EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 150 No. 8
August 2021

Research in Computing Science

Research in Computing Science, Año 20, Volumen 150, No. 8, agosto de 2021, 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 agosto de 2021.

Research in Computing Science, year 20, Volume 150, No. 8, August 2021, is published monthly by the Center for Computing Research of IPN.

Volume 150(8)

Geographic Information Systems Latin America (GIS LATAM)

Roberto Eswart Zagal Flores
Abraham Efraim Rodriguez Mata
Jose Antonio León Borges
Miguel Félix Mata Rivera (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2021

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

Convolutional Neural Networks Comparison in Covid Incidence Detection

Abel Robles Montoya[1], Alec Guerrero Gallardo[1], Jaqueline P.[1], David Alejandro T.[1], Victor G.[1], Volodymyr Ponomaryov[2]

[1] Tecnológico Nacional de México, campus Culiacán, Mexico

[2] Instituto Politecnico Nacional, Mexico

alx3416@hotmail.com

Abstract. This paper aims to compare the different state-of-the-art neural network performances in binary detection of the novel covid-19. We will be using Xception, VGG16, MobileNetV3, EficientNetB0, and NasNetMobile to make our comparison, working with approximately 4,000 Ches X-ray (CXR) images as our dataset. We obtain the quality measures (Accuracy, Precision, Recall, F1-score), confusion matrix, the receiver operating characteristic curve (curve), and saliency maps to observe the legitimacy of each network output. Xception gives us the best performance (at the expense of having the highest parameter volume) with a general accuracy of 0.97, a recall of 0.97 on the positive class, a F1-score of 0.97, and Area Under the Curve (AUC) of 0.995 on the ROC curve. EfficientNetB0 obtain a marginal 0.01 difference with respect to Xception in both recall and F1-score archiving a general accuracy of 0.97, a recall of 0.96 on the positive class, a F1-score of 0.96, and an AUC of 0.996, with the huge benefit of having about 17.3 millions fewer parameters. We present a comparison with previous works, where we can observe that our best model obtain a relatively reliable performance taking into consideration the bigger number of CXRs compared to the rest of the researches. Finally, we conclude with a discussion and future related work.

Keywords: Transfer learning, machine learning, deep learning, COVID-19.

1 Introduction

The appearance of the new coronavirus disease (covid-19) that was detected for the first time in Wuhan, China on December 31, 2019, and that has caused thousands of infections around the world have incentivizes the use of multiple new technologies in order to detect news incidences. To carry out the detection of covid-19, different techniques have been implemented such as nucleic acid

detection (poly-merase chain reaction or PCR)[14], antigen detection and antibody detection. The most widely used novel coronavirus (COVID-19) detection technique is real time PCR. However, considering its high cost and confirmation time, the rapid spread of the disease, the increment of demand for tests and the fact that the less sensitivity version of real time PCR provides high false-negative results, have caused other alternatives to be sought to complement the detection of covid 19. To resolve this problem, radiological imaging techniques such as chest X-rays (CXR), computed tomography (CT) [13] and the implementation of artificial intelligence has been used not only for detection but also to study and understand the behavior of this disease.

Deep learning is successfully used as a tool for machine learning, where a neural network is capable of automatically discern features. This makes a big contrast versus traditionally hand-crafted features. Among deep learning techniques, deep convolutional networks are actively used for medical image analysis. This includes applications such as segmentation, abnormality detection, disease classification, computer-aided diagnosis, and retrieval[4]. These techniques have been used to address the problem of discriminating benign cysts from malignant masses in breast ultrasound (BUS) images based [34] and for covid-19 detection using CXR [8].

This project aims to implement different Convolutional Neural Network (CNN) architectures to detect covid-19 using CXR images and to make a comparison between them to know which of them is better for this type of data and highly reduce the number of false negatives. For this proposes we implement different pre-trained models: MobileNet V3, EfficientNetB0, NasNet-Mobile, Xception, and VGG16. Previous studies have implemented models of deep learning through transfer learning to detect covid-19 such as in [17], where five pre-trained CNN-based models were proposed for detection of coronavirus pneumonia-infected patients using CXR radiographs. Three different binary classifications with four classes (COVID-19, normal (healthy), viral pneumonia, and bacterial pneumonia) were implemented by using five-fold cross-validation. Considering the performance results obtained, it has been seen that the pre-trained ResNet50 model provides the highest classification performance. Research [2] presented a novel method to improve the performance of transfer learning when the target domain data is not only scarce, but it also has a slightly different modality. This technique uses an ensemble of deep learning models that are modified and hierarchically fine-tuned to the target domain. They tested their technique by using pre-trained models of natural images (ImageNet) and transferring them to the domain of chest radiography images. The study [28] aimed at evaluating the effectiveness of the state-of-the-art pre-trained CNNs on the automatic diagnosis of COVID-19 from CXRs. The results showed that deep learning with X-ray imaging is useful in collecting critical biological markers associated with COVID-19 infections. In [6], the performance of state-of-the-art CNNs architectures proposed over the recent years for medical image classification was evaluated. The results suggest that Deep Learning with X-ray imaging may extract significant biomarkers related

to the Covid-19 disease. Another work where transfer learning was used to detect covid 19 via CXR was in project[13] by using the extreme version of the Inception (Xception) model. Extensive comparative analyses show that the proposed model performs significantly better as compared to the existing models. And also [19] proposed a model that depends on the working of deep learning-based CNN known as nCOVnet. it can classify the COVID-19 patient correctly with 97.97 % confidence.

In [5], the-state-of-the-art CNN called MobileNet was employed and trained from scratch to investigate the importance of the extracted features for the classification task. The results suggest that training CNNs from scratch may reveal vital biomarkers related but not limited to the COVID-19 disease, while the top classification accuracy suggests further examination of the X-ray imaging potential. In study[24] the deep learning based methodology is suggested for detection of coronavius-infected patient using X-ray images. In [29] the proposed approach was designed to provide fast and accurate diagnostics for COVID-19 diseases with binary classification (COVID-19, and No-Findings) and multi-class classification (COVID-19, and No-Findings, and Pneumonia). The proposed method achieved an accuracy of 97.24%, and 84.22% for binary class, and multi-class, respectively. In the study [31], developed COVID-Net a deep CNN was designed tailored for the detection of COVID-19 cases from chest CXR images that is open source and available to the general public. And also, deep learning framework have been developed; namely COVIDX-Net to assist radiologists to automatically diagnose COVID-19 in X-ray images[15].

However, our work differs from some of the studies. We use a dataset with 2000 positive images and 2000 negative images, we don't use data augmentation, and our models are binary classifiers. Our research will help us to know more information about CNN pre-training models, which of these architectures give us more precise results, and which of them are more efficient according to the parameters it needs and its performance to solve these kinds of problems.

This paper is organized as follows. In section 2, we describe the state-of-the-art, previous studies that are related to this work. In section 3, the methodology that we follow to carry out this project. The results and the comparison of the quality measures of the different models are described in section 4. And finally, in section 5, we present the conclusions.

2 State of the Art

Previously comparisons have been made for different image classification techniques, and on some occasions, results obtained by different network architectures were directly compared, [30] develop a research project where InceptionV3, VGG16, ResNet50, and Xception networks were evaluated, which although they are not exactly the networks that will be compared below if data were obtained that helped as a starting point for this research. [30] classifies objects from the Cifar10 dataset using deep learning and implemented in an embedded device such as the Nvidia Jetson TX2, to compare the TensorFlow, Caffe, and Pytorch

frameworks for deep learning. This research compares the aforementioned architectures, concluding that the Xception network had a higher precision (93.3%) than Inception V3 network (85.8 %), similar to VGG16 (91.9%), but lower than ResNet50 (98.6%) [30].

In 2018, [34] made a comparison between some models, and consist in evaluate three transferred models InceptionV3, ResNet50, and Xception, a CNN model with three convolutional layers (CNN3), and traditional machine learning-based model with hand-crafted features were developed for differentiating benign and malignant tumors from breast ultrasound data. The transfer learning model with the pre-trained InceptionV3 network scored the best performance with a maximum accuracy of 85.13%. The accuracy obtained with the ResNet50 and Xception models was slightly lower, 84.94% and 84.06%, respectively. The authors concluded that the transferred InceptionV3 model achieved the best accuracy among the models compared.

In 2020, [3] proposed a more sophisticated transfer learning that not only systematically modifies the original network architecture, but also hierarchically refines it on augmented target domain data. In addition, they use a set of modified networks to make a prediction. They established the effectiveness of their technique with the ChestXray-14 radiography data set.

Their experimental results showed more than 50% reduction in the error rate using their method as compared to the baseline transfer learning technique. After that, they applied their technique to a COVID-19 data set for binary and multi-class classification tasks. Their technique achieved 99.49% accuracy for the binary classification, and 99.24% for multi-class classification.

With COVID-19 dataset, they compared four models, DenseNet201, ResNet50, Inception-V3 and VGG-16, and an ensemble of them. The accuracy mentioned above was from the ensemble of models.

Following a brief description of the principal networks we are using to conduct our experiment.

The NasNetMobile model is a mobile version of NasNet search which is a scalable CNN architecture consisting of blocks optimized for reinforced learning[16], this mobile version consists of 12 blocks with 5.3 million parameters[23], in this work we will use this version of NasNet with the weights obtained when training with the ImageNet dataset. MobileNet V3 it is the new entry in the MobileNet series and it is also based on network architecture search (NAS). Implementing complimentary search techniques and a novel architecture, this network promises to give the best results yet [16].

MobileNetV3 has two versions, which target are different resource use cases. In this work, we will use the MobileNetV3-Large version. Xception architecture is entirely based on depthwise separable convolution layers with residual connections, which is an extreme interpretation of the Inception model, its structure makes it easy to define and modify, and in some comparations performed with the JFT and ImageNet dataset, Xception shows a better performance in classification.[9] Despite this architecture not represent a big change from which

is based, their characteristics make it a good option to compare with other state of art architectures for image classification.

In 2019, [27] proposed a new way of thinking in model Scaling for CNNs introducing a novel method that uniformly scales all dimensions of depth/width/resolution(DWR). One of the observations the authors make state that: *Scaling up any dimension of network width, depth, or resolution improves accuracy, but the accuracy gain diminishes for bigger models.* This lead the to think that, in order to maximize accuracy and efficiency, it crucial to balance the DWR of the network. The result was the family of networks knows as EfficientNets, which score a state-of-the-art 84.3% top-1 accuracy on ImageNet on the 7th version while being smaller and faster than most convolutional networks. Finally VGG16 model was proposed to investigate the effects of the convolutional network depth and the performance on the localization and classification tasks. The authors Karen Simonyan and Andrew Zisserman presented this architecture in [22] in 2014, winning first and second place in the localization and classification categories, which proves to be an excellent vision model architecture [26].

Briefly, we use 5 models: MobileNetV3 Large, EfficientNetB0, Xception, NasNetMobile and VGG16. We used these models because the libraries that implement them run natively in Google Colaboratory, those libraries are TensorFlow and Keras. Despite having mentioned the ResNet50 network and seeing that it has been one of the best performing networks, we decided not to use that model, because we realized that the ResNet50 network has already been extensively tested against other networks, and we wanted to test the performance of networks that were not as widely used and that were already implemented in Google Colaboratory.

3 Metodology

Now, we will describe how we select the data, the general process and how our neural networks were set up. In this project, we work with pre-trained models and we modify some configurations of the different architectures until adapting them with the data which we decide to work, some parameters were modified to get better performance, such as the optimization function, the layers of the neural networks, the activation function, the loss function, and some other parameters that will be shown below.

3.1 Dataset

This dataset was developed by Lida Wang, Zhong Qiu Lin, and Alexander Wong, and its name is COVIDx, it is comprised of a total of 13,975 CXR images across 13,870 patient cases. And it was generated combining and modifying five different publicly available data repositories[32, 12, 11, 10, 21, 18, 33]: 1) COVID-19 Image Data Collection. 2) COVID-19 Chest X-ray Dataset Initiative. 3) ActualMed COVID-19 Chest X-ray Dataset Initiative, established in collaboration with ActualMed. 4) RSNA Pneumonia Detection Challenge dataset (which used publicly

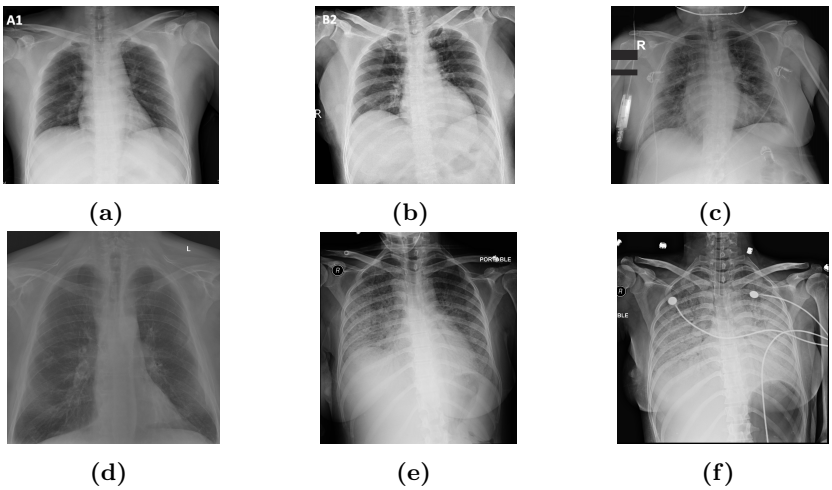


Fig. 1. Sample dataset images labeled with negative results of covid-19 with, a), b) and c) are positive, d), e) and f) are negative.

available CXR data from hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases). 5) COVID-19 radiography database.

The choice of these five datasets from which to create COVIDx, the authors says, is guided by the fact that all five of the datasets are open source and fully accessible to the research community and the general public, and as datasets grow they will continue to grow COVIDx accordingly. More specifically, Wang et. al. combined and modified the five data repositories to create the COVIDx dataset by leveraging the following types of patient cases from each of the data repositories [32, 12, 11, 10, 21, 18, 33]: • Non-COVID19 pneumonia patient cases and COVID-19 patient cases from the COVID-19 Image Data Collection. • COVID-19 patient cases from the COVID-19 Chest X-ray Dataset Initiative. • COVID-19 patient cases from the ActualMed COVID-19 Chest X-ray Dataset Initiative. • Patient cases who have no pneumonia (i.e., normal) and non-COVID19 pneumonia patient cases from RSNA Pneumonia Detection Challenge dataset. • COVID-19 patient cases from COVID-19 radiography database.

We used only 4000 images from COVIDx dataset because of hardware limitations, moreover we only had 2000 negative case images in the chosen dataset, so we took the same number of positive case images to make it balanced. The input images were of 224 pixels per side and RGB, so their dimensions were 224x224x3.

Below, in figure 1, is a sample of what images labeled with positive and negative results look like in the dataset.

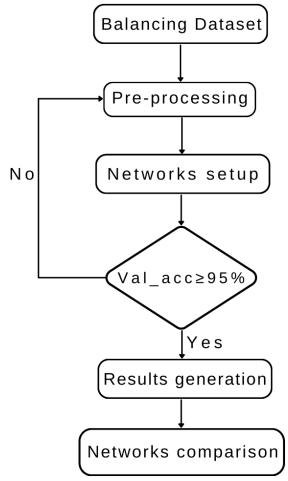


Fig. 2. General process of the experiment.

Next, in figure 2, we present the general flow we used to conduct this experiment. As previously mentioned, we balance the data set in two sets of approximately 2000 samples for class, then we pre-process it as the network requires it (0-1 or 0-255 or No normalization), and set up the network,(this will be described in much more detail in the next section). Continuing, we train the networks making use of callbacks for monitoring the metric valLoss, to change the learning rate or early stop the training if no substantial learning was archive in the few last epochs. We set a threshold of ValAccuracy $\geq 95\%$ in order to proceed to the next step in the process, if it did not fulfill the condition, we return to the image pre-processing, searching for new ways to represent the information or we re-tune the network. When such threshold was cover, we proceed to estimate our quality measures, confusion matrix and ROC curve for further comparison.

3.2 Setting up the networks

`MobileNet V3 Large:` We used the large version of `MobileNet V3` and we configured this model with the initial weights from ImageNet and after the base model, we added the layers, first we added a `GlobalAveragePooling2D` layer, after that a `Dropout` layer of 50%, then a Flatten layer, and finally a `Dense` layer with 1 neuron and a *sigmoid* activation function. We use an `Adam` optimizer, which combines the best of the `AdaGrad` and `RMSProp` optimizers, with a learning rate of 0.001 and the function loss was `binary crossentropy`, this architecture gives us a total of params of 4,227,713 which 4,203,313 are trainable. This model was trained with a generator as a training dataset of 3884 images which were resized to 224x224 pixels and are `RGB` and a batch size of 60. During the fitting, we use a callback to get a view on internal states and statistics of the model.

`EfficientNetB0:` We drop the top classifications layers and replace them with a `GlobalAveragePooling2D` layer, a `Dropout` (50%), `Flatten`, and finally a `Dense` 2 neurons layers with a *softmax* activation function. We train the whole network almost from scratch, reusing only the 2% of the ImageNet weights. Because the normalization process is a part of the network itself, we introduce the data into the network without any pre-processing (`int64`) in `RGB` mode and 60 image batches.

`Xception:` The model was configured with the initial weights from and after the base model, we added a `GlobalAveragePooling2D` layer, a dense layer with 256 neurons and `Relu` activation function, a `Dropout` layer of (50%) and at the end, a `Dense` layer with 2 neurons and `softmax` activation function. We use an `RMSprop` optimizer with a learning rate of 0.0001 and the function loss was `categorical crossentropy`, this architecture gives us total params of 21,386,538 which 21,332,010 are trainable. This model was trained with a generator as a training dataset of 3884 images which were resized to 224x224 pixels and are `RGB`, 60 of batch size, and class mode categorical. During the fitting, we use a callback to monitor the loss and reduce the learning rate where needed.

`NasNetMobile:` In this model, we use the ImageNet initial weights as well and implement the `NasNetMobile` base model without including the top layers, after the base model we added some layers like `Dropout` layer of 50%, `Flatten`, `BatchNormalization`, and `Dense` of 256 neurons, a total of 14 layers before the output layer which has 2 neurons and `softmax` activation function. This model had 17,858,582 total params which 13,483,842 are trainable and was compiled with `categorical crossentropy` as loss function.

`VGG16:` We configurated this model with initial weights from ImageNet, we added a `Flatten` layer, a `Dropout` layer of 50% which helps prevent overfitting, and a `Dense` layer with *sigmoid* activation, the function loss was `binary crossentropy` and we use an `Adam` optimizer. This architecture gives us a total of params of 14,739,777 which 25,089 are trainable. This model was trained with a generator as a training dataset of 3884 which were resized to 224x224 pixels `RGB` and a batch size of 60. During the fitting, we use a callback to monitor the loss and reduce the learning rate.

4 Results

In this section, we present the main result of the previously discussed methodology for each network, a comparison, and a summary of the best performance. To compare the models, we evaluate some quality measures like accuracy, precision, recall, F_1 score using the results of the confusion table and ROC curve of each neural network. These results will be shown below.

4.1 Evaluation Metrics

To present the results and compare the models, we consider the following quality measures: Accuracy is the percentage that indicates the ratio of correct predic-

tions compared to the test data, as shown in (1). Precision is the probability, given a positive label, of how many of them are actually positive, as shown in (2). Recall, or sensitivity, is the ratio of correctly predicted positive observations to the all observations in actual class, como se describe en (3). Finally, F_1 score is the weighted average of Precision and Recall, as shown in (4).

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

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

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

$$F_1 score = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{4}$$

Where TP refers to true positives, TN to true negatives, FP to false positives and FN to false negatives.

4.2 Results

Confusion matrix: Figure 3 shows the confusion matrix obtained for each network where we can see the predictions compared with the actual values of the class to which the image belongs. In this context, we are looking to reduce to the minimum the false-positives (FP) and especially the false-negative (FN) outcomes as it could give the flawed belief that a person is non-contaminated and therefore exposing the community further to viral exposition. Considering this, Xception achieves the best performance as it only had seven FN and six FP, followed by EfficientNetB0 with nine FN and seven FP. The rest of the models, although having relatively high accuracy and recall, have a few more FN and FP, resulting in less reliable networks especially in the case of the mobileNetV3, which has the highest number of misclassifications.

Saliency Maps: The next two figures, show us the regions of the input image that contributes the most to the output value(label prediction), using the tool provied by [25], in order to corroborate the network correctness and its reliability. Figure 4 shows two positive cases (1 each row) and Figure 5 shows two negative cases, the first column in each figure is the original image and the rest belongs to each of the networks.

Comparison table: Table 1 show the quality measures previously mentioned and the total number of trainable parameters. Furthermore, we highlight the positive recall column as we considered the most meaningful result in this context, where a false negative could have far-reaching consequences. Here we can observe very similar outcomes, giving the central stage to VGG16 for archiving a relatively good performance with the lowest number of trainable parameters, approximately 3.97 million less than EfficientNetB0, the next network in the volume tier.

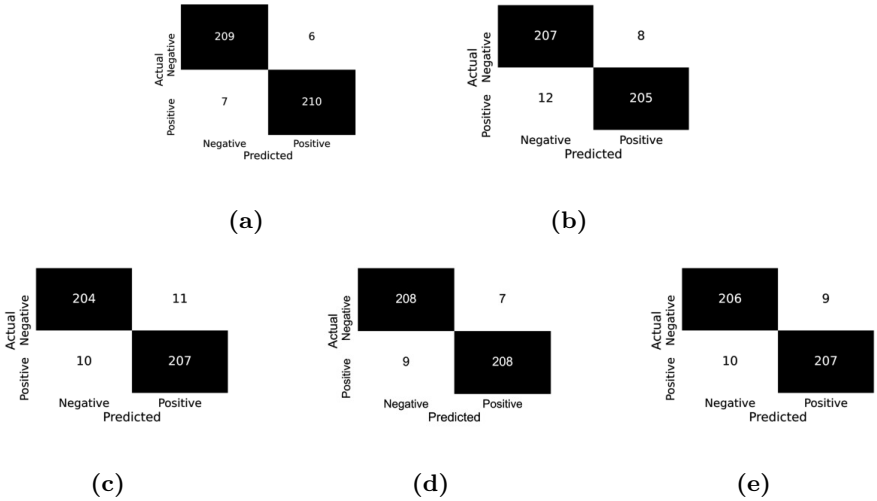


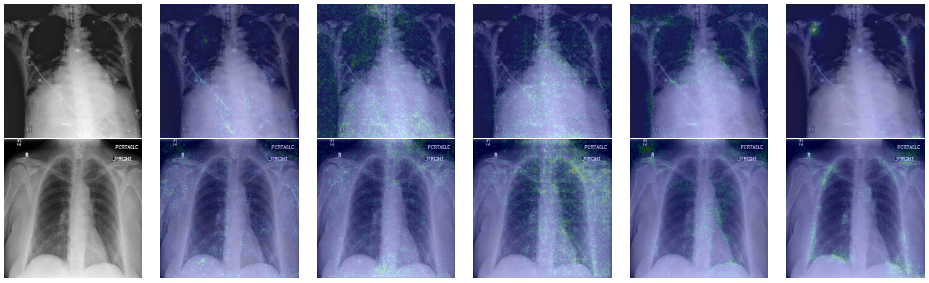
Fig. 3. This figure shows the confusion table for each network with, a) Xception, b) NasNetMobile, c) MobileNetV3, d) EfficientNetB0 and, e) VGG16

However, using the proper tuning and the fact that VGG16 were almost completely reused, it would be unfair to say, VGG16 had the best performance, as is possible to lower the other network's parameter number. In addition to this, we find that, oddly, being training with all its parameters, VGG16 falls behind. From this we consider that VGG16 has a promising reuse value in it is pre-train weights, as for the rest of the networks, they were, for the most part, trained from scratch so its reuse capability is off discussion in this work. We also examine that, considering the good achievement mobiles networks (EfficientNetB0, MobileNetV3Large, NasNetMobile) had relative to Xception (a large one), we can argue that, using this kind of models without transfer learning may be overkill for binary classification with the current number of samples.

That being said, Xception was the model that gives us the best performance in terms of pure statistics, archiving a general accuracy of 0.97 %, a recall of 0.97 % on the positive class, a F1-score of 0.97 %, and AUC of 0.995 on the ROC curve, follow by EfficientNetB0 with accuracy 0.97 %, recall 0.96 % positive class, F1-score 0.96 %, and AUC 0.996.

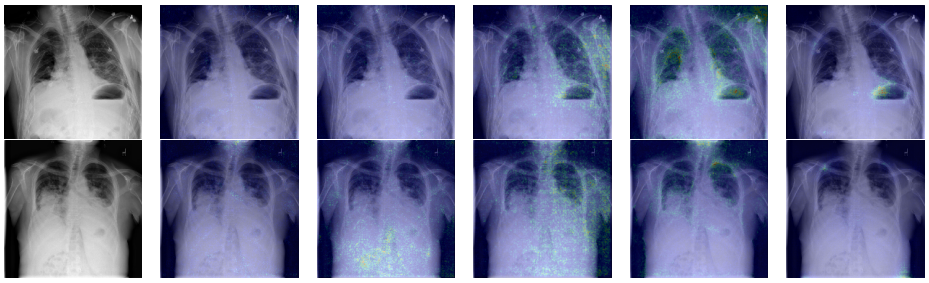
In table 1 we expose the main result of our experiment, showing the precision, recall, F1-score, general accuracy, and number of trainable parameters. We also highlight the positive recall column, as it is the quality measure of highest interest in this work.

