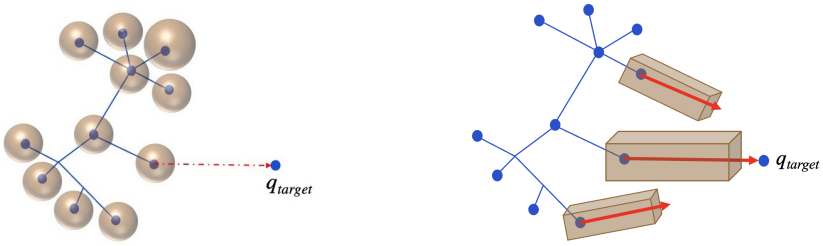


Fig. 3. Dynamic Range Proposed. (a) The Dynamic Range used in [17] uses ranges defined by spheres of radius R. (b) The Dynamic Range proposed in our 3D-RRT uses a box with length ϵ, focused in the direction of the target.

geometry. An example of the box encapsulation model in an urban environment is presented in Figure 2, where it can see that one box could contain a single house, a building, or a group of buildings.

Two features of the RRT algorithm proposed in [7] are considered in our proposal method, which is presented in Figure 4. The first refers to the minimal bias for random function using vector distance in the target direction to generate dynamic ranges that speed up the calculations and converge closer to the target point. The second one refers to an increment of minimal ϵ segments used in the proposed algorithm [7] with vectorial distance as cost function during calculations, allowing exploring the 3D space randomly.

Previous implementations of the RRT algorithms for robot and drone navigation in 3D spaces usually generate the path in the 2D plane then join it to the Z-axis, adding a vertical value. Although this approach can generate a valid route, it does not take advantage of most points through the 3D space available for other routes in multiple configurations approaches.

In this context, our proposal method presents a new form of calculating a valid path in a 3D space. Specifically, the main contributions of this paper are presented in lines 3 (RANDCONF function) and 4 (EXTEND function) of the RRT Algorithm 1 presented in [10]. The detail of EXTEND function used in our 3D-RRT proposed in the tree generation is presented in Figure 3.

The new 3D-RRT algorithm proposed are illustrated in the block diagram of Figure 4. Is important to say, that in RANDCONF were introduced the proposed modifications for the new 3D-RRT algorithm (green boxes in Fig4). The RANDCONF function generates coordinate values in each dimension in the 3D space. This function uses a range [m, n] to generate a pseudo-random value.

The proposed change considers the target's direction, using a vector pointing from the start point to the target point. The range is calculated for each dimension in space C on the target's bias, generating a cubic volume of random points in C. The start point, used to calculate the direction, is updated to the previous nearest value in each iteration; this keeps the quality of segment added.

Previous approaches used the dynamic domain for RRT [17], considering the volume of Voronoi regions with spheres of the previously defined radius. The proposed novelty

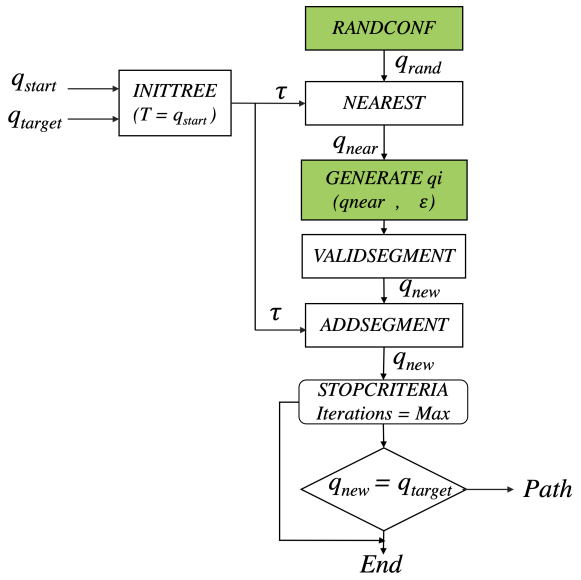


Fig. 4. Proposed 3D-RRT algorithm.

in our model speeds up the algorithm, reduces convergence over Voronoi volumes selected and proximal to the target point, reduces the number of computing operations to build trees; as a result, a worthy improvement on reducing wasting calculations over space in distant regions focus calculations.

The modified algorithm still converges as original RRT [9] without loose effective calculations. This new feature helps on drones implementations where the battery is an essential resource in 3D implementation, and at the same time is impacting in a reduction in the total number of computing operations, in consequence, the algorithm 3D-RRT contributes in the energy consumption saving energy from the battery.

On the other hand, the VALIDSEGMENT function implemented is used to determine if the segment is in collision with an obstacle. The path found q_{start} - q_{target} avoid obstacles using an encapsulation approach implementing boxes as a modular shape to run calculations.

The drone is encapsulated by a box that preserves the integrity and reduces complexity on calculations, as shown in Figure 2. A system of boxes also models obstacles, then the collision is determined by the calculation of box collision. Figure 2 shows an approximation of the box model and the drone encapsulated; in implementations of the proposed technique, a drone can use sensors to measure the height of the box, consequently the height of the obstacle.

For path planning in drones, it is necessary to use a couple of sensors above and under the chassis to detect the obstacle edges and then the box dimensions, observing that the flying or the grounded obstacles can be avoided using this approach.

The approach introduced in this paper uses a box model in the direction or target and dynamic range over this volume to run random exploration in space C. The NEAREST function will set the dimension of the box depending on q_{near} and q_{target}. This function

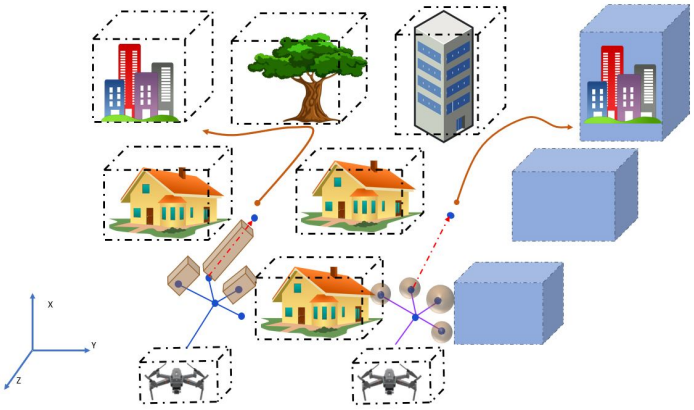


Fig. 5. Proposed 3D-RRT algorithm using boxes as modular shape.

will help to the algorithm search on the more likely and closer configuration. It is important to note that the dynamic box range occurs on each iteration on nearest q_i, so the algorithm keeps covering all space C, focusing on the target's direction.

Line 2 in Algorithm 2, shows detail of incremental ϵ from q_{near} calculated with vectorial distance to q_{rand}. In our model, the distance calculation and the random points are calculated using 3D coordinates in RANDCONF function, and q_{rand_i} points denote a component of each axis in C (where $i \in C$) were generated with dynamic box range technique as is showed in Fig 3. Thus, q_i points added to τTREE are closer to the distance vector, which represents an advantage in narrow spaces between obstacles.

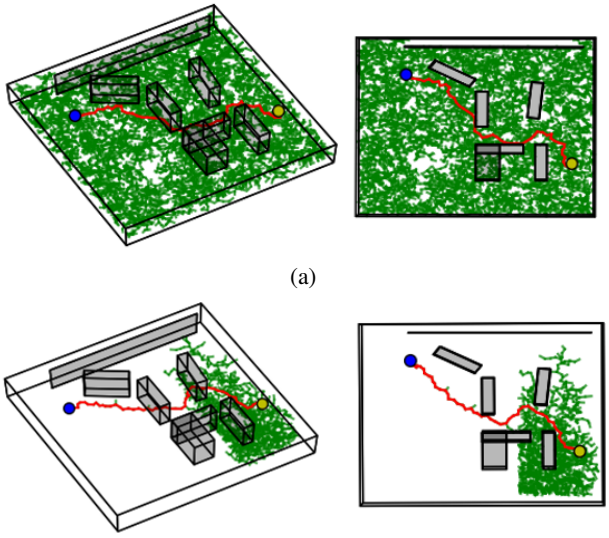
Using focused boxes to calculate new points in Tree and boxes to encapsulate obstacles help to processing the environment to generate a valid path, at first glance Figure 5 shows that dynamics boxes fits better the environments modeled with boxes, similar to maze of boxes, results found out a reductions in time to find a path and number of iteration needed; this approach could be used in real time systems where number of iterations and processing time are critical.

5 Simulation Results

In order to validate the proposed 3D-RRT algorithm, simulations with different scenarios, different numbers of obstacles, and forms of 3D obstacles, which were randomly located, have been tested using Python.

The validation results for different scenarios and forms of 3D obstacles are presented in Figure 8. These figures show the comparison between the results of the RRT algorithm proposed by [10] (Figs. 5, 5 and 5) and our version 3D-RRT algorithm (Figs. 5, 5 and 5). The starting point (q_{start}) is shown in blue and the target point (q_{target}) is shown in yellow.

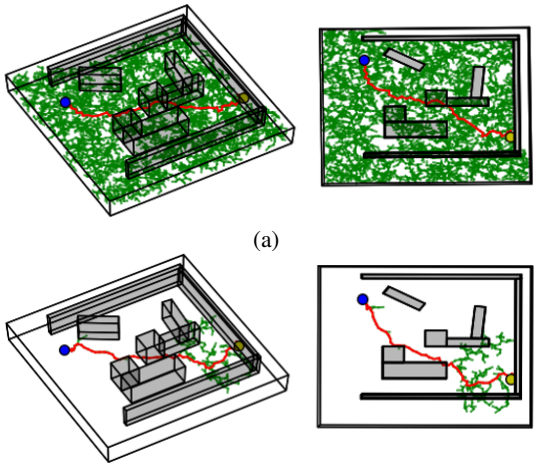
Note that the RRT algorithm proposed by [10] executes searching of new points to build paths in farther areas, even in regions previously explored, while the results of our



(a)

(b)

Fig. 6. Comparative between the results of the RRT algorithm of [10] and our proposed 3D-RRT algorithm for the case 1. Figs 5, 5 and 5 presented the results of RRT algorithm of [10], and Figs 5, 5 and 5 presents the results of our proposed 3D-RRT algorithm.

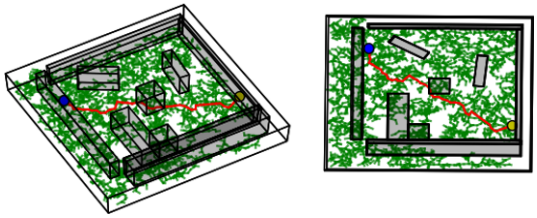


(a)

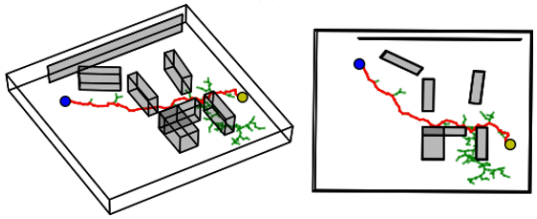
(b)

Fig. 7. Comparative between the results of the RRT algorithm of [10] and our proposed 3D-RRT algorithm for the case 2. Figs 5, 5 and 5 presented the results of RRT algorithm of [10], and Figs 5, 5 and 5 presents the results of our proposed 3D-RRT algorithm.

3D-RRT algorithm show the giving orientation on the exploring steps with the fixed boxes, it reduces the number of computational operations and processing time to obtain a valid path free of obstacles.



(a)



(b)

Fig. 8. Comparative between the results of the RRT algorithm of [10] and our proposed 3D-RRT algorithm for the case 3. Figs 5, 5 and 5 presented the results of RRT algorithm of [10], and Figs 5, 5 and 5 presents the results of our proposed 3D-RRT algorithm.

Table 1. Comparison of the performance between the RRT algorithm in [10] and our 3D-RRT proposal algorithm.

Test Cases		Time (seconds)		Iterations (number)	
Case	**obstacles**	**RRT**	**3D-RRT**	**RRT**	**3D-RRT**
1	7	204.3432	9.7121	7292	4413
2	9	45.2837	0.6609	3593	384
3	9	97.2242	1.0178	5466	556
4	6	86.2973	0.517	4557	276
5	6	160.2973	2.2086	6548	1084

Several simulation cases were executed to test our algorithm and the results are summarized in Table 1 where five cases are presented. It is important refer that although in some cases there is the same number of obstacles, the location of them is different.

This feature produces a different result in terms of performance of the algorithm, because in closes spaces is necessary more computational processing to obtain a valid path. However, we can see that in all test cases, our algorithm has a better performance in terms of execution time and number of iterations that the basic RRT algorithm.

6 Discussion

Based on the simulation results is possible to see that trees growth shows a strong bias in the target point's direction. No branches are exploring lateral edges over limits of space simulated or other areas with few probabilities to add a valid path; this approach avoids executing computational operations in regions most likely too far from the target.

The 3D exploration occurs between and over obstacles; it is not restricted to a 2D plane to increase probabilities to find valid paths. Simulations use different

configurations of obstacles to show the response of the 3D algorithm to variations in space C.

The validation results of our method show that the 3D-RRT proposed model has better performance due to the reduction of the time execution and the reduction of the total number of computational operations. Although there are few works on other versions of the 3D-RRT algorithm, all of them explore areas with classic perspective, as mentioned before.

This could include wasting exploration time on farther areas. In our model, a valid path free of obstacles is found, and the use of the box encapsulation technique saves time and computational operations compared with the classic models. The proposed changes saves time and computational operations that produce energy saving in implementation in real world examples like use of drones with 3D-RRT algorithm.

7 Conclusions and Future Works

This paper proposes a novel path planning method using the RRT algorithm for 3D surfaces in Python. The validation results for different randomly located obstacles show that it successfully finds a good valid trajectory, avoiding collisions, and a higher reduction in the execution time and in the number of iterations or computational operations. The 3D exploration gives benefits on computational cost and improvement in narrowed spaces.

Furthermore, an encapsulation approach implements a box model for both obstacles, and the drone is used. The dynamic box encapsulation model generates a bias volume through the space with the most valuable points, as presented, using length as a function of cost. The box could be spread across different Voronoi regions without losing a random exploration feature, which is ideal for spaces like streets and forests.

Future Works could cover the implementations in not urban scenarios like forest, where terrain and obstacles became more irregular, in other hand use of algorithm proposed in laboratory scenarios could help to implement different sensors and strategies to obstacles sampling. The proposed algorithm use fixed obstacles, this mean size and number of obstacles don't change during path calculation, this became an limitation to consider in real implementations. This could be address with a sampling method that generate a fresh environment samples for real implementations but that analysis is out of scope for current proposal.

Acknowledgment. This work was funded by COECYTJAL (Consejo Estatal de Ciencia y Tecnología del Estado de Jalisco) with the found FODECIJAL, Project number 7817, and the UAG (Universidad Autónoma de Guadalajara).