ROC curve: Because our data set was well balanced, we did not see the necessity of the precision-recall plot as the ROC curve was a suitable indicator of the diagnostic ability of our classifier. Area Under the Curve (AUC) was calculated using the trapezoidal rule, supported in [20].



Original | MobileNetV3 | NasnetMobile | Xception | VGG16 | EfficientNetB0

Fig. 4. Saliency maps for covid clasification. Each row represents a positive example of covid and the activation of the points that contributed the most to the clasification with each of the networks.



Original | MobileNetV3 | NasnetMobile | Xception | VGG16 | EfficientNetB0

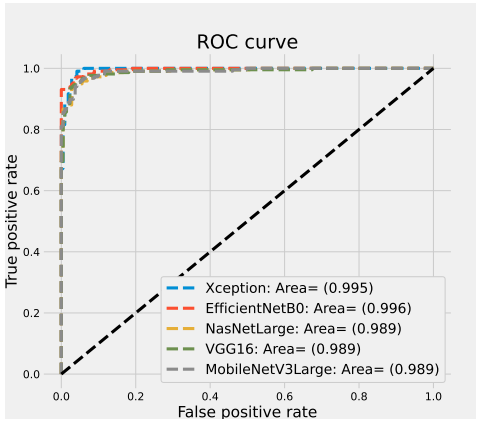
Fig. 5. Saliency maps for covid clasification. Each row represents a negative example of covid and the activation of the points that contributed the most to the clasification with each of the networks.

Table 1. Quality measures.

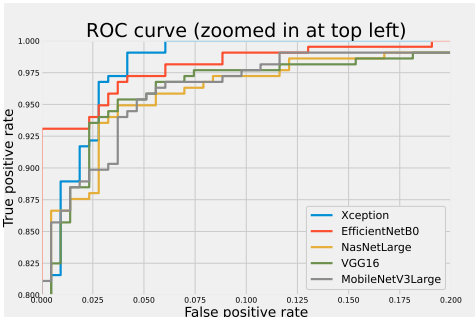
	Precision	**Recall**	F1-Score	Precision	Recall	F1-Score	Accuracy	Parameters (Millions)
Xception	0.97	**0.97**	0.97	0.97	0.97	0.97	0.97	21.3
EfficientNetB0	0.97	**0.96**	0.96	0.96	0.97	0.96	0.96	3.996
NasNetMobile	0.96	**0.95**	0.96	0.95	0.96	0.96	0.96	13.483
VGG16	0.96	**0.95**	0.96	0.95	0.96	0.96	0.96	0.02508
MobileNetV3	0.95	**0.95**	0.95	0.95	0.95	0.95	0.95	4,203

In figure 6a we expose the Receiver Operating Characteristic curve, where we can see the general performance of the diagnostic ability for each network, and in figure 6b we can observe a zoom in the top-left area for a better appreciation. Black serve as reference line. In general, we can see the same performance around all the models, this favorite of course the low parameters networks, due to the

gain in training time and the number of the parameters.



ROC



zoom

Fig. 6. Networks comparison a) ROC Curve b) ROC Curve zoomed in a top left.

Comparison with state-of-the-art: Using the reports published by [17, 28, 1], we present table 2, which summarizes the best performance of different recent articles, using similar datasets of Covid-19 and the same binary problem, to compare this work achievement. The table also includes studies that tackle the problem of discerning Covid-19 incidences on CXR using CNNs from a three classes perspective, being the third-class CXR of patients who suffer viral or bacterial pneumonia. Furthermore, we introduce the sample number of Covid-19 CXR, Healthy or Normal CXR (a.k.a, no-findings), and viral or bacterial pneumonia CXR.

Best architecture with its accuracy (%) is also present for each research. Our method was characterized by containing a relatively high number of CXRs in both Covid-19 and healthy units and adopt a binary approach. We consider that our best model archive a relatively solid result compared with these other

networks, contemplating that our dataset was bigger. That being said, we do consider that there is still room for improvement, using a bulkier dataset, employing data augmentation or image enhancement techniques, and adding pneumonia CXR specimens for network validation.

Table 2. General comparison of this work best results with the state-of-the-art.

Study	No. of samples	Architecture	Number of clases	Accuracy (%)
[17]	50 Covid-19, 50 Normal	ResNet-50	2	98
[28]	1,200 Covid-19, 1,341 Normal, 1,345 viral Pneumonia	VGG-16	3	98.29
[7]	224 Covid-19, 700 Pneumonia, 504 Healthy	VGG-19	3	98.75
[15]	25 Covid-19, 25 Normal	VGG-19, DenseNet-121	2	90
[24]	25 Covid-19, 25 Normal	ResNet-50	2	95.38
[29]	1,050 Covid-19, 1,050 Normal	CapsNet	2	97.24
[19]	192 Covid-19, 145 Normal	nCOVnet	2	97.62
[31]	358 Covid-19, 8,066 Normal, 5,538 Pneumonia	COVID-Net	3	93.3
[13]	125 Covid-19, 500 Normal, 500 Pneumonia	Xception	3	97.40
[5]	224 Covid-19, 504 Normal, 700 bacterial Pneumonia	VGG-19	3	93.48
This work	**2,158 Covid-19, 2,158 Normal**	**Xception**	**2**	**97**

5 Conclusions

In this paper, a comparison was made between five neural network models, which were pre-trained with the ImageNet dataset and then individually configured to be trained with chest X-ray images to identify covid incidences which were obtained from the COVIDx dataset and only approximately 4000 were used due to hardware limitations, each network was implemented using the base model of the corresponding architecture, but in some cases with the necessary adaptations to obtain better results.

This comparison shows that, although the five models obtain close results in terms of the recall measure, which is the most relevant for the problem at hand, some models such as VGG16 achieve relatively good results with a very small number of parameters compared to those that obtained the best results but with a considerably larger number of parameters such as Xception which obtained the highest recall or EfficientNetB0 with very close results but a larger area under the curve exceeding Xception with 0.01, which demonstrates the impact that can be achieved using transfer learning before training the model for a specific dataset.

Depending on the problem, it may be worth sacrificing a certain percentage of efficiency to greatly improve the performance of the network. The main contributions are: first, the transfer learning used in each model, adapting the architecture and parameters to the specific problem. Second, the comparison between some of the most popular models evaluated quantitatively with quality metrics and the general performance as a classifier through the roc curve. Third, the implementation of these models for covid detection using radiographs.

In future work, we can analyze variance to ensure that with a smaller part of the dataset very similar results are obtained, also improve the architecture with fewer parameters preserving the results and improve the most relevant quality metrics by adding images of cases with pneumonia as it is often confused with the covid and even use a larger dataset or data augmentation to get the train the network with a larger number of samples, finally we would also add a preprocessing stage of the images where everything that is not relevant for classification is removed.

This work is available at the Github repository[3].

References

1. Altaf, F., Islam, S.M., Janjua, N.K.: A novel augmented deep transfer learning for classification of COVID-19 and other thoracic diseases from X-rays. Neural Computing and Applications 0 (2021), https://doi.org/10.1007/s00521-021-06044-0
2. Altaf, F., Islam, S.M., Janjua, N.K.: A novel augmented deep transfer learning for classification of covid-19 and other thoracic diseases from x-rays. Neural Computing and Applications pp. 1–12 (2021)
3. Altaf, F., Islam, S.M., Janjua, N.K.: A novel augmented deep transfer learning for classification of covid-19 and other thoracic diseases from x-rays. Neural Computing and Applications pp. 1–12 (2021)
4. Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K.: Medical image analysis using convolutional neural networks: a review. Journal of medical systems 42(11), 1–13 (2018)
5. Apostolopoulos, I.D., Aznaouridis, S.I., Tzani, M.A.: Extracting Possibly Representative COVID-19 Biomarkers from X-ray Images with Deep Learning Approach and Image Data Related to Pulmonary Diseases. Journal of Medical and Biological Engineering 40(3), 462–469 (2020)
6. Apostolopoulos, I.D., Mpesiana, T.A.: Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine 43(2), 635–640 (2020)
7. Apostolopoulos, I.D., Mpesiana, T.A.: Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine 43(2), 635–640 (2020), https://doi.org/10.1007/s13246-020-00865-4
8. Bassi, P.R., Attux, R.: A deep convolutional neural network for covid-19 detection using chest x-rays. Research on Biomedical Engineering pp. 1–10 (2021)
9. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
10. Chung, A.: Actualmed covid-19 chest x-ray data initiative. https://github.com/agchung/Actualmed-COVID-chestxray-dataset (2020)
11. Chung, A.: Covid-19 chest x-ray data initiative. https://github.com/agchung/Figure1-COVID-chestxray-dataset (2020)

[3] https://github.com/Abel-RM/Deteccion-Covid.git

12. Cohen, J.P., Morrison, P., Dao, L., Roth, K., Duong, T.Q., Ghassemi, M.: Covid-19 image data collection: Prospective predictions are the future. arXiv preprint arXiv:2006.11988 (2020)
13. Das, N.N., Kumar, N., Kaur, M., Kumar, V., Singh, D.: Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information (January) (2020)
14. Grupo Directivo de Organización Mundial de la salud: Pruebas diagnósticas para el sars-cov-2 (2020)
15. Hemdan, E.E.D., Shouman, M.A., Karar, M.E.: COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images (2020), http://arxiv.org/abs/2003.11055
16. Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Le, Q., Adam, H.: Searching for mobileNetV3. Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
17. Narin, A., Kaya, C., Pamuk, Z.: Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. Pattern Analysis and Applications pp. 1–14 (2021)
18. Pan, I., Cadrin-Chênevert, A., Cheng, P.M.: Tackling the radiological society of North America pneumonia detection challenge. American Journal of Roentgenology 213(3), 568–574 (2019)
19. Panwar, H., Gupta, P.K., Khubeb, M., Morales-menendez, R.: Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information (January) (2020)
20. 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, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
21. Radiological Society of North America: Covid-19 radiography database. https://www.kaggle.com/tawsifurrahman/covid19-radiography-database (2019)
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252 (2015)
23. Saxen, F., Werner, P., Handrich, S., Othman, E., Dinges, L., Al-Hamadi, A.: Face attribute detection with mobilenetv2 and nasnet-mobile. In: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA). pp. 176–180. IEEE (2019)
24. Sethy, P.K., Behera, S.K., Ratha, P.K., Biswas, P.: Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. International Journal of Mathematical, Engineering and Management Sciences 5(4), 643–651 (2020)
25. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR abs/1312.6034 (2014)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

27. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
28. Taresh, M.M., Zhu, N., Ali, T.A.A., Hameed, A.S., Mutar, M.L.: Transfer learning to detect covid-19 automatically from x-ray images using convolutional neural networks. International Journal of Biomedical Imaging 2021 (2020)
29. Toraman, S., Burak, T., Turkoglu, I.: Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information (January) (2020)
30. Valderrama Molano, J.A., et al.: Clasificación de objetos usando aprendizaje profundo implementado en un sistema embebido. Universidad Autónoma de Occidente (2017)
31. Wang, L., Lin, Z.Q., Wong, A.: COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Scientific Reports 10(1), 1–12 (2020), https://doi.org/10.1038/s41598-020-76550-z
32. Wang, L., Lin, Z.Q., Wong, A.: Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports 10(1), 19549 (Nov 2020), https://doi.org/10.1038/s41598-020-76550-z
33. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2097–2106 (2017)
34. Xiao, T., Liu, L., Li, K., Qin, W., Yu, S., Li, Z.: Comparison of transferred deep neural networks in ultrasonic breast masses discrimination. BioMed research international 2018 (2018)

ISSN 1870-4069

Oil Fractions Impact on Mexican Soils Evaluation Web Application

Vladimir Avalos-Bravo[1], Chadwick Carreto Arellano[2], Macario Hernández Cruz[3], Blanca Barragán-Tognola[4], Mónica Fernanda Barragán-Tognola[5]

[1] Instituto Politécnico Nacional, Direccion de Educacion Virtual, SEPI-ESIQIE, UPIEM, SARACS ESIME, Mexico

[2] Instituto Politécnico Nacional, Direccion de Educacion Virtual, SEPI-ESCOM, Mexico

[3] Instituto Politécnico Nacional, Direccion de Educacion Virtual, Mexico

[4] Instituto Politécnico Nacional, SEPI-ESIQIE, Mexico

[5] Universidad Politécnica del Golfo de México, Mexico

monica.barragan@updelgolfo.mx, chadcarreto@gmail.com, tognola6@hotmail.com, {ravalos, mahernandezc} @ipn.mx

Abstract. This paper presents an Evaluation Web Application designed to provide essential technical information about environmental damage caused by light fraction, medium fraction and heavy impact fraction contamination by oil and its effect on Mexican soils. According to NOM-138-SEMARNAT-2012 "Maximum permissible limits of hydrocarbons in soils and guidelines for sampling in the characterization and specification for remediation", published at Federation Official Gazette in Mexico on September 10th, 2013. The application developed on Java Script Object (JSON) technology for a non-relational database, shows effects on every different soil when a hydrocarbon spill happens, and suggest appropriate mitigation and remediation strategies for every soil present at Mexico municipalities.

Keywords: Oil fractions, polluted soils, web application, Mexican soils.

1 Introduction

In Petroleum Industry, it is of fundamental importance to minimize risks in industrial operation, maximizing occupational safety and health of all the processing work personnel, like storage and distribution facilities, at the wholesale or retail level of petroleum products and petrochemicals. as well as functional and operational integrity of facilities and related equipment [1]. In Mexico, gasoline and diesel clandestine takes in transportation pipelines are a serious problem with social, economic, environmental and operational safety impacts. As a result of the so-called Energy Reform of our country (2013-2014) [2], the responsibility for Industrial Hydrocarbons Sector Management was concentrated in specialized government agencies: National Hydrocarbons Commission (CNH) for oil resources exploration and exploitation; Energy Regulatory Commission (CRE) for petroleum and petrochemicals products operation, distribution and commercialization; besides Security, Energy and Environment Agency (ASEA) for Industrial Safety and Environmental Protection of the Hydrocarbons National Sector. The activities carried out in the Oil Value Chain (OVC) three segments, begin with the drilling of wells in land and seabed and the extraction of crude oil, going through natural resource industrial transformation to convert it into oil products and other derivatives; liquids and gases, which have to be stored, transported, distributed and marketed until they reach final consumers, which can be industries and the general public [6]. Dangerous substances are handled throughout activities sequence, since they are chemical compounds that can form explosive, flammable and toxic mixtures, consequences of an unexpected release into the environment can cause irreparable damage, as well as considerable economic losses [3].

Considering the magnitude and complexity of the activities carried out on a daily basis in the Petroleum Industry, major accidents related to the handling of hazardous substances and materials occur infrequently; however, social, environmental and economic cost is often high [8, 9]. The resulting loss depends on the capacity and resilience of the system and environment under study to withstand the disaster and, in turn, the disaster occurs when hazards and risks are linked to vulnerability [5].

Oil spills to the ground are conveniently studied by classifying them according to the activities carried out in a given segment of the oil value chain (OVC). The spills in the upstream segment focus on the effects on the soil and offshore water, due to the spillage of oil, drilling fluids and materials extracted from the subsoil by drilling bits in onshore and offshore producing reservoirs, as well as in transfer ports [12]. The spills in the intermediate current segment (midstream) correspond to those that occur at sites where oil, gas and its derivatives storage and distribution facilities are located, gas and liquid transportation pipelines and natural gas liquefaction and regasification facilities [7]. In this classification there are also mobile sources, freight trains and tank trucks. Spills in the downstream segment correspond to those that occur at sites where oil refineries and petrochemical facilities are located, including facilities with a large storage capacity for petroleum products and petrochemicals. Industrial facilities for processing petroleum derivatives called petrochemicals are also included in this segment [12].

Table 1. Distribution of fires and explosions in industries that use or process hydrocarbon [16].

Typology	Proportion %
Fire	32
Explosions inside equipment due to air input	11
Explosions within equipment due to uncontrolled reaction, or explosive decomposition	23
Explosions outside equipment, but inside building	24
Explosions outdoors	3
Container explosions (due to corrosion, third party damage, overheating or overpressure)	7

1.1 Explosions and Fire in the Oil and Gas Industry

An explosion can be defined as the generation of a pressure wave in the air as a consequence of the extremely rapid release of energy. Within this broad definition there are various physical and chemical phenomena that, with a certain probability, can occur in process industry that uses hazardous substances [13, 14, 15]. In these industries, explosions represent, together with fires, the most frequent and destructive accidents, highlighting those dedicated to the manufacture of explosives or pyrotechnic materials, those that use flammable gases or Installations that, without having capable substances If they cause an explosion by themselves, they have containers where high pressures can be generated which, when they explode, release the contained energy in a violent way. Table 1 provides an overview of the importance of explosions in the chemical industries [16].

Fire is a chemical reaction that involves an element oxidation or rapid combustion causing contamination in Mexican soils, although there is no presence of fire, hydrocarbons also affect soils nature and contaminate them.

1.1.1 Contaminated Soils in Mexico

In Mexico there is a great variety of soils due to various factors interaction, such as complex topography caused by the Cenozoic volcanic activity, altitudinal gradient (from zero to 5,600 meters above sea level), the presence of four of the Five types of climates and the landscape diversity and rocks type in territory, in Mexico there are 26 of 32 soil groups recognized by International World Reference Base System for Soil Resources (IUSS). Leptosols (28.3% of territory), Regosols (13.7%), Phaeozems (11.7%), Calcisols (10.4%), Luvisols (9%) and Vertisols (8.6%) dominate all over the country, together occupy 81.7% of national surface [20].

Contaminated soil is all that whose characteristics have been negatively altered by the presence of dangerous chemical components of human origin, in a concentration such that it poses an unacceptable risk to human health or the environment. Oil spills affect the physical properties of soils and especially, natural populations of microorganisms. In particular, they cause a decrease in free-living nitrogen-fixing bacteria, responsible for assimilating and recycling nutrients in biogeochemical cycles [10].

In addition to soil contamination effects produced by hydrocarbons, oil interferes in parameters determination, such as texture, organic matter, real density and porosity

Table 2. Products associated with hydrocarbon spills, for which maximum permissible limits of contamination in soils are established [11].

Contaminant Product	Hydrocarbons				
	Heavy Fraction	Medium Fraction	HAP	Light Fraction	BTEX
Mix of unknown petroleum products	X	X	X	X	X
Crude Oil	X	X	X	X	X
Fuel Oil	X		X		
Paraffin	X		X		
Petrolatum	X		X		
Derived Oils	X		X		
Gas Oil		X	X		
Diesel		X	X		
Jet Fuel		X	X		
Kerosene		X	X		
Creosote		X	X		
Gas Plane				X	X
Solvent Gas				X	X
Gasoline				X	X
Naphta Gas				X	X

Table 3. Maximum permissible limits for hydrocarbon fractions in soils [11].

Hydrocarbons Fractions	Predominant Soil Usage (mg/kg) Dry Base			Analytic Method
	Agricultural, forestry, livestock and conservation	Residential and recreational	Industrial and commercial	
Light	200	200	500	NMX-AA-105-SCFI-2008
Medium	1200	1200	5000	NMX-AA-145-SCFI-2008
High	3000	3000	6000	NMX-AA-134-SCFI-2008

[18]. Therefore, soil pollutant's introduction can result in damage or some functions loss and affectation of water, particularly groundwater. The concentration of dangerous pollutants in soils above certain levels entails a large number of negative consequences for human health, as well as for all types of ecosystems and others.

For this reason, contaminated soils constitute the most urgent and important problem still unsolved in environmental matters [17].

1.2 NOM-138-SEMARNAT/SSA1-2012

This standard establishes the maximum permissible limits for hydrocarbons in soils and sampling guidelines for characterization and specifications for remediation, as well as, products associated with oil spills, for which maximum limits are established are listed in Table 2 [11].

Maximum permissible limits for hydrocarbons fractions in soils listed in Table 3 [11], and Maximum Permissible Limits for specific hydrocarbons in the soil in Table 4 [11].

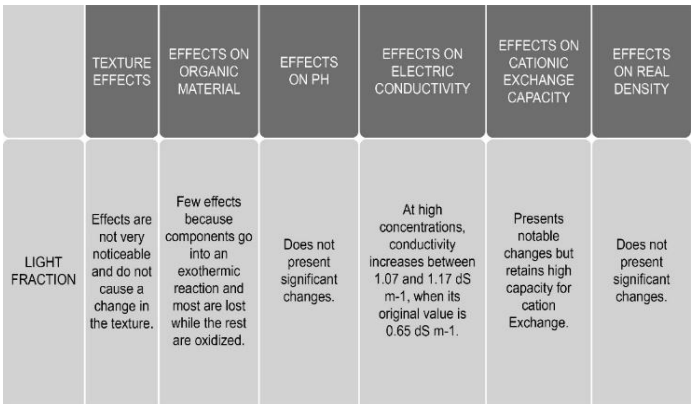


Fig. 1. Oil Impacts effects on soils.

Table 4. Maximum permissible limits for specific hydrocarbons in soils [11].

Specific Hydrocarbons	Predominant Soil Usage (mg/kg) Dry Base			Analytic Method
	Agricultural, forestry, livestock and conservation	Residential and recreational	Industrial and commercial	
Benzene	6	6	500	NMX-AA-141-SCFI-2007
Toluene	40	40	5000	NMX-AA-141-SCFI-2007
Ethylbenzene	10	10	6000	NMX-AA-141-SCFI-2007
Xylenes (sum of isomers)	40	40		NMX-AA-141-SCFI-2007
Benzo [a] pyrene	2	2		NMX-AA-146-SCFI-2008
Dibenzo [a, h] anthracene	2	2		NMX-AA-146-SCFI-2008
Benzo [a] anthracene	2	2		NMX-AA-146-SCFI-2008
Benzo [b] fluoranthene	2	2		NMX-AA-146-SCFI-2008
Benzo [k] fluoranthene	8	8		NMX-AA-146-SCFI-2008
Indene (1,2,3-cd) pyrene	2	2		NMX-AA-146-SCFI-2008

2 Oil Fractions Impact on Mexican Soils Evaluation Web Application Development

In order to be able to develop the correlation between oil fractions impact and the effect on the soils, a characterization study was used to determine soil damage [18]. It was decided that based on the effects produced according to oil fractions, it would be classified according to most noticeable changes that have occurred in soils, such a texture effects, organic matter effects, pH effects, electrical conductivity effects, cation exchange capacity effects, real density effects, apparent density effects and porosity effects, as shown in Figure 1.

In general, no matter soil type, sand values tend to increase and clay to decrease in medium and heavy fraction presence, variance between clay and concentration factors and hydrocarbon type indicates a non-significant difference. ($P < 0.05$) due to type of hydrocarbon effect that can be observed when comparing effects caused by the three types of fraction that is manifested when the concentration of hydrocarbon in the soil is very high. The texture is modified due to the adsorption of the middle fraction and the heavy fraction by the soil particles through electrostatic interactions of the type of Van der Waals forces, hydrogen bridges, water bridges and cationic bridges, causing effects on the sedimentation rate, which affects the density of the mud formed by the soil and water during the analysis, inducing a reading of particles. with the Bouyocos hydrometer, which is interpreted as altered.

Organic matter is one of the parameters with important variations that increase in proportion to the fraction's concentration, whether light, medium or heavy. It can also be seen that for the middle fraction there is a greater increase when hydrocarbons concentrations are present. Statistically between the organic matter parameter and the hydrocarbon type and concentration factors, it was determined that there is a significant difference ($P < 0.05$) for both factors, which is corroborated with the levels that increase in direct proportion to the concentration of hydrocarbons of light fraction ($r = 0.97$), medium fraction ($r = 0.88$) and heavy fraction ($r = 0.97$). Increase in organic matter is not beneficial, since said increase is due to petrogenic and not biogenic material. This may actually represent an ecotoxic risk due to the presence of polynuclear aromatic hydrocarbons contained in a greater proportion in medium and heavy fraction.

PH practically does not have a variation in any concentration presence and hydrocarbon type, so it remains medium to slightly acidic. The analysis of variance between the pH parameter, against the factors concentration and type of hydrocarbon, determined that there is no significant difference ($P < 0.05$), that is, the soil subjected to different levels of contamination from the different fractions, produces minimal effects on pH, in such a way that no effect is perceived in this variable. Electrical conductivity has an irregular variation, since values increase and decrease in the presence of the three types of fractions at different concentrations. However, there is a slight tendency to increase when the pollutant is light fraction. When analyzing the variance between the electrical conductivity against the factors concentration and fraction, it was determined that there is no significant difference ($P < 0.05$), this means that the contaminated soil at different levels of contamination by different hydrocarbons does not they prevent the solubilization of the salts present in the soil, which is manifested in the similar determinations of conductivity in the extract, similar determinations of conductivity in the extract of the saturation paste.

There is an irregular variation in the cationic exchange capacity values, since they increase and decrease regardless of the concentration and type of hydrocarbon, so the trend is indefinite. The most notable changes correspond to the soil contaminated with gasoline, still retaining a high cation exchange capacity. The variations are related to the effect of the adsorption of hydrocarbons on the surface of the mineral soil particles, interfering in the cation exchange sites and by electrostatic interactions. It was determined that there is no significant difference ($P < 0.05$) between the cation exchange capacity parameter against the concentration factors and type of fraction, which is

observed as an effect on the values with an irregular tendency to increase and decrease around the average value of uncontaminated soil, clay soil real density variation with respect to the concentration and type of hydrocarbon, where the medium fraction and the light fraction show minimal variation even in high concentrations of hydrocarbon.

It should be noted that the heavy fraction does cause a variation in the real density tending to decrease from high concentrations, causing a downward trend. This value must be considered with great care for the purpose of selecting some soil remediation technology, however, between real density against the factors concentration and type of hydrocarbon, there is no significant difference ($P < 0.05$). This is due to the relationship with the determination process, since it takes into account the weight and volume only of the mineral particles in the soil, which at the time of determining their weight, this is modified by the combination of particles with higher density with the hydrocarbide with lower density. Variations in apparent density are minimal, so different fractions at different concentrations do not influence this parameter, the analysis between variance between the apparent density parameter against the concentration factors by fraction type, they indicate that there is no significant difference ($P < 0.05$) for the type of hydro-carbide.

Porosity does not vary significantly when the soil is contaminated with light fraction or medium fraction hydrocarbons, however, with heavy fraction hydrocarbons, this parameter decreases drastically. This trend originates from the effect on the decrease in the real density, since there is a mathematical relationship between both parameters. Statistically between porosity and hydrocarbon type, there is no significant difference ($P < 0.05$) for light fraction and medium fraction.

2.1 Database Construction

The necessary information was obtained for the creation of the database in Excel® software by consulting NOM-138-SEMARNAT/SSA1-2012 rule [11]. Because Excel® is a registered trademark of Microsoft®, the use of freeware was considered for database migration, it was necessary to import all data with help of simple text format for data exchange JSON and will be linked to a Mexican edaphology database [19]. It is important to mention that it is due to this data collection that a non-relational database will be generated.

As a first phase, maximum permissible limits for hydrocarbon fractions in soils were obtained from PROFEPA, although the type of pollutant, in this case, oil derivatives [4]. including hydrological information, edaphologic information, geologic information and economic information of every place, production modes among other facts. Only the necessary information that would be required for the database elaboration had to be captured and some fields that help to its exploration in order to obtain desired results, once information had been reviewed, it was then synthesized into State, number of municipalities and soils percentage, classified in an edaphological way by each of them, as well as the use of land in corresponding municipality (urban, agriculture and pasture). as shown in Figure 2.

HYDROCARBONS FRACTIONS	PREDOMINANT SOIL USE (mg/kg DRY BASE)			ANALYTIC METHOD
	Agricultural, forestry, livestock and conservation	Residential and recreational	Industrial and commercial	
Light	200	200	500	NMX-AA-105-SCFI-2008
Medium	1 200	1 200	5 000	NMX-AA-145-SCFI-2008
Heavy	3 000	3 000	6 000	NMX-AA-134-SCFI-2006

Fig. 2. NOM-138-SEMARNAT/SSA1-2012 Database created in Excel®.

```
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title></title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="lib/materialize/css/materialize.css">
    <link rel="stylesheet" href="css/estilos.css">
</head>

<body>
    <!--[if lt IE 7]>
        <p class="browsehappy">You are using an <strong>outdated</strong> browser. Please <a href="#">upgrade your browser</a> to improve your experience.</p>
    <![endif]-->
    <nav>
        <div class="nav-wrapper lime accent-4 ">
            <a href="#" class="brand-logo blue-grey-text text-darken-3"><img src="img/tierra.png" class="logo responsive-img" />
                <B>EDAFOLOGÍA MEXICANA</B>
            </a>

            <ul id="nav-mobile" class="right hide-on-med-and-down ">
                <li><a class="blue-grey-text text-darken-3" href="sass.html">Salir</a></li>
            </ul>
        </div>
    </nav>
    <section class="blue lighten-5">

        <div class="row">
            <div class="col s4 m3 ">
                <ul class="collection white-text ">
                    <li class="collection-item blue-grey lighten-3 center-align lime accent-4">Estado</li>
                    <li class="collection-item blue-grey lighten-3 blue-grey lighten-3 truncate">Aguascalientes</li>
                    <li id="bc" class="collection-item blue-grey lighten-3 blue-grey lighten-3">Baja California</li>
                    <li class="collection-item blue-grey lighten-3  blue-grey lighten-3">Baja California Sur</li>
                    <li class="collection-item blue-grey lighten-3  blue-grey lighten-3">Campeche</li>
                    <li class="collection-item blue-grey lighten-3">Coahuila</li>
                    <li class="collection-item blue-grey lighten-3">Colima</li>
                    <li class="collection-item blue-grey lighten-3">Chiapas</li>
                    <li class="collection-item blue-grey lighten-3">Chihuahua</li>
                    <li class="collection-item blue-grey lighten-3">Distrito Federal</li>
                    <li class="collection-item blue-grey lighten-3">Durango</li>
                    <li class="collection-item blue-grey lighten-3">Guanajuato</li>
                    <li class="collection-item blue-grey lighten-3">Guerrero</li>
                    <li class="collection-item blue-grey lighten-3">Hidalgo</li>
                    <li class="collection-item blue-grey lighten-3">Jalisco</li>
                    <li class="collection-item blue-grey lighten-3">México</li>
                    <li class="collection-item blue-grey lighten-3">Michoacán</li>
                    <li class="collection-item blue-grey lighten-3">Morelos</li>
```

Fig. 3. Converting Excel® to JSON format using JavaScript®.

The first step will be to convert the Excel® database to Java Script Object (JSON), format. JSON is a simple text format that is used for data exchange because it is open source and has no cost, it is also a subset of the independent - language JavaScript object literal notation and has many advantages as it is used as a data exchange format.

In order to add the values contained in the Excel® database to the compilation code, it must be carried out through transformation and adaptation to JSON, constructing and referencing the values contained in each cell linearly so that they can be represented in the tool.

It is important to mention that the no relational database or the NoSQL query language, works under the principle that they do not contain an identifier that serves as

```
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title></title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="lib/materialize/css/materialize.css">
    <link rel="stylesheet" href="css/estilos.css">
</head>

<body>
    <!--[if lt IE 7]>
        <p class="browsehappy">You are using an <strong>outdated</strong> browser. Please <a href="#">upgrade your browser</a> to improve your experience.</p>
    <![endif]-->
    <nav>
        <div class="nav-wrapper lime accent-4 ">
            <a href="#" class="brand-logo blue-grey-text text-darken-3"><img src="img/tierra.png" class="logo responsive-img" />
                <B>EDAFOLOGÍA MEXICANA</B>
            </a>

            <ul id="nav-mobile" class="right hide-on-med-and-down ">
                <li><a class="blue-grey-text text-darken-3" href="sass.html">Salir</a></li>
            </ul>
        </div>
    </nav>
    <section class="blue lighten-5">

        <div class="row">
            <div class="col s4 m3 ">
                <ul class="collection white-text ">
                    <li class="collection-item blue-grey lighten-3 center-align lime accent-4">Estado</li>
                    <li class="collection-item blue-grey lighten-3 blue-grey lighten-3 truncate">Aguascalientes</li>
                    <li id="bc" class="collection-item blue-grey lighten-3 blue-grey lighten-3">Baja California</li>
                    <li class="collection-item blue-grey lighten-3  blue-grey lighten-3">Baja California Sur</li>
                    <li class="collection-item blue-grey lighten-3  blue-grey lighten-3">Campeche</li>
                    <li class="collection-item blue-grey lighten-3">Coahuila</li>
                    <li class="collection-item blue-grey lighten-3">Colima</li>
                    <li class="collection-item blue-grey lighten-3">Chiapas</li>
                    <li class="collection-item blue-grey lighten-3">Chihuahua</li>
                    <li class="collection-item blue-grey lighten-3">Distrito Federal</li>
                    <li class="collection-item blue-grey lighten-3">Durango</li>
                    <li class="collection-item blue-grey lighten-3">Guanajuato</li>
                    <li class="collection-item blue-grey lighten-3">Guerrero</li>
                    <li class="collection-item blue-grey lighten-3">Hidalgo</li>
                    <li class="collection-item blue-grey lighten-3">Jalisco</li>
                    <li class="collection-item blue-grey lighten-3">México</li>
                    <li class="collection-item blue-grey lighten-3">Michoacán</li>
                    <li class="collection-item blue-grey lighten-3">Morelos</li>
```

Fig. 4. Edaphology and NOM-138-SEMARNAT/SSA1-2012 non-relational database.

a data set and others, the information is organized in tables or documents and it is very useful when there is not an exact scheme of what is going to be stored, the program code is simple, as shown in Figure 3.

Next, the Materialize® Software is used, Materialize® is a modern responsive front-end framework based on Material Design, specially conceived for projects that make Material Design their flag. This CSS framework allows you to save time and effort when implementing and optimizing web projects. Not only does it contain a multitude of CSS classes already configured, but by incorporating JavaScript® code, the values from the database can be added to our interface.

On the one hand, the advantages of Materialize® CSS framework are:

- The development time is less, since most of the code is already written.
- A CSS framework is applied to achieve a better aesthetic in future projects.
- Design is more robust and the final aesthetic is more homogeneous.
- Materialize® occupies only 140 KB with its CSS, to which must be added 180 KB with Java Script® for a low space and resources demand.

At present, edaphology and NOM-138-SEMARNAT/SSA1-2012 non-relational database is fully operational and is available online via http://148.204.111.72/edafologia2/ web page as shown in Figure 4.

3 Conclusions

Depending on the type of hydrocarbons that are handled in each state and municipality of the country, the authorities must know what type of unwanted event may occur, that is, fire, explosion, formation of toxic clouds (leak) or spill, in order to locate the population near risk areas and determine the resources they have, in addition to the emergency services available to develop an emergency response plan based on computer tools such as the one developed for this purpose.

Information systems allow making decisions as a timely manner to mitigate the unwanted events effects.

Although NOM-138-SEMARNAT-2012 and NOM-138-SEMARNAT / SSA1-2012 standards contemplate techniques to determine soil impact by heavy, medium or light fractions, the amount of petroleum products is quite significant, so this standard should be expanded with more information to be applied in storage plants that spill pollutants on the ground.

Being a knowledge tool that helps to improve the understanding of all types of hydrocarbons pollution effects on soils over the Mexican republic in order to mitigate them will reduce population affectation.

This type of information systems will serve as a reference for design of remediation and restoration processes on refineries, gas stations, storage and distribution plants adjacent land, as well as lands where these types of facilities have existed and could have affected soils, in addition to serving as a reference for the development of standards related to characterization and sanitation of soils in the proximity of gas stations, refineries and storage and distribution terminals.

Acknowledgments. This project was funded under the following grants: SIP-IPN: No-20210303 and the support of DEV-IPN Instituto Politecnico Nacional.

References

1. Crowl, D. A., Louvar, J. F.: HAZOP Studies. Chemical Process Safety, Fundamentals with Applications Second Edition, pp. 450–453 (2002)
2. GOB.MX: Reforma Energetica 2013. Modernizacion del sector energetico del pais (2021) url: https://www.gob.mx/cms/uploads/attachment/file/10233/Explicacion_ampliada_de_la_Reforma_Energetica1.pdf.
3. NOM-010-STPS-1999: Condiciones de seguridad e higiene en los centros de trabajo donde se manejen, transporten, procesen o almacenen sustancias químicas capaces de generar contaminación en el medio ambiente laboral (1999)
4. SEMARNAT: Informe de Actividades de la PROFEPA (2017)
5. NRF-045-PEMEX-2010: Requisitos técnicos y documentales que se deben cumplir en la contratación y/o para la adquisición de los Sistemas Instrumentados de Seguridad aplicables

a los Sistemas de Paro por Emergencia en las instalaciones de procesos industriales de Petróleos Mexicanos y Organismos Subsidiarios (2010)
6. Arcos, S., Izcapa, C.: Identificación de peligros por almacenamiento de sustancias químicas en industrias de alto riesgo en México Centro Nacional para la Prevención de Desastres 1ª. Edición (2003)
7. INFOMEX: A través de PEMEX Refinación, Subdirección de Logística de Hidrocarburos y Derivados de la Dirección Corporativa de Operaciones, solicitud Nº 1857200017614, Datos Estadísticos con respecto a fallas o percances en operación y mantenimiento en ductos (2018)
8. Jensen, H.J.: Self-Organized Critically Cambridge Lecture Notes in Physics, Cambridge University Press (1998)
9. Lees, F.P.: Loss Prevention in the Process Industries. Butterworth-Heinemann, Oxford (1996)
10. Vázquez-Luna: Impacto del Petróleo Crudo en Suelo Sobre la Microbiota de Vida Libre Fijadora de Nitrógeno, Tropical and Subtropical Agroecosystems,13, pp. 511–523 (2011)
11. NOM-138-SEMARNAT-2012: Límites máximos permisibles de los hidrocarburos en suelos (2021)
12. NOM-117-SEMARNAT-2006: Especificaciones de protección al ambiente durante las actividades de instalación, mantenimiento mayor y abandono, de los sistemas para la conducción de hidrocarburos y petroquímicos en estado líquido y gaseoso (2006)
13. Flood, R.L.: The Relationship of "Systems Thinking to Action Research", handbook of action Research-Participative Inquiry & Practice, Peter Reason & Hillary Brandbury (2001)
14. Garrison, W.G.: One Hundred Largest Losses-A Thirty-Year Review of Property Damage Losses in the Hydrocarbon-Chemical Industries. M&M Protection Consultant (1988)
15. Horálek, V.: EGIG Pipeline Incident database: Safety performances determine the acceptability of cross country (2001)
16. Horálek, V.: Safety performances determines the acceptability of cross country gas transmission systems.HSE (2002) The Dangerous substances and Explosives Atmospheres Regulations, url: http://www.hse.gov.uk/fireandexplosion/dsear.htm (2001)
17. IChemE: Institution of Chemical Engineers, Explosions in the Process Industries. Major Hazards Monograph IchemE (1994)
18. Martínez, V.E., López S. M.: Efecto de hidrocarburos en las propiedades físicas y químicas de suelos arcillosos Terra Latinoamericana, 19, pp. 9–17 (2001)
19. Avalos V., et al.: Mexican Edaphology Database (2021)
20. INEGI: Carta Edafológica de la República Mexicana (2007)

ISSN 1870-4069

Automatic Recognition of Indigenous Languages from Different Mexican Geographic Regions Using Long-Term Average Spectrum

Luis David Huerta-Hernández[1], Ricardo Melchor[1], Julio Cesar Ramírez-Pacheco[3], Homero Toral-Cruz[3], Khalid S. Aloufi[2], José Antonio León-Borges[3]

[1] Universidad del Istmo, Mexico

[2] Taibah University, College of Computer Science and Engineering, Saudi Arabia

[3] Universidad de Quintana Roo, Mexico

{luisdh2, ricardo.melchor}@bianni.unistmo.edu.mx, alawfikhalid@gmail.com, {jleon, juliocr, htoral}@uqroo.edu.mx

Abstract. In this paper, a method for automatic recognition of Mexican Indigenous Languages (MIL): Maya, Mixteco, Zapoteco, Mixe, Nahuatl, Tarahumara, Mazahua, Tseltal, Chichimeco, and Huichol is proposed. The Long-Term Average Spectrum (LTAS) is employed as a feature for the recognition process. Moreover, the performance of classifiers such as Multi-Layer Perceptron (MLP), Sequential Minimal Optimization (SMO), Random Forest (RF), Naive Bayes, and K-Nearest Neighbor (KNN) are also highlighted. In order to reduce speech vector features, LTAS sequences, extracted from audios are first passed on *BestFirst* filters. In the experiments, high performance for MIL recognition was achieved, using a simplified codification scheme of the voice, with vectors features with a low number of values. The method is remarkable for its simplicity and effectivity, bringing away on no-tested languages in the speech processing area.

Keywords: Speech processing, recognition, data mining, long term average spectrum, classifiers, Mexican indigenous languages.

1 Introduction

The continuous development of technologies requests for more robust applications capable to generalize tasks, including most possible cases found in the real world. In this sense, the Automatic Speech Recognition (ASR) area, since some years ago, awoke the interest between researchers for the Automatic Dialect Identification (ADI) [1].

Many countries have been dedicating efforts to the research of ADI [1, 2], recent studies have been presented for a variety of dialects [2], Arabic [3], and Malayalam [4].

Many are the benefits for integrate with ASR the languages representatives for each country, which the importance is high too. Many people could be benefited from more robust ASR, including dialect languages. Applications as the automatic call center and improves enriched indexed of spoken documents are discussed in [1], improving and enriching of ASR engines, as well as characterizing of speaker traits and Data Mining processes [2].

In socio-political context, the United Nations declares about the importance and promoting of multilingualism: “it is urgent to *take action to promote multilingualism*, in other words, to encourage the development of coherent regional and national language policies which give opportunity for the appropriate and harmonious use of languages in a given community and country" [5].

In Mexico, the efforts to research new technologies and their endemic languages are scarce. Although there was focused research on the textual analysis of indigenous languages as *Nahuatl* [6], there are not enough research aimed to integrate the indigenous languages with the ASR.

On the other hand, most of the literature makes the ADI using the typical features as MFCC, leading in well-detailed information extracted from the speech, but on the other side, the number of parameters is high. These values of the speech scheme codification must be processed by filters and classifiers, affecting the time of values extraction, filtering, and training, as well as the precision of the classification [7, 8], for mention some disadvantages. As well, many efforts are done to reduce speech parameters [9] [10]. So, we need to test alternative speech codifications. Speech codifications scheme as LTAS have been scarcely explored on language recognition, hence, in this work, a method that uses LTAS as a speech codification scheme is presented. One direction of this study is to determine if LTAS is highly useful to make the ADI at the same time the number of extracted values is reduced.

One motivation of this study is to start the basis for ADI of Mexican Languages, encouraging this research area in the country, due to the limited or null studies in this sense, at the same time to propose a new method, using a low number of speech values for the classification, reducing the complexity of data and timely response, as well as in-crease the probability to enhance the classifiers accuracy. As well, is important to experiment with the recognition based on the vocal tract, the way the speech is produced, where the LTAS fits in this sense.

The aim of this study is to propose a method to make ADI, using some representative Mexican Indigenous Languages of different geographical regions, answering the following questions: Is LTAS useful to recognize MIL? The use of LTAS promotes the use of a reduced number of values extracted from speech, in ADI?, Which of the classifiers are best suited in order to classify MIL?

The paper is organized as follows: Section 2, exposes the related works citing the classifiers and features used. Section 3, a brief explanation of the common features used on ADI, is presented. Section 4 exposes the database description and the geographical representation of the experimental dialects. Section 5, the experimental results of MIL recognition, are presented. Section 6, the Discussion, presenting the similarity of the classifiers used in state-of-art works and the present study; remarking the low number

of values used to perform recognition, for the presented method. Section 7. Conclusions and Future works.

2 Related Works

In this section, related works about ADI are exposed. In respect to Mexican Indigenous Languages recognition, there are not works reported in the literature of the state of art.

The works will be revised briefly, with respect to the methods, the features, and the number of values used to recognize the dialects. These works will be cited too, in the Discussion section. The methods used above the 300 values per second and MFCC features, Mel energies, energy, pitch, or a mix of them.

2.1 ADI based on MFCC as the Main Feature

A comparative study for classifiers to recognize Malayalam dialect has been reported in [4], using Thrissur and Kozhikode corpora. Malayalam dialect is spoken in the south of India, concretely, in Kerala state. The study is based on two of the fourteen Malayalam dialects. Speech files of 15 speakers for each dialect were used, who read 30 sentences with 3 three or a maximum of six words.

The features used were MFCC (detailed in *Speech Features* section), energy, and pitch. The speech signals were pre-processed before the feature extraction, to enhance the accuracy and efficiency of the extraction process. In this work, the MFCC extraction is exposed and citing the use of 10-30 *ms* size frames. In each frame several coefficients are extracted, typically 12-13 coefficients. Since the precise number of coefficients and frame size are not detailed, assuming the use of 12 coefficients and 20 *ms* frame size, for each second there are 50 coefficients vectors with length 12. Simple calculating, 12×50, having 600 values of MFCC per second.

This method, additionally uses energy and pitch features, resulting in above those 600 values per second. Artificial Neural Net classifier produced a recognition accuracy of 90.2 %, Support Vector Machine of 88.2%, and Naive Bayes of 84.1%.

The study reported in [2], is approached to analyze the differences between reading versus spontaneous speech Arabic dialect. To analyze the differences, a Gradient Mixture Model (GMM) was used. The method is based on MFCC features, using a 20 *ms* frame size, estimating 600 values per second.

2.2 ADI based on Mel Filter Banks as the Main Feature

Automatic Arabic dialect identification was reported in [3], were proposed methods to discriminate between the five most widely used dialects of Arabic, with an accuracy of 59.2%. Using classifiers as Naïve Bayes, Support Vector Machines, and Deep Neural Net. Different dimension vectors in the range 300-1600 were used. For instance, DNN with 5 hidden sigmoid layers, where the first layer used 23 critical band energies obtained from Mel scaled filter-bank. Since the Mel scale uses frames too, and frame size is about 20-40 *ms*, if a 40 *ms* frame is used, then there is 23×25, 575 values per second (estimated).

Chinese dialect recognition was reported in [7], identifying the 10 most wide spreads Chinese dialects. Recurrent Neural Nets and 40-dimensional Mel filter bank coefficients with a frame size of 25 *ms* (40 per second), were used. Estimating 40×40, 1600 coefficients per second. The dialect recognition achieved approximately 90% of accuracy.

In this section, was observed the common use of MFCC and Mel filter bank coefficients to perform ADI, an approach based on the auditory human system. At the difference, the study reported here was done using the LTAS, which models the vocal tract and a simple and reduced number of values per signal, detailed in the Experiments section.

3 Speech Features

The extracted features from audios are fundamental in the ASR performance, for this reason, it takes a great relevance to know and make mention about the benefits and usage of them.

3.1 Time-Domain Speech Features

Energy is a common and frequently used speech time-domain feature. The energy is obtained by the summary of the signal amplitude squares, and its measure unit is given in *Pa2 s* [8]. being *x(t)* the amplitude of the sound, given on Pascal's *Pa*, then the energy is defined in (1):

$$\int_{-\infty}^{\infty} x^2(t). \tag{1}$$

The fundamental frequency (F0) is another time domain feature, also known as pitch, is the vibratory frequency of the vocal cords. The number of air pressure oscillations per second determines the sound pitch. The pitch is sampled into many frames centered on equally spaced time.

3.2 Mel Spectrums

The Mel spectrums vector is obtained as a result of passing a signal through a filter bank. Every spectrum contained in the vector is the result of filtering the input spectrum through an individual filter, being the length of the vector equal to the number of filters. The triangular filters are centered over the frequency axis, distributed on the no lineal Mel scale; this scale was initially proposed by Stevens and Volkman in 1940 [9].

The filters banks emulate the perceptual critical band (Fig 1.), accentuating the low frequencies. The edges of the filters coincide with the axis of the adjacent filters. A common model for the relations of Mel frequencies and lineal scales is expressed in (2):

$$Mel\ Frequency = 2595 \times log_{10}\left(1 + \frac{Lineal\ Frequency}{700}\right). \tag{2}$$

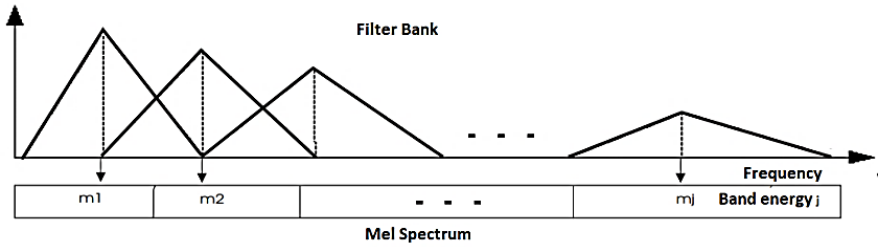


Fig. 1. Mel filter bank.

The audio codification scheme has been used in phonetic speech segmentation [10], also the Mel scale is the base for another widely used scheme in speech processing, the Mel Frequency Cepstral Coefficients (MFCC).

3.3 Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients is one of the most used sound representations in ASR and audio engineering. The process to obtain the MFCC starts applying a Hamming window, to avoid spectral distortions, fragmenting the continual signal in frames, where these frames typically are of 20 or 30 *ms*, because in this length the signal assumes to be stationary.

The logarithm of the energy in each filter is calculated and accumulates before to be applied the discrete cosine transform, producing the vector features of MFCC. The cepstral analysis denotes an unusual treatment of the signal in the frequency domain as if it were on the time domain [11]. The unit measure in the cepstral domain is in seconds, but they represent the spectral variations of frequencies.

Most of the audio processing studies have used this scheme codification: in emotion recognition, speech, and speakers. Studies for ADI, MFCC were used too in [2, 4].

3.4 Long Term Average Spectrum

The Long-Term Average Spectrum (LTAS) represents the power spectral density (PSD), expressed on *dB*/*Hz* relative to $2x10^{-5}$ *Pa* [8].

The spectrum is computed based on the PSD which is obtained from overlapped FFT series. Typically, the FFT length is 4096, step 2048. The segments of X are obtained applying the Hann-Window. The PSD average is smoothed with Gaussian function [12]. The average sound power between a range of time (t_1, t_2), is obtained by (3).

$$\int_0^F PSD(f)df = \frac{1}{T}\int_{t_1}^{t_2} |x(t)^2| \cdot dt. \tag{3}$$

The LTAS brings a representation for the location of the vocal tract resonances and the glottal source [13], which is more suitable to study the way that the people speak. On the other hand, the MFCC represents a human auditory system, the way that we perceive sounds.

Fig. 2. The study area of Mexican indigenous languages.