References

1. Khan, A., Noreen, I., Habib, Z.: On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges. Journal of Information Science and Engineering, vol. 33, no. 1, pp. 101–121 (2017)

2. Nazarahari, M., Khanmirza, E., Doostie, S.: Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Systems with Applications, vol. 115, pp. 106–120 (2019). DOI: 10.1016/j.eswa.2018.08.008.
3. Radmanesh, M., Kumar, M., Guentert, P. H., Sarim, M.: Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study. Unmanned systems, vol. 6, no. 2, pp. 95–118 (2018). DOI: 10.1142/S2301385018400022.
4. Wu, Q., Lin, H., Jin, Y., Chen, Z., Li, S., Chen, D.: A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability. Soft Computing, vol. 24, no. 3, pp. 2369–2380 (2020). DOI: 10.1007/s00500-019-04067-3.
5. Aguilar, W., Morales, S.: 3D environment mapping using the Kinect V2 and path planning based on RRT algorithms. Electronics, vol. 5, no. 4 (2016). DOI: 10.3390/electronics5040070.
6. Vagale, A., Oucheikh, R., Bye, R., Osen, O., Fossen, T.: Path planning and collision avoidance for autonomous surface vehicles I: A review. Journal of Marine Science and Technology, vol. 26, pp. 1292–1306 (2021). DOI: 10.1007/s00773-020-00787-6.
7. Garcia, N., Rosell, J., Suárez, R.: Aplicacion de algoritmos RRT en la aplicación de movimientos óptimos en robótica. In: Proceedings of the 12th Metaheuristics International Conference, pp. 953–962 (2017)
8. Chen, L., Shan, Y., Tian, W., Li, B., Cao, D.: A fast and efficient double-tree RRT*-like sampling-based planner applying on mobile robotic systems. IEEE/ASME Transactions on Mechatronics, vol. 23, no. 6, pp. 2568–2578 (2018) DOI: 10.1109/TMECH.2018.2821767.
9. Karaman, S., Frazzoli, E.: Incremental Sampling-based Algorithms for Optimal Motion Planning. Robotics Science and Systems VI, vol. 142, no. 2 (2010). DOI: 10.48550/arXiv.1005.0416.
10. LaValle, S.: Rapidly-exploring random trees: A new tool for path planning. Iowa State University, pp. 1–4 (1998)
11. Naderi, K., Rajamäki, J., Hämäläinen, P.: RT-RRT*: A Real-time Path Planning Algorithm Based on RRT*. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, pp. 113–118 (2015). DOI: 10.1145/2822013.2822036.
12. Dong, Y., Camci, E., Kayacan, E.: Faster RRT-based Nonholonomic Path Planning in 2D Building Environments Using Skeleton-constrained Path Biasing. Journal of Intelligent and Robotic Systems, vol. 28, no. 3, pp. 387–401 (2018). DOI: 10.1007/s10846-017-0567-9.
13. Aiswarya, L. Chowdhury, A.: Human Aware Robot Motion Planning Using RRT Algorithm in Industry 4.0 environment. In: 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), pp. 351–358 (2021). DOI: 10.1109/ISR50024.2021.9419511.
14. Noreen, I., Khan, A., Asghar, K., Habib, Z.: A Path-planning Performance Comparison of RRT*-AB with MEA* in a 2-dimensional environment. Symmetry, vol. 11, no. 7, pp. 945–958 (2019). DOI: 10.3390/sym11070945.
15. Wang, J., Chi, W., Li, C., Wang, C., Meng, M.: Neural RRT*: Learning-based Optimal Path Planning. IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1748–1758 (2020). DOI: 10.1109/TASE.2020.2976560.
16. Wang, J., Chi, W., Shao, M., Meng, M.: Finding a High-quality Initial Solution for the RRTs Algorithms in 2D Environments. Robotica, vol. 37, no. 10, pp. 1677–1694 (2019). DOI: 10.1017/S0263574719000195.
17. Yersova, A., Jaille, L., LaValle, S.: Dynamic-domain-RRTs Efficient Exploration by Controlling the Sampling Domain. In: Proceedings of International Conference on Robotics and Automation, pp. 3856–3861 (2005). DOI: 10.1109/ROBOT.2005.1570709.
18. Pérez-Higueras, N., Jardón, A., Rodríguez, Á., Balaguer, C.: 3D Exploration and Navigation with Optimal-RRT Planners for Ground Robots in Indoor Incidents. Sensors, vol. 20, no. 1 (2020). DOI: 10.3390/s20010220.

19. Jin, H., Cui, W., Fu, H.: Improved RRT-connect Algorithm for Urban Low-altitude UAV Route Planning. Journal of Physics: Conference Series, vol. 1948 (2021). DOI: 10.1088/1742-6596/1948/1/012048.
20. Wang, Z., Li, G., Ren, J.: Dynamic Path Planning for Unmanned Surface Vehicle in Complex Offshore Areas Based on Hybrid Algorithm. Computer Communications, vol. 166, pp. 49–56 (2021). DOI: 10.1016/j.comcom.2020.11.012.
21. Kiani, F., Seyyedabbasi, A., Aliyev, R., Gulle, M. U., Basyildiz, H., Shah, M. A.: Adapted-RRT: Novel Hybrid Method to Solve Three-dimensional Path Planning Problem Using Sampling and Metaheuristic-based Algorithms. Neural Computing and Applications, vol. 33, pp. 15569–15599 (2021). DOI: 10.1007/s00521-021-06179-0.
22. Wu, X., Xu, L., Zhen, R., Wu, X.: Biased Sampling Potentially Guided Intelligent Bidirectional RRT Algorithm for UAV Path Planning in 3D Environment. Mathematical Problems in Engineering, vol. 2019 (2019). DOI: 10.1155/2019/5157403.
23. Penicka, R., Scaramuzza, D.: Minimum-time Quadrotor Waypoint Flight in Cluttered Environments. IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 5719–5726 (2022). DOI: 10.1109/LRA.2022.3154013.

ISSN 1870-4069

Time Series Prediction of Pregnant Women with COVID–19 in Mexico

Paula Hernández–Hernández, Norberto Castillo–García

Tecnológico Nacional de México/Instituto Tecnológico de Altamira,
Department of Engineering,
Mexico

{paulahdz314,norberto_castillo15}@hotmail.com

Abstract. This study focuses on the prediction of the daily number of pregnant women infected with SARS–CoV–2 in Mexico. In particular, we developed a Fuzzy Time Series Model (FTSM) from 910 historical observations. The accuracy of the proposed model was measured by the well–known Root Mean Square Error (RMSE) index. Specifically, our FTSM obtained a RMSE value of 34.45 units. This indicates that the forecasted values fit relatively well the real data. Thus, taking into account the empirical evidence, we conclude that the values forecasted by our FTSM are really close to the real values and could be helpful in medical decision–making.

Keywords: COVID–19, pregnant patients, fuzzy time series.

1 Introduction

Pregnant women constitute an important group of risk in the COVID–19 pandemic. This population group needs to take additional precautions due to the high risk of vertical transmission, that is, the transmission of SARS–CoV–2 from the mother to the offspring. Furthermore, the presence of diseases during pregnancy such as gestational diabetes mellitus or preeclampsia increase the risk of COVID–19 infection since the immune system might be weaken [8, 9].

To June 28, 2022, in Mexico there have been $50,303$ pregnant women infected with SARS–CoV–2 and 377 of these patients sadly passed away. In the literature, some works have been mainly focused on the maternal mortality [2, 6].

Due to the aforementioned issues, it is important to know in advance the estimated number of cases for the population group under study. The estimated number could be used by the Health Minister to make appropriate decisions to prevent, control and manage COVID–19 during pregnancy. In this paper we propose a Fuzzy Time Series Model (FTSM) to forecast the daily number of pregnant women with COVID-19 in Mexico.

The main motivation to use fuzzy logic is due to the successful applications in several domains reported in the literature [7, 5, 4]. The proposed FTSM uses the average–based method to partition the universe of discourse. We use the publicly available data from the Mexican Federal Government.

The data include the documented cases from the beginning of the pandemic to date. In order to assess our FTSM, we use the Root Mean Square Error (RMSE) index. The RMSE value for the proposed FTSM was 34.45 units. This value strongly suggest a high accuracy of the model.

The remainder of this paper is organized as follows. In Section 2 we utterly describe the proposed Fuzzy Time Series Model, including the data acquisition. Section 3 reports the empirical validation of the model. Finally, in Section 4 we present the conclusions of this research.

2 Forecasting Model

2.1 Time Series Computation

The Mexican Government maintains a website devoted to share information about the dynamics of the COVID–19 pandemic [6]. The information is updated on a daily basis and distributed in a CSV (comma–separated values) file. At the time of this research, the database contains over 16.7 millions of records.

In order to obtain the time series of interest, we computed the daily number of pregnant women infected with SARS–CoV–2 from January 1, 2020 to June 28, 2022. This computation gave us a time series consisting of 910 observations. Figure 1 shows the time series under study.

2.2 Universe of Discourse

In the context of fuzzy time series, the universe of discourse is the closed interval $U = [D_{\text{min}} - D_1,\ D_{\text{max}} + D_2]$, where D_{min} and D_{max} are the minimum and maximum values observed in the time series; and D_1 and D_2 are two positive numbers conveniently determined [3]. In this study, $D_{\text{min}} = 0$, $D_{\text{max}} = 660$, $D_1 = 0$ and $D_2 = 15$. Therefore, the universe of discourse considered for this study is $U = [0,\ 675]$.

2.3 Data Partitioning

The data partitioning consists in computing a collection of sub–intervals u_i (for $i = 1, 2, \ldots, n$) from the universe of discourse U. The sub–interval lengths can be of either equal or different size [1]. In this study we partition the universe of discourse with equally–sized sub–intervals. More precisely, we use the so–called average–based partition method [10]. This method is deterministic and operates in the following way.

The first step is to compute the average of the absolute differences between each pair of consecutive observations, that is:

$$\text{avg} = \left\lfloor \frac{|0-0| + |0-0| + \cdots + |251-77| + |224-251|}{909} \right\rfloor = \left\lfloor \frac{16,514}{909} \right\rfloor = 18.$$

Then, we have to divide the average by 2, i.e., $\text{half}_{\text{avg}} = \text{avg}/2 = 9$. As can be observed, the value of half_{avg} has one digit. In this case, the partition length ℓ must be the value of half_{avg}, that is, $\ell = 9$.

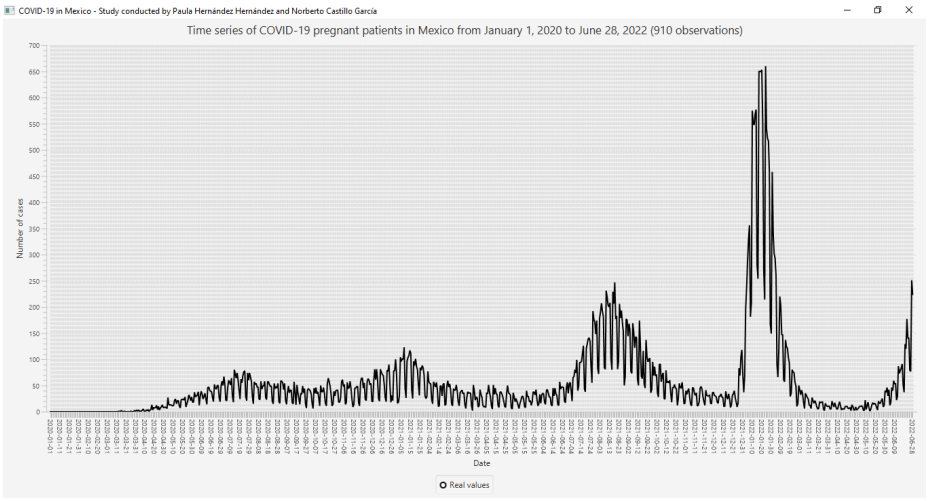


Fig. 1. Time series of pregnant women with COVID–19 in Mexico from January 1, 2020 to June 28, 2022.

The next step is to determine the number of sub–intervals n considering the bounds of the universe of discourse $U_{\text{min}} = 0$ and $U_{\text{max}} = 675$ as well as the lengths of the intervals $\ell = 9$ as follows:

$$n = \left\lfloor \frac{U_{\text{max}} - U_{\text{min}}}{\ell} \right\rfloor = \left\lfloor \frac{675}{9} \right\rfloor = 75.$$

This means that our FTSM will have $n = 75$ sub–intervals. The domain of each sub–interval is given by:

$$u_i = [U_{\text{min}} + (i-1) \times \ell,\ U_{\text{min}} + i \times \ell] \quad \forall i = 1, \ldots, n.$$

All the sub–intervals are shown in Table 1. Notice that the length of each sub–interval $u_1, \ldots, u_{75}$ has the same size ($\ell = 9$ units) and they cover the entire universe of discourse $U = [0,\ 675]$.

2.4 Fuzzification

The fuzzification consists in determining the fuzzy sets and their corresponding membership functions for the fuzzification model defined on the universe of discourse U. In this study, we adopt the approach proposed in [3], which exclusively uses triangular membership functions. Formally, triangular membership functions $T : U \rightarrow [0, 1]$ assign an element of the universe of discourse $x \in U$ to a real number between

Table 1. Sub–intervals of our Fuzzy Time Series Model.

$u_1 = [0,\ 9]$	$u_{20} = [171,\ 180]$	$u_{39} = [342,\ 351]$	$u_{58} = [513,\ 522]$
$u_2 = [9,\ 18]$	$u_{21} = [180,\ 189]$	$u_{40} = [351,\ 360]$	$u_{59} = [522,\ 531]$
$u_3 = [18,\ 27]$	$u_{22} = [189,\ 198]$	$u_{41} = [360,\ 369]$	$u_{60} = [531,\ 540]$
$u_4 = [27,\ 36]$	$u_{23} = [198,\ 207]$	$u_{42} = [369,\ 378]$	$u_{61} = [540,\ 549]$
$u_5 = [36,\ 45]$	$u_{24} = [207,\ 216]$	$u_{43} = [378,\ 387]$	$u_{62} = [549,\ 558]$
$u_6 = [45,\ 54]$	$u_{25} = [216,\ 225]$	$u_{44} = [387,\ 396]$	$u_{63} = [558,\ 567]$
$u_7 = [54,\ 63]$	$u_{26} = [225,\ 234]$	$u_{45} = [396,\ 405]$	$u_{64} = [567,\ 576]$
$u_8 = [63,\ 72]$	$u_{27} = [234,\ 243]$	$u_{46} = [405,\ 414]$	$u_{65} = [576,\ 585]$
$u_9 = [72,\ 81]$	$u_{28} = [243,\ 252]$	$u_{47} = [414,\ 423]$	$u_{66} = [585,\ 594]$
$u_{10} = [81,\ 90]$	$u_{29} = [252,\ 261]$	$u_{48} = [423,\ 432]$	$u_{67} = [594,\ 603]$
$u_{11} = [90,\ 99]$	$u_{30} = [261,\ 270]$	$u_{49} = [432,\ 441]$	$u_{68} = [603,\ 612]$
$u_{12} = [99,\ 108]$	$u_{31} = [270,\ 279]$	$u_{50} = [441,\ 450]$	$u_{69} = [612,\ 621]$
$u_{13} = [108,\ 117]$	$u_{32} = [279,\ 288]$	$u_{51} = [450,\ 459]$	$u_{70} = [621,\ 630]$
$u_{14} = [117,\ 126]$	$u_{33} = [288,\ 297]$	$u_{52} = [459,\ 468]$	$u_{71} = [630,\ 639]$
$u_{15} = [126,\ 135]$	$u_{34} = [297,\ 306]$	$u_{53} = [468,\ 477]$	$u_{72} = [639,\ 648]$
$u_{16} = [135,\ 144]$	$u_{35} = [306,\ 315]$	$u_{54} = [477,\ 486]$	$u_{73} = [648,\ 657]$
$u_{17} = [144,\ 153]$	$u_{36} = [315,\ 324]$	$u_{55} = [486,\ 495]$	$u_{74} = [657,\ 666]$
$u_{18} = [153,\ 162]$	$u_{37} = [324,\ 333]$	$u_{56} = [495,\ 504]$	$u_{75} = [666,\ 675]$
$u_{19} = [162,\ 171]$	$u_{38} = [333,\ 342]$	$u_{57} = [504,\ 513]$	–

zero and one according to Equation (1):

$$T(x; a, b, c) = \begin{cases} 0 & x \leq a, \\ \dfrac{x-a}{b-a} & a \leq x \leq b, \\ \dfrac{c-x}{c-b} & b \leq x \leq c, \\ 0 & x \geq c. \end{cases} \tag{1}$$

In the piecewise function of Equation (1), parameters a and b define the points in U where the function T reaches its lowest value: zero. Furthermore, parameter b represents the point in U where T reaches its largest value: one. Typically, this parameter is placed in the middle of the membership function, i.e., $b = (a + c)/2$.

The fuzzification model adopted in this research considers $n + 1 = 76$ fuzzy sets to entirely cover the universe of discourse. Each fuzzy set $A_1, \ldots, A_{76}$ has a corresponding triangular membership function $T_{A_1}, \ldots, T_{A_{76}}$ with its own parameter values. Table 2 shows the parameter values for each membership function.

With the fuzzification model fully defined, the next step consists in fuzzifying all of the 910 observations of the time series. The fuzzification process is performed by computing the degrees of membership of all the observations to each fuzzy set according to Equation (1). Thus, each observation belongs to fuzzy sets $A_1, \ldots, A_{76}$ with a certain level of membership. The observations are fuzzified to the fuzzy sets for which their level of membership is maximum [10]. Mathematically:

$$A_j^\star = \underset{A_i:\ i=1,\ldots,76}{\arg\max} \{\mu_{A_i}(Y(t))\} \qquad \forall t = 1, \ldots, 910,$$

Table 2. Parameter values for the triangular membership functions $T_{A_1}, \ldots, T_{A_{76}}$.

T_{A_j}	a	b	c	T_{A_j}	a	b	c	T_{A_j}	a	b	c
T_{A_1}	-18	0	18	$T_{A_{27}}$	216	234	252	$T_{A_{53}}$	450	468	486
T_{A_2}	-9	9	27	$T_{A_{28}}$	225	243	261	$T_{A_{54}}$	459	477	495
T_{A_3}	0	18	36	$T_{A_{29}}$	234	252	270	$T_{A_{55}}$	468	486	504
T_{A_4}	9	27	45	$T_{A_{30}}$	243	261	279	$T_{A_{56}}$	477	495	513
T_{A_5}	18	36	54	$T_{A_{31}}$	252	270	288	$T_{A_{57}}$	486	504	522
T_{A_6}	27	45	63	$T_{A_{32}}$	261	279	297	$T_{A_{58}}$	495	513	531
T_{A_7}	36	54	72	$T_{A_{33}}$	270	288	306	$T_{A_{59}}$	504	522	540
T_{A_8}	45	63	81	$T_{A_{34}}$	279	297	315	$T_{A_{60}}$	513	531	549
T_{A_9}	54	72	90	$T_{A_{35}}$	288	306	324	$T_{A_{61}}$	522	540	558
$T_{A_{10}}$	63	81	99	$T_{A_{36}}$	297	315	333	$T_{A_{62}}$	531	549	567
$T_{A_{11}}$	72	90	108	$T_{A_{37}}$	306	324	342	$T_{A_{63}}$	540	558	576
$T_{A_{12}}$	81	99	117	$T_{A_{38}}$	315	333	351	$T_{A_{64}}$	549	567	585
$T_{A_{13}}$	90	108	126	$T_{A_{39}}$	324	342	360	$T_{A_{65}}$	558	576	594
$T_{A_{14}}$	99	117	135	$T_{A_{40}}$	333	351	369	$T_{A_{66}}$	567	585	603
$T_{A_{15}}$	108	126	144	$T_{A_{41}}$	342	360	378	$T_{A_{67}}$	576	594	612
$T_{A_{16}}$	117	135	153	$T_{A_{42}}$	351	369	387	$T_{A_{68}}$	585	603	621
$T_{A_{17}}$	126	144	162	$T_{A_{43}}$	360	378	396	$T_{A_{69}}$	594	612	630
$T_{A_{18}}$	135	153	171	$T_{A_{44}}$	369	387	405	$T_{A_{70}}$	603	621	639
$T_{A_{19}}$	144	162	180	$T_{A_{45}}$	378	396	414	$T_{A_{71}}$	612	630	648
$T_{A_{20}}$	153	171	189	$T_{A_{46}}$	387	405	423	$T_{A_{72}}$	621	639	657
$T_{A_{21}}$	162	180	198	$T_{A_{47}}$	396	414	432	$T_{A_{73}}$	630	648	666
$T_{A_{22}}$	171	189	207	$T_{A_{48}}$	405	423	441	$T_{A_{74}}$	639	657	675
$T_{A_{23}}$	180	198	216	$T_{A_{49}}$	414	432	450	$T_{A_{75}}$	648	666	684
$T_{A_{24}}$	189	207	225	$T_{A_{50}}$	423	441	459	$T_{A_{76}}$	657	675	693
$T_{A_{25}}$	198	216	234	$T_{A_{51}}$	432	450	468	–	–	–	–
$T_{A_{26}}$	207	225	243	$T_{A_{52}}$	441	459	477	–	–	–	–

where:

- $Y(t)$ stands for the observation at time t in the time series,
- $\mu_{A_i}(Y(t))$ is the degree of membership of $Y(t)$ to fuzzy set A_i, and
- $A_j^\star$ represents the fuzzy set for which the membership level of $Y(t)$ is the largest.

Given the large number of observations in the time series, we only show the first and last five fuzzified observations in Table 3.

2.5 Fuzzy Logical Relationships

The fuzzy logical relationship $A_i \rightarrow A_j$ establishes the following. If the value at time $t-1$ is A_i then the value at time t is A_j. Thus, in this fuzzy rule, A_i is the antecedent and A_j is the consequent. In order to obtain the fuzzy logical relationships (FLRs) for our FTSM, we must relate all consecutive pair of fuzzified observations. For instance, the fuzzy logical relationship of observations $Y(906)$ and $Y(907)$ is $A_{17} \rightarrow A_{10}$ (see Table 3).

Table 3. First and last five fuzzified values of the time series under study.

t	$Y(t)$	fuzzy set	t	$Y(t)$	fuzzy set
1	0	A_1	906	141	A_{17}
2	0	A_1	907	79	A_{10}
3	0	A_1	908	77	A_{10}
4	0	A_1	909	251	A_{29}
5	0	A_1	910	224	A_{26}

Table 4. First and last four fuzzy logical relationships for the time series under study.

$A_1 \to A_1$	$A_1 \to A_1$	$A_1 \to A_1$	$A_1 \to A_1$	$\cdots$
$\cdots$	$A_{17} \to A_{10}$	$A_{10} \to A_{10}$	$A_{10} \to A_{29}$	$A_{29} \to A_{26}$

Since there are 910 observations, there will be 909 FLRs. Due to the large number of relations, we show the first and last four FLRs for our fuzzy time series model in Table 4.

Sometimes when dealing with large time series, there are $k > 1$ fuzzy logical relationships with the same antecedent, i.e., $A_i \to A_{j1}, A_i \to A_{j2}, \ldots, A_i \to A_{jk}$. In this case, the k fuzzy logical relationships can be grouped into one single fuzzy logical relationship group (FLRG), that is, $A_i \to A_{j1}, A_{j2}, \ldots, A_{jk}$.

For example, from Table 4 we can observe that $A_{10} \to A_{10}$ and $A_{10} \to A_{29}$. In this example both relations have the same antecedent (A_{10}). Thus, the (incomplete) FLRG for this antecedent is $A_{10} \to \cdots, A_{10}, A_{29}$. In fact, this group has 28 fuzzy sets in its consequent. It is important to mention that there will be one group for each fuzzy set $A_1, \ldots, A_{76}$.

Therefore, our FTSM has 76 FLRGs. Due to the large numbers of groups and fuzzy sets in their consequent, we only show 15 FLRGs in Table 5. Notice that some groups do not have any fuzzy set in their consequent (e.g., Group 70), some groups have one fuzzy set (e.g., Group 19), and some groups have two or more fuzzy sets (e.g., Group 12).

2.6 Defuzzification and Accuracy

Defuzzification obtains a crisp value from a fuzzy value. In the context of fuzzy time series models, defuzzification computes the forecasted value from the fuzzy logical relationship groups. In this study, we use the three principles proposed by Chen [3]:

1. If the fuzzy logical relationship has exactly one fuzzy set in the consequent, i.e., $A_i \to A_j$, then the output value is the middle point of A_j.
2. If the fuzzy logical relationship has $k \geq 2$ fuzzy sets in the consequent, i.e., $A_i \to A_{j1}, \ldots, A_{jk}$, then the output value is the average of the middle points of the fuzzy sets in the consequent.
3. If the fuzzy logical relationship does not have any fuzzy set in the consequent, i.e., $A_i \to \emptyset$, then the output value is the middle point of the antecedent A_i.

In order to illustrate the defuzzification process under the previous principles, let us consider three different scenarios, one for each principle.

The first scenario consists in predicting the number of pregnant women with COVID–19 on January 30,2022 (forecasted output) knowing that there were 166 pregnant women positive for COVID–19 on January 29, 2022.

Table 5. Some fuzzy logical relationship groups for the time series under study.

	$\vdots$
Group 12:	$A_{12} \rightarrow A_9, A_{12}, A_{12}, A_{15}, A_{12}, A_{13}, A_{11}, A_{11}, A_5, A_{13}, A_{10}, A_{24}, A_{12}, A_{12}, A_5, A_9$
Group 13:	$A_{13} \rightarrow A_{12}, A_{14}, A_6, A_{15}, A_{12}, A_{14}, A_{11}$
Group 14:	$A_{14} \rightarrow A_{13}, A_{15}, A_7, A_9, A_{25}, A_{13}, A_{21}$
Group 15:	$A_{15} \rightarrow A_6, A_6, A_{16}, A_{17}, A_9, A_{14}, A_{14}$
Group 16:	$A_{16} \rightarrow A_{17}, A_{11}, A_{13}, A_{15}$
Group 17:	$A_{17} \rightarrow A_{16}, A_{20}, A_{20}, A_{17}, A_8, A_{17}, A_8, A_{17}, A_{10}$
Group 18:	$A_{18} \rightarrow A_7, A_{10}, A_{15}, A_{52}$
Group 19:	$A_{19} \rightarrow A_{18}$
	$\vdots$
Group 70:	$A_{70} \rightarrow \emptyset$
Group 71:	$A_{71} \rightarrow \emptyset$
Group 72:	$A_{72} \rightarrow \emptyset$
Group 73:	$A_{73} \rightarrow A_{73}, A_{73}, A_{74}$
Group 74:	$A_{74} \rightarrow A_{62}, A_{61}$
Group 75:	$A_{75} \rightarrow \emptyset$
Group 76:	$A_{76} \rightarrow \emptyset$

Firstly, we must fuzzify the crisp value $Y(760) = 166$. The fuzzy set associated to this observation is A_{19} since the degree of membership $\mu_{A_{19}}(166) = 0.778$ is the largest. According to Table 5, the FLRG for A_{19} (Group 19) has one fuzzy set in the consequent: A_{18}. Thus, the first principle must be applied.

This principle states that the forecasted value is the middle point of A_{18}, which is the value of parameter b for $T_{A_{18}}$ (see Table 2). Therefore, the forecasted output is 153. It is important to mention that the actual value is $Y(761) = 149$. Thus, the predicted output had an absolute error of only 4 units.

In the second scenario we consider observation $Y(751) = 653$ corresponding to January 20, 2022. The fuzzy set associated to this observation is A_{74} since $\mu_{A_{74}}(653) = 0.778$ is the largest. In this case the FLR to be used is $A_{74} \rightarrow A_{62}, A_{61}$ (Group 74). Since the consequent has two fuzzy sets, the second principle applies. Thus, the forecasted output is the average of the middle points of A_{62} and A_{61}, which is $\lfloor (549 + 540)/2 \rfloor = 544$. Therefore, our FTSM predicts 544 pregnant women positive for COVID–19 on January 21, 2022. According to official data, the actual value is $Y(752) = 552$, which implies a relatively low error of 8 units.

Finally, in the third scenario we consider a hypothetical observation $x = 619$. This observation belongs to fuzzy set A_{70} with the maximum degree of membership, i.e., $\mu_{A_{70}}(619) = 0.889$. The fuzzy logical relationship to be used is $A_{70} \rightarrow \emptyset$ (Group 70).

Notice that there is no any fuzzy set in the consequent, and hence, the third principle applies. According to this principle, the forecasted output is the middle point of A_{70}. Therefore, the predicted number of pregnant women with COVID–19 for the following day is 621. Any prediction of time series can be assessed in terms of its accuracy.

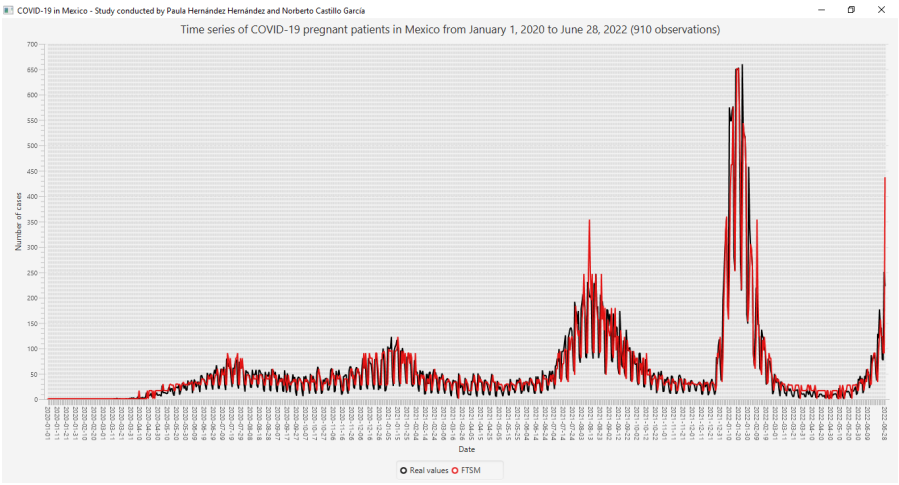


Fig. 2. Real values (black) against forecasted values (red) of the daily number of pregnant women with COVID–19 in Mexico (RMSE = 34.45).

In this paper, we assess the accuracy of our proposed Fuzzy Time Series Model by means of the well–known Root Mean Square Error (RMSE) according to Equation (2):

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{T} \left(Y(t) - \widehat{Y}(t) \right)^2}{T}}, \tag{2}$$

where:

- $Y(t)$ represents the value observed at time t in the time series (real value),
- $\widehat{Y}(t)$ stands for the value forecasted by the FTSM for time t, and
- T is the number of paired values (real and forecasted).

It is important to mention that low values of RMSE indicate a good accuracy while high values denote a poor accuracy.

3 Model Validation and Discussion

Our Fuzzy Time Series Model (FTSM) was implemented in Java (JDK 1.8.0_121) and executed on a workstation with an AMD Ryzen Threadripper 3960X 24–Core Processor at 3.80 GHz and 128 GB of RAM. Figure 2 depicts the actual and forecasted values for the time series under study.

From Figure 2, we can observe that the forecasted values fit relatively well to the original time series. This observation is numerically consistent with the RMSE value (34.45 units). This value is low since it is closer to the minimum RMSE value than the maximum. The minimum RMSE value is zero and is obtained when each pair of observations have the same value, i.e., $Y(t) = \widehat{Y}(t)$ for all t.

The maximum value is obtained when all the forecasted values are equal to the lower bound of the universe of discourse, i.e., $\widehat{Y}(t) = 0$ for all t. Thus, the maximum possible RMSE value for this time series is 103.08. Clearly, the RMSE value of our model is low, and hence, it has good accuracy.

4 Conclusions

In this paper, we developed a Fuzzy Time Series Model (FTSM) to forecast the daily number of pregnant women who tested positive for COVID–19 in Mexico. The raw data were collected from the official web site of the Mexican Government. The time series was obtained by computing the daily number of pregnant women with COVID–19 in Mexico from January 1, 2020 to June 28, 2022, totaling 910 observations.

The proposed FTSM uses the average–based method to partition the universe of discourse and the three principles of Chen to perform the defuzzification process. Our forecasting model was implemented in Java and its accuracy was assessed by the Root Mean Square Error (RMSE).

Our model obtained a RMSE value of 34.45 units. This value is low and therefore the accuracy of the model is high. According to the empirical results, we conclude that the FTSM proposed in this research produces reliable predictions and could be used in benefit of the important group of risk studied here.

Acknowledgments. The authors would like to thank *Tecnológico Nacional de México* (TecNM) and the National Council for Science and Technology of Mexico (CONACYT) for their support in this research.

References

1. Bose, M., Mali, K.: Designing fuzzy time series forecasting models: A survey. International Journal of Approximate Reasoning, vol. 111, pp. 78–99 (2019). DOI: 10.1016/j.ijar.2019.05.002.
2. Carvalho-Sauer, R. C. O., Costa, M., Teixeira, M. G., Nascimento, E. M., Silva, E. M. F., Barbosa, M. L., Silva, G., Santos, T. P., Paixao, E. S.: Impact of Covid-19 Pandemic on Time Series of Maternal Mortality Ratio in Bahia, Brazil: Analysis of period 2011–2020. BMC Pregnancy and Childbirth, vol. 21, no. 1, pp. 1–7 (2021). DOI: 10.1186/s12884-021-03899-y.
3. Chen, S. M.: Forecasting Enrollments Based on Fuzzy Time Series. Fuzzy Sets and Systems, vol. 81, no. 3, pp. 311–319 (1996). DOI: 10.1016/0165-0114(95)00220-0.
4. Hernández-Hernández, P., Castillo-García, N.: Optimization of Route Planning for the package Delivery Problem Using Fuzzy Clustering. Technological and Industrial Applications Associated with Intelligent Logistics, pp. 239–252 (2021). DOI: 10.1007/978-3-030-68655-$0_1$2.
5. Hernández-Hernández, P., Castillo-García, N., Rodríguez-Larkins, E., Guerrero-Ruiz, J. G., Morales-Díaz, S. V., Resendiz, E. S.: A Fuzzy Logic Classifier for the Three Dimensional Bin Packing Problem Deriving from Package Delivery Companies Application. Handbook of research on metaheuristics for order picking optimization in warehouses to smart cities, pp. 433–442 (2019). DOI: 10.4018/978-1-5225-8131-4.ch025.