4 Database and Study Area

The audio samples were obtained from [14]. The audios have a length between 1 and 3 seconds, speaking phrases as: "come in", "good morning", "good afternoon", "good night", "welcome", "thank you for visiting us", "enjoy this meeting", "excuse me", "come back soon". Maya, Mixteco, Tseltal, Mixe, and Mazahua were spoken by female natives. Nahuatl, Zapoteco, Huichol, Tarahumara and Chichimeco were spoken by males' natives. The audios were recorded with a sampling frequency of 22100 Hz. The samples used were: 12 samples of Maya, Mixteco and Mixe respec-tively; 14 samples of Nahuatl, Tseltal, Huichol, Tarahumara, Mazahua and Chi-chimeco for each one; and 15 samples of Zapoteco. The phrases were spoken by five men and five women (one speaker per language). The indigenous languages used in this study, cover every zone of the Mexican territory. The geographical study area of Mexican indigenous languages is shown in Fig. **2**: Maya, Mixteco, Zapoteco, Mixe, Nahuatl, Tarahumara, Mazahua, Tseltal, Chichimeco, and Huichol.

5 Experiments

This study is based on the premise, every language adjusts the vocal tract in a particular way, to express its words and elocutions. This idea is supported by previous studies, for example, because "aerodynamic and anatomical properties of the vocal tract, influence in shape and patterning of the speech sounds" [14]. In this sense, the use of the LTAS fits as the main feature to recognize the patterns of languages, in our case, the Mexican Indigenous Languages.

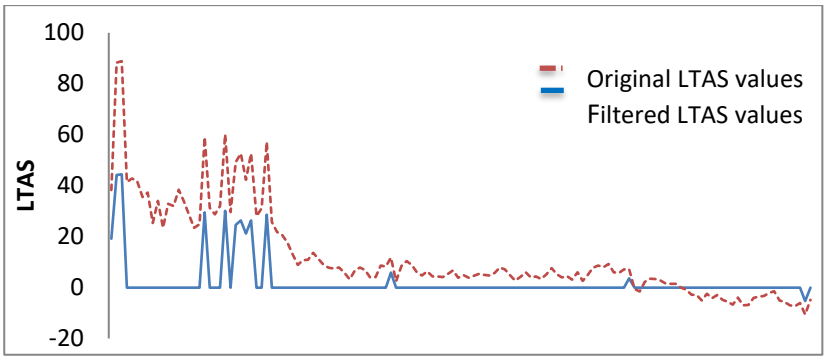


Fig. 3. LTAS values *vs* LTAS filtered values.

Table 1. Classifier's performance using five classes and *bin width* variations.

Classifier	80 Hz 276 LTAS values	100 Hz 221 LTAS values	140 Hz 158 LTAS values
J48	95.52	95.52	77.61
KNN3	91.04	89.55	88.05
KNN	89.55	94.02	94.02
MLP	94.02	95.52	94.02
RF	89.55	89.55	89.55
Random Tree	58.2	59.7	68.65
SMO	94.02	94.02	95.52
Naive Bayes	88.05	85.07	88.05
Bayes Net	79.1	79.1	77.61

The Long-Term Average Spectrum (LTAS) was extracted from speech signals using PRAAT software version 6.0.39 [8]. The classifiers are provided by the Waikato Environment for Knowledge Analysis version 3.8.5, as well as the attribute filter called *BestFirst* detailed in [15].

5.1 Experimental Classifiers Settings

The settings of the classifiers with the best performance, used in these experiments are presented. The same settings were used for five and 10 classes recognition.

The KNN was set using *Euclidean distance, no distance weighting, with 1 and 3 Neighbors* denoted as KNN and KNN3 respectively.

The MLP was set 0.3 on *learning rate, 0.2* on *momentum*, and *500 epochs*. One single hidden layer was used in the experiments. The *(classes + input size)/2* criteria, in order to determine the number of neurons in the hidden layer was used. So, for instance, using five classes and 140 *Hz* for LTAS extraction, there were 13 features and 5 classes (Table 2), in this case, the neurons in hidden layers were 9.

The SMO used a *multinomial logistic regression*, the complexity parameter was set in 1, an *epsilon of 1.0E-12* for round-off error.

For the Naïve Bayes classifier, an especial setting is not required.

Table 2. Classifier's performance with filtered values, using five classes and *bin width* variations.

Classifier	80 Hz 22 LTAS values	100 Hz 21 LTAS values	140 Hz 13 LTAS values
J48	95.52	95.52	80.59
KNN3	98.5	98.5	**98.5**
KNN	98.5	97.01	**98.5**
MLP	98.5	98.5	**100**
RF	98.5	98.5	**98.5**
Random Tree	83.58	74.62	76.11
SMO	98.5	98.5	**98.5**
Naive Bayes	97.01	95.5	97.01
Bayes Net	98.5	98.5	95.52

Table 3. Classifier's performance over 10 classes.

Classifier	140 Hz 158 LTAS values	140 Hz 18 LTAS filtered values
J48	77.77	82.22
KNN3	85.18	91.11
KNN	90.37	90.37
MLP	92.59	**94.07**
RF	90.37	93.33
Random Tree	62.22	80.00
SMO	91.85	**94.07**
Naive Bayes	87.4	**94.07**
Bayes Net	74.81	88.14

5.2 Five Languages Classification

In this experimental phase, the LTAS from speech audios were extracted and used as the only feature in the classification process. In the first instance, the classification for indigenous languages as Maya, Zapoteco, Mixteco, Tsoltil, and Nahuatl, was experimented with. The previous languages selection was aleatory.

The performance of different classifiers tested with five classes and three sampling frequencies is observed in Table 1. The bandwidth (frequency step) used were 80Hz, 100Hz and 140 Hz, obtaining 276, 221, and 158 spectrums per signal, respectively. The classifier's performance difference with the variations of spectrums number was minimal.

To enhance the classification performance, a *BestFirst* filter was applied to the LTAS values of the signal. This filter gets the relevant values of LTAS, those forming the peaks, discarding those values with light changes (flat lines). The peaks are plotted

on the blue line (down), the original values of LTAS are plotted on the red line (dotted), see Fig. 3.

Testing with filtered LTAS values, the Table 2 results were obtained, where a performance increase for all classifiers is observed, mainly Bayes Net followed by KNN. The *BestFirst* filter reduces the number of LTAS values (less than 10% respect the original number of them) down to 22, 21, and 13 LTAS values respectively. The MLP performance is notorious, classifying correctly the 100% of the samples, using only 13 values per signal, followed by the SMO, Random Forest with 98.5% of correct classification.

There is high performance for almost every one of the classifiers.

5.3 Ten Languages Classification

In this phase, the method was passed over a more rigorous test, to observe its behavior adding five classes over the existing.

The languages added were Huichol, Mixe, Tarahumara, Mazahua, Chichimeco. Starting on the previous results in Table 2, where high performance was using a 140 Hz *bin width*, and minimal values were required. The experiments with 10 classes were done only on this *bin width.* The results of this phase are shown in Table 3, where the best classifiers were MLP, SMO, and Naive Bayes with 94.07% of correct classification.

5.4 Overall Performance for Each Language

The languages were complicated to classify, we expose Table 3. The most recognizable language, using this set of classifiers was *Chichimeco* and the least recognizable was *Mazahua*. On the header of the columns, the total of samples and the letter assigned for each language is shown, so we obtained the performance per language with a simple division between the total value (bottom of the table) and the number of samples on the header, this las value multiplied by 10 (number of classifiers).

The idea of this simple overview is, to have an initial perspective about the complexity of recognition for each language.

6 Discussion

Similar research has been done to recognize different dialect languages. The discussion is about the methods used, although an exact comparative is difficult because of the different databases used.

A Malayan dialect recognition system was exposed in [4]. The Thrissur and Kozhikode were the two kinds of Malayan dialects used in the recognizer task. The MFCC, energy, and pitch were the features extracted, and then processed for the classifiers Artificial Neural Net, Support Vector Machine, and Naive Bayes, reaching

Table 4. Classifier's performance over 10 classes.

Classifier	a 12	b 12	c 14	d 14	e 15	f 14	g 12	h 14	i 14	j 14
J48	9	11	9	12	13	11	9	13	10	14
KNN3	11	10	11	14	14	13	9	14	13	14
KNN	10	10	13	13	12	13	9	14	14	14
MLP	10	12	14	14	13	13	11	13	13	14
RF	10	11	13	13	15	13	10	14	13	14
RT	9	10	11	11	12	12	8	10	11	14
SMO	11	11	13	14	14	13	10	14	13	14
NBAYES	10	11	14	14	13	13	11	13	14	14
BNET	10	10	12	11	13	12	10	14	13	14
Total	90	96	110	116	119	113	87	119	101	126
Performance	75.00	80.00	78.57	82.86	79.33	80.71	72.50	85.00	72.14	90.00

a=Maya,b= Mixteco, c=Nahuatl, d=Tseltal, e=Zapoteco, f=Huichol, g=Mixe, h=Tarahumara, i=Mazahua, j=Chichimeco

a recognition accuracy of 90.2%, 88.2%, and 84.1% respectively. Our experiments match exactly with the same kind of classifiers: MLP, SMO, and Naive Bayes; citing SMO is an algorithm used to train Support Vector Machines.

Arabic automatic dialect detection was presented in [3], reported results where they discriminated between the five most widely used dialects of Arabic: namely Egyptian, Gulf, Levantine, North African, and MSA, with an accuracy of 59.2%. Experimenting with classifiers like Naive Bayes and SVM (between others), obtaining the best performances with the second, achieved 100% of accuracy for binary classification between English vs Modern Standard Arabic. Highlight the use of SVM and Naive Bayes, as classifiers selected for dialect recognition.

Also, a dialect database that contains 10 types of Chinese dialects was used in [7], using a Recurrent Neural Network with an accuracy of 90.04%.

On the side features, the LTAS has been useful in the recognition, in topics as recognition of speaker [13], gender, age, including diseases [16], and forensic usage [17], in a general way, LTAS are speech encoding related with the voice quality [18]. In this study, LTAS brings a high performance for MIL recognition, using 158 numeric values per signal, and only 18 values using the *BestFisrt* filter.

7 Conclusion

In this work a method based on Long Term Average Spectrum was presented. The proposed method can be used to recognize the main Mexican Indigenous Languages.

Our studies have revealed that LTAS can be considered as a promising feature of dialect or language recognition because used a low number of parameters to describe the speech audios, which leads to short vector inputs for classifiers. The LTAS parameters can be optimized by considering the peaks of its values, with the aim of reducing the information from 158 values per signal to 18 values and consequently decrease the load on the classifiers. There are various works related to languages recognition, however these works use above 300 values per second as input on the classifiers, while our method uses below of 30 values per signal. As a result of our experiments, we found that MLP, SMO, and Naïve Bayes bring the best performance with an accuracy of

94.07%, in the 10 MIL recognition task. Besides, Networks, Support Vector Machines and Naive Bayes were found as common classifiers in state of art studies presented here and our experiments, to perform dialect recognition. As future work we will test on a most robust database, increasing the number of samples per language, adding more languages and we will test other types of classifiers as deep learning, hybrid approaches, filters, and new codification scheme of the speech.

References

1. Gray, S., Hansen, J.: An Integrated Approach to the Detection and Classification of Accents/Dialects for A Spoken Document Retrieval System. In: IEEE ASRU-2006, pp. 35–40 (2006)
2. Liu, G., Lei Y., Hansen, J.: Dialect Identification: Impact of Differences Between Read Versus Spontaneous Speech. In: EUSIPCO´10: European Signal Processing Conference, pp. 2003–2006 (2010)
3. Ali, A., Dehak, N., Cardinal, P.: Automatic Dialect Detection in Arabic Broadcast Speech. Proc. Interspeech, pp. 2934–2938 (2016)
4. Sunija, A.P., Rajisha, T., Riyas, K.: Comparative Study of Different Classifiers for Malayalam Dialect Recognition System. ScienceDirect, 24, pp. 1080–1088 (2016)
5. United Nations: United Nations (2008) https://www.un.org/en/events/iyl/multilingualism.shtml.
6. Martínez, C., Zempoalteca, A., Soancatl, V.: Computer Systems for Analysis of Nahuatl. Research in Computing Science, (47), pp. 11–16 (2012)
7. Pappu V., Pardalos, P. M.: High Dimensional Data Classification. Springer Optimization and Its Applications, pp. 34 (2013)
8. Othman, A., Hasan, T.: Impact of Dimensionality Reduction on the Accuracy of Data Classification. In: 3rd International Conference on Engineering Technology and its Applications (IICETA), pp. 128–133 (2020) doi: 10.1109/IICETA50496. 2020.9318955.
9. Hassan, M., Nath B., Bhuiya, M.: Bengali Phoneme Recognition: A New. In: 6th International Conference on Computer and Information Technology (2003)
10. Cheng, H., Ma X., Yugong, X.: A Study of Speech Feature Extraction Based on Manifold Learning. In: Journal of Physics: Conference Series, 1187(5), pp. 052021 (2019)
11. Zongze, R., Guofu Y., Shugong, X.: Two-stage Training for Chinese Dialect Recognition. Proc. Interspeech (2019)
12. Boersma P., Weenink, D.: Praat: Doing Phonetics by Computer [Computer program]. Version 6.1.50 (2020) url: http://www.praat.org/.
13. Volkmam J., Steven, S.: The Relation of Pitch to Frequency. American Journal of Psychology (1940)
14. Huerta, L., Huesca J., Contreras, J.: Speech Segmentation Algorithm based on Fuzzy Memberships. International Journal on Computer Science and Information Security, pp. 229–234 (2010)
15. Tukey, J., Bogert, P., Healy, M.: The Quefrency Analysis of Time Series for Echoes: Cepstrum, Psuedo-Autocovariance, Cross-Cepstrum and Saphe Cracking. Proceedings of the Symposium on Time Series Analysis (2006)

16. Hummersone, C.: Calculate the Long-Term Average Spectrum of a Signal (2021) https://github.com/IoSR-Surrey/MatlabToolbox.
17. Kinnunen, T., Hautmaki, V., Franti, P.: On the Use of Long-Term Average Spectrum in Automatic Speaker Recognition. In: International Synposium on Chinese Spoken Language Processing (ISCSLP) (2006)
18. Ohala, J.: The Origin of Sound Patterns in Vocal Tract Constraints. The Production of Speech, Springer, pp. 189–216 (1983)
19. Kohavi R., George, J.H.: Wrappers for Feature Subset Selection. Artificial Intelligence, 97 (1) pp. 273–324 (1997)
20. Cukier-Blaj, S., Camargo Z., Madureira, S.: Longterm Average Spectrum Loudness Variation in Speakers with Asthma, Paradoxical Vocal Fold Motion and Without Breathing Problems. In: Proceedings of the Fourth Conference on Speech Prosody, No. 9780616220, 44, pp. 41 (2008)
21. Rose, P.: Forensic Speaker Identification. CRC Press (2002)
22. Pittam, J.: Voice in Social Interaction: An interdisciplinary approach. London: SAGE Publications (1994)
23. Sunija, A., Rajisha T., Riyas, K.: Comparative Study of Different Classifiers for Malayalam Dialect. In: International Conference on Emerging Trends in Engineering, Science and Technology, 1088, pp. 1080 (2015)
24. Insituto Nacional de Lenguas Indígenas: Prontuarios de frases de cortersía de Lenguas Indígenas (2010) https://site.inali.gob.mx/Micrositios/Prontuarios/index.html.

ISSN 1870-4069

Runtime Prediction of Filter Unsupervised Feature Selection Methods

Teun van der Weij[1], Venustiano Soancatl-Aguilar[1], Saúl Solorio-Fernández[2]

[1] University of Groningen
Netherlands

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico

mailvanteun@gmail.com, v.soancatl.aguilar@rug.nl, ssolori1@asu.edu

Abstract. In recent years, the speed and quality of data analysis have been hindered by an increase in data size, an increase in data dimensionality, and the expensive task of data labeling. Much research has been conducted in the field of Unsupervised Feature Selection (UFS) to counteract this hindrance. Specifically, filter UFS methods are popular due to their simplicity and efficiency in counteracting performance problems in unlabeled data analysis. However, this popularity resulted in a great variety of filter UFS methods, each with their own advantages and disadvantages, making it hard to choose an appropriate method for a particular problem. Unfortunately, an inappropriate method choice can lead to a decrease in research or project quality, and it can render data analysis unfeasible due to time constraints. Importantly, terminating a method's analysis before completion means in most cases that no partial results are obtained either. Previous works on the evaluation of filter UFS methods focused mainly on assessing clustering and classification performance. Although very useful, choosing an appropriate method often requires knowledge about the method's runtime as well. In this paper, we study the runtimes of six popular filter UFS methods using synthetic and real-world datasets. Runtime prediction models were trained on 114 synthetic datasets and tested on 29 real-world datasets. The models showed good performance on four out of the six methods. Finally, we present general runtime guidelines for each method. To the best of our knowledge, this is the first paper that investigates methods' runtimes in this fashion.

Keywords: Feature selection, unsupervised feature selection, runtime prediction, execution time prediction, filter methods.

1 Introduction

Feature Selection, also known as Attribute or Variable Selection, concerns selecting a subset of the most relevant features from a dataset. Selecting the

most relevant features can be useful to achieve three main goals: improve prediction accuracy, faster predictions, and a better understanding of the phenomena that the data represent [1]. The importance of Feature Selection increases as the data grows in the number of objects and especially in the number of features, yielding all sorts of problems relating to the "curse of dimensionality" [2]. A high number of features requires more computational resources; if many features need to be analyzed, the speed of both the training and the predictions of a learning algorithm decrease. Furthermore, an excess of features reduces generalization capabilities and may negatively affect predicting performance [3]. Additionally, it is harder to understand the underlying mechanisms that the data describes when many irrelevant features clutter the relevant ones [1]. Feature Extraction is another closely related dimensionality reduction strategy with similar advantages to Feature Selection. However, Feature Extraction, which includes methods such as principal component analysis, unclearly transforms the relevant features, complicating the interpretation of the data [4, 5].

According to the availability of information in the data, datasets can be classified as completely labeled, partially labeled, or completely unlabeled. Fully labeled datasets require supervised methods, partially labeled require semi-supervised methods, and unlabeled datasets require unsupervised feature selection methods. The labels of objects in a dataset can be categorical, ordinal, or continuous [6]. These labels can, for example, describe what kind of animal the features represent, the place a bowler got in a bowling competition, or how happy a person says she is. Such labels are often not available, especially where high-dimensional data is present, such as in text mining, bioinformatics, and social media [3, 7]. Moreover, data labeling is expensive in both time and money because the labels need to be accurate, requiring qualified human labor [8]. Therefore, for unlabeled data, Unsupervised Feature Selection (UFS) methods are often used. Other important advantages of UFS methods include that these methods perform well when prior knowledge is unavailable and that they are less prone to overfitting [1]. UFS methods can be subdivided further into three categories: filter, wrapper, and hybrid methods [6, 9]. Filter methods are the fastest and most scalable methods, and they work independently of the classifier. Wrapper methods use a classifier or learning algorithm to evaluate a subset of features, which generally makes it much more computationally expensive. Moreover, wrapper methods need to be entirely retrained when a different classifier is used. Hybrid (embedded) methods aim to be a mixture of filter and wrapper methods, trying to balance the two approaches to get the benefits of both [9]. However, the integration of filter and wrapper approaches is generally insufficient, leading to lower classification performance [7].

Because of the advantages of the filter approach, many methods have been developed in this UFS category [6]. As a consequence, choosing an appropriate method for the task at hand can be time-consuming and difficult. A choice of a UFS method is important because of two reasons. First, one wants to obtain

the best possible insight from the data, which entails optimal understanding, optimal clustering and classification performance. Missed or uncertain insights might result in less fruitful research. Second, there is limited time available for all research. Depending on how limited the time is, a method must be selected that operates within these time constraints. Problems arise especially if it is unknown a priori how long these methods take to analyze a dataset. The runtime of a method analyzing a certain dataset could take several days, and larger datasets might take months or longer, even with fast hardware and software. The setup of a research project must be adjusted to the runtime of a method, which can mean sacrificing clustering and classification performance for runtime gains.

Solorio-Fernández *et al.* [7] saw the lack of and need for a comprehensive empirical study to enable users to choose an appropriate filter UFS method. The authors systematically analyzed the performance of 18 filter UFS methods, which were applied on 75 datasets. They also scored the methods based on clustering and classification performance. Consistent with the literature, the authors found that statistical-based methods generally had the worst clustering and classification performance, but they were the quickest methods. On the other hand, multivariate spectral/sparse-learning-based methods had significantly higher scores for clustering and classification, but they were substantially time-consuming. Furthermore, Solorio-Fernández *et al.* [7] reported the runtimes for every method ran on a dataset, which illustrated that some methods analyze a dataset in fractions of a second and some take more than seven days. As time constraints affect research quality, and Solorio-Fernández *et al.* [7] showed that there is a high variation in runtimes between methods, further research on method runtimes is needed to make a good a priori decision for a certain UFS method. Additionally, the number of objects also affects the runtime, as methods need to analyze more data. Moreover, datasets with millions and trillions of features already exist, for example, the MovieLens dataset (over 20 million objects) and the Google Books Ngram dataset (over 10 billion objects) [10, 11]. Furthermore, the feature sizes are very likely to further increase according to Bolón-Canedo *et al.* [12]. So, even if the runtimes shown by Solorio-Fernández *et al.* [7] are not problematic with maximums of 12960 objects and 2283 features, runtime problems are bound to arise with much bigger dataset sizes. Moreover, terminating a running method before completion means that no partial results can be obtained unless complicated changes to the methods are made.

To help users choose an appropriate method with respect to these runtimes issues, we investigate six popular UFS methods by predicting their runtimes based on the number of objects and features of a dataset. As a result, we contribute to the runtime knowledge of filter UFS methods by providing prediction models and general runtime guidelines. We examine the runtime performance of the six filter UFS methods available in the scikit-feature package created by Li *et al.* [3], which contains the implementation of some classical, relevant and more cited methods in the literature. We now present a brief overview of these six methods. The runtime prediction of the six methods will be discussed in the Methodology section.

The rest of the paper is organized as follows: Section 2 describes the filter UFS methods analyzed in this study. Section 3 describes the evaluation methodology used in our experiments. Section 4 reports the experimental results. Section 5 discusses the main insights and the general runtime observations derived from our experiments. Finally, Section 6 concludes the paper and provides some directions for future work.

2 Filter UFS Methods

According to Alelyani *et al.* [4] and Solorio-Fernández *et al.* [6, 7], filter UFS methods can be categorized into univariate and multivariate methods. We describe the key characteristics and the corresponding methods of both categories in Sections 2.1 and 2.2.

2.1 Univariate Methods

Univariate methods evaluate features separately and score a feature based on a certain criterion. Consequently, these methods do not have to solve the computationally expensive combinatorial optimization problem of selecting a feature subset [13]. Therefore, relevant features are found relatively quick. However, redundant features (those highly similar to other features) cannot be filtered out because features are not compared to other features, potentially leading to superfluous features in the selected set of features.

Low Variance: This relatively simple method ranks the features based on their variance [5, 14]. The underlying idea is that features that differ more in value are more relevant to uncover the underlying mechanisms in a dataset and to help differentiate instances between different classes [3]. Features with a low variance often do not carry much relevant information and do not differentiate between classes [7].

Laplacian Score: This method developed by He *et al.* [15] scores the importance of a feature by analyzing how well it preserves the locality. The Laplacian matrix is derived from the distance between data points, so the method can capture and analyze the local structure in the data space, which is often more important than the global structure [15]. Each feature is individually scored, and the top k features with the lowest Laplacian Score are selected [3].

SPEC: SPECtrum decomposition, created by Liu *et al.* [5], extends on the Laplacian Score method and is also built on a similar idea: "a feature that is consistent with the data manifold structure should assign similar values to instances that are near each other" [3]. This method ranks the features based on a consistency score calculated by three different criteria [5].

2.2 Multivariate Methods

A multivariate approach entails that subsets of the feature set are evaluated and scored together. Because of this, they are able to filter out both irrelevant and redundant features. However, selecting a (sub)optimal subset of a set of features is computationally expensive, as illustrated by the "subset sum" problem [16]. Even though the multivariate methods try to approach this problem efficiently, multivariate methods are generally much slower than univariate methods [4, 6, 7].

MCFS: The Multi-Cluster Feature Selection method developed by Cai *et al.* [13] selects features "that can cover the multi-cluster structure of the data where spectral analysis is used to measure the correlation between different features" [3]. As with previous methods, a Laplacian Matrix is constructed. The MCFS method takes the first k eigenvectors of this Laplacian matrix and calculates the importance of features by a regression model with l_1 norm regularization [13]. After solving the regression problems, a coefficient is computed where a high MCFS score means that the feature is important [3].

UDFS: Yang *et al.* [17] propose the Unsupervised Discriminative Feature Selection method, which uses both the discriminative information and feature correlations to select features [3, 7]. UDFS efficiently optimizes the $l_{2,1}$ norm regularized minimization problem with orthogonal constraint Yang *et al.* [17]. The UDFS method trains a linear classifier which obtains the highest local discriminative score for all features [3]. As with MCFS, the higher the score, the more important the feature is [17].

NDFS: The Nonnegative Discriminative Feature Selection method by Li *et al.* [18] first uses spectral analysis with nonnegative and orthogonal constraints to learn pseudo-class labels, and these labels are defined as nonnegative real values. Afterwards, the authors introduce a novel iterative algorithm to efficiently solve the $l_{2,1}$ norm regularization problem NDFS creates. The top k features that most relate to the pseudo-class labels are selected [18].

3 Methodology

The general protocol of our study is as follows: First, we measured the runtime of the six UFS methods on 114 synthetic datasets. Then, we used four different regression models to fit these runtimes based on the number of objects and features of a dataset. Subsequently, we evaluated the performance of these regression models using 10-fold cross-validation. Finally, we tested the best method per model on 29 real-world datasets.