6. Lumbreras-Marquez, M. I., Fields, K. G., Campos-Zamora, M., Rodriguez-Bosch, M. R., Rodriguez-Sibaja, M. J., Copado-Mendoza, D. Y., Acevedo-Gallegos, S., Farber, M. K.: A Forecast of Maternal Deaths with and Without Vaccination of Pregnant Women Against COVID-19 in Mexico. International Journal of Gynaecology and Obstetrics, vol. 154, no. 3, pp. 566–567 (2021). DOI: 10.1002/ijgo.13788.
7. Melliani, S., Castillo, O.: Recent Advances in Intuitionistic fuzzy logic systems (2019). DOI: 10.1007/978-3-030-53929-0.
8. Mirzadeh, M., Khedmat, L.: Pregnant Women in the Exposure to COVID-19 Infection Outbreak: The Unseen Risk Factors and Preventive Healthcare Patterns. The Journal of Maternal-Fetal and Neonatal Medicine, vol. 35, no. 7, pp. 1377–1378 (2022). DOI: 10.1080/14767058.2020.1749257.
9. Takemoto, M. L., Menezes, M. O., Andreucci, C. B., Knobel, R., Sousa, L. A., Katz, L., Fonseca, E. B., Magalhães, C. G., Oliveira, W. K., Rezende-Filho, J., Melo, A., Amorim, M.: Maternal Mortality and COVID-19. The Journal of Maternal-Fetal and Neonatal Medicine, vol. 35, no. 12, pp. 2355–2361 (2022). DOI: 10.1080/14767058.2020.1786056.
10. Yu, H. K.: Weighted Fuzzy Time Series Models for TAIEX forecasting. Physica A: Statistical Mechanics and its Applications, vol. 349, no. 3-4, pp. 609–624 (2005). DOI: 10.1016/j.physa. 2004.11.006

ISSN 1870-4069

Exploración de representaciones para identificar relaciones entre atributos

Alfredo Piero Mateos-Papis[1], Christian Sánchez-Sánchez[1], Héctor Jiménez-Salazar[1], Natalí N Guerrero-Vargas[2,3], Alberto Manuel Ángeles-Castellanos[2,4], Carolina Escobar[2,5]

1 Universidad Autónoma Metropolitana, México

2 Universidad Nacional Autónoma de México, Facultad de Medicina, México

{csanchez, amateos, hjimenez}@cua.uam.mx
natalinadi@facmed.unam.mx, mangeles_castellanos@unam.mx
cescobar@comunidad.unam.mx

Resumen. La motivación de este trabajo es el desarrollo alarmante del sobrepeso y obesidad en adultos jóvenes, lo cual no se explica tan solo por aspectos individuales como dieta, sedentarismo, estrés, u otros conocidos, sino que recientemente se ha encontrado explicación en aspectos sociales de la vida moderna, caracterizados por la actividad que se prolonga hacia la noche, que afectan al sistema circadiano. En este trabajo se analizan datos obtenidos de un cuestionario aplicado a jóvenes, alumnos de la carrera médico cirujano, sobre hábitos diarios que afectan al sistema circadiano, como sueño-vigilia, alimentación por la noche, exposición a luz nocturna, y uso de aditamento luminosos por la noche, además de datos como el índice de masa corporal (IMC), que determina la condición de sobrepeso u obesidad, para buscar la relación entre dichos hábitos y el IMC (atributos). Hasta donde los autores conocen, hay pocos trabajos que se apoyen en el Análisis de Componentes Principales (PCA), como sí lo hace el presente trabajo, donde la cantidad de atributos considerados entorpecen la obtención de dependencias de interés entre tales atributos. Así, a partir de buscar la relación entre el IMC con los otros atributos, aquí se propone un criterio para identificar los componentes que mejor muestran las relaciones entre un atributo, en este caso el IMC, y los otros. Es importante mencionar que con las técnicas tradicionales, desde las más simples, como la correlación entre las variables, hasta la representación directa en componentes principales, o la separación no lineal a través de Kernel PCA, como coseno o sigmoidal, no se pudieron identificar relaciones entre el IMC y otros atributos, pero esto sí se logró a través de una búsqueda que maximizara las proyecciones del atributo IMC en algunos de los componentes principales. En el trabajo se muestran, visualmente, los atributos que +más tienen relación con el IMC.

Palabras claves: Componentes principales, identificación de dependencias, ritmo circadiano.

Exploring Representations for Identification of Relations between Attributes

Abstract. The motivation for this work is the alarming increase in overweight and obesity in young adults. This trend is not only explained by individual factors such as diet, sedentary lifestyle, stress, or other known factors. It has also recently been explained by social aspects of modern life, characterized by nighttime activity, which affects the circadian system. This work analyzes data obtained from a questionnaire administered to young students of medical and surgical studies regarding daily habits that affect the circadian system, such as sleep-wakefulness, nighttime eating, exposure to light at night, and use of luminous devices at night. This questionnaire also analyzes data such as the body mass index (BMI), which determines overweight or obesity, to identify relationships between these habits and BMI (attributes). To the authors' knowledge, few studies rely on Principal Component Analysis (PCA), as this study does. The number of attributes considered makes it difficult to determine relevant dependencies between these attributes. Thus, by searching for the relationship between BMI and other attributes, a criterion is proposed here to identify the components that best reflect the relationships between one attribute, in this case BMI, and the others. It is important to mention that traditional techniques, from the simplest, such as correlation between variables, to direct representation in principal components, or nonlinear separation through Kernel PCA, such as cosine or sigmoidal, failed to identify relationships between BMI and other attributes. However, this was achieved through a search that maximized the projections of the BMI attribute on some of the principal components. The attributes most closely related to BMI are visually shown in this paper.

Keywords: Principal components, identification of dependencies, circadian rhythm.

1. Introducción

El aumento progresivo y alarmante que ha tenido en los últimos 20 a 30 años el sobrepeso y la obesidad, en México y en algunos países del Mundo, no se logra explicar por las causas más exploradas en el ámbito individual, como la dieta, el sedentarismo, el estrés, alteraciones hormonales, fármacos, factores genéticos, etc. Este aumento puede deberse a la actividad y trabajo 24 h / 7 días del estilo de vida moderno [1-2], donde las personas: 1) prolongan su actividad hacia la noche reduciendo las horas dedicadas a dormir [3]; 2) tienen horarios irregulares de alimentación [4] y consumen más alimentos por la noche [5]; y 3) se exponen a contaminación lumínica por la noche [6-7]. Estas condiciones afectan al sistema circadiano de las personas, que tiene como función principal adaptar a los organismos a un ambiente cíclico dado por el día y la noche, y regular la homeostasis. Entre los grupos sociales mayormente expuestos a este nuevo estilo de vida están los adolescentes y adultos jóvenes, y los profesionales de la

salud [8, 9, 6], donde se identifica el "Jetlag social" [10]. Estudios clínicos y experimentales recientes señalan que la disrupción circadiana es un factor de riesgo para desarrollar enfermedades crónicas, incluyendo al sobrepeso y obesidad y enfermedades metabólicas [11].

Con el fin de conocer los factores que afectan al sistema circadiano que más inciden en el desarrollo de sobrepeso y obesidad, un grupo de investigadores de la Facultad de Medicina de la UNAM [12] aplicó un cuestionario a jóvenes alumnos de la carrera de médico cirujano, colectando datos de peso y altura; y hábitos diarios sobre sueño-vigilia, alimentación, exposición a luz nocturna y uso de aditamentos electrónicos por la noche, para generar datos con decenas de atributos, entre los cuales está el "Índice de Masa Corporal" (IMC), donde $\text{IMC} = Kg\ de\ peso/(metros\ de\ altura)^2$, que representa al sobrepeso y la obesidad. Este grupo pidió apoyo para el análisis de datos a un grupo de estudio de ciencia de datos y computación. Estos dos grupos son los autores de este trabajo.

La mayor parte de los estudios sobre los factores que disrumpen al sistema circadiano, que pueden causar alteraciones como sobrepeso y obesidad en las personas, obtienen sus datos de cuestionarios, o de pruebas con grupos de personas en condiciones controladas de laboratorio, y usan técnicas de análisis estadístico para obtener resultados; pero a conocimiento de los autores, hay pocos trabajos que apliquen el Análisis de Componentes Principales, PCA[1], como técnica de análisis estadístico, que ayuda en situaciones como las del presente trabajo, donde tanto la cantidad de atributos considerados, aunque sean "solamente" decenas, como su alta correlación, entorpecen o impiden la obtención de resultados. El objetivo de este trabajo es buscar la relación o dependencia entre el IMC con los otros atributos de los datos, usando PCA.

Con relación al PCA, ésta es una técnica proviene del año 1900 [13], que se ha desarrollado con intensidad recientemente por el poder del cómputo. Con PCA se realiza una secuencia de transformaciones lineales de los datos "originales", cada transformación con relación a lo que se llama un componente principal[2] (PC[3]), donde los datos "transformados" con relación al primer componente principal (PC (1)) son los más cercanos a los datos "originales"; los datos transformados con relación al segundo componente principal (PC (2)) son los siguientes más cercanos, y así sucesivamente. Se dice que PC (1) recaba la mayor información de los datos, y así sucesivamente [14]. Los datos en torno a algún PC, con relación a los datos en torno a cualquier otro PC, no están correlacionados. Estas características de PCA permiten seleccionar los primeros PC, desechando los restantes y reteniendo una mayor información sobre los datos originales, para reducir la cantidad de datos "transformados" y así simplificar el análisis. En la literatura sobre PCA se recomienda seleccionar los PC con más información [15-17].

En este trabajo no se tuvieron resultados claros con el uso de los PC con más información; en contraste, sí se obtuvieron resultados más claros al utilizar otros PC con menos información que sin embargo sí tenían una mayor información considerando

[1] Principal Component Analysis.
[2] Con relación al significado de PC ver el pie de página 6.
[3] Principal Component.

solo al IMC. Los autores consideran que este resultado es un hallazgo que ofrece una opción atractiva para el análisis de datos. El balance entre la información global y la información relativa a un solo atributo (el IMC) se realizó con el uso de un indicador propuesto en este trabajo. Los resultados de este trabajo se limitan a realizar operaciones que resuelven un problema particular, con observación visual en una gráfica que se llama Biplot, que presenta gráficamente los resultados del PCA.

En el Apéndice (Sección 8) se presenta la lista de los atributos correspondientes a los datos usados en este trabajo, que incluyen el identificador de dos letras con que se representan en el cuerpo del trabajo.

Las siguientes secciones de este trabajo son: Sección 2 - Trabajos Relacionados; Sección 3 - Descripción de los datos; Sección 4 - Método de análisis (e hipótesis); Sección 5 - Resultados; Sección 6 - Conclusiones y trabajo futuro.

2. Trabajos relacionados

Esta sección presenta algunos trabajos relacionados que vinculen al IMC con factores del estilo de vida, indicando qué métodos de análisis utilizan.

Subu et al. [18], en su estudio buscan si existe relación entre la adicción a los videojuegos y el IMC, enfocado a 250 alumnos de secundaria en Indonesia. El análisis usa la herramienta SPSS, determinando dos clases de nivel de adicción: bajo y alto. El estudio mostró que de 177 estudiantes con nivel bajo de adicción a los videojuegos 126 tenían IMC normal, mientras que estudiantes con mayor nivel de adicción tendían a tener sobrepeso; sin embargo reportan que con sus datos la relación entre el nivel de adicción y el IMC fue débil e inversa, y plantean en el trabajo futuro tratar de entender esa relación y en su caso si existe causalidad alguna.

Johnson et al. [19] estudiaron la asociación del insomnio, o duración corta del sueño menor a 6 horas, con hipertensión, diabetes tipo 2 e IMC. Aunque en su estudio no encontraron evidencia de que el insomnio, o dormir menos de 6 horas, se asociara con el IMC, menciona que se apoyaron principalmente en otro estudio publicado por Crönlein et al. [20]. Es importante mencionar que ellos utilizaron 2,345 artículos, extraídos de 5 bases de datos, con 11 estudios individuales para una revisión cualitativa y 10 estudios para un metaanálisis, donde el tamaño de la muestra varió de 30 hasta 4,994 participantes; sin embargo, reportan que son muy pocos los estudios que busquen dicha relación.

Contrario a esto, los investigadores Bocicor et al. [21] también realizaron un estudio sobre la relación entre desorden de sueño y el incremento de IMC en pacientes adultos. En el estudio se sondearon 84 pacientes, entre 18 y 65 años, quienes respondieron un cuestionario (SAQ©) que ayudaba a determinar el nivel de desorden del sueño, donde un desorden medio de sueño lo obtienen quienes alcanzan entre 4 a 5 puntos y un severo desorden de sueño es obtenido con 6 a 7 puntos. La asociación entre el desorden de sueño y el IMC fue probada estadísticamente (Chi cuadrada y Prueba exacta de Fisher), y los investigadores reportan que sí encontraron una asociación entre el desorden de sueño y el IMC.

Por otro lado, se han presentado algunos estudios usando PCA y Biplot para estudiar problemas de salud, tal es el caso de Durankova et al. [22], que presentan un estudio de los efectos del estilo de vida en la salud física de alumnos de la Universidad en Eslovaquia. Los autores estudiaron el IMC en relación con nueve factores (lugar, comida, actividad, cantidad de actividad física, tipo de actividad física, consumo de alcohol, consumo de tabaco, estrés y dormir), en una muestra de 200 estudiantes (108 hombres y 92 mujeres). Los datos fueron obtenidos mediante un cuestionario. Los resultados mostraron diferencias entre ambos sexos, ya que mientras la actividad diaria y la actividad física fueron asociados son mayor obesidad en los hombres para las mujeres lo fue el consumo del tabaco.

También Imran et al. [23] presentan un estudio sobre el patrón de prevalencia de sobrepeso y obesidad y factores de riesgo asociado en la división de Malakand en Pakistán, con una muestra de 562 individuos, de los cuales obtuvieron datos sobre su estilo de vida, alimentación, presión arterial, consumo de tabaco, enfermedades y actividades de su vida. Para realizar revisar la relaciones los autores utilizaron análisis multivariado, análisis de clusters y técnicas de análisis de correspondencia canónica, reportan que la obesidad es mayor para mujeres casadas y en hombres la obesidad se relaciona por la comida y su estilo de vida.

Es importante mencionar que actualmente el uso de la computación puede ayudar a determinar relaciones entre diversos factores y el IMC, por ejemplo Zhang et al. [24] desarrollaron una revisión de artículos que buscan determinar el IMC basado en el análisis de los rostros humanos. Se presentan enfoques basados en predicciones matemáticas y aprendizaje automático (Machine Learning), principalmente clasificación, que toman en cuenta características geométricas de los rostros, extraídos de diferentes fuentes, reportando buenos resultados en la mayoría de los trabajos incluidos.

Otro ejemplo del uso de la tecnología para determinar el IMC es el presentado por Mark [25], donde a través de realizar una un dispositivo, basado en IOT, compuesto por un microcontrolador y un dispositivo para medir el ancho del cuerpo y otro para medir la altura, obtienen los datos que permiten determinar el IMC del usuario, donde se utilizó un algoritmo de regresión gaussiana que puede predecir si el peso de una persona es bajo, o peso normal, o sobrepeso, u obeso, con una exactitud reportada de 99.18%.

3. Descripción de los datos

Entre los años 2018 y 2019 se aplicó el cuestionario de Cronobiología: “Hábitos circadianos”, a los estudiantes de primero y segundo año de la Licenciatura “Médico Cirujano”, de la Facultad de Medicina de la Universidad Nacional Autónoma de México (UNAM) [12]. Este estudio fue aprobado por la comisión de Investigación y Ética de la División de Investigación Facultad de Medicina UNAM, FM/DI/097/2018.

El cuestionario colectó datos generales, entre ellos edad, peso y altura; y además colectó datos de hábitos diarios sobre patrones de: 1) sueño-vigilia; 2) horarios de alimentación; 3) exposición a luz nocturna; 4) uso de aditamentos electrónicos por la

noche. Las preguntas; se orientan tanto a patrones en días entre semana y en días de fines de semana. Se excluyeron los cuestionarios que aportaran información errónea en las unidades de peso y talla, o que tuvieran datos incompatibles en cuanto a horarios. Con los valores provenientes de las preguntas, se obtuvieron valores derivados comparativos, por ejemplo el *Jetlag* de la hora de despertar, que proviene calcular el valor absoluto de la resta entre la hora de despertar entre semana y la hora de despertar en fin de semana. Al final, se obtiene una matriz con 67 columnas, de los cuales se seleccionaron 34. Cada columna, para cada individuo, tiene un dato que le da al individuo una característica o rasgo, por eso se llama atributo.

El sobrepeso y obesidad se obtienen a través del IMC (Índice de Masa Corporal), que es proporcional al peso (en Kg), e inversamente proporcional al cuadrado de la estatura (en metros), tal que los investigadores clasifican el peso como: 1) Bajo peso $(\text{IMC} < 18)$, 2) peso normal $(18 \leq \text{IMC} < 25)$, 3) sobrepeso $(25 \leq \text{IMC} < 30)$, y 4) obesidad $(\text{IMC} \geq 30)$.

Con los cuestionarios y datos seleccionados, se forma una matriz de datos de 2485 renglones por 34 columnas, que equivale a 2485 cuestionarios (uno por individuo) de 34 atributos. En la matriz de datos, el individuo asociado a cada renglón se identifica con la matrícula de la UNAM, representada por las sigla: ID; y cada uno de los 34 atributos tiene una sigla. Las 34 siglas son: G1, G2, G3, G4, G5, S1, S2, S3, S4, S5, S6, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, L1, L2, L3, D1, D2, D3, D4, D5, D6, donde el atributo G4 es el IMC, que es el atributo de interés en este trabajo, por indicar la condición de sobrepeso y obesidad. Los atributos se refieren según la letra de inicio de la sigla: G → información general; S → sueño (horas de dormir); C → consumo (alimentación); L → luz (iluminación).

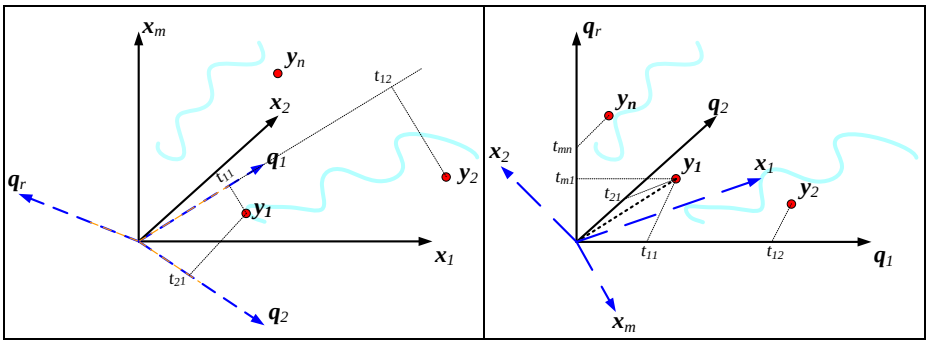
4. Método de análisis

La hipótesis de este trabajo es que mediante el análisis de componentes principales, y el B176iplot, se puede encontrar una relación que hay entre factores asociados (atributos) a la disrupción del sistema circadiano, y la presencia de sobrepeso y obesidad.

Con las variables m y n se representa el número de atributos, y el número de individuos, respectivamente (o sea $m = 34$ y $n = 2485$). Así, la matriz de datos tiene n renglones y m columnas, y se representa como $Y_{n\times m}$.

4.1 Análisis de componentes principales

Para estudiar la relación entre los atributos se puede representar gráficamente un espacio cartesiano de m dimensiones, con ejes coordenados ortonormales $\boldsymbol{x}_i\ i = 1, \dots, m$, donde cada eje coordenado corresponde a una dimensión y representa a un atributo. En este espacio se colocan n puntos, $\boldsymbol{y}_1, \dots \boldsymbol{y}_n$, cada punto representando a un individuo, y cada punto teniendo las coordenadas de uno de los renglones, $\boldsymbol{y}_{1_{1\times m}}, \dots, \boldsymbol{y}_{n_{1\times m}}$, de la matriz de datos $Y_{n\times m}$.



a *b*

Fig. 1. Representación de: *a*) El plano original m–dimensional con ejes coordenados ortonormales, $\boldsymbol{x}_1, \ldots \boldsymbol{x}_m$ (correspondientes a los m atributos), con n puntos, $\boldsymbol{y}_1 \ldots, \boldsymbol{y}_n$ (representando a n individuos), y r vectores, $\boldsymbol{q}_1, \ldots \boldsymbol{q}_r$ (representando a las r columnas de la matriz de transformación $\boldsymbol{Q}_{\boldsymbol{m}\times r}$ (ver **ecuación 1**)); *b*) El plano alternativo r–dimensional con ejes coordenados ortonormales, $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r$, con proyecciones de los n puntos, $\boldsymbol{y}_1 \ldots, \boldsymbol{y}_n$, y proyecciones de los m vectores, $\boldsymbol{x}_1, \ldots \boldsymbol{x}_m$.

Cuando m es "grande", es difícil observar y obtener conclusiones de la representación en este espacio que llamamos "original". Por lo anterior, es interesante poder encontrar un espacio cartesiano "alternativo", con menos dimensiones, que represente "cercanamente" al espacio "original", y por eso se plantea una transformación lineal matricial, que se puede lograr multiplicando la matriz $Y_{n\times m}$ por una matriz de transformación $Q_{m\times r} = [\boldsymbol{q}_{1_{m\times 1}} \quad \boldsymbol{q}_{2_{m\times 1}} \quad \cdots \quad \boldsymbol{q}_{r_{m\times 1}}]$, donde $r \leq m$, para obtener la matriz $T_{n\times r}$, tal como lo indica la **ecuación 1**. Al multiplicar $Y_{n\times m}$ por cada columna, $\boldsymbol{q}_{i_{m\times 1}}$, de $Q_{m\times r}$ se produce la transformación $\boldsymbol{t}_{i_{n\times 1}}$:

$$Y_{n\times m}Q_{m\times r} = \begin{bmatrix} \boldsymbol{y}_{1_{1\times m}} \\ \boldsymbol{y}_{2_{1\times m}} \\ \vdots \\ \boldsymbol{y}_{n_{1\times m}} \end{bmatrix} [\boldsymbol{q}_{1_{m\times 1}} \quad \boldsymbol{q}_{2_{m\times 1}} \quad \cdots \quad \boldsymbol{q}_{r_{m\times 1}}] =$$

$$T_{n\times r} = [\boldsymbol{t}_{1_{n\times 1}} \quad \boldsymbol{t}_{2_{n\times 1}} \quad \cdots \quad \boldsymbol{t}_{r_{n\times 1}}] = \begin{bmatrix} t_{1_1} & t_{2_1} & \cdots & t_{r_1} \\ t_{1_2} & t_{2_1} & \cdots & t_{r_2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{1_n} & t_{2_n} & \cdots & t_{r_n} \end{bmatrix}. \quad (1)$$

La **Figura 1(a)** presenta al plano de m dimensiones con los puntos, $\boldsymbol{y}_1 \ldots, \boldsymbol{y}_n$, y las $\boldsymbol{r}$ columnas, $\boldsymbol{q}_{1_{m\times 1}}, \ldots \boldsymbol{q}_{r_{m\times 1}}$ representadas como los vectores $\boldsymbol{q}_1, \ldots \boldsymbol{q}_r$. Hay que notar que tanto vectores como puntos tienen m dimensiones.

La transformación pretende ver a los puntos $\boldsymbol{y}_1, \ldots \boldsymbol{y}_n$ desde el punto de vista del espacio alternativo r dimensional, que tiene ejes coordenados que son los vectores $\boldsymbol{q}_1, \ldots \boldsymbol{q}_r$ (que como se explica más adelante al calcularlos resultan ortonormales). En el espacio alternativo también se puede ver como vectores a los ejes coordenados originales $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, correspondientes a los m atributos.

Antes de la transformación, es importante "normalizar" los valores de cada columna de $Y_{n\times m}$ (para tener media cero y variancia uno en cada columna de la matriz), para no sesgar la transformación hacia uno o más atributos. Por facilidad, la matriz normalizada también se representa como $Y_{n\times m}$.

En la transformación, en el espacio "original" se pretende minimizar el promedio del error entre la ubicación de los puntos $\boldsymbol{y}_1 \dots, \boldsymbol{y}_n$, y la ubicación de sus proyecciones sobre cada uno de los vectores $\boldsymbol{q}_1, \dots \boldsymbol{q}_r$ (se dice que se minimizan los residuos [17]). De la minimización se obtiene la dirección y sentido de cada uno de los vectores $\boldsymbol{q}_1, \dots \boldsymbol{q}_r$, representados en el espacio "original". La primera minimización se hace con relación a $\boldsymbol{q}_1$, la segunda con relación a $\boldsymbol{q}_2$ dado que se tiene la minimización con a $\boldsymbol{q}_1$, y así sucesivamente. Los últimos vectores $\boldsymbol{q}_i$ resultan ser los que tienen mayor residuo.

Se puede mostrar [17] que estas minimizaciones equivalen a maximizar la variancia de las proyecciones de los puntos $\boldsymbol{y}_1 \dots, \boldsymbol{y}_n$ sobre los vectores $\boldsymbol{q}_1, \dots \boldsymbol{q}_r$. Esto resulta en que la variancia de las proyecciones sobre $\boldsymbol{q}_1$ sea la mayor; la variancia de las proyecciones sobre $\boldsymbol{q}_2$ sea la siguiente mayor; y así sucesivamente.

Después de maximizar (ver [17]) resulta que los vectores $\boldsymbol{q}_1, \dots \boldsymbol{q}_r$, $r \leq m$, son los eigenvectores de la matriz $Y_{m\times n}^T Y_{n\times m}$ (la matriz de covariancias de $Y_{n\times m}$), tal que $Y_{m\times n}^T Y_{n\times m} \boldsymbol{q}_{i_{m\times 1}} = \varphi_1^2 \boldsymbol{q}_{i_{m\times 1}}$, $i = 1, \dots, r$, donde el eigenvalor de $\boldsymbol{q}_{1_{m\times 1}}$ es $\lambda_1 = \varphi_1^2$, que es el máximo eigenvalor para esa matriz; el eigenvalor de $\boldsymbol{q}_{2_{m\times 1}}$ es $\lambda_2 = \varphi_2^2$, que es el segundo mayor eigenvalor, y así sucesivamente (estos valores aparecen en la **ecuación 2**).

Por ser $Y_{m\times n}^T Y_{n\times m}$ una matriz simétrica, los eigenvectores son reales, y por ser matriz de covariancia todos los eigenvalores son positivos [17]. Además, por ser $\boldsymbol{q}_{i_{m\times 1}}, i = 1, \dots r, r \leq m$, eigenvectores, todos ellos son ortogonales cuando sus correspondientes eigenvalores son diferentes, y los eigenvectores que pudiesen corresponder a iguales eigenvalores se pueden seleccionar para ser ortogonales entre sí, de tal forma que se puede obtener un conjunto completo de eigenvectores: $\boldsymbol{q}_{i_{m\times 1}}, i = 1, \dots m$, ortonormales[4].

De la **ecuación 1** y la **Figura 1(*a*)**, es importante entender que cada columna $\boldsymbol{t}_{i_{n\times 1}}$ de $T_{n\times r}$ tiene las coordenadas de proyección de los puntos $\boldsymbol{y}_1 \dots, \boldsymbol{y}_n$ sobre $\boldsymbol{q}_i$, y como resulta que la media y la variancia de $\boldsymbol{t}_{i_{n\times 1}}$ son iguales a cero y a φ_i^2/n , respectivamente, para $i = 1, \dots, r$, $r \leq m$, entonces los valores φ_i^2 se pueden asociar a las variancias de las proyecciones sobre $\boldsymbol{q}_i$, $i = 1, \dots, r$, $r \leq m$, y las variancias relativas de las proyecciones se calculan como $\varphi_i^2 / \sum_{j=1}^{m} \varphi_j^2$ $(r = m)$. Estas variancias relativas se suelen dar de forma porcentual. Entonces, cada vector $\boldsymbol{q}_i$ tiene un valor de variancia relativa[5]. A las variancias relativas se les asocia con cantidad de información [14].

La **Figura 1(*b*)** presenta una gráfica en un espacio "alternativo" r–dimensional, que tiene como ejes coordenados ortonormales a los vectores $\boldsymbol{q}_i$, $i = 1, \dots, r$, y con la que

[4] Es decir $Q_{r\times m}^T Q_{m\times r} = I_r$. Y en caso de que $r = m$ se cumple que $Q_{m\times m}^T Q_{m\times m} = I_m$, donde I_r e I_m representan a matrices identidad de tamaño r y m, respectivamente.

[5] Además, la suma de las variancias de las columnas de $T_{n\times r}$, cuando $r = m$, es igual a m, o sea que la transformación no altera el valor total de la suma de variancias. También, la correlación entre cualesquiera dos diferentes columnas de $T_{n\times r}$ es igual a cero.

se pretende facilitar la observación de los datos para encontrar la relación entre atributos, por ser un espacio de dimensión reducida, manteniendo la mayor cantidad de "información original".

Esta técnica de transformación se llama *análisis de componentes principales* (PCA por sus siglas en inglés) [16,15], donde la palabra Componente Principal, se designa con las iniciales PC (por sus siglas en inglés).

En este trabajo, como en otros [17], cada componente principal PC(i), corresponde al eje coordenado $\boldsymbol{q}_i$ del plano "alternativo", así que a cada PC(i) corresponde una variancia relativa (cantidad de información). En otros trabajos la nomenclatura cambia[6].

4.2 Descomposición en valores singulares

Esta subsección presenta brevemente la "descomposición en valores singulares", para introducir el Biplot, que es la herramienta que se usa en este trabajo para descubrir las relaciones entre el IMC y otros atributos.

La "descomposición en valores singulares" se relaciona con PCA, y muestra que toda matriz de tamaño $n \times m$, como lo es $Y_{n\times m}$, se puede factorizar en un producto de tres matrices ([26] Pág. 284), ([14], Eq. 7.14), como lo indica[7] la **ecuación 2**, donde $n > m$:

$$Y_{n\times m} = P_{n\times n}\Phi_{n\times m}Q^T_{m\times m} = P_{n\times n}\begin{bmatrix} \varphi_1 & 0 & \cdots & 0 \\ 0 & \varphi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi_m \\ \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} Q^T_{m\times m.} \qquad (2)$$

La **ecuación 2** se puede reducir como en la **ecuación 3**:

$$Y_{n\times m} = P_{n\times m}\Phi_{m\times m}Q^T_{m\times m} = P_{n\times m}\begin{bmatrix} \varphi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varphi_m \end{bmatrix} Q^T_{m\times m.} \qquad (3)$$

Los valores φ_i, $i = 1 \ldots m$, se llaman los valores singulares de $Y_{n\times m}$. Los elementos de las ecuaciones 2 y 3 se pueden ordenar, tal que $\varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_m$. De las ecuaciones 2 y 3 resulta la ecuación 4:

[6] Hay trabajos en donde la palabra "componente" se refiere a la proyección del punto $\boldsymbol{y}_i, i = 1, \ldots, n$, sobre el vector $\boldsymbol{q}_j$, $j = 1, \ldots, r, r \leq m$ (ver por ejemplo [15] Pág. 3), es decir que la proyección de $\boldsymbol{y}_i$ sobre $\boldsymbol{q}_j$ sería el componente j de $\boldsymbol{y}_i$. Cada elemento de esa proyección se llama *score*. Además, cada elemento de $\boldsymbol{q}_j$ escalado se llama *loading* ([15] Pág. 3).