First, we discuss the synthetic and the real-world datasets. Second, we elaborate on the four regression models and how we evaluate them. Lastly, we discuss the software and hardware specifications used in this study, including the parameter settings for the methods, the way of measuring runtimes, and details on the CPUs.

3.1 Datasets

To construct good models, good training data is needed. In our case, good training data means a considerable number of datasets with spread-out dimensions. Runtime patterns become more apparent for both humans and runtime prediction models when they evenly cover a larger part of the space of the number of objects and features. Therefore, to get training data that meets these conditions, we generated synthetic datasets SD with stepwise differing dimensions in both the number of objects and the number of features. These dimensions range from 500 objects and 500 features until, but not including, 10,000 objects and 10,000 features with a step size of 500. However, we constrained the maximum size per dataset to 10,000,000 data points. The number of data points is estimated as the product of the number of objects and the number of features. This means that datasets were only generated when this product is lower than or equal to the maximum size. Mathematically:

$$\begin{aligned} SD = \{D_{ij} \mid & i = 500, 1000, 1500, \ldots, 9500; \\ & j = 500, 1000, 1500, \ldots, 9500; \\ & i \times j \leq 10,000,000\}, \end{aligned} \tag{1}$$

where D_{ij} represents a dataset composed of i objects and j features. For example, a dataset of 8,000 objects and 4,000 features would not be generated, but a dataset with 4,000 objects and 2,500 features would have been generated. This resulted in 114 synthetic datasets in total.

The maximum size constraints for objects and features were chosen to keep the runtimes within the time and resource limits of the study. Furthermore, the datasets were generated with certain parameters, which are described in detail in Table 1. Note that the hypercube size value indicates the multiplication factor of the hypercube. This factor influences the spread of clusters/classes, which might make it easier for methods to converge. The effect of the hypercube size multiplication factor on the runtime is not part of our research. However, we varied this factor to improve the generalization of our runtime prediction models. Furthermore, the dataset generation parameter values were designed in a way to mimic real-world datasets. As a last note, all the default settings[3] were used for the parameters not described in the Table 1.

In addition to the synthetic datasets, we tested the runtime prediction models on real-world datasets to verify their performance. These were taken from the ASU Feature Selection Repository [3]. These datasets are all different types of data: text, face images, handwritten images, biological, amongst others. Further details of these real-world datasets can be found in Table 2. Both the synthetic and the real-world datasets were standardized to have a mean of 0 and a standard deviation of 1, as recommended by [14].

[3] Further general description and default settings of the parameters can be found on https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html.

Table 1. Synthetic dataset details.

Index	Objects	Features	Informative features	Redundant features	Classes	Clusters per class	Randomly labeled	Hypercube size
1	500	500	231	43	28	3	0.018	1.533
2	1000	500	221	174	19	1	0.435	0.615
3	1500	500	89	71	27	1	0.222	2.451
4	2000	500	8	171	3	3	0.256	2.365
5	2500	500	141	116	10	2	0.167	2.008
6	3000	500	189	144	27	1	0.225	0.872
7	3500	500	8	31	29	1	0.154	1.247
8	4000	500	112	163	3	2	0.258	0.987
9	4500	500	110	5	11	1	0.262	1.234
10	5000	500	83	89	18	3	0.416	0.917
11	5500	500	165	176	10	2	0.338	1.693
12	6000	500	158	221	18	3	0.137	2.032
13	6500	500	183	227	27	1	0.281	1.431
14	7000	500	75	23	3	2	0.189	1.319
15	7500	500	121	181	10	3	0.465	1.470
16	8000	500	179	196	3	2	0.049	1.185
17	8500	500	183	78	13	1	0.470	2.369
18	9000	500	122	78	6	3	0.497	1.009
19	9500	500	107	238	3	2	0.286	1.633
20	500	1000	58	291	3	3	0.278	1.384
21	1000	1000	333	140	8	3	0.371	1.213
22	1500	1000	98	361	16	2	0.467	2.117
23	2000	1000	85	128	23	2	0.437	0.744
24	2500	1000	357	406	8	1	0.375	1.575
25	3000	1000	262	46	16	1	0.323	0.886
26	3500	1000	481	188	13	1	0.020	0.946
27	4000	1000	478	344	8	1	0.088	0.747
28	4500	1000	38	31	2	2	0.310	0.808
29	5000	1000	373	466	22	3	0.065	1.299
30	5500	1000	125	334	30	1	0.040	0.736
31	6000	1000	319	196	22	3	0.232	0.675
32	6500	1000	387	472	12	1	0.348	0.608
33	7000	1000	429	294	10	3	0.097	2.313
34	7500	1000	69	200	18	1	0.059	0.737
35	8000	1000	195	181	16	2	0.045	0.546
36	8500	1000	184	353	8	3	0.013	0.837
37	9000	1000	370	113	10	2	0.420	2.442
38	9500	1000	322	485	21	3	0.059	1.728
39	500	1500	541	182	24	1	0.049	1.962
40	1000	1500	398	689	18	2	0.072	0.687
41	1500	1500	288	138	2	1	0.117	1.993
42	2000	1500	190	564	8	2	0.459	2.122
43	2500	1500	184	330	20	3	0.485	0.568
44	3000	1500	703	285	12	3	0.243	2.411
45	3500	1500	312	598	17	2	0.305	2.156

Table 1 continued from previous page.

Index	Objects	Features	Informative features	Redundant features	Number of classes	Clusters per class	Randomly labeled	Hypercube size
46	4000	1500	654	118	11	2	0.286	1.310
47	4500	1500	412	31	24	1	0.267	2.109
48	5000	1500	581	455	9	3	0.288	1.150
49	5500	1500	219	73	30	3	0.223	1.824
50	6000	1500	51	504	6	3	0.083	1.339
51	6500	1500	656	586	4	3	0.313	1.653
52	500	2000	546	537	9	1	0.108	0.502
53	1000	2000	379	388	25	1	0.303	1.181
54	1500	2000	217	776	10	1	0.075	0.620
55	2000	2000	726	721	25	2	0.439	1.016
56	2500	2000	829	711	3	3	0.451	0.669
57	3000	2000	485	258	22	3	0.477	1.955
58	3500	2000	778	506	29	1	0.248	2.167
59	4000	2000	413	495	19	3	0.467	1.704
60	4500	2000	238	526	19	1	0.200	1.840
61	5000	2000	238	514	29	2	0.234	1.230
62	500	2500	606	677	20	3	0.227	1.454
63	1000	2500	1056	1025	21	3	0.037	2.085
64	1500	2500	1116	990	8	3	0.433	1.157
65	2000	2500	623	1208	16	2	0.296	0.744
66	2500	2500	501	757	8	1	0.379	1.178
67	3000	2500	257	468	5	3	0.221	0.800
68	3500	2500	1191	840	10	1	0.499	2.377
69	4000	2500	399	281	9	1	0.470	2.127
70	500	3000	826	732	19	3	0.139	2.468
71	1000	3000	654	245	16	3	0.422	1.896
72	1500	3000	646	967	6	1	0.339	2.058
73	2000	3000	765	631	25	2	0.011	2.286
74	2500	3000	1048	668	19	1	0.236	1.702
75	3000	3000	1006	1140	16	3	0.261	2.370
76	500	3500	1445	41	10	3	0.309	2.364
77	1000	3500	26	1685	17	1	0.134	1.301
78	1500	3500	574	451	17	1	0.269	1.729
79	2000	3500	1047	280	8	1	0.051	1.645
80	2500	3500	162	407	21	2	0.192	1.441
81	500	4000	1445	1043	29	2	0.243	2.244
82	1000	4000	233	57	27	2	0.063	0.848
83	1500	4000	1474	222	11	3	0.453	1.054
84	2000	4000	1727	488	20	1	0.067	2.186
85	2500	4000	1387	1555	8	3	0.389	1.156
86	500	4500	73	1693	6	2	0.198	2.403
87	1000	4500	75	1074	3	2	0.364	1.640
88	1500	4500	699	1067	7	1	0.125	0.544
89	2000	4500	839	209	21	3	0.257	1.706
90	500	5000	1496	1588	2	2	0.058	0.789
91	1000	5000	2122	1717	4	3	0.499	0.654

Table 1 continued from previous page.

Index	Objects	Features	Informative features	Redundant features	Number of classes	Clusters per class	Randomly labeled	Hypercube size
92	1500	5000	2357	1980	4	2	0.164	0.972
93	2000	5000	384	1143	14	2	0.295	2.265
94	500	5500	875	1162	3	3	0.028	1.483
95	1000	5500	2526	77	3	1	0.217	1.124
96	1500	5500	1439	1493	23	2	0.475	1.664
97	500	6000	2028	879	28	2	0.497	1.459
98	1000	6000	933	2944	5	3	0.068	1.112
99	1500	6000	2248	2935	3	2	0.179	1.283
100	500	6500	480	636	27	2	0.255	1.326
101	1000	6500	453	3142	29	2	0.206	1.425
102	1500	6500	2920	2502	20	1	0.402	1.608
103	500	7000	1317	1030	25	3	0.068	1.294
104	1000	7000	45	589	22	2	0.060	1.066
105	500	7500	187	3440	18	3	0.037	1.815
106	1000	7500	2835	323	18	2	0.334	1.335
107	500	8000	266	690	21	2	0.470	1.573
108	1000	8000	2605	2334	14	1	0.434	0.574
109	500	8500	2630	2774	25	3	0.440	1.432
110	1000	8500	3451	3706	23	1	0.037	0.742
111	500	9000	661	2287	18	3	0.337	1.524
112	1000	9000	2150	3700	19	3	0.266	2.206
113	500	9500	4749	3582	14	3	0.482	1.719
114	1000	9500	4330	2531	19	2	0.127	1.931

3.2 Runtime Prediction Models

For our experiments, we use four models, namely simple linear regression, multiple linear regression, power regression, and exponential regression. We use these relatively simple models for the three following reasons:

1. The number of objects and the number of features of a dataset are the only two independent variables in our study. Therefore, models that sophistically select or independently weight variables are excessive.
2. We assume that users generally want to know a good runtime approximation of a method in terms of seconds, hours, days, months, or years. Therefore, a runtime approximation would be sufficient to help users in choosing an appropriate filter UFS method, which simpler models can give. How many seconds or days exactly it will take will likely be less relevant.
3. Precise runtime prediction of UFS methods running in different environments is out of the scope of this paper because it is expensive in both time and hardware resources. Runtime predictions vary based on the environment in which the methods are run due to different hardware arrangements and other tasks being run in that environment. Simple models can more easily use and adapt to their own environment.

Table 2. Real-world datasets.

Index	Name	Number of objects	Number of features	Number of classes
1	Isolet	1560	617	26
2	Yale	165	1024	15
3	OLR	400	1024	40
4	WarpAR10P	130	2400	10
5	Colon	62	2000	2
6	WarpPIE10P	201	2420	10
7	Lung	203	3312	5
8	COIL20	1440	1024	20
9	Lymphoma	96	4026	9
10	GLIOMA	50	4434	4
11	ALLAML	72	7129	2
12	Prostate-GE	102	5966	2
13	TOX-171	171	5748	4
14	Leukemia	72	7070	2
15	Nci9	60	9712	9
16	Carcinom	174	9182	11
17	Arcene	200	10000	2
18	Orlraws10P	100	10304	10
19	Pixraw10P	100	10000	10
20	RELATHE	1427	4322	2
21	PCMAC	1943	3289	2
22	BASEHOCK	1993	4862	2
23	CLL-SUB-111	111	11340	3
24	GLI-85	85	22283	2
25	SMK-CAN-187	187	19993	2
26	USPS	9298	256	10
27	Madelon	2600	500	2
28	Lung-small	73	325	7
29	Gisette	7000	5000	2

The three reasons above-mentioned lead us to use simple models that are easy to understand. From the simpler models, we selected those which were expected to perform well based on visual inspection of the runtimes. Additionally, we selected models based on the time complexity of a method, which were only available for the SPEC and MCFS methods.

In the following subsections, we describe the runtime prediction models used in our experiments. It is important to mention that for simple and multiple linear regression models, the objective functions were given, whereas we merely present the model's predicted runtime per sample for power regression and exponential linear regression. Moreover, for all models, we do not estimate a y-intercept because we expect the runtime to approach 0 when the number of objects and features approach 0. Using a y-intercept might result in overfitting on the training data and worse values for the remaining parameters.

Lastly, all the runtimes and the corresponding errors are presented and calculated in seconds throughout the whole paper.

Simple linear regression Let y_i and βx_i ($i = 1, ..., n$, with n denoting the number of samples) be the true and the fitted runtime, respectively. A sample consists of a dataset and the corresponding true runtime of one method. The criteria of fitting S is when the sum of squared residuals of the linear regression model is minimal, i.e.:

$$S(\beta) = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \sum_{i=1}^{n} (y_i - \beta x_i)^2, \tag{2}$$

where the coefficient β is the slope of the linear regression line tuned to minimize the residual sum of squares between the true and fitted runtimes, $\hat{\varepsilon}_i^2$ denotes the squared fit error of sample i, and x_i is defined as the product of the number of objects and features of the dataset of sample i (the total number of data points of a dataset).

Multiple linear regression Similar to Equation 2, the objective function S of the multiple linear regression model is defined as:

$$S(\beta_1, \beta_2) = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2, \tag{3}$$

where the coefficients β_1 and β_2 denote the corresponding slopes of the regression lines of x_{i1} and x_{i2} which are fitted to minimize the residual sum of squares, y_i and $\hat{\varepsilon}_i^2$ are defined the same as in Equation 2, and x_{i1} denotes the number of objects and x_{i2} represents the number of features of the sample dataset i.

Power regression The power regression model best models situations where the runtime equals the independent predictor variables raised to a power. As previously described, Equation 4 represents the fitted runtime of one sample. Consequently, the power regression model is defined as:

$$\hat{y} = \beta_1 x_1^{\beta_2} x_2^{\beta_3}, \tag{4}$$

where $\hat{y}$ denotes the fitted runtime, x_1 and x_2 denote the number of objects and features of a dataset, respectively, and the parameters β_1, β_2 and β_3 are minimized with the Trust Region Reflective algorithm [19].

This function is inspired by both visual inspection of the method runtimes and the time complexity of the SPEC and MCFS methods. It allows different effects of both object and feature numbers through β_2 and β_3, but they still influence each other because they are multiplied. In other words, the effect of the number of objects on the runtime is partially determined by the number of features and vice versa.

Exponential regression This model combines exponential and linear regression. As with Equation 4, Equation 5 represents the fitted runtime. The Exponential and Linear regression model is defined as:

$$\hat{y} = \beta_1 e^{(\beta_2 x_1)} \beta_3 x_2, \tag{5}$$

where $\hat{y}$ denotes the fitted runtime, x_1 can either be the number of objects or the number of features of a dataset and x_2 is the remaining option, and β_1, β_2 and β_3 are minimized with the Trust Region Reflective algorithm.

The design of this model is set up to let x_1 have a strong exponential influence on the runtime prediction and x_2 to have a secondary linear role. These roles are inspired by the plots presented in the results section and more detailed investigation of the effect on the runtime by only changing either objects or features sizes. From now on, we refer to the exponential regression model with x_1 denoting the number of objects and x_2 denoting the number of features as $\text{Exp}_{\text{objects}}$. Similarly, we refer to the exponential regression model with x_1 denoting the number of features and x_2 denoting the number of objects as $\text{Exp}_{\text{features}}$.

3.3 Model Evaluation Criteria

The performance of the runtime prediction models on the synthetic datasets is evaluated by 10-fold cross-validation, as recommended in the literature [20–22]. Additionally, we evaluate the performance of the best runtime prediction models tested on the real-world datasets. Both the cross-validation folds and the performance on the test set are scored with two error measures, namely Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE).

Let y_i and $\hat{y}_i$ ($i = 1, ..., n$, with n denoting the number of samples) be the true and the fitted runtime, respectively. The MAE and RMSE measures are defined as follows:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{6}$$

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{7}$$

where y and $\hat{y}$ denote all n true and fitted runtimes of one method, respectively. MAE represents the average prediction error and is not prone to outliers. In contrast, RMSE is sensitive to outliers because the error is squared initially [22, 23]. The combination of MAE and RMSE provides information on the origin of the error values. If the MAE and the RMSE are relatively close together, the prediction errors are relatively even in size across the test sets. If the error measure values lie relatively far apart, it means that some runtime prediction errors were much bigger than others.

3.4 Software and Hardware Specifications

For our experiments, we use the six UFS methods available on the ASU Feature Selection Repository [3], and we adapted[4] them to make them suitable for the newer versions of Python and the machine learning package scikit-learn [24]. As mentioned before, these methods belong to the most used and most cited methods in the literature, and they are a good representation of the variety among filter UFS methods (see the taxonomy in Solorio-Fernández *et al.* [6]). We used the default parameter settings given in the ASU Feature Selection Repository for the six methods [3]. Additionally, each method selects 100 features from the given dataset, apart from the Low Variance method. The Low Variance method selects the features with a variance higher than $p(1-p)$, where p is the variance threshold. We used the default setting of $p = 0.1$. The runtime of a method on a dataset is determined by the time it took the methods to return the selected features from the time the data was passed to the method. The timing was done with the time function of the time module in Python.

The experiment was run on the Peregrine compute cluster of the University of Groningen, which made it possible to run the methods on the quantity and dimensions of the datasets as previously described. Moreover, a compute cluster provides the additional advantage of more reliable runtimes because the system is less cluttered by other tasks demanding a machine's resources, such as software updates and antivirus programs. To run something on a compute cluster, one must create a job script to specify what needs to be done. The univariate methods were used to analyze all datasets in one job, which in our case means that the methods analyzed all the datasets consecutively on the same node and CPU. For the slower multivariate methods, we submitted many individual jobs where the methods analyzed one to three datasets at a time, depending on expected and observed runtime. This division of datasets onto many jobs allowed us to get the runtime data in a reasonable period of time.

All these jobs were run on Intel Xeon E5 2680v3 CPUs @ 2.5 GHz. Because not all cores are in use all the time, the clock speed could be as high as 3.3 GHz. Furthermore, the jobs were run with 8 GB of reserved memory. The only exceptions to this are the jobs for UDFS and NDFS, where they analyzed the GLI-85 and the SMK-CAN-187 datasets for which they could use up to 64 GB of memory. 8 GB memory resulted in memory shortage errors. The OS of the Peregrine high-performance cluster when running the experiment was CentOS Linux, release 7.8.2003. The versions of the Python packages are available in the requirements text file on the GitHub page of this paper.

4 Experimental Results

In this section, we present the evaluation of the runtime prediction models. First, we describe the figures and tables. Afterwards, the methods are discussed in the

[4] Further details on the specific requirements and adaptations can be found on the GitHub repository of this paper https://github.com/FeatureSelection/UFS

same order as in Section 2, i.e., Low Variance, Laplacian Score, SPEC, MCFS, UDFS, and NDFS. Each method is discussed based on the observed patterns on Figures 1, 2 and 3, which helped in the design of the models. Afterwards, we assessed the models by using the error criteria presented in Table 3. Finally, we present the final runtime prediction model for each method based on this analysis and test them on the real-world datasets visible in Figure 4 and Tables 4 and 5.

4.1 Description of Figures and Tables

The plots with the runtimes of each method applied on the synthetic datasets are shown in Figures 1, 2 and 3. The position of the points on the vertical axis represents the time a method took to analyze a dataset. Additionally, there are legends encoded by color which represent the category of the points in the figure. Figures 1, 2, and 3 differ in what the x-axis represents. Figure 1 has the number of objects of the analyzed dataset on the x-axis. The number of features of the same dataset is represented by the size of the point with the principle of the bigger the point, the higher the number of features. In Figure 2, the x-axis represents the number of features and the point size represents the number of objects. Lastly, in Figure 3 the x-axis represents the number of data points of a dataset (the product of objects and features).

For Figures 1 and 2 it is important to realize that as the x-axis increases in value, the number of runtime points gradually decreases. This is the effect of creating the synthetic datasets with a maximum of 10,000,000 data points per dataset. As a result, in Figures 1 and 2 the runtime does not seem to increase as quickly as it perhaps should. To combat this potentially misleading representation, the third variable (the number of features or the number of objects) is represented by the size of the data point, as described in the previous paragraph. Figures 1, 2 and 3 also include the fitted runtimes of the best runtime prediction model for each method. These fitted runtimes are only plotted in the figure with the most runtime determining factor as value on the x-axis. The best runtime prediction models will be discussed in Section 4.2. Figure 4 shows the true runtimes of each method applied on the real-world datasets and the predicted runtime by the best model(s). For Low Variance, UDFS and NDFS, more information was needed to pick the best model, so the predictions of those models are both plotted in Figure 4. Notice that the x-axis differs per plot; the independent variable with the most impact on the runtime is presented on the x-axis.

Table 3 shows the mean scores of the 10-fold cross-validation procedure for each combination of method and prediction model for the synthetic datasets. Notice that all but the UDFS and NDFS methods are rounded to three decimals. UDFS and NDFS scores are integers because decimals are unnecessary with numbers over a thousand in our case. Furthermore, the nonlinear model has two scores, one where x_1 = number of objects ($\text{Exp}_{\text{objects}}$) and one where x_1 = number of features ($\text{Exp}_{\text{features}}$) as defined in Equation 5. Table 4 complements Figure 4 by representing the performance of the runtime prediction models. The

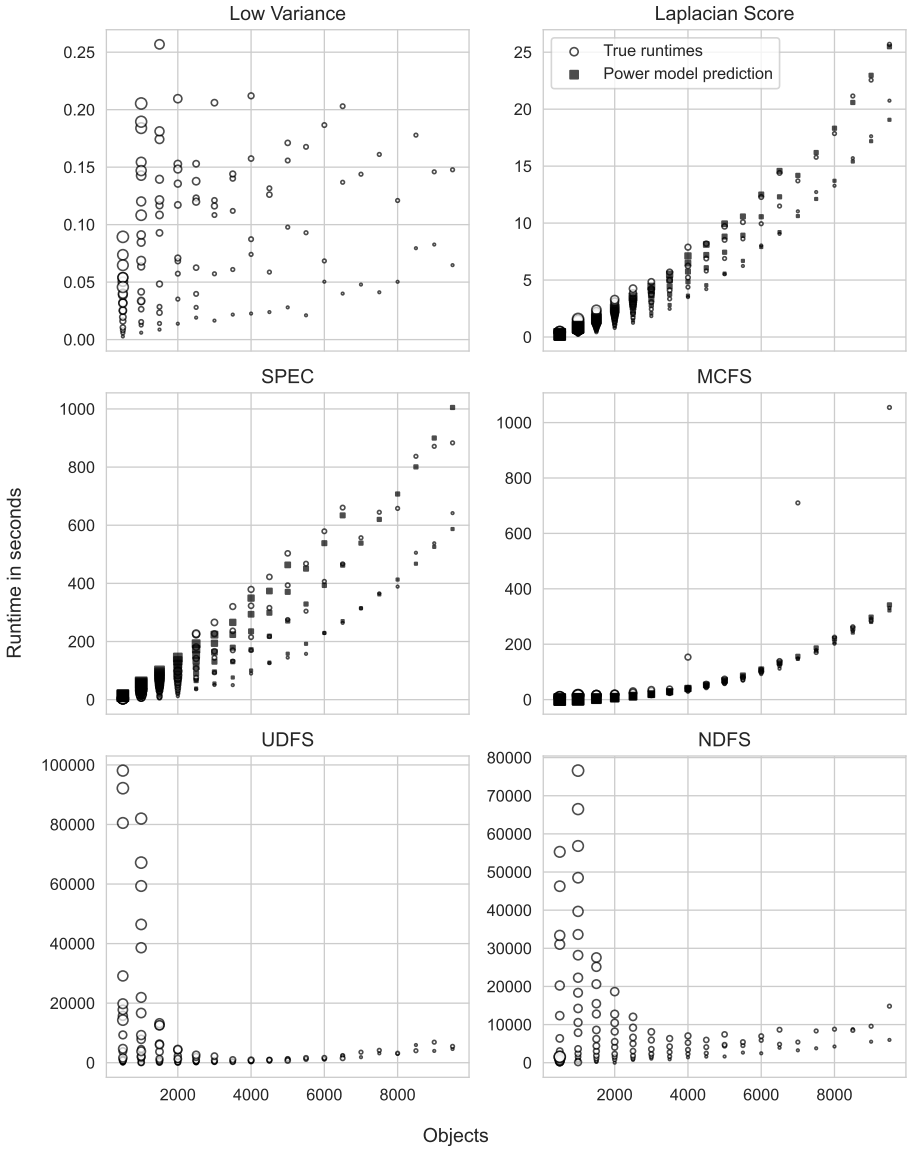


Fig. 1. Runtime results in seconds with objects as x-axis for each method. The legend in one plot describes every plot in the figure. The model that best predicts a method's runtime is also shown. For more details, see Section 4.1.

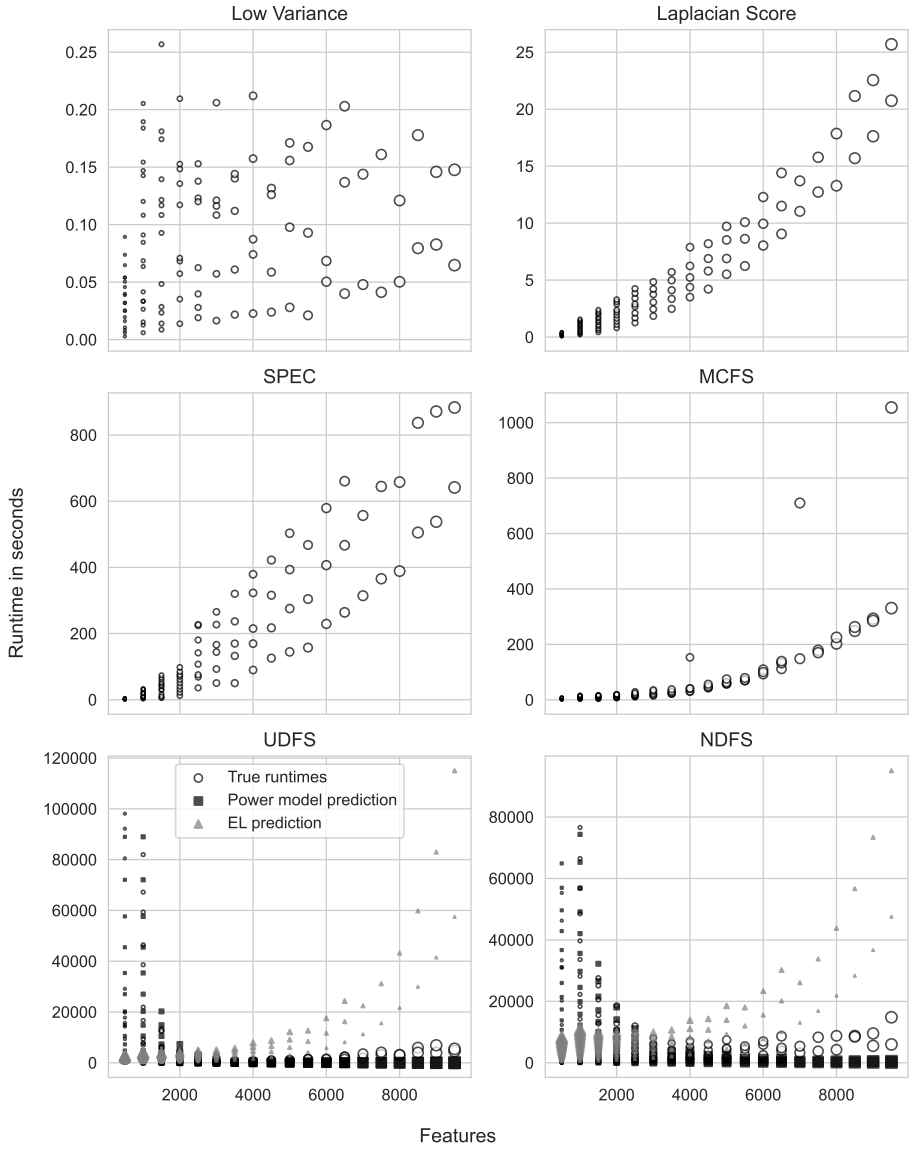


Fig. 2. Runtime results in seconds with features as x-axis for each method. The legend in one plot describes every plot in the figure. The models that best predict a method's runtime are also shown. For more details, see Section 4.1.

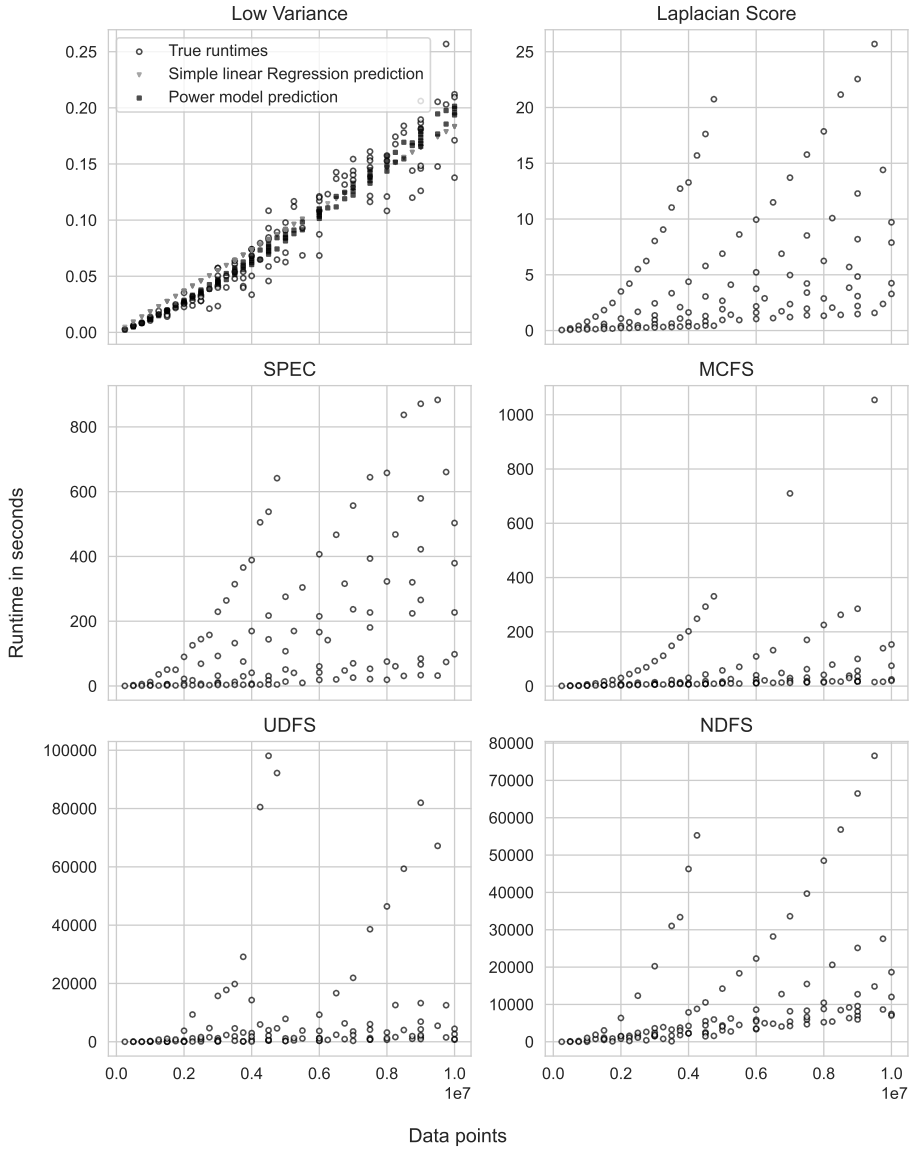


Fig. 3. Runtime results in seconds with objects as x-axis for each method. The legend in one plot describes every plot in the figure. The models that best predict a method's runtime are also shown. For more details, see Section 4.1.

Table 3. Synthetic data error scores for the prediction models in seconds.

Method	Mean Runtime	Measure	Simple linear	Multiple linear	Power	Exponential x_1 =objects	Exponential x_1 =features
Low Variance	0.087	MAE	0.017	0.041	0.009	0.032	0.033
		RMSE	0.021	0.049	0.011	0.038	0.039
Laplacian Score	4.424	MAE	3.625	1.510	0.187	1.295	1.453
		RMSE	4.822	1.960	0.229	1.559	1.761
SPEC	163.221	MAE	139.523	77.251	11.192	56.749	70.658
		RMSE	183.213	94.154	14.095	67.328	83.991
MCFS	43.223	MAE	42.632	26.558	3.940	8.583	15.372
		RMSE	59.515	33.614	4.834	12.782	18.696
UDFS	7997	MAE	10103	9439	1794	6092	2639
		RMSE	15413	12960	2567	7800	4039
NDFS	10244	MAE	8556	5508	1585	4508	1867
		RMSE	12146	7302	2169	5434	2814

table shows the true and predicted runtimes with a high number of objects or features. This allows for a better and zoomed-in representation of the other runtime predictions. Table 5 is similar to Table 3, but it describes the best runtime prediction models applied on the real-world datasets instead of all the runtime prediction models applied on the synthetic datasets.

Notice that these error scores have major flaws in capturing the performance of the runtime prediction models on the real-world test data. This is mostly due to the greatly varying dataset dimensions. There are many smaller datasets and some bigger ones, as visible in Table 2. This distorts the means, and as a result, the error measures do not accurately represent the performance of a model. A better insight can be gained by scrutinizing the plots.

4.2 Runtime Analysis

In the following, we provide a brief description and analysis of the best runtime prediction models for each UFS method applied to the datasets of Tables 1 and 2.

Low Variance For the Low Variance method, the influence of the number of objects and features on the runtime is best represented by having the number of data points on the x-axis, as shown in Figure 3. We see a mostly linear relationship with some variation.

The error scores in Table 3 partially confirm this idea. The MAE and RMSE scores for the simple linear regression model are 0.017 and 0.021, respectively, but the power model has even lower error scores with 0.009 and 0.011. Although the power model has better scores, a linear model might generalize better to datasets with differing dimensions, whereas the power model can be overfitted on the training data.

This is confirmed by testing both models on the real-world data visible in Figure 4. Therefore, the model that best predicts the runtimes of the Low

Variance method in our experiment is:

$$\hat{y} = 1.833 \times 10^{-8} x, \tag{8}$$

where $\hat{y}$ is the predicted runtime and x is the number of data points of a dataset. Figure 4 and the corresponding Table 4 show that the runtimes are predicted well, with the connotation that the predictors are consistently slightly higher than the true runtimes.

Laplacian Score The Laplacian Score plot in Figure 1 shows a nonlinear relation between the number of objects and the runtime. Additionally, it shows that the number of features affects the runtime too, because bigger points have higher runtimes (see Figures 1 and 2). Therefore, we suspect that the power model will perform the best. Our error measures support this, since the error scores for the power model are nearly seven times smaller than the nearest competitor (see Table 3). Therefore, the model that best predicts the runtime of the Laplacian Score method is:

$$\hat{y} = 3.335 \times 10^{-8} x_1^{1.918} x_2^{0.418}, \tag{9}$$

where $\hat{y}$ again denotes the predicted runtime, x_1 denotes the number of objects, and x_2 the number of features of a dataset. Figure 4 and Table 4 show that the power regression model accurately predicts the runtimes of the Laplacian Score method.

SPEC The SPEC plot in Figure 1 shows a similar pattern as the Laplacian Score plot. We see a nonlinear relation between the number of objects and runtimes, and we see that features have a clear influence on the runtime as well. However, there seems to be more variation in runtimes, partially caused by a relatively bigger influence of features on the runtime than with the Laplacian Score method. We hypothesize that the power model will perform the best among the models. This hypothesis is supported by the error measures for which the power model has some five times lower scores than the nearest model, as shown in Table 3. Therefore, the resulting prediction model for the SPEC method is:

$$\hat{y} = 3.507 \times 10^{-8} x_1^{2.044} x_2^{0.776}, \tag{10}$$

with the same definitions as with the Laplacian Score method. Similar to the Laplacian Score method, Figure 4 and Table 4 show that the power regression model accurately predicts the method's runtimes.

MCFS We see in Figure 1 that the MCFS method shows a strong nonlinear relation between objects and runtimes, with a seemingly minimal role of feature numbers (also see Figure 2). Therefore, we hypothesize that the power model suits the data best. Moreover, we see three outliers. We ran the experiment for a second time, and the same three outliers remained. To improve generalization,

we removed these three outliers to fit the models and calculate the MAE and RMSE scores.

As we can observe in Table 3 the error scores provide support for our hypothesis. The scores for the power model are 3.940 and 4.834 for MAE and RMSE, respectively, where the closest contender is the $\text{Exp}_{\text{objects}}$ model with MAE and RMSE scores of 8.583 and 12.782. Thus, the final prediction model for the MCFS method is:

$$\hat{y} = 1.183 \times 10^{-8} x_1^{2.564} x_2^{0.087}, \tag{11}$$

again with $\hat{y}$ denoting the predicted runtime, x_1 denoting the number of objects, and x_2 denoting the number of features of a dataset. In Figure 4 we observe that the runtimes of the MCFS are not predicted well for smaller object sizes, because the model underestimates the effect of the number of features on the runtime. However, the runtimes are predicted with more accuracy when the number of objects increases (see especially Table 4).

UDFS In the UDFS plot in Figure 2 we see a nonlinear relationship between feature numbers and runtimes. The number of objects seems to have a relatively minor effect on the runtime. Lastly, there seems to be relatively much variation as feature sizes increase. These observations lead us to suspect that the power model performs best and that the $\text{Exp}_{\text{features}}$ will perform well too. This suspicion is made more certain by the error measures in Table 3. The scores for the power model are 1794 and 2567 for MAE and RMSE, and the scores for the $\text{Exp}_{\text{features}}$ model are 2639 and 4039. These scores do fit the power model better, but, to make a better substantiated choice, we plotted both models in Figure 4 and show the extra information in Table 4.

In Figure 4 we see that for datasets with relatively low number of features, the runtime predictions are quite accurate. However, when the number of features increases, the prediction quality rapidly decreases. In Table 4 we see that for the datasets with around 20,000 features, the runtime predictions are much larger than the true runtimes. Predictions are especially inaccurate for the $\text{Exp}_{\text{features}}$ model. Therefore, the runtime prediction model for the UDFS method is:

$$\hat{y} = 2.676 \times 10^{-11} x_1^{0.000542} x_2^{3.902}, \tag{12}$$

where the same definitions apply as with the last three models. Notice that in contrast to the last three runtime prediction models, β_1 is much lower than β_2, indicating that the number of features determines the runtime much more than the number of objects. This is in line with the plots we see in Figures 1, 2, 3 and 4.

NDFS Similar to the UDFS method, the NDFS plot in Figure 2 shows a relatively strong nonlinear relationship between feature numbers and runtimes, with a relatively small effect of the number of objects on the runtime. NDFS

Table 4. Measured and predicted runtimes with a high number of objects or features for the real-world datasets (Fig 4).

Method	Dataset	Objects	Features	True Runtime	Model	Predicted Runtime	Methods	Predicted Runtime
Low Variance	Gisette	7000	5000	0.422	Simple linear	0.642	Power	0.914
Laplacian Score	Gisette	7000	5000	41.008	Power	27.808	-	-
Laplacian Score	USPS	9298	256	17.335	Power	13.839	-	-
SPEC	Gisette	7000	5000	2092.370	Power	1882.422	-	-
SPEC	USPS	9298	256	414.148	Power	335.067	-	-
MCFS	Gisette	7000	5000	263.291	Power	179.319	-	-
MCFS	USPS	9298	256	332.579	Power	286.716	-	-
UDFS	SMK-CAN-187	187	19993	157422.626	Power	1,624,588	$\text{Exp}_{\text{features}}$	20,309,840
UDFS	GLI-85	85	22283	484025.000	Power	2,479,277	$\text{Exp}_{\text{features}}$	41,182,695
NDFS	SMK-CAN-187	187	19993	19170.060	Power	322,335	$\text{Exp}_{\text{features}}$	4,223,331
NDFS	GLI-85	85	22283	19163.363	Power	358,746	$\text{Exp}_{\text{features}}$	6,274,883

seems to have less variance than UDFS. Again, we hypothesize that the power model performs best and that the $\text{Exp}_{\text{features}}$ model will perform well too.

Similar to MCFS, for NDFS, some outliers were produced. We did not consider these outliers for fitting the models and calculating the error scores, as we did with the outliers in MCFS. Notice that these outliers are not too problematic because they finish much quicker than regular predicted runtimes. The number of outliers is noteworthy, however, with 7 out of 114 runtimes classified as outlier. Running the experiment for a second time resulted in the same outliers.

The error measures in Table 3 share the observation of the power model and the $\text{Exp}_{\text{features}}$ model performing best. The scores for the power model are 1585 and 2169 for MAE and RMSE, and the scores for the $\text{Exp}_{\text{features}}$ model are 1867 and 2814. These scores do fit the power model better, but to further investigate, we plotted both results of the runtime prediction models in Figure 4 and present the extra information in Table 4.

Figure 4 and Table 4 show that both the power model and the $\text{Exp}_{\text{features}}$ model do not predict the runtimes well. Table 4 shows that while the true runtimes for two high dimensional datasets are around 19,000 seconds, whereas the power predicts around 322,000 and 358,000 seconds respectively, and the $\text{Exp}_{\text{features}}$ model predicts around 4,000,000 and 6,000,000, respectively. Therefore, the power model predicts the runtimes best for the NDFS method, with the model being:

$$\hat{y} = 4.890 \times 10^{-6} \times x_1^{0.196} x_2^{2.412} \tag{13}$$

where the same definitions and remarks apply as with the UDFS method.

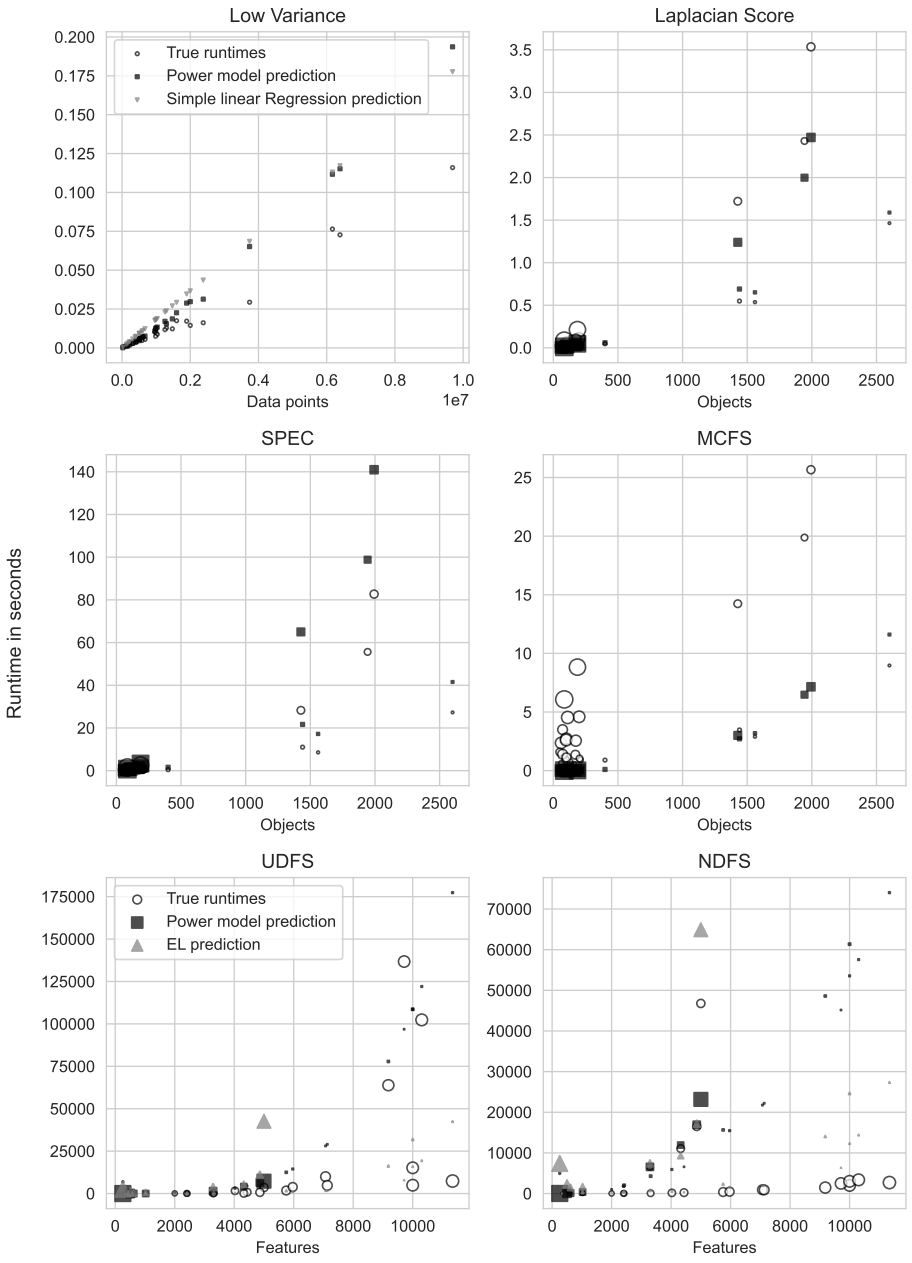


Fig. 4. Runtime results in seconds with objects as x-axis for each method. The top legend describes the top four plots, and the bottom legend the bottom two. The models that best predict a method's runtime are shown as well. For more details, see Section 4.1.

Table 5. The error scores for the best prediction models in seconds.

Method	Mean Runtime	Best model	Error	Score	Contender Model	Error	Scores
Low Variance	0.018	Simple linear	MAE RMSE	0.027 0.009	Power	MAE RMSE	0.021 0.002
Laplacian Score	0.473	Power	MAE RMSE	0.682 6.480	-	MAE RMSE	- -
SPEC	9.656	Power	MAE RMSE	16.360 2055.675	-	MAE RMSE	- -
MCFS	4.487	Power	MAE RMSE	7.771 340.712	-	MAE RMSE	- -
UDFS	10219	Power	MAE RMSE	137506 2.122×10^{11}	$\text{Exp}_{\text{features}}$	MAE RMSE	2118501.956 7.155×10^{13}
NDFS	3950	Power	MAE RMSE	37653.890 7.822×10^{9}	$\text{Exp}_{\text{features}}$	MAE RMSE	349876.063 1.800×10^{12}

5 Discussion

In this section, we first discuss the performance of the runtime prediction models, followed by the section on general runtime observations. Then, we address related literature. Finally, we discuss some limitations of our research.

5.1 Performance of Runtime Prediction Models

The runtime prediction models performed better for the Low Variance, Laplacian Score, SPEC, and MCFS methods than for the UDFS and NDFS methods. The low error scores and near predictions presented in Tables 3, 4 and 5 and in Figures 1, 2, 3 and 4 illustrated the quality of prediction models for the four methods. Especially relevant are Figure 4 and the associated Table 4, which indicate how well the models trained on the synthetic datasets generalize to the real-world datasets. Clearly, the runtime predictions are not perfectly accurate. However, this was not the goal of this paper. More importantly, these predictions can give users a priori runtime information of a method. Additionally, these predictions expose the influence of the number of objects and features on the runtime, which will be discussed in Section 5.2.

The performance of the runtime prediction models for the UDFS and NDFS methods is worse than for the other four methods. Although the power model predicts the runtimes of the synthetic datasets well, the prediction performance on some real-world datasets decreases, particularly when the number of features is high. In general, our four regression models fail to capture the influence of the number of objects and features of a dataset on the runtime for both methods. More specifically, we believe that the runtime prediction models for UDFS and NDFS have low performance because of the following observation. When the number of objects increases and the number of features is fixed, we see a reasonably clear nonlinear relationship between an increase in the number of

objects and the increase in the runtime. It is likely that our models would be able to capture this interaction. However, this interaction depends on the number of features too. For example, the difference in runtime between dataset A (2000, 1000) and dataset B (2500, 1000) is different between the runtime differences in dataset C (2000, 2000) and dataset D (2500, 2000). Although the increase of object numbers between A & B are the same as between C & D, the runtime will not increase with the same amount (even while ignoring runtime differences caused by the environment itself). Of course, we see a similar phenomenon when the roles of objects and features are switched. Our models likely failed to capture these important and more complex interactions. This deficiency becomes evident when the number of objects and the number of features of a dataset greatly differ from the synthetic training set, which is the case for some real-world datasets. The prediction errors we see in Table 4 are likely the result of this deficiency.

5.2 Runtime Observations

The runtime prediction results are generally in line with the theory on univariate and multivariate methods, which would suggest that the number of features is less important for univariate methods than for multivariate methods. Whereas univariate methods analyze the features separately, multivariate methods aim to find an optimal subset of features. This requires solving the "subset sum" problem that gets increasingly difficult as the whole set of features increases in size. Indeed, we observed that the number of features affects the runtimes most for the UDFS and NDFS methods, which is not the case for the univariate methods. An exception to this is the multivariate MCFS method, where the number of objects has a stronger influence on the runtime than the number of features. Furthermore, we clearly see differences in the order of runtimes.

The Low Variance and the Laplacian Score methods are the quickest methods and can be employed to analyze large datasets. The Laplacian Score method is particularly quick in analyzing highly dimensional datasets where the number of objects is relatively low and the number of features is much higher. On the other hand, the SPEC and MCFS methods are clearly slower than the previous two methods, but in our experiment, they operated in the order of seconds and minutes for larger datasets. Consequently, they can typically execute within most time constraints of research projects. The SPEC method is best applied when the number of objects of a dataset is not too high, while the number of features can be large. The same holds for the MCFS method, although the number of features has a relatively higher effect on the runtimes than with the SPEC method.

Lastly, the UDFS and NDFS methods are the slowest methods. Although our models have low performance in predicting their runtimes, the runtime data is still useful to provide runtime guidelines. In our environment, both UDFS and NDFS take a couple of days to complete when the number of features exceeds 10,000. We have seen that the number of features has a strong linear effect on the runtime; thus, the runtimes of these methods might easily take weeks and months when the number of features exceeds 10,000. Do notice that this is based only on datasets with around 100 objects. It is unknown what runtimes to expect

with different object sizes. Nonetheless, long runtimes should be considered when these methods are applied on high-dimensional datasets.

5.3 Related Literature

As far as we know, this experiment is unique in the field of Unsupervised Feature Selection. However, method runtimes have been examined in other research fields, especially in optimizing job scheduling in high-performance clusters [22, 25]. Random forest regression models often perform well in these fields because they can take many independent variables into account. However, we focus on two independent variables in our study. Moreover, random forest has difficulties with extrapolating from training data, which in our case means that the runtime predictions will not be accurate when datasets have considerably different dimensions than the dimensions of synthetic training datasets [22]. Unfortunately, we cannot train our models on all the potential dataset dimensions users might have because of these two reasons; therefore, the random forest algorithm is not suitable for this study.