[7] Donde $P^T_{n\times n}P_{n\times n} = I_n$ y $Q^T_{m\times m}Q_{m\times m} = I_m$.

$$Y^T_{m\times \mathrm{n}} Y_{n\times m} Q_{m\times m} = Q_{m\times m}\Phi^2_{m\times m}$$
$$\Phi^2_{m\times m} = \begin{bmatrix} \varphi_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varphi_m^2 \end{bmatrix}. \quad (4)$$

Poniendo la **ecuación 4** con más detalle resulta la **ecuación 5**:

$$Y^T_{m\times \mathrm{n}} Y_{n\times m}[\boldsymbol{q}_{1_{m\times 1}} \quad \boldsymbol{q}_{2_{m\times 1}} \quad \cdots \quad \boldsymbol{q}_{r_{m\times 1}}] =$$
$$= [\boldsymbol{q}_{1_{m\times 1}} \quad \boldsymbol{q}_{2_{m\times 1}} \quad \cdots \quad \boldsymbol{q}_{r_{m\times 1}}] \begin{bmatrix} \varphi_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varphi_m^2 \end{bmatrix}. \quad (5)$$

En la **ecuación 5** se observa que los valores φ_i^2 son los eigenvalores de $Y^T_{m\times n} Y_{n\times m}$, siendo las columnas $\boldsymbol{q}_{i_{m\times 1}}$, de $Q_{m\times \mathrm{m}}$, los correspondientes eigenvectores.

En la **ecuación 5** los valores φ_i, $i = r+1, \ldots m$, se suelen desechar para obtener la matriz $\Phi'_{m\times m}$, similar a $\Phi_{m\times m}$ pero con $\varphi_i = 0$, $i = r+1, \ldots, m$, obteniendo la aproximación $Y_{(r)_{n\times m}}$ de $Y_{n\times m}$, expresada en la **ecuación 6**:

$$Y_{(r)_{n\times m}} = P_{n\times m}\Phi'_{m\times m} Q^T_{m\times m}. \quad (6)$$

La **ecuación 6** se puede simplificar aún más, como lo muestra la **ecuación 7**, donde $\Phi'_{r\times r}$ es la parte de $\Phi'_{m\times m}$ que tiene los elementos de la diagonal diferentes de cero:

$$Y_{(r)_{n\times m}} = P_{n\times r}\Phi'_{r\times r} Q^T_{r\times m}$$
$$Y_{(r)_{n\times m}} Q_{m\times r} = P_{n\times r}\Phi'_{r\times r}. \quad (7)$$

El segundo renglón de la **ecuación 7** resulta de post multiplicar la ecuación por $Q_{m\times r}$. Resulta que $Y_{(r)_{n\times m}}$ tiene rango r. La "falta de ajuste" entre $Y_{n\times m}$ y $Y_{(r)_{n\times m}}$ es la suma de los cuadrados de todos los elementos de la matriz $\left(Y_{n\times m} - Y_{(r)_{n\times m}}\right)$, y se representa como $\left\|Y_{n\times m} - Y_{(r)_{n\times m}}\right\|^2$, y es igual a $\sum_{i=r+1}^{m} \varphi_i^2$, así que mientras más pequeños sean los valores $\varphi_i, i = r+1, \ldots, m$, la falta de ajuste es menor.

Se puede mostrar que $Y_{n\times m} Q_{m\times r} = Y_{(r)_{n\times m}} Q_{m\times r}$, y por eso, considerando la ecuaciones 1 y 7, podemos escribir la **ecuación 8**:

$$T_{n\times r} = Y_{n\times m} Q_{m\times r} = Y_{(r)_{n\times m}} Q_{m\times r} = P_{n\times r}\Phi'_{r\times r.} \quad (8)$$

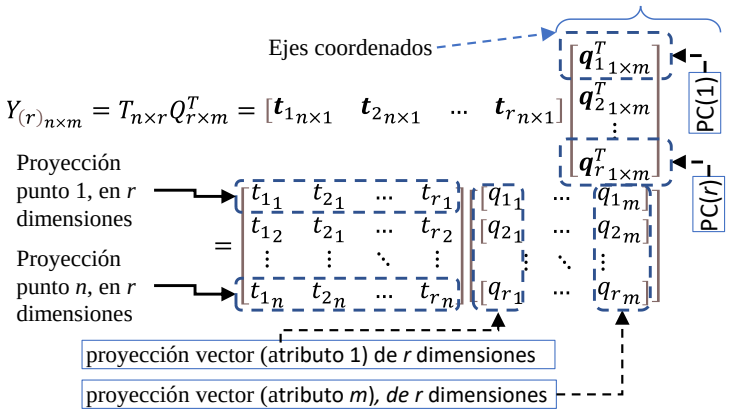
4.3 Biplot

La matriz $Y_{(r)_{n\times m}}$, por tener rango r, $r \leq m$, se puede representar como la multiplicación de dos matrices de rango r, que llamamos, matriz derecha, y matriz izquierda. Para esto, tomamos la **ecuación 7**, considerando también la **ecuación 8**, para formar la **ecuación 9**, donde $L_{n\times r} = P_{n\times r}\Phi'_{r\times r} = T_{n\times r}$, y $R_{m\times r} = Q_{m\times r}$ (lo mismo

que $R^T_{r\times m} = Q^T_{r\times m}$), y donde las letras L proviene de *Left* (izquierda) y R proviene de *Right* (derecha) [27]. Hay más formas de conformar las matrices L y R (ver ecuaciones 2 a 5 en [28])[8]:

$$Y_{(r)_{n\times m}} = P_{n\times r}\Phi'_{r\times r}Q^T_{r\times m} = [P_{n\times r}\Phi'_{r\times r}]Q^T_{r\times m} = T_{n\times r}Q^T_{r\times m} = L_{n\times r}R^T_{r\times m}. \quad (9)$$

La **ecuación 9** la podemos representar como en la **ecuación 10**, donde se indica el significado de las partes:



(10)

La **ecuación 10** contiene elementos que son proyecciones en r dimensiones de los puntos $\boldsymbol{y}_1 \dots, \boldsymbol{y}_n$, m–dimensionales, y de los vectores $\boldsymbol{q}_1, \dots \boldsymbol{q}_r$, m–dimensionales. Estas proyecciones se ven en la gráfica de la **Figura 1-b**, que representa un espacio r–dimensional con r ejes coordenados ortonormales m–dimensionales $\boldsymbol{q}_j$, $j = 1, \dots, r, r \leq m$, es decir, hay m ejes coordenados ortonormales $\boldsymbol{q}_j$, pero solamente se usan r de ellos para la gráfica, la cual se llama Biplot [29,27,30]. El prefijo "bi" se refiere a que la gráfica tiene dos tipos de proyecciones: 1) "puntos" (individuos), y 2) "vectores" (atributos). El prefijo "bi" no quiere decir que un Biplot sea una gráfica bidimensional [30], aunque es común usar un Biplot bidimensional, es decir, con $r = 2$. Es frecuente que alguno de los dos grupos de elementos se grafique escalado para lograr tener en la gráfica elementos de un tamaño visual similar. Un Biplot que usa $r = 2$ con $\boldsymbol{q}_1$ *y* $\boldsymbol{q}_2$, se dice que es un Biplot con PC(1) y PC(2).

La forma de interpretar un Biplot es observar las proyecciones de los m vectores que representan a los m atributos. Si dos proyecciones son colineales en el mismo sentido, los atributos correspondientes se refuerzan; si las proyecciones son colineales en sentido contrario, los atributos correspondientes se contraponen; si las proyecciones son perpendiculares, los atributos correspondientes no están relacionados. La relación

[8] Con relación a la **ecuación 9**, considerando que $r < m$, no hay que caer en la tentación de volver a sustituir en círculo, tal que quedara $Y_{(r)_{n\times m}} = T_{n\times r}Q^T_{r\times m} = Y_{(r)_{n\times m}}Q_{m\times r}Q^T_{r\times m}$ pues parecería que $Q_{m\times r}Q^T_{r\times m}$ es igual a I_m, y no lo es, solamente se cumple que $Q^T_{r\times m}Q_{m\times r} = I_r$. Lo mismo pasa para la matriz $P_{n\times r}$, donde resulta que $P_{n\times r}P^T_{r\times n} \neq I_n$, y solamente se cumple que pues $P^T_{r\times n}P_{n\times r} = I_r$. Cuando $r = m$ sí se cumplen todas estas expresiones.

de colinealidad y sentido se puede obtener mediante el producto punto de las proyecciones.

En el Biplot, también es de interés observar si las proyecciones de los puntos (individuos) clasificados según uno o más atributos de interés, se ubican en zonas dependiendo de los atributos elegidos. Estas zonas pueden tener una relación con la dirección y sentido de los diversos vectores (atributos), lo que es útil también para relacionar a los atributos.

Es conveniente mencionar que para la realización de las operaciones matemáticas aquí presentadas, se usa el lenguaje Python con sus diversos componentes, como son los paquetes NumPy [31] y Pandas [32] y la biblioteca Scikit-learn (Sklearn) [33] de la cual se usan los paquetes: 1) preproceessing (del cual se usa la clase StandardScaler), y 2) decomposition (del cual se usa la clase PCA( ) que contiene algoritmos de aprendizaje de máquina no supervisado para reducción de dimensiones y la clase KernelPCA para trabajo con espacios no lineales). También se usa el módulo csv y otros paquetes para hacer representación gráfica, como Seaborn [34] y Matplotlib [35].

5. Resultados

5.1 Exploración de las relaciones entre los atributos

Se tienen $m = 34$ atributos, y $n = 2485$ puntos (individuos), que forman la matriz de datos. En la **Figura 2** se observa el "mapa de calor" entre atributos, derivado de la matriz de datos, que indica las correlaciones entre los atributos. Con relación al atributo de interés G4 (IMC), no se observa relación con los demás atributos, excepto con G2 (peso).

A partir de la matriz de datos, se calculan los eigenvalores $\varphi_i^2, i = 1, \dots, m$. Por ejemplo $\varphi_1^2 = 15{,}685.748$, $\varphi_1^2/n = 6.312$, que es la variancia de la primera columna de

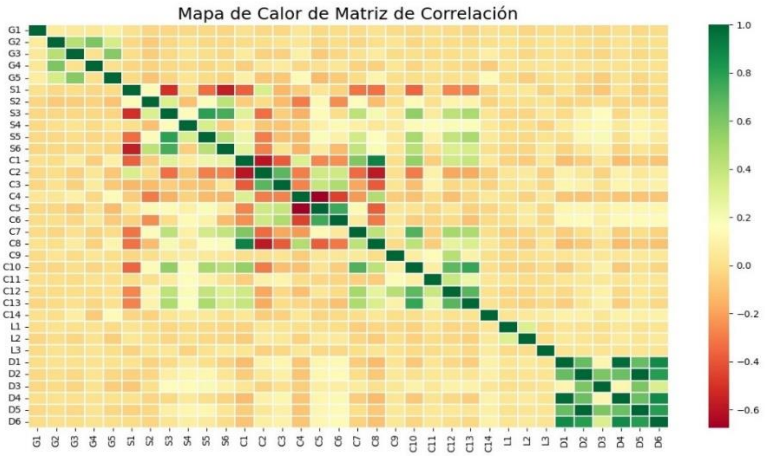


Fig. 2. "Mapa de calor" entre atributos. Con relación al atributo G4 (IMC), no se observa claramente la relación que hay con los demás atributos, excepto con G2 (peso).

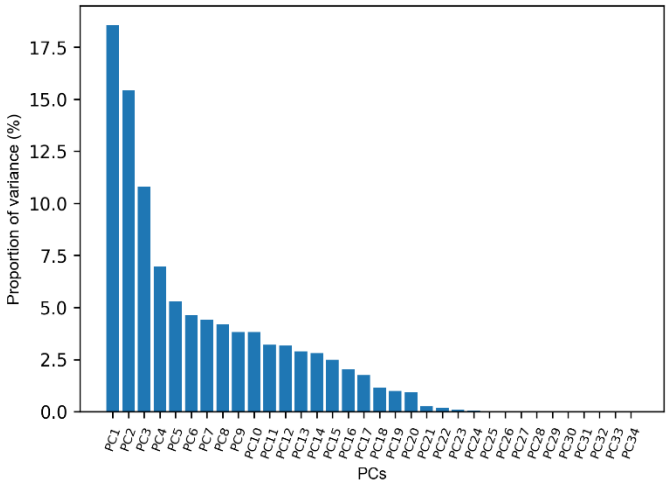


Fig. 3. *Sreeplot*: Gráfica de barras de los valores relativos de los componentes principales.

$T_{n\times r}$ (de la ecuación 1), y $\varphi_1^2/\sum_{j=1}^{m}\varphi_j^2 = 0.186$, que es el valor relativo porcentual de φ_1^2 (o sea la información de PC(1)) es 18.6%. El valor relativo porcentual de φ_2^2 (la información de PC(2)) es 15.447%, …, y al final, el valor relativo porcentual de φ_{34}^2 (la información de PC(34)) es 1.046×10^{-31} %. Estos valores se representan en la gráfica de la **Figura 3**, que se llama *Screeplot* [9], que muestra los valores en forma de barras.

Los dos componentes principales, PC(1) y PC(2) reúnen un 34% de información (variancia relativa porcentual). Para reunir un 50% de información se requieren los primeros 4 componentes, y para reunir un 90% de información se requieren los primeros 14 componentes.

5.2 Resultados del análisis con los componentes PC (1) y PC (2)

El Biplot bidimensional ($r = 2$) normalmente se realiza con los primeros 2 componentes PC(1)–PC(2), por ser los componentes de más información[10]. La **Figura 4** presenta el Biplot bidimensional con PC(1)–PC(2), con las proyecciones de $m = 34$ vectores (que representan a 34 los atributos), y las proyecciones en dicho plano de $n = 2485$ puntos (que representan a los individuos). Hay que notar que aun cuando los 34 vectores correspondientes a los atributos son ortonormales, sus proyecciones en el Biplot se ven oblicuas entre sí, y de tamaños diferentes. Los identificadores de algunas proyecciones de vector (atributo), que tienen similar dirección y sentido se

[9] La forma de la gráfica se parece a la forma de las piedras que se amontonan al lado de una montaña, que van bajando de altura conforme la distancia, que se llaman, en conjunto, *Scree* (del inglés).

[10] Dado que estos componentes no reúnen un porcentaje de información que llegue al 50%, se podría generar desconfianza sobre la validez representativa de los resultados.

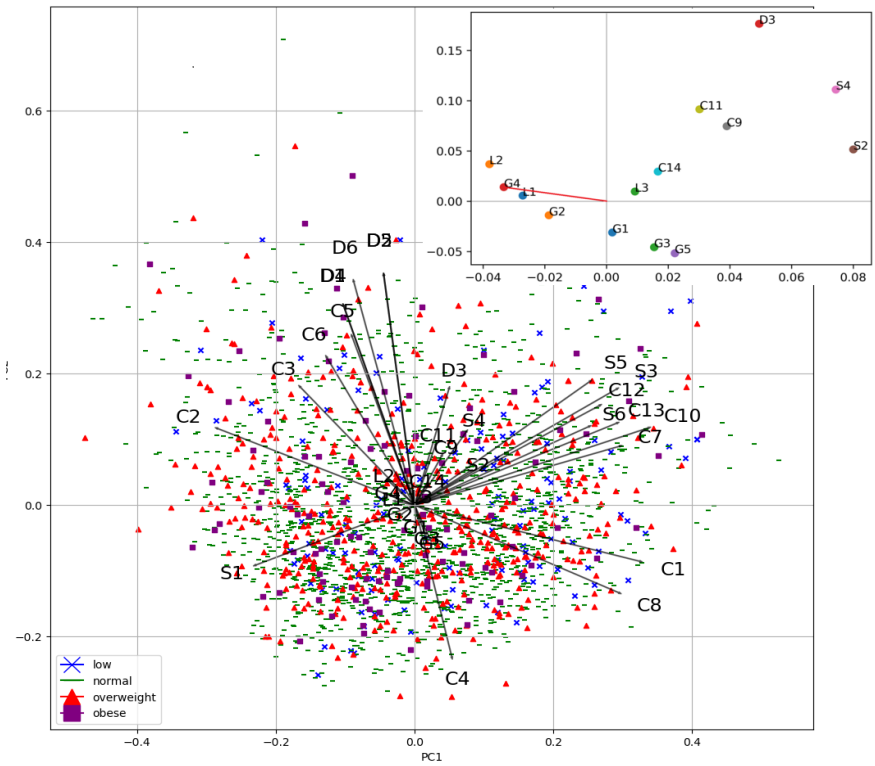


Fig. 4. Biplot bidimensional con PC(1)–PC(2)). Se observan las proyecciones de los 34 vectores que representan a los atributos, y de los 2485 puntos, que representan a los individuos. El símbolo de las proyecciones depende de su clasificación según el atributo G4 (IMC). La simbología es: Bajo peso × ; Peso normal – ; Sobrepeso ▲ ; Obesidad ■ .

enciman: éste es el caso de los identificadores C7, C10, S3, S5, S6. Otros identificadores no se distinguen por estar muy aglomerados muy cercanos al centro del Biplot, como lo son L1, L2, L3, G1, G2, G3, G4, G5 y C14. Con relación a las proyecciones de los puntos, (individuos), se observa que hay mayor dispersión a lo largo del componente PC(1), que a lo largo del componente PC(2).