On the other hand, we used a benchmarking approach in this paper, but another possible approach was to focus even more on a mathematical analysis of methods. The Big O notation is often used to represent the time and space complexity of a method [26]. The Big O notation describes the general computational operations that a method performs, but it ignores important factors for runtime analysis, such as the machine, the programming language, and the compiler the method runs in. Moreover, the time complexity is only available for two out of the six methods, namely the SPEC and MCFS methods. Still, time complexity analysis such as in Cai *et al.* [13] and Zhao & Liu [27] can be useful to examine the interaction of object and feature numbers on the runtime. In this study, we did use the time complexity of the SPEC and the MCFS method to create runtime prediction models.

Finally, it is noteworthy that the runtime prediction for the UDFS and NDFS methods can probably best be improved by analyzing the time complexity of the methods by examining the original paper where the methods are presented (Yang *et al.* [17] and Li *et al.* [18], respectively). However, the time complexity has not been provided by the authors themselves and analyzing their time complexity was out of the scope of this project.

5.4 Limitations

Our experiment is also subject to some limitations. Most notably, as we stated in Section 3.2, generalizing the runtime findings from our experiment environment to a user's environment brings along complications. Although this was not part of our research objective, it does interfere with the extrapolation of our findings to the environment of a user. On the other hand, a missed insight that is relevant to our research goal are the outliers of the MCFS and NDFS methods, visible in Figures 1 and 2. It is unknown to us why exactly these outliers exist and the effect it has on the clustering and classification performance. For the NDFS method,

the outliers are less likely to be problematic, as the method analyzes much faster than expected. The outliers for the MCFS methods can be problematic as the runtime is multiple factors higher than expected.

Additionally, the models and their parameters do not necessarily represent an optimal fit. It could be that other simple models, such as polynomial ones, fit the runtime data better. As for the model parameters, Trust Region Reflective algorithm, used to optimize the parameters in the nonlinear models, does not converge to a global minimum [19]. Consequently, it could be the case that with different initial parameter guesses, the models would have better-fitted parameters. Furthermore, the effect of the hypercube size multiplication factor, used in generating the synthetic datasets, on the runtime is unknown. It could be that a higher factor speeds up some methods when, for example, pseudo-class labels are learned with spectral analysis in the NDFS method.

Lastly, the effects of method parameters, such as the number of selected features and the number of nearest neighbours, used in the MCFS method, for example, are left unstudied.

6 Concluding Remarks and Future Work

In this study, we have presented runtime prediction models for six relevant and classical filter UFS methods of the state-of-the-art. The runtime prediction models and the general guidelines for each of the six methods can be particularly useful for professionals and practitioners in this research field. Moreover, our results, in line with previous work on the evaluation of filter UFS methods [7], could be useful to assist users in choosing an appropriate method for a particular problem. From the results presented in the previous sections and the analysis performed, we contribute to the runtime knowledge of filter UFS methods by providing some insights and guidelines:

- The Low Variance, Laplacian Score, SPEC, and MCFS methods are much faster than the UDFS and NDFS methods.
- The Low Variance method is the quickest method, which can be applied on large datasets in most cases without runtime problems. The runtime is best determined by the number of data points of a dataset.
- The Laplacian Score method can be applied on large datasets as well, and is especially efficient in analyzing high-dimensional datasets.
- The SPEC method is considerably slower than the previous two methods, but its runtimes will still often be manageable. The SPEC method is best at analyzing high-dimensional datasets. However, a large number of objects increase the runtime considerably.
- The MCFS method is the fastest multivariate method, even faster than the univariate SPEC method. Surprisingly, the runtimes of the MCFS method are most influenced by the number of objects, meaning that it handles high-dimensional datasets well.

- The UDFS method is substantially slower than the previous four methods. UDFS should be used carefully with large datasets, especially when datasets have many features (roughly $> 10,000$).
- The NDFS method has similar runtimes and should be used with the same care as the UDFS method.

Finally, future work of this research includes the following:

- Analyzing the time complexity of the UDFS and NDFS methods and building corresponding runtime prediction models to improve the current predictions.
- Performing experiments on datasets with different shapes to generalize and improve the general performance of the runtime prediction models.
- Investigating runtime prediction in other environments could improve the usability of our research. Future research similar to, or in combination with, a paper by Sidnev [28] might be fruitful.
- A similar study like ours can be used to examine the runtimes of other filter methods, and it can be extended to wrapper and hybrid (embedded) UFS methods as well.

Acknowledgments. We want to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high-performance computing cluster.

References

1. Iguyon, I. & Elisseeff, A. *An introduction to variable and feature selection* 2003.
2. Bellman, R. Combinatorial processes and dynamic programming. https://apps.dtic.mil/sti/pdfs/AD0606844.pdf (1958).
3. Li, J. *et al. Feature selection: A data perspective* 2017. arXiv: 1601.07996. https://doi.org/10.1145/3136625.
4. Alelyani, S., Tang, J. & Liu, H. *Feature selection for clustering: a review* (eds Aggarwal, C. C. & Reddy, C. K.) 29–60. ISBN: 9781728128474 (Taylor & Francis, 2014).
5. Liu, L., Kang, J., Yu, J. & Wang, Z. A comparative study on unsupervised feature selection methods for text clustering. *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, IEEE NLP-KE'05* **2005,** 597–601 (2005).
6. Solorio-Fernández, S., Carrasco-Ochoa, J. A. & Martínez-Trinidad, J. F. A review of unsupervised feature selection methods. *Artificial Intelligence Review* **53,** 907–948. ISSN: 15737462. https://doi.org/10.1007/s10462-019-09682-y (2020).
7. Solorio-Fernández, S., Carrasco-Ochoa, J. A. & Martínez-Trinidad, J. F. A systematic evaluation of filter Unsupervised Feature Selection methods. *Expert Systems with Applications* **162.** ISSN: 09574174 (2020).

8. Fredriksson, T., Mattos, D. I., Bosch, J. & Olsson, H. H. *Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12562 LNCS** (Springer Science and Business Media Deutschland GmbH, 2020), 202–216. ISBN: 9783030641474. https://doi.org/10.1007/978-3-030-64148-1%7B%5C_%7D13.
9. Dong, G. & Liu, H. *Feature engineering for machine learning and data analytics* 192–194. ISBN: 9781138744387 (Taylor & Francis, 2018).
10. Goldberg, Y. & Orwant, J. A Dataset of Syntactic-Ngrams over Time from a Very Large Corpus of English Books. **SEM 2013 - 2nd Joint Conference on Lexical and Computational Semantics* **1,** 241–247 (2013).
11. Harper, F. M. & Konstan, J. A. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* **5,** 1–19. ISSN: 21606463 (2015).
12. Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. *Artificial Intelligence: Foundations, Theory, and Algorithms Feature Selection for High-Dimensional Data - Chapter 3: A Critical Review of Feature Selection Methods* 1–10. ISBN: 9783319218571. www.springer.com/series/ (Springer, 2016).
13. Cai, D., Zhang, C. & He, X. Unsupervised feature selection for Multi-Cluster data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 333–342 (2010).
14. Dy, J. G. & Brodley, C. E. *Feature Selection for Unsupervised Learning* tech. rep. (2004), 845–889.
15. He, X., Cai, D. & Niyogi, P. Laplacian Score for feature selection. *Advances in Neural Information Processing Systems,* 507–514. ISSN: 10495258 (2005).
16. John, G. H., Kohavi, R. & Pfleger, K. Irrelevant Features and the Subset Selection Problem. *Machine Learning Proceedings 1994,* 121–129 (1994).
17. Yang, Y., Shen, H. T., Ma, Z., Huang, Z. & Zhou, X. l2,1-Norm regularized discriminative feature selection for unsupervised learning. *IJCAI International Joint Conference on Artificial Intelligence,* 1589–1594. ISSN: 10450823 (2011).
18. Li, Z., Yang, Y., Liu, J., Zhou, X. & Lu, H. Unsupervised feature selection using nonnegative spectral analysis. *Proceedings of the National Conference on Artificial Intelligence* **2,** 1026–1032 (2012).
19. Branch, M. A., Coleman, T. F. & Li, Y. Subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal of Scientific Computing* **21,** 1–23. ISSN: 10648275 (1999).
20. Hastie, T., Tibshirani, R. & Friedman, J. The Elements of Statistical Learning. *The Mathematical Intelligencer* **27,** 241–247. ISSN: 03436993 (2017).
21. Arlot, S. & Celisse, A. A survey of cross-validation procedures for model selection. *Statistics Surveys* **4,** 40–79. ISSN: 19357516. arXiv: 0907.4728 (2010).

22. Hutter, F., Xu, L., Hoos, H. H. & Leyton-Brown, K. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence* **206,** 79–111. ISSN: 00043702. www.elsevier.com/locate/artint (2014).
23. Chai, T. & Draxler, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* **7,** 1247–1250. ISSN: 19919603 (2014).
24. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Machine Learning Research* **12,** 2825–2830 (2011).
25. McKenna, R., Herbein, S., Moody, A., Gamblin, T. & Taufer, M. Machine learning predictions of runtime and IO traffic on high-end clusters. *Proceedings - IEEE International Conference on Cluster Computing, ICCC,* 255–258. ISSN: 15525244 (2016).
26. Russel, S. & Norvig, P. *Artificial Intelligence A Modern Approach (4th Edition)* **9.** ISBN: 9788578110796 (Pearson, 2020).
27. Zhao, Z. & Liu, H. Spectral feature selection for supervised and unsupervised learning. *ACM International Conference Proceeding Series* **227,** 1151–1157 (2007).
28. Sidnev, A. A. Runtime prediction on new architectures. *ACM International Conference Proceeding Series* **23-24-Octo,** 1–6 (2014).

ISSN 1870-4069

Geoinformatics Based Mapping and Assessment of Flood Risk in Periyar River Basin, South India: A Pairwise Weighted Approach

Sreelakshmi A.R.[1], Sumith Satheendran S.[2], Latha P. [3]

[1] College of Engineering Trivandrum,
Department of Civil Engineering,
Environmental Engineering Division,
India

[2] Dr. R. Satheesh Centre for Remote Sensing and GIS,
School of Environmental Sciences,
India

Mahatma Gandhi University Kottayam, Kerala,
India

ar.sreelakshmi95@gmail.com

Abstract. Flood risk assessment is important to be carried out in areas susceptible to flood so that proper urban planning can be done accordingly. Periyar is the longest river in the state of Kerala which is known as the "Lifeline of Kerala". The Periyar river basin experienced an increasing number of flood events over years and highlighted the need for flood-related studies. GIS-based multi-criteria evaluation was performed to determine the flood risk assessment in the present study. The identification of flood risk zones was done by the Analytical Hierarchy Process and pairwise weighted approach. The major parameters used in this study were Flood vulnerability, Flow accumulation, Topographical Wetness Index and Rainfall. From this study, 20.78 % area of the Periyar River basin was identified with high flood risk, 71.35% with moderate flood risk and 7.86% with low flood risk. The flood risk assessment will be helpful to figure out the mitigation measures and reduce the chances of future flooding damages.

Keywords: Flood risk, topographic wetness index (TWI), pairwise weightage.

1 Introduction

Flood is a natural disaster caused due to climate change, rapid urbanization, and population growth. Floods happen in different places in varying magnitude and their influence on people and the environment differs. Kerala is highly vulnerable to natural disasters and floods are the most common natural hazard in the state. Nearly 14.5% of the state's land area is prone to floods. The mega-floods of 2018 have impacted the economy and society of Kerala in many different ways. Floods and subsequent landslides caused extensive damages to houses, roads, railways, bridges, power supplies, communication networks, infrastructures, agriculture, and also the lives and livelihoods of millions of people in the state were affected. In the Periyar river basin,

the increasing number of flood events over the years highlighted the need for flood-related studies.

1.1 State of Art

Flood is the most common natural hazard in the world and has caused serious loss of life and property (Xiao, Y. et al., 2017). Assessment of flood-prone areas is of great importance for watershed management and reduction of potential loss of life and property. Multi-Criteria Analysis (MCA) incorporating Geographic Information System (GIS), Fuzzy Analytic Hierarchy Process (AHP) and spatial Ordered Weighted Averaging (OWA) method are popular for flood hazard assessment. In risk assessment, the factors associated with geographical, hydrological and flood-resistant characteristics of the basin are selected as evaluation criteria.

For natural phenomena such as floods, G.I.S proves to be taking an effective role to make assessments. Multi-criteria decision-making can be adopted after considering different favoring factors that impact the flood. Analytical Hierarchy Process (AHP) has been performed in many research works.

The terms ‘‘flood risk’’ and ‘‘flood hazard’’ are often used synonymously (A. Shalikovskiy and K. Kurganovich, 2016). Recently two main approaches to risk assessment have been developed in the world. The first approach was based on an understanding that risk can be defined as the combination of hazard and vulnerability. Another approach relies on priority conception, according to which the qualitative methods of flood risk assessment are used.

For endangering natural hazards like floods, GIS can play a critical role with its visualization capacity to make results from assessments (Mohamad et al., 2019). For natural phenomena such as floods, G.I.S proves to be taking an effective role to make assessments. Multi-criteria decision-making can be adopted after considering different favoring factors that impact the flood. Analytical Hierarchy Process (AHP) has been performed in many research works. Assessment of flood risks, especially extreme floods, is required for any location that is faced with recurrent flooding events for proper implementation of proactive measures (Zulkhairi, MD et al., 2019).

Susceptibility factors such as elevation, slope, flow accumulation, topographic wetness index (TWI), drainage density, etc., are very useful while assessment and flood risk mapping through AHP (Kumar and Agrawal, 2020). Basan et al., (2019), consider the flood-affected LULC and its area extend determines the economic loss the flood caused. The flood risk map after the AHP process can be used to identify the high risk, medium risk, and low-risk areas.

Wondim, Y. K. (2016), conducted a study on Awash River, Ethiopia considered the application of GIS and Remote Sensing techniques on flood hazard/risk management process. Flood causative factors such as slope, elevation, drainage density, soil type and land cover were developed in the GIS environment.

Assessing risk from flood using composite hazard and vulnerability index has been a widely recognized tool that acts as an important element for the formulation of policies aiming at flood risk reduction (Ghosh and Kumar, 2019). To assess risk due to flooding using Analytical Hierarchical Processes, incorporating flood hazard elements and vulnerability indicators in the geographical information system environment. Flood hazard map has been prepared using selected morphological and hydro-meteorological

elements whereas the vulnerability map has been produced using demographic, socio-economic and infrastructural elements.

Salam, A. and Rubeena, T.A, (2020), examines the temporal correlation of flood damages that occurred in Kerala and analyzed the peripheral land-use change of Periyar River for a stretch of 25 km flowing through Aluva and Paravur Taluks of Ernakulam District, Kerala. There is a huge expansion of the built-up area within 10 years around the river, which is highly prone to flood.

Flood risk assessment is important to be carried out in areas susceptible to flood so that proper urban planning can be done accordingly. This can also be used in forecasting and warning of floods. Due to the development of GIS technology, this can be used in spatial policy and space management systems, including the determination of ways to develop and use flood risk land in a planned way.

The present study aimed at flood risk assessment using Analytical Hierarchy Process (AHP), a pairwise weighted approach.

1.2 Study Area

Periyar (Big river) is the longest river in the state of Kerala with the largest discharge potential. Periyar is known as the "Lifeline of Kerala". Periyar River originates from the forest land of Sivagiri hills, 80 km South of Devikulam (2438 MSL) (CWC, 2018). Periyar River is about 244 km in length with a catchment area of 5398 km2. Periyar river mouth is Vembanad Lake and the Arabian Sea. The study area extends between 9º 0'0" - 10º 30'0" North latitudes, 76º 0'0" - 77º 30'0" East longitudes as shown in Fig 1.

1.3 Data Collection

The study area was delineated from Survey of India toposheets numbered 58B3, 58B7, 58B11, 58B15, 58F3, 58F7, 58B4, 58B8, 58B12, 58B16, 58F4, 58F8, 58C1, 58C5, 58C13, 58G11, 58G5, 58C14, 58G2, 58G6, 58C14, 58G2, 58G2, 58G6, 58G3 and 58G7. The remotely sensed satellite imageries of Sentinel-2 (10m spatial resolution) on February 7 and 12, 2021, were used to interpret the present land use/land cover of the study area. ALOS DEM, a 12.5 m elevation data from the Japanese Exploration Agency, was used to develop elevation, slope, aspect, curvature and stream density layer. Road networks were developed after vectorization from Google Earth. Soil data for the study area was developed from the Benchmark soils of Kerala (Soils of Kerala), Demographic data was taken from the Census of India 2011 report. Rainfall data for the study was sourced from the Global Precipitation Measurement Mission of NASA.

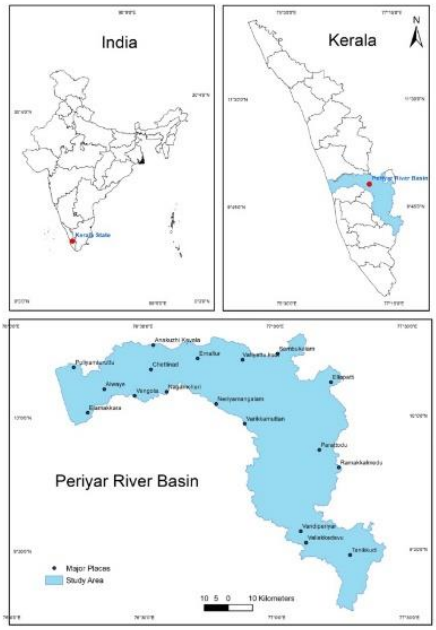


Fig. 1. Index map of the study area: Periyar river basin.

2 Methodology

In the present study, flood risk assessment through the Analytical Hierarchy Process, of the Periyar river basin was performed. The flood vulnerability assessment was done firstly by weighted overlay analysis and then this result was incorporated in the determination of flood risk. Different layers were selected depending on the topography of the area, referring to pieces of literature and also considering expert opinion.

2.1 Flood Vulnerability Assessment Using Weighted Overlay Analysis

Influencing thematic layers in raster form for determining the flood vulnerability includes: Elevation, Slope, Aspect, Curvature, Population density, Land use Land cover, Soil type, Stream density, Distance from road networks, and Distance from water body as shown in Fig 2.

In the flood vulnerability assessment, ten different influencing thematic layers were considered mainly Elevation, Slope, Aspect, Curvature, Population density, Land use Land cover, Soil type, Stream density, Distance from road networks, and Distance from the water body.

Elevation. The elevation is an important factor that affects flooding; low-level areas are more prone to flooding and were given more weightage. Elevation was categorized into different classes and weights were assigned. In the study area, the elevation ranged from 0 to 2705 m above the mean sea level.

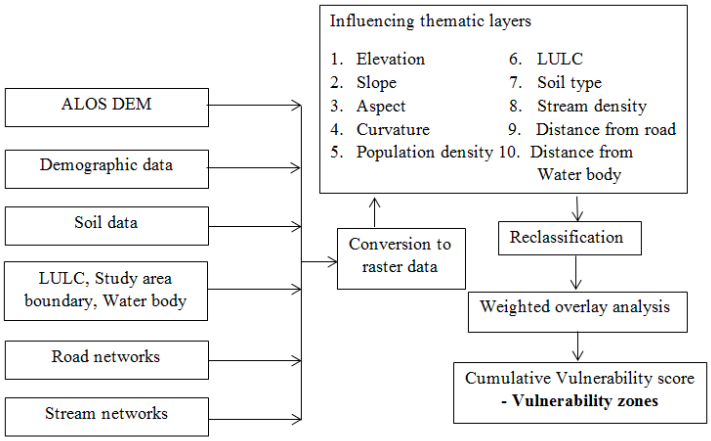


Fig. 2. Flow chart for flood vulnerability assessment.

Slope. Low slope areas are more prone to flooding. The slope is directly related to the runoff velocity, vertical percolation, and stream power in the downstream portion. The value ranged from 0 to 76.7872. The slope was derived from DEM using the slope function in the Arc toolbox.

Aspect. Aspect is the direction of the slope. The aspect was derived from DEM using the Aspect function in the Arc toolbox. Periyar River originates from the Sivagiri hills of Western Ghats and is located along the North-East border of the Idukki district in Kerala state. The mouth of the Periyar River is the Arabian Sea and Vembanad Lake. So the river flows from the North East to the West direction. Thus, priority was assigned in such a way that along the entire Periyar River basin, the West direction was given more preference. In the North West direction, there is flat land that is very vulnerable to flooding and thus highest priority was given accordingly.

Curvature. The curvature of Earth considers the topological characteristics of the study area. The curvature layer was developed from DEM and the values range between 1.40093 e^11 to 1.27303e^11. As the values changes from negative, zero to positive, curvature changes from concave, flat, and convex.

Population density. Taluk-wise population data of Ernakulam and Idukki district from Census 2011, after population forecasting using geometric method were used to develop population density layer. There were nearly 50 locations of major Census towns taken throughout the study area. Total population, male, female and children population in the age of 0 to 6 years were attributed. The population density layer was developed from the density tool of the Arc toolbox. The population growth rate was calculated using the formula:

$$Population\ growth\ rate\ = \left(Increase\ in \frac{population}{Original} population\right) \times 100. \quad (1)$$

Forecasted population of 2021:

$$〚P2021 = P2011(1+r/100)〛^{n}, \quad (2)$$

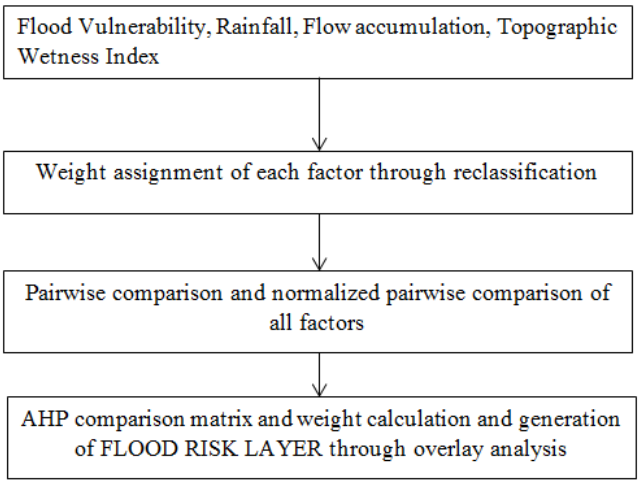


Fig. 3. Flow chart for flood risk assessment.

where, P2021 is forecasted population of 2021,
P2011 is population as per 2011 Census,
n is number of decades, here n = 1.

The population density was considered per km2 basis using density function in Arc toolbox. The values range between 0 to 16807.3.

Land use land cover (LULC). Land use/land cover affects the infiltration rate. Urban areas have low infiltration compared to forests. Land use pattern thus affects the flooding. An urbanized area generates more runoff and is thus more flood-prone. Iso-cluster unsupervised classification was performed to identify the different land use/land cover classes. For unsupervised classification, satellite image scenes of Sentinel 2 dated 07/2/2021 (2 scenes), 12/2/2021 (2 scenes) of the US Geological Survey were used. After unsupervised classification, different classes obtained were water body, paddy field, barren land and built-ups, vegetation, mixed crop, rocky land, and agricultural land.

Soil type. The nature of the soil in the particular area is very important while considering the infiltration characteristics. If infiltration is more there is less probability the water gets inundated and thus less flooding occurs. So, the soil type was taken district-wise into account in the present study and then finally considered as the entire Periyar river basin after topology correction. Soil data was sourced from Benchmark soils of Kerala (Soils of Kerala). The predominant soil types found were Alluvium, Black cotton soil, Acid saline soil, Red soil, Forest soil, Laterite soil, and Hill soil.

Stream density. The density of the stream is an important characteristic to be considered. As the stream density increases, there is more chance of easy drainage as thus less probability of flooding.

Distance from the road. As the distance from the road increases, there are fewer chances the area gets affected by the flood. In urban areas, the sewer systems are laid

along the roads. Improper management of the drainages across the urban areas causes congestions and trouble for natural drainages and causes urban flooding.

Distance from the water body. As the distance from the water body decreases, the chance of flooding increases. After LULC classification, the water body was extracted separately and then converted to vector format using Raster to Polygon function in Arc GIS. The distance from the river was then calculated by using the Euclidean distance tool in the Arc toolbox. The distance varied from 0 to 17.516 km from the river to different points in the study area.

For generating a flood vulnerability map, layers were prepared for all the parameters, and reclassification was performed to provide weights. Table 1 shows the weights for different parameters in a hierarchical manner.

The reclassified influencing layers were assigned relative weights depending upon the influence in flooding. The order of priority is fixed after referring to the literatures:

$$Cumulative\ vulnerability = F1 * W1 + F2 * W2 + \ldots\ldots + Fn * Wn, \tag{3}$$

where, F is contributing factor reclassed with flood vulnerability score,

W is the relative weight applied to each contributing factor:

$$\begin{gathered} Cumulative\ vulnerability = 15*Elevation + 10*Slope + 10*Aspect + \\ 10*Curvature + 5*Population\ density + 10*Stream\ density + 10*LULC + 10*Soil \\ type + 5*Distance\ from\ road + 15*Distance\ from\ water\ body. \end{gathered} \tag{4}$$

The cumulative flood vulnerability value ranged from 2 to 6. The cumulative vulnerability output thus generated, equally sized and then reclassified into four zones viz., Low, Moderate, High and Very highly vulnerable as shown in Table 2.

2.2 Flood Risk Mapping through Analytical Hierarchy Process – A Pairwise Weighted Approach

Analytical Hierarchy Process (AHP) is a powerful method of decision making considering all the influencing factors, assigning weights depending upon the priority and influence of that particular parameter on the matter considered.

Many parameters that help to evaluate the flood situation were considered as different layers. Flood vulnerability, Flow accumulation, Topographical wetness index (TWI), and Rainfall are considered in this study. Each parameter is divided into different classes and weights were assigned after pairwise comparison as given in Table 4. Normalized pairwise classification, as shown in Table 5, ensured all the weights are between 0 and 1 so that the relative importance is scaled and easily understandable. A consistency check was also performed to ensure that the pairwise comparison aided the right result. Finally, the flood risk layer was prepared through the weighted overlay technique. The cumulative flood risk obtained was reclassified into three zones i.e., High, Moderate, Low.

Table 1. Influencing thematic layers and Weights.

Influencing Thematic Layer	Weights
Elevation	15
Slope	10
Aspect	10
Curvature	10
Population density	5
Stream density	10
LULC	10
Soil type	10
Distance from the road	5
Distance from the water body	15

Table 2. Cumulative Flood Vulnerability.

Cumulative Flood Vulnerability	
2	Very high
2 – 3	High
3 – 4	Moderate
4 - 6	Low

Four influencing thematic layers used for flood risk assessment include: Flood vulnerability, Flow accumulation, Topographic Wetness Index and Rainfall.

Flood vulnerability. The flood vulnerability ranged from 2 to 6 were reclassified into different zones namely, very high (2), High (2–3), Moderate (3–4), and Low (4–6) as shown in Fig. 4. This is an important layer used for flood risk assessment that considers all the topographical features, land use classes, soil characteristics, and distance from the water body into consideration.

Flow accumulation. This is a crucial parameter for flood mapping. The hydrology tool of Arc Toolbox was used to generate this layer. More amount of accumulated flow naturally leads to increased runoff in a low elevated area and thus contributing to more flooding (Kumar and Agarwal, 2020). This layer was also classified into four classes and assigned weights as shown in Fig. 4. The value ranges from 0 to 4346195, with the highest score value at the highest flow accumulation.

TWI (Topographical Wetness Index). TWI is defined as, $TWI = \ln(\alpha/\tan\beta)$, where α is flow accumulation and β is the slope in radian. There is a strong relationship between TWI and geomorphology as well as the hydrographic position of an area. This index evaluation is capable of predicting saturated land surfaces that reserves the potential of producing overland flow (Kumar and Agarwal, 2020). TWI layer is useful to demonstrate flood inundated regions due to produced overflow at the time of high precipitation intensity. TWI was calculated using the raster calculator of ArcGIS. The value range of TWI varies from 1.033213 - 24.718002 as shown in Fig. 4.

Rainfall. Average annual rainfall from 2015 to 2020 was considered for the study. Monthly data source was from the Global Precipitation Mission of NASA.

Annual rainfall was calculated from monthly data of each year using a raster calculator in Arc GIS 10.8. Finally, the average annual rainfall for all five years was computed using a raster calculator. Average annual rainfall for five years varied from

Table 3. Influencing thematic layers, their classes, score value.

Influencing Thematic Layer	Score Value
Flood Vulnerability	
2	1
2 - 3	2
3 - 4	3
4 – 6	4
Flow accumulation	
0 - 10000	7
10000 - 25000	6
25000 - 50000	5
50000 - 75000	4
75000 - 100000	3
100000 - 125000	2
125000 - 4346195	1
TWI	
1.033213 - 3	7
3 - 6	6
6 - 9	5
9 - 12	4
12 - 15	3
15 - 18	2
18 - 24.718002	1
Rainfall	
1308.21 - 2000	7
2000 - 2500	6
2500 - 3000	5
3000 - 3500	4
3500 - 4000	3
4000 - 4500	2
4500 - 4682.39	1

1308.21 mm to 4682.39 mm as shown in Fig. 4. Score value increases as the annual average rainfall increases.

In the AHP process, relative weights for each influencing factor were fixed after pairwise comparison as shown in Table 4. The consistency check assures the weights assigned are correct (Kumar and Agarwal, 2020). The factors which were of very high importance, equal importance and less importance were assigned weights according to Saaty's scale (Wondim, Y. K., 2016). All pairwise comparisons are considered to be correct when the Consistency Ratio (CR) is within the permissible limit of inconsistency i.e., <10%.

In AHP, pairwise comparisons are correct when the Consistency ratio is within permissible limit of inconsistency (<10%):

$$\textit{Consistency ratio} = (\textit{Consistency index} / \textit{Random index}), \quad (5)$$

$$\textit{Consistency index} = (\lambda max - n) / (n - 1). \quad (6)$$

where, λmax = average of priority vector:

n = 4, Random index = 0.9 as per Table 6.

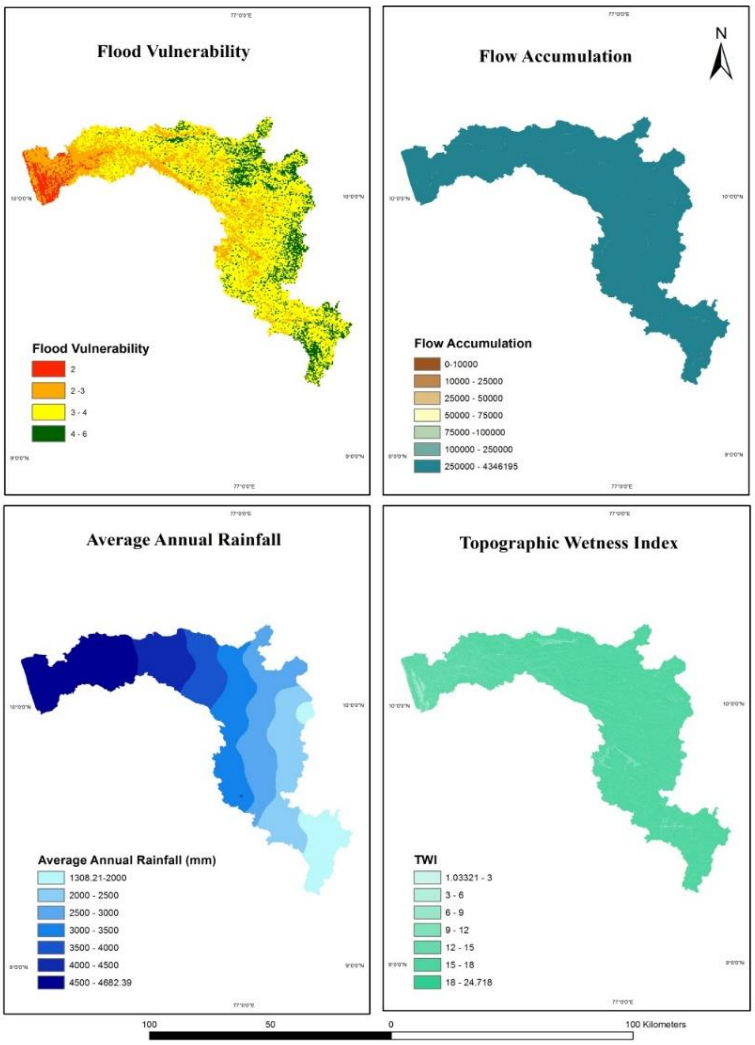


Fig. 4. Influencing thematic layers for flood risk assessment.

Table 4. Pairwise comparison matrix.

	Flood Vulnerability	**Flow Accumulation**	**Rainfall**	**TWI**
Flood Vulnerability	1.00	5.00	4.00	7.00
Flow Accumulation	0.20	1.00	0.50	3.00
Rainfall	0.25	2.00	1.00	3.00
TWI	0.14	0.33	0.33	1.00
SUM	1.59	8.33	5.83	14.00

Table 5. Normalized pairwise comparison matrix.

	Flood Vulnerability	**Flow Accumulation**	**Rainfall**	**TWI**	**Criteria Weights**
Flood Vulnerability	0.629	0.600	0.686	0.500	0.60
Flow Accumulation	0.126	0.120	0.086	0.214	0.14
Rainfall	0.157	0.240	0.172	0.214	0.20
TWI	0.090	0.040	0.057	0.071	0.06
SUM					1.00

Table 6. Random index (Dutta D., and Mahanty B., 2020).

n	**1**	**2**	**3**	**4**
Random index	0.00	0.00	0.58	0.9

Table 7. Criteria weights of factors for flood risk assessment.

Factors	**Weights**
Flood vulnerability	60
Flow accumulation	14
Rainfall	20
TWI	6

Here, consistency ratio obtained was 0.0431 < 10%, therefore the pairwise weight assignment for influencing factors was correct.

Weighted overlay analysis of different influencing thematic layers was performed of which the weighs obtained are given in Table 7. In this method, the cumulative flood risk score was derived similarly by using Equation 7:

$$Cumulative\ flood\ risk = 60 * Flood\ vulnerability + 14 * Flow\ accumulation + 20 * Rainfall + 6 * TWI, \quad (7)$$

Finally, the flood risk layer was generated through a weighted overlay; the values ranged from 2 to 5. The cumulative flood risk output thus generated was equally sized and then reclassified into three zones viz., High, Moderate, and Low as shown in Table 8.

3 Results and Discussion

Flood risk assessment is an important environmental management tool. It is very helpful in case of disaster risk reduction. In the present study flood risk assessment is carried out using AHP, which is an important multi-criteria decision-making tool widely used.

Cumulative flood risk zones, and their area extend in the entire Periyar river basin can be understood from Table 9. Cumulative flood risk in the range of 2 to 3 was considered as a high flood risk zone with 46.09 km2, which is 20.79% of the entire study area.

As seen in Fig. 5, the high flood risk areas are covered in Cochin and urban parts of Ernakulam district which was already recognized as very high flood vulnerable in the

Table 8. Cumulative Flood Risk

Cumulative Flood Risk	
2 – 3	High
3 – 4	Moderate
4 - 5	Low

Table 9. Flood risk zones

Flood risk zones	Percentage	Area (km^2)
High	20.79%	46.09
Moderate	71.35%	158.21
Low	7.86%	17.41

flood vulnerability study. The low elevation is one of the main reasons for the area being flooded. Another reason is that the area is adjacent to the backwaters of Vembanadu Lake along the coastline. There are chances that excess water gets gathered in low-lying areas and thus waterlogging can take place. The Ernakulam town center is very highly populated so there are more chances a lot of people get affected. Drainage block sites in the urban parts are another reason. The highland, mid-land, lowland regions of the land terrain is one most influential factor causing flood in Ernakulam.

Also, land use/land cover pattern change and rapid urbanization another potential reasons for making the area flood-prone. In the Idukki district, the portions of Mullaperiyar and Cheruthoni dam were identified in the High-risk zone. This may be due to more flow accumulation and TWI in the particular area. Other places under high flood risk zone include Idukki, Adimaly, Uputhara, Rajakumari, and Rajakkad.

Cumulative flood risk in the range of 3 to 4 was considered as a moderate flood risk zone with 158.21km2, which is 71.35% of the entire study area. In Ernakulam district, Aluva, Perumbavoor, Angamaly, Cochin International Airport Nedumbassery, Neeleshwaram, Vengoor, Kalamassery were identified under Moderate flood risk.

Cumulative flood risk in the range of 4 to 5 was considered as Low flood risk zone with 17.41 km2, which is 7.86% of the entire study area. This area includes the hilly regions of the Idukki district at the northern portion of the study area. The low flood vulnerability zones like Shivagiri hills, Chathurangappara, Idamalayar dam, etc., come under the high flood risk zone due to the higher values of flow accumulation and Topographic Wetness Index, also accompanied with the rainfall.

The flood risk map generated in the study helps to identify various flood vulnerable areas and places. This can be used to take measures of pre and post-monsoon disaster management in the study area.

4 Conclusions

Flood risk assessment is important to be carried out in areas susceptible to flood so that proper urban planning can be done. This can also be used in forecasting and warning of floods. In the Periyar river basin, the increasing number of flood events over years highlighted the need for flood-related studies. The geospatial approach was made use of in the preparation of flood vulnerability, flood risk maps. Flood risk map generated can be used to identify the high, medium, and low-risk areas. A combination of different

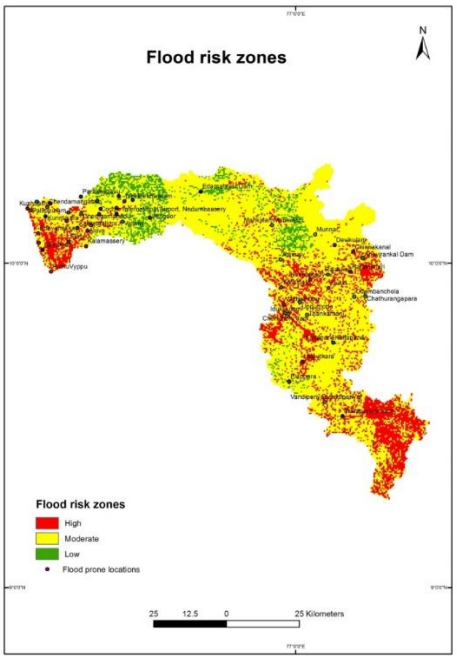


Fig. 5. Flood risk zones.

influencing factors like flood vulnerability, rainfall, TWI, flow accumulation gave a realistic picture which finally led to the computation of flood risk. In the Periyar river basin, 20.78% of the area as per the study comes under high flood risk, 71.35% under moderate flood risk, and 7.86% under low flood risk.

Extreme floods cannot be prevented but a lot of investment in the reducing flood risks is absolutely necessary. In the present study, flood maps generated are useful for emergency management. The areas with high flood risks can undergo several mitigation strategies. In the urban area, the permission of new constructions in the high or moderate risk zones must be given only after ensuring the flood safety measures. Locations for the operation of flood relief camps, i.e., low flood risk areas, can be identified through the flood map generated through the study. The present study is very useful for emergency disaster management, disaster preparedness, readiness, and responses.

References

1. Ghosh A., Kumar Kar, S.: Application of Analytical Hierarchy Process (AHP) for Flood Risk Assessment: A Case Study in Malda District of West Bengal, India. Natural Hazards, 94, pp. 349–368 (2018)
2. Ameen-Salam, A., Rubeena, T.A.: Damage Assessment and Changing Nature of Land Use with Special Reference to the Flood Plain Areas of Aluva and Paravur Taluks, Kerala. Journal of Aquatic Biology & Fisheries, 6, pp. 1–8 (2020)

3. Kumar-Saha A., Agrawal, S.: Mapping and Assessment of Flood Risk in Prayagraj District, India: A GIS and Remote Sensing Study. Nanotechnology for Environmental Engineering, 5(11), pp. 1–18 (2020)
4. Krzywoszanska, A., Mrowczynska, M., Tronte, S.: GIS Technology, 3D Models and Mathematical Models as a Tool for Assessing Development Capabilities of Flood Risk Land to Make Arrangements of Municipal Planning Documents. Journal of Ecological Engineering, 20, pp. 25–33 (2019)
5. Springer: LNCS Homepage (2020) http://www.springer.com/lncs.
6. Censusindia: Census of India 2011. Kerala, Series-33, Part XII-B, District Census Handbook Idukki, Directorate of Census Operations Kerala, Government of India (2020)
7. Censusindia: Census of India 2011, Kerala, Series-33 Part XII-B, District Census Handbook Ernakulam, Directorate of Census Operations Kerala, Government of India (2020)
8. CWC: Study Report Kerala Floods of August 2018. Government of India Central Water Commission Hydrological Studies Organization Hydrology (S) Directorate, Government of India (2019)
9. Dutta, D., Mahanty B.: Role of Consistency and Random Index in Analytic Hierarchy Process - A New Measure. Numerical Optimization in Engineering and Sciences, 979, pp. 233–239 (2020)
10. Mousavi, S., Roostaei, S., Rostamzadeh, H.: Estimation of Flood Land Use/Land Cover Mapping by Regional Modelling of Flood Hazard at Sub-Basin Level Case Study: Marand basin. Geomatics, Natural Hazards and Risk, 10(1), pp. 1155–1175 (2019)
11. Soils of Kerala: http://www.keralasoils.gov.in/index.php/2016-04-27-09-26-39/soils-of-kerala, Department of Soil Survey and Soil Conservation, Government of Kerala (2017)
12. Xiao, Y., Yi, S., Tang, Z.: Integrated Flood Hazard Assessment Based on Spatial Ordered Weighted Averaging Method Considering Spatial Heterogeneity of Risk Preference. Science of the Total Environment, 599-600, pp. 1034–1046 (2017)
13. Wondim Y.K.: Flood Hazard and Risk Assessment Using GIS and Remote Sensing in Lower Awash Sub-basin, Ethiopia. Journal of Environment and Earth Science, 6(9), pp. 69–86 (2016)
14. Wondim, Y.K.: Flood Hazard and Risk Assessment Using GIS and Remote Sensing in Lower Awash Sub-basin, Ethiopia. Journal of Environment and Earth Science, 6, pp. 7–86 (2016)
    Zulkhairi, M.D., Azman T.: A Multicriteria Index and Analytical Hierarchy Process on Flood Risk Assessment: Application in Niger State, Nigeria. International Journal of Innovative Technology and Exploring Engineering, 8(8S), pp. 674–678 (2020)

ISSN 1870-4069

Data Collection and Data Processing System for Traffic Studies

Hector Rodriguez-Rangel[1], Rafael Imperial Rojo[1], Luis Alberto Morales Rosales[2], Sofia Isabel Fernandez Gregorio[3], Abraham Efraim Rodríguez Mata[4], Peter Lepej [5]

[1] Tecnólogico Nacional de México, Campus Culiacán, Mexico

[2]Conacyt-Universidad Michoacana de San Nicolás de Hidalgo, Mexico

[3] Instituto Tecnológico Superior de Martínez de la Torre, Mexico

[4] Tecnólogico Nacional de México, Campus Chihuahua, Mexico

[5]University of Maribor, Slovenia

hector.rr@culiacan.tecnm.mx, imperial435@gmail.com, imperial435@gmail.com, lamorales@conacyt.mx, sfernandez@tecmartinez.edu.mx, abraham.rm@chihuahua.tecnm.mx, peter.lepej@guest.um.si

Abstract. The usage of vehicles boosted urban mobility, which resulted in a rise in traffic accidents. Traffic studies (TS) using specialized equipment are necessary to investigate this phenomenon. It is now feasible to carry out these tasks without significantly altering the urban infrastructure because of technology advancements, artificial intelligence (AI), and films. The design of AI-based solutions requires generating public databases that provide reliable videos for the calibration and development of solutions that perform TS efficiently. This paper presents a system to generate a dataset from videos taken at an observation point on a roadway. The major highlight of these films is that they are not shot straight line with the camera. With the development of this system, we can assist in constructing various datasets to infer an object's speed and extract the attributes that the user believes to be the most essential. In addition, the system can detect, track, and offer object value information and combine it with their speed.

Keywords: Data base, vehicle speed, artificial vision.

1 Introduction

Automobiles are a necessity in our everyday life. More than 1.3 million people are killed in road accidents each year, according to the Association for Safe International Road Travel (ASIRT), with another 20 to 50 million wounded or handicapped ([15]). Traffic accidents are more common in the city center and suburbs than elsewhere, especially in Mexico. According to the National Institute of Statistics and Geographic Information (INEGI, 2011), accidents in the region grew by 72.7 percent in both urban and rural areas between 1997 and 2009 ([2]). The driver of an automobile carries the most responsibility in the case of an accident ([10]). Speeding, distracted driving, road obstructions, inadequate signaling, road infrastructure quality, and illumination are major contributors to traffic accidents. Because speeding is the most lethal element in road accidents, traffic studies are the primary source of information for identifying potential causes. These investigations necessitate the use of a reliable speed monitoring system. Furthermore, control systems such as Advanced Driver Assistance Systems (ADAS) help the driver while driving on the roadways [2].

The employment of special equipment has an impact on the usual technique of performing traffic studies. Traditional methods of collecting traffic data include those placed below the road surface (typically used for weigh-in-motion). For instance, inductive spires, magnetic field sensors (also above the road), galvanic contact devices/axle counters, capacitive and piezoelectric sensors (usually used for axle counting), or scales with sensor plates.

On the other hand, some sensors can be mounted above the road. For example, video surveillance cameras, microwave radar detectors, laser radar detectors, magnetic field sensors, passive and active infrared sensors, ultrasonic sensors, and other sound or video image processing instruments. A traffic study involves skilled experts and sensors at various places to properly analyze the sections that create congestion and/or accident concerns for examination inside cities and roads.

Video security cameras that provide real-time information are sensors that do not require any changes to the city's infrastructure. It is feasible to predict traffic flow parameters (e.g., vehicle density, peak hours, average speed, accident rate) by analyzing data from cameras, as well as identify congestion, accidents, and watch driver behavior ([12]). Besides, we can carry out processes such as traffic count, traffic modelling, traffic analysis, traffic prognosis, traffic safety studies (inspection and audit), and street network configuration from the video analysis. The main parameters are the speed and the trajectories of the vehicles. The traffic estimations can be provided to users and police patrols via road panels or integrated vehicle monitors to help in planning exits and avoiding traffic bottlenecks ([12]).

Artificial intelligence (AI) is changing the way we live in the modern world. Researchers and developers actively develop autonomous driving based on deep learning and monitoring techniques of the cars and road safety industries. However, before any artificial intelligence algorithm can be used in a production vehicle, it must first undergo a thorough functional safety evaluation and

database construction. Therefore, we can see some advances and opportunities to solve problems of incorporating deep learning into autonomous cars in [6], as well as the construction of a database for training such neural networks.

This paper describes a technique for creating a dataset from movies collected at a roadside observation site. The major highlight of these films is that they are not shot perpendicular lines with the camera. With the development of this system, we can assist in constructing various datasets to infer object speed and extract the attributes that the user believes to be the most essential. Databases that gather diverse circumstances to monitor behaviors and calibrate their results are required to develop solutions for the autonomous creation of traffic studies. Besides, we want to remark that no credible public databases are available to assist with the Mexican traffic flow research.

2 Background

There are now several studies that calculate object speed; each one proposes a unique method, utilizing corpus such as the KITTI dataset [1]. This section highlights some of the works that were used as inspiration for this project.

Physics equations may be used with data from videos to estimate object's speed (1), acceleration (2), and angle (3) with a fast response. However, the straightforward application of these equations restricts their usage to a two-dimensional plane or the movement of vehicles parallel to the camera view.

$$Speed = \frac{Distance}{Time}, \tag{1}$$

$$Aceleration = \frac{Speed\,Final - Speed\,Initial}{Time\,Final - Time\,Initial}, \tag{2}$$

$$Angle = \tan^{-1} \frac{Point\,Final\,Y - Point\,Initial\,Y}{Point\,Final\,X - Point\,Initial\,X}. \tag{3}$$

In contrast, [8] proposes to determine the speed (equation 4), acceleration rate (equation 5) and the angle of an object (6)with data from a sequence of images with equation. These authors consider that the following characteristics must be satisfied:

- An approximate range of RGB values of the object is known.
- The rate at which the camera is taking images is known.
- The approximate size of the image is known.
- The image is represented in a uniform color.
- The background has uniform color and is not of the color of the object.

They use two images to calculate the acceleration of the vehicle. In the first and second images, a point is drawn at the centroid of the item to determine the desired values. The following formulae can be used to determine the object's speed, acceleration, and angle from both centroids:

$$Speed = \frac{\sqrt{(x2 - x1)^2 + (y2 - y1)^2},}{FPS} \tag{4}$$

$$Acceleration = \frac{Speed\,Final - Speed\,Initial}{FPS}, \tag{5}$$

$$Angle = \tan^{-1} \frac{y2 - y1}{x2 - x1}. \tag{6}$$

In [4], authors use a multilayer perceptron neural network that was trained using picture sequences to detect a vehicle's relative speed, achieving good results with an average error of 1.12 m/s, which is equivalent to a LiDAR radar's error of 0.71 m/s. The vehicle track in a 2D plane, the depth, and optical flow estimations between successive pictures were recognized as three key elements for determining the speed. When recognizing a car in a succession of photos, we can see that its size changes based on its distance from the observer. This process impedes neural network learning; therefore, it was decided to divide the data into three categories: close, medium, and far, and train a model for each.

In [14], [13], authors determine the speed with the use of stereo cameras which are fixedly located in a strategic place in the city. The stereo cameras are responsible for identifying the license plate, headlights, and logo, as these characteristics are taken as main and always appear in a vehicle; these are used as a reference to determine the speed. In [9], a similar strategy is used by only identifying the license plate as the main character and using it as a reference to determine the speed of the vehicle. However, in this stage, only one camera is used instead of the stereo cameras used in [14, 13].

In [5] determine the speed of moving vehicles using drone-mounted stereo cameras. Their method integrates Mask-R-CNN and K-Means with the Lucas Kanade pyramid algorithm. In their experiments, they show that the combined use of Lucas Kanade, Mask-R-CNN and K-Means, improve their results compared to the use of these methods separately or the combined use of only 2 of them, even though they have performed experiments under normal circumstances, with many moving objects, and in low light circumstances.

In [3] develop a module for measuring the distance of vehicles in several lanes simultaneously using a drone. The drone is mounted with two LiDAR sensors, and each sensor emits a front point and a rear point, which vehicles are detected. The vehicle speed is estimated using the calculated distance between the front and rear point through which the vehicle passes and the time it takes to pass through both points.

3 Methodology

As stated in the introduction, this project aims to generate information on a road's traffic flow from video sequences. When conducting research, it was discovered that no dataset engaged the project's requirements. Thus