La **Figura 4** enfoca las proyecciones de los vectores más centrales (más pequeños), entre los que se encuentra G4 (IMC), cuya proyección es relativamente pequeña, lo que indica que este vector **es en gran medida perpendicular al plano PC(1)–PC(2)**. De aquí surgió la pregunta si este plano es significativo para observar lo que afecta a este atributo, aun cuando los componentes PC(1) y PC(2) son aquellos con más información.

Se dibujaron (marcaron) los puntos según los valores del atributo G4 (IMC), y no se observó que las proyecciones de los puntos en el Biplot se separaran según el valor de este atributo.

Las proyecciones, por separado, de los puntos (individuos) según G4 (IMC) en el Biplot bidimensional con PC(1)–PC(2)) se muestran la **Figura 5**.

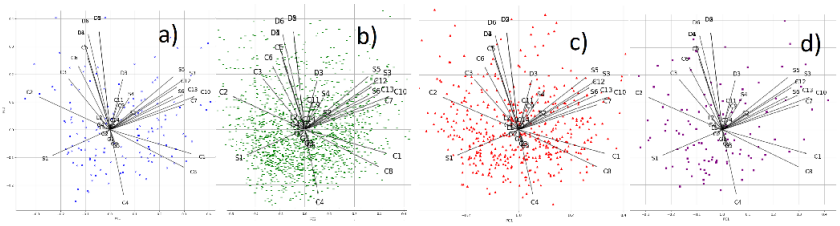


Fig. 5. Cuatro imágenes de Biplot bidimensional con PC(1)–PC(2)), con las proyecciones de los 34 vectores (atributos) en cada imagen. Cada imagen contiene, además, las proyecciones de los puntos (individuos) clasificados según el atributo G4 (IMC). Las imágenes corresponden a: a) Bajo peso × ; b) Peso normal – ; c) Sobrepeso ▲ ; d) Obesidad ■ .

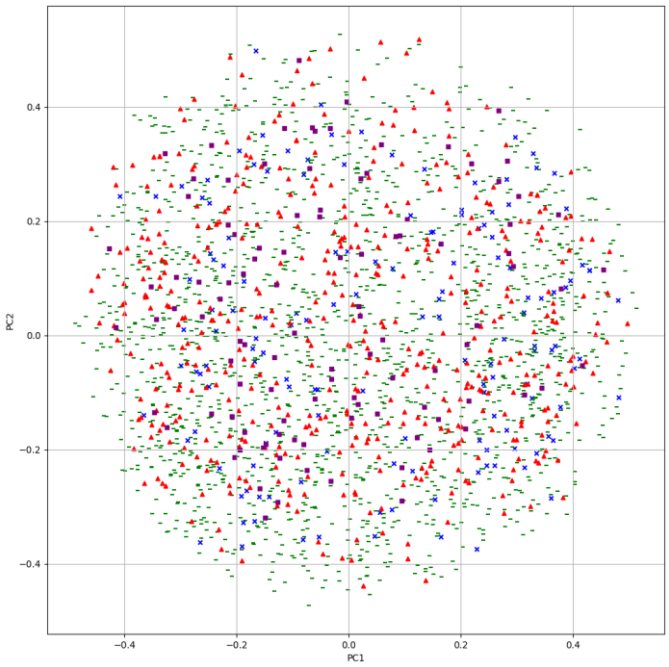


Fig. 6. Biplot bidimensional con PC(1)–PC(2)), donde se observan las proyecciones de los 2485 puntos (que representan a los individuos), transformadas con el Kernel PCA Coseno. El símbolo de las proyecciones depende de su clasificación según el atributo G4 (IMC). La simbología es: Bajo peso × ; Peso normal – ; Sobrepeso ▲ ; Obesidad ■ . No se observa separación de las proyecciones según el valor del atributo G4 (IMC).

Para buscar en el Biplot que las proyecciones de los puntos se separaran según el valor del atributo G4 (IMC), se usaron diversos Kernel PCA, que generalizan el análisis de PCA a dominios no-lineales. Los Kernels fueron: 1) Coseno, 2) Linear, 3) RBF, 4) Sigmoidal y 5) Polinomial, pero en ningún caso se logró obtener la separación buscada. En la Figura 6 se observa, como ejemplo, el Biplot referente al Kernel Coseno.

Tabla 1. Lista parcial de resultados del producto punto, de la proyección del vector del atributo G4 (IMC), con la proyección del vector de cada uno de los demás atributos, en el plano PC(1)– PC(2).

Sigla	Prod. punto	Nombre de atributo
C2	0.01108	Tiempo entre cena y punto medio de sueño. (ideal más de 6 horas).
C3	0.00800	Tiempo entre hora de cena y hora de dormir.
…	…	…
L3	-0.000169	¿Si te despiertas por la noche prendes la luz? "No"=0; "Otro"=1.
C9	-0.000244	ABS(*Jet Lag* de comida principal).
…	…	…
C10	-0.009552	ABS(*Jet Lag* de desayuno).
C8	-0.011672	Promedio ventanas alimentación entre semana y fin de semana.
C1	-0.012005	Ventana de alimentación.

Aún sin haber logrado observar alguna separación de las proyecciones de los puntos (individuos), con los componentes PC(1) y PC(2), según los valores del atributo G4 (IMC), en el Biplot, aquí se comparan las proyecciones de los vectores correspondientes a los atributos, para obtener un resultado de la relación de los atributos con relación al atributo G4 (IMC).

Para comparar la colinealidad y sentido de la proyección, en el plano PC(1)–PC(2), del vector correspondiente al atributo G4 (IMC), con la de los demás atributos, realizamos el producto punto de las proyecciones. La **Tabla 1** presenta una lista parcial de los resultados, tomados en orden descendente.

De acuerdo con la **Tabla 1**, podemos concluir que los atributos más influyentes en G4 son C2 y C3, los no relacionados son L3 y C9, y los más contrapuestos son C1, C8, C10.

5.3 Búsqueda de los componentes más significativos para G4 (IMC)

Ya que la proyección del vector correspondiente al atributo G4 (IMC) sobre el plano PC(1)–PC(2) fue más perpendicular a ese plano que la mayoría de los vectores de los otros atributos, buscamos otro plano en donde el vector correspondiente al atributo G4 (IMC) se proyecte mayormente.

Para esto, observamos la matriz $Q^T_{r\times m}$, para el caso de $r = m$ (o sea $Q^T_{m\times m}$), que indica proyecciones. Observamos la cuarta columna de la matriz $Q^T_{r\times m}$, que indica las proyecciones del vector relativo a G4 (IMC) sobre cada uno de componentes a PC(1), …, P(34). Algunas proyecciones relevantes se ven en la **Tabla 2**, que contiene, para cada PC(i), su información (variancia relativa porcentual). La suma de los cuadrados de las 34 proyecciones es igual a 1.

Encontramos que donde se tiene la mayor proyección es en PC(26), de 0.5707, pero este eje coordenado tiene una variancia de $\varphi^2_{26}/n = 0.00398$, para una variancia relativa porcentual de 0.0117%, que es un valor de información "demasiado" pequeño, con relación a lo que tienen otros PC(i). Encontramos una mayor variancia relativa en PC(8) y PC(4), con valores de proyección, respectivamente, de -0.4823 y de 0.3668,

Tabla 2. Proyección del vector relativo al atributo G4 (IMC), sobre algunos de los 34 PC(i), seleccionados. Se ordenan los renglones por valor del indicador que se propone en este trabajo. $\mathrm{G4}_{\mathrm{PC}(i)}$es la proyección.

Consecutivo	Variancia Relativa (%) $100 \times \varphi_i^2 / \sum_{j=1}^{m} \varphi_j^2 = 100 \times \varphi_i^2/m$	Indicador(i)	PC(i)	$\mathbf{G4_{PC(i)}}$
1	4.2091%	5.44E-01	PC(8)	-0.482330806
2	6.9690%	4.89E-01	PC(4)	0.366788251
3	3.8271%	3.68E-01	PC(9)	-0.337078093
11	0.0117%	9.05E-02	PC(26)	0.570684596
12	18.5577%	6.15E-02	PC(1)	-0.033285789
15				
16	15.4472%	2.45E-02	PC(2)	0.0140993
32	0.0000%	1.43E-26	PC(32)	-4.16439E-16
33	0.0000%	9.49E-27	PC(33)	-2.78152E-16
34	0.0000%	4.90E-27	PC(34)	-1.48889E-16

que tienen una variancia relativa, respectivamente, de 4.2091% y de 6.9690%. Por lo anterior, para establecer un balance entre proyección y variancia relativa, proponemos un indicador con la expresión 11, donde $\left|\mathrm{G4}_{\mathrm{PC}(i)}\right|$ significa la magnitud de la proyección del vector correspondiente al atributo G4 (IMC) sobre PC(i). La razón de aplicar raíz cúbica al valor φ_i^2/n es disminuir su relevancia por ser muy pequeño. Según este indicador, los valores mayores fueron para PC(8) y PC(4), respectivamente. Los renglones de la **Tabla 2** se ordenan según el valor de este indicador:

$$Indicador(i) = \sqrt[3]{\varphi_i^2/n} \times \left|\mathrm{G4}_{\mathrm{PC}(i)}\right|. \tag{11}$$

Aun cuando los ejes coordenados PC(4) y PC(8) tienen una cantidad de información en conjunto de 11.18%, ésta es una información que se sospecha ser más significativa debido a la mayor proyección del vector correspondiente al atributo G4 (IMC) en el plano PC(4)-PC(8).

La **Figura 7** presenta el Biplot bidimensional con PC(4)–PC(8). **Es sorprendente** observar que en este Biplot sí se distinguen, "a simple vista", zonas de localización para las proyecciones de los puntos por cada categoría de IMC (bajo peso, peso normal, sobrepeso y obesidad).

Las proyecciones de los puntos de bajo peso se concentran lejanas del centro y arriba a la izquierda, las de peso normal están centradas arriba a la izquierda, las de de sobrepeso están centradas abajo a la derecha, y las de obesidad están lejanas del centro y abajo a la derecha. Se observa que la proyección del vector relativo al atributo G4 (IMC) apunta hacia las zonas donde están las proyecciones de los puntos de sobrepeso, y de obesidad.

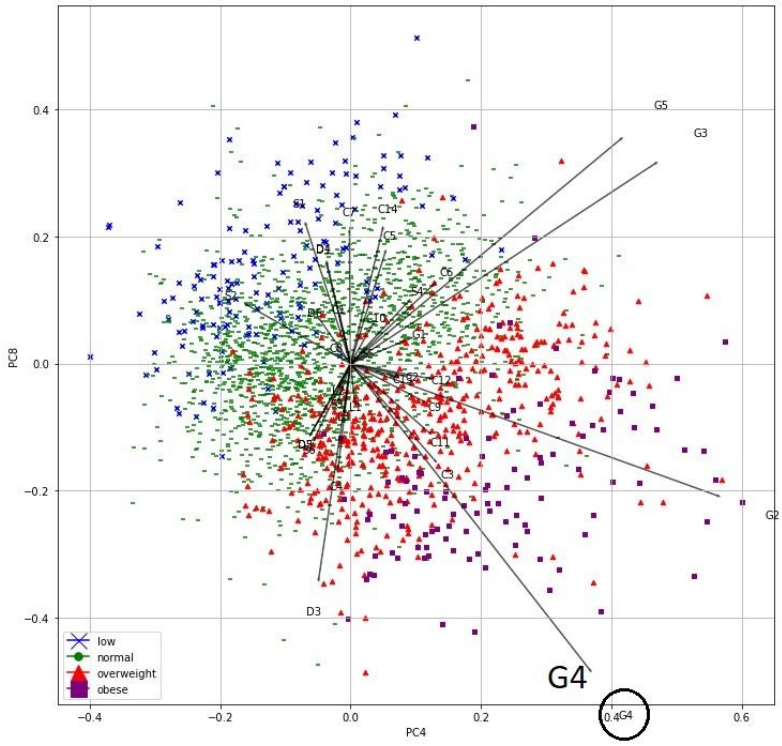


Fig. 7. Biplot bidimensional con PC(4) y PC(8). Se observa la proyección de los 34 vectores que representan a los atributos, y de los 2485 puntos que representan a los individuos, clasificados según el atributo G4 (IMC). Para las proyecciones de los puntos (individuos) la simbología es: Bajo peso × ; Peso normal ● ; Sobrepeso ▲ ; Obesidad ■ .

Realizamos la operación producto punto de la proyección, en el plano PC(4)–PC(8), del vector correspondiente al atributo G4 (IMC) por cada una de las proyecciones de los vectores correspondientes a los demás atributos. La **Tabla 3** presenta una lista parcial de resultados de esta operación, que no incluye el resultado relativo al atributo G4 (IMC), que tiene el mayor valor: 0.3671.

De acuerdo con la **Tabla 3**, podemos concluir que los atributos más influyentes en G4 (IMC) son G2, D3, C3, los no relacionados son L2, S5, C6, y los más contrapuestos son C7, S2 y S1.

Observamos que los resultados de proyectar en el plano PC(1)–PC(2) son diferentes a aquellos de proyectar en el plano PC(4)–PC(8).

6. Conclusiones y trabajo futuro

La descomposición por componentes principales podría no arrojar resultados útiles tan solo seleccionando a los componentes de más información, pero puede sí puede

Tabla 3. Lista parcial de resultados de la operación producto punto de la proyección del vector correspondiente al atributo G4 (IMC), en el plano PC(4)–PC(8), con la proyección de cada uno de los vectores correspondientes a los demás atributos.

Sigla	Prod. punto	Nombre de atributo
G2	0.306754286	Peso Kg.
D3	0.145614712	Diferencia, en cama, antes dormir, uso celular o disp. Elec (entre semana a fin de semana), minutos.
C3	0.120905029	Tiempo entre hora de cena y hora de dormir (entre semana).
C11	0.096623535	ABS(*Jet Lag* cena).
…	…	…
L2	0.011971867	Luz al dormir: "'Obscuro"=0; "Tenue"=1; "Luz"=2; "Brillante"=3.
S5	0.00201177	ABS(*Jet Lag* punto medio sueño).
C6	-0.013398287	Tiempo entre hora de cena y hora de dormir (fin de semana).
…	…	…
C7	-0.10035604	Diferencia ventanas alimentación (entre semana vs fin de semana).
S2	-0.102823015	Horas dormido (fin de semana).
S1	-0.130563624	Horas dormido (entre semana).

hacerlo al seleccionar componentes principales más enfocados al atributo que sea de interés. Esto pasó en este trabajo al seleccionar los componentes principales PC(4) y PC(8), que reunían apenas poco más del 11% de la información, en comparación de seleccionar los a los componentes principales con la mayor información, PC(1) y PC(2), que reunían el 34%, de la información. La utilidad se apreció en el Biplot bidimensional con PC(4)–PC(8), donde las proyecciones de los puntos (que representan a los individuos) ocuparon zonas según el atributo de interés, que indica el sobre peso y la obesidad, G4 (IMC).

Esta ubicación por zonas no se observó al trabajar con un Biplot bidimensional con PC(1) y PC(2). La clave para seleccionar a PC(4) y PC(8) fue el balance entre la proyección del vector asociado al atributo G4 (IMC), con la información de esa proyección. Para esto, en este trabajo se propuso un indicador que hiciera ese balance.

En próximos trabajos se puede realizar un estudio en un Biplot de tres dimensiones. También, se puede seguir analizando la utilidad del identificador que hemos presentado en la ecuación 11.

Según los autores del grupo de estudio de la ciencia de medicina, los resultados del análisis en las dos proyecciones coinciden con observaciones de estudios clínicos que indican que diversos hábitos como factores de riesgo para la salud promueven el desarrollo de sobrepeso y obesidad [36]. El atributo C3 (Tiempo entre hora de cena y hora de dormir entre semana) sale como reforzante en el análisis de las dos proyecciones, lo cual ubica como un factor más relevante entre los demás analizados.

Es importante considerar también que la disrupción circadiana no necesariamente incide en ganar peso; en algunos individuos puede incidir en perderlo. Esta situación puede confundir. Por esto, es importante perfeccionar las consideraciones del estudio.

Apéndice. Lista de atributos de los datos

Cada atributo está numerado y entre paréntesis identificado por una letra que puede contener una ecuación si el atributo se deriva de otros. Si hay otro paréntesis es que el atributo se seleccionó para análisis: el paréntesis contiene el número de atributo seleccionado y una sigla de identificación de dos letras. La letra de inicio de la sigla indica el tipo de atributo: G → información general; S → sueño (horas de dormir); C → consumo (alimentación); L → luz (iluminación).

- **Atributos de identificación.** 1- Fecha (A); 2- Carrera (B); 3- Matricula (C) (ID);
- **Atributos de información general.** 4- Edad años (D) (1- G1); 5- Peso Kg (E) (2- G2); 6- Estatura metros (F) (3- G3); 7- IMC ($G = E/F^2$) (4- G4); 8- Sexo: Fem.=1, Masc.=2 (H) (5- G5);
- **Atributos marcadores de disrupción circadiana (sueño entre semana).** 9- Me despierto a las: (I); 10- Me duermo a las: (J); 11- Horas despierto ($K = J - I$); 12- Horas dormido ($L = 24 - K$) (6- S1); 13- Punto medio de sueño ($M = J + L/2$); 14- Punto medio de sueño, horas de la mañana ($N = M - 24$);
- **Atributos marcadores de disrupción circadiana (sueño fin de semana).** 15- Me despierto a las: (O); 16- Me duermo a las: (P); 17- Horas despierto ($Q = P - O$); 18- Horas dormido ($R = 24 - Q$) (7- S2); 19- Punto medio de sueño ($S = P + R/2$); 20- Punto medio de sueño horas de la mañana ($T = S - 24$);
- **Atributos marcadores de disrupción circadiana (sueño en general).** 21- Jet Lag hora de despertar ($U = O - I$); 22- ABS(Jet Lag hora de despertar) ($V = ABS(U)$) (8- S3); 23- Jet Lag hora de dormir ($W = P - J$); 24- ABS(Jet Lag hora de dormir) ($X = ABS(W)$) (9- S4); 25- Jet Lag punto medio de sueño ($Y = S - M$); 26- ABS(Jet Lag punto medio de sueño) ($Z = ABS(Y)$) (10- S5); 27- Diferencia de horas dormido ($AA = R - L$) (11- S6);
- **Atributos marcadores de disrupción circadiana (consumo entre semana).** 28- Desayuno a las: (AB); 29- Como a las: (AC); 30- Ceno a las: (AD); 31- Ventana de alimentación ($AE = AD - AB$) (12- C1); 32- Punto medio de alimentación ($AF = AB + AE/2$); 33- Tiempo entre cena y punto medio de sueño (ideal más de 6 horas) ($AG = M - AD$) (13- C2); 34- Tiempo entre hora de cena y hora de dormir ($AH = J - AD$) (14- C3);
- **Atributos marcadores de disrupción circadiana (consumo fin de semana).** 35- Desayuno a las: (AI); 36- Como a las: (AJ); 37- Ceno a las: (AK); 38- Ventana de alimentación ($AL = AK - AI$) (15- C4); 39- Punto medio de alimentación ($AM = AI + AL/2$); 40- Tiempo entre cena y punto medio de sueño (ideal más de 6 horas) ($AN = S - AK$) (16- C5); 41- Tiempo entre hora de cena y hora de dormir ($AO = P - AK$) (17- C6);
- **Atributos marcadores de disrupción circadiana (consumo en general).** 42- Diferencia ventanas alimentación ($AP = AE - AL$) (18- C7); 43- Promedio

ventanas alimentación ($AQ = (AE \times 5 + AL \times 2)/7$) (19- C8); 44- Jet Lag de comida principal ($AR = AJ - AC$); 45- ABS(Jet Lag de comida principal) ($AS = ABS(AR)$) (20- C9); 46- Jet Lag de desayuno ($AT = AI - AB$); 47- ABS(Jet Lag de desayuno) ($AU = ABS(AT)$) (21- C10); 48- Jet Lag cena ($AV = AK - AD$); 49- ABS(Jet Lag cena) ($AW = ABS(AV)$) (22- C11); 50- Promedio de Jet Lag metabólico ($AX = (AS + AU + AW)/3$) (23- C12); 51- Jet Lag metabólico punto medio de alimentación ($AY = AM - AF$); 52- ABS(Jet Lag metabólico punto medio de alimentación) ($AZ = ABS(AY)$) (24- C13); 53- ¿Si te despiertas por la noche comes? (BA); 54- ¿Si te despiertas por la noche comes? (numérico): "Nunca"=0, "A veces"=1 (BB) (25- C14);

- **Atributos marcadores de disrupción circadiana (iluminación en general –solo por las noches).** 55- ¿La luz de calle ilumina tu cuarto? (BC); 56- ¿La luz de calle ilumina tu cuarto? (numérico): "No"=0, "Un poco"=1, "Sí"=2 (BD) (26- L1); 57- Luz al dormir (BE); 58- Luz al dormir (numérico): "'Obscuro"=0, "Tenue"=1, "Luz"=2, "Brillante"=3 (BF) (27- L2); 59- ¿Si te despiertas por la noche prendes la luz? (BG); 60- ¿Si te despiertas por la noche prendes la luz? (numérico): "No"=0, "Otro"=1 (BH) (28- L3); 61- Puntuación total exposición luz ($BI = BD + BF + BH$);
- **Atributos marcadores de disrupción circadiana (iluminación de dispositivos en general).** 62- En cama, antes dormir, uso celular o dispositivo electrónico, minutos (entre semana) (BJ) (29- D1); 63- En cama, antes dormir, uso celular o dispositivo electrónico, minutos (fines semana) (BK) (30- D2); 64- Diferencia, en cama, antes dormir, uso celular o dispositivo electrónico, minutos ($BL = BK - BJ$) (31- D3); 65- En cama, antes dormir, total uso celular o dispositivo electrónico, minutos (entre semana) ($BM = BJ \times 5$) (32- D4); 66- En cama, antes dormir, total uso celular o dispositivo electrónico, minutos (fines semana) ($BN = BK \times 2$) (33- D5); 67- En cama, antes dormir, total uso celular o dispositivo electrónico, minutos ($BO = BM + BN$) (34- D6).

Referencias

1. Shrestha, P., Ghimire, L.: A Review about the Effect of Life Style Modification on Diabetes and Quality of Life. Global Journal of Health Science, vol. 4, no. 6, pp. 185–90 (2012). DOI: 10.5539/gjhs.v4n6p185.
2. Egger, G., Dixon, J.: Beyond Obesity and Lifestyle: A Review of 21st Century Chronic Disease Determinants. BioMed Research International, vol. 2014, pp. 731685 (2014). DOI: 10.1155/2014/731685.
3. Stoner, L., Castro, N., Signal, L., Skidmore, P., Faulkner, J., Lark, S., Williams, M.A., Muller, D., Harrex, H.: Sleep and Adiposity in Preadolescent Children: The Importance of Social Jetlag. Childhood obesity (Print), vol. 14, no. 3, pp. 158–164 (2018). DOI: 10.1089/chi.2017.0272.
4. Gill, S., Panda, S.: A Smartphone App Reveals Erratic Diurnal Eating Patterns in Humans that Can Be Modulated for Health Benefits. Cell metabolism, vol. 22, no. 5, pp. 789–798, (2015). DOI: 10.1016/j.cmet.2015.09.005.
5. Escobar, C., Ángeles-Castellanos, M., Espitia Bautista, E.N., Buijs, R.M.: Food During the Night is a Factor Leading to Obesity. Revista Mexicana de Trastornos Alimentarios, vol. 7, no. 1, pp. 78–83 (2016). DOI: 10.1016/j.rmta.2016.01.001.

6. Touitou, Y., Reinberg, A., Touitou, D.: Association Between Light at Night, Melatonin Secretion, Sleep Deprivation, and the Internal Clock: Health Impacts and Mechanisms of Circadian Disruption. Life Sciences, vol. 173, pp. 94–106 (2017). DOI: 10.1016/j.lfs.2017.02.008.
7. Guerrero-Vargas, N.N., Ángeles-Castellanos, M., Escobar Briones, C.: Los efectos adversos de la luz artificial por la noche. Revista Digial Universitaria, vol. 19, no. 3, pp. 1–18 (2018). DOI: 10.22201/codeic.16076079e.2018.
8. Ayala, G.X., Baquero, B., Klinger, S.: A Systematic Review of the Relationship Between Acculturation and Diet Among Latinos in the United States: Implications for Future Research. Journal of the American Dietetic Association, vol. 108, no. 8, pp. 1330–1344 (2008). DOI: 10.1016/j.jada.2008.05.009.
9. Miller, M.A., Kruisbrink, M., Wallace, J., Ji, C., Cappuccio, F.P.: Sleep Duration and Incidence of Obesity in Infants, Children, and Adolescents: A Systematic Review and Meta-Analysis of Prospective Studies. Sleep, vol. 41, no. 4 (2018). DOI: 10.1093/sleep/zsy018.
10. Roenneberg, T., Allebrandt, K.V., Merrow, M., Vetter, C.: Social jetlag and obesity. Current Biology: CB, vol. 22, no. 10, pp. 939–943 (2012). DOI: 10.1016/j.cub.2012.03.038.
11. Escobar, C., Ángeles-Castellanos, M., Miñana-Solís, M.C., Salgado Delgado, R.: Disturbance of Circadian Rhythms as a Predisposing Factor for Obesity and Metabolic Disease, de Advances in Obesity-diabetes. Research at UNAM, El Manual Moderno, pp. 1–17 (2010)
12. UNAM: Oferta Académica UNAM. Médico Cirujano, UNAM, 2022. [En línea]. Available: http://oferta.unam.mx/medico-cirujano.html. [Último acceso: 21 10 2022].
13. Stewart, G.W.: On the Early History of the Singular Value Decomposition, SIAM Review, vol. 35, p. 551–566 (1993). DOI: 10.1137/1035134.
14. Borg, I., Groenen, P.J.F.: Modern Multidimensional Scaling. Theory and Applications, Ney York: Springer (2005)
15. Jolliffe, I.T., Cadima, J.: Principal Component Analysis: A Review and Recent Developments, Philosophical Transactions of the Royal Society A: Mathematical. Physical and Engineering Sciences, vol. 374, p. 20150202 (2016). DOI: 10.1098/rsta.2015.0202.
16. Shlens, J.: A Tutorial on Principal Component Analysis, CoRR, vol. abs/1404.1100 (2014). DOI: 10.48550/arXiv.1404.1100.
17. Shalizi, C.R.: Advanced Data Analysis from an Elementary Point of View (Principal Component Analysis chapter), Carnegie Mellon University, [on line]. Available: https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ [Último acceso: 18 10 2022].
18. Subu, M.A., Al-Yateem, N., Waluyo, I., Dias, J.M., Rahman, S.A., Agustino, R., Ahamed, I.S., Al Marzooqi, A.: Relationship Between Internet Gaming Addiction and Body Mass Index Status Among Indonesian Junior High School Students. IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) pp. 1391–1393 (2021). DOI: 10.1109/COMPSAC51774.2021.00200.
19. Johnson, K.A., Gordon, C.J., Chapman, J.L., Hoyos, C.M., Marshall, N.S., Miller, C.B., Grunstein, R.R.: The Association of Insomnia Disorder Characterised by Objective Short Sleep Duration with Hypertension, Diabetes and Body Mass Index: A Systematic Review and Meta-analysis. Sleep Medicine Reviews, vol. 59, pp. 101456 (2021). DOI: 10.1016/j.smrv.2021.101456.
20. Croenlein, T., Langguth, B., Busch, V., Rupprecht, R., Wetter, T.: Severe Chronic Insomnia is not Associated with Higher Body Mass Index. Journal of Sleep Research, vol. 24, no. 5, pp. 514–517 (2015). DOI: 10.1111/jsr.12294.

21. Bocicor, A.E., Buicu, G., Sabau, D., Varga, A., Tilea, I., Gabos-Grecu, I.: Association Between Sleep Disorder and Increased Body Mass Index in Adult Patients. Acta Marisiensis - Seria Medica, vol. 62, p. 221–224 (2016). DOI: 10.1515/amma-2016-0015.
22. Duranková, S., Csanády, A., Ždiľová, A., Bernasovská, J., Buková, A.: Effects of lifestyle on Physical Health in Slovak University Students. Anthropological Review, vol. 83, pp. 129–142 (2020). DOI: 10.2478/anre-2020-0010.
23. Imran, M., Khan, N., Shah, A.A., Ahmad, I.: Overweight and Obesity Prevalence Pattern and Associated Risk Factors Among the People of Malakand Division, Khyber Pakhtunkhwa Pakistan. Arabian Journal for Science and Engineering, vol. 44 (2018)
24. Zhang, Y., Li, W.: A Survey of The Relationship Between Human Faces And Body Mass Index (BMI), IEEE Globecom Workshops, pp. 1–6 (2021). DOI: 10.1109/GCW kshps52748.2021.9682060.
25. Mark, G.M., Bakunzibake, P., Mikeka, C.: Design of an IoT-based Body Mass Index (BMI) Prediction Model. 4th International Seminar on Research of Information Technology and Intelligent Systems, pp. 629–634 (2021). DOI: 10.1109/ISRITI54043.2021.9702866.
26. Leon, S.J.: Linear Algebra with Applications, 1 éd., New York: Macmillan Publishing Co., p. 338 (1980)
27. Gabriel, K.R.: The Biplot Graphic Display of Matrices with Application to Principal Component Analysis, Bimoetriks, vol. 58, no. 3, pp. 453–467 (1971)
28. Greenacre, M.J.: Biplots: the Joy of Singular Value Decomposition, Wiley Interdisciplinary Reviews: Computational Statistics, vol. 4, no. 4, pp. 399–406 (2012). DOI: 10.1002/wics.1200
29. Bedre, R.: Principal Component Analysis (PCA) and Visualization Using Python (Detailed Guide with Example), Creative Commons (https://creativecommons.org/), 07 11 2021. [on line]. Available: https://www.reneshbedre.com/blog/principal-component-analysis.html. [Último acceso: 07 07 2022].
30. Greenacre, M.: Biplots in Practice, B. Foundation, Ed., Bilbao: Spain (2010)
31. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array Programming with NumPy, Nature, vol. 585, pp. 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
32. McKinney, W. et al.: Data Structures for Statistical Computing in Python. In: Proceedings of the 9th Python in Science Conference (2010). DOI: 10.25080/Majora-92bf1922-00a.
33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, vol. 12, pp. 2825–2830 (2011)
34. Waskom, M.L.: Seaborn: Statistical Data visualization, Journal of Open Source Software, vol. 6, pp. 3021 (2021). DOI: 10.21105/joss.03021.
35. Hunter, J.D.: Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, vol. 9, p. 90–95 (2007). DOI: 10.1109/MCSE.2007.55.
36. McHill, A.W., Wright, K.P.J.: Role of Sleep and Circadian Disruption on Energy Expenditure and in Metabolic Predisposition to Human Obesity and Metabolic Disease, Obesity Reviews: An Official Journal of the International Association for the Study of Obesity, vol. 18, 1, pp. 15–24 (2017). DOI: 10.1111/obr.12503.

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación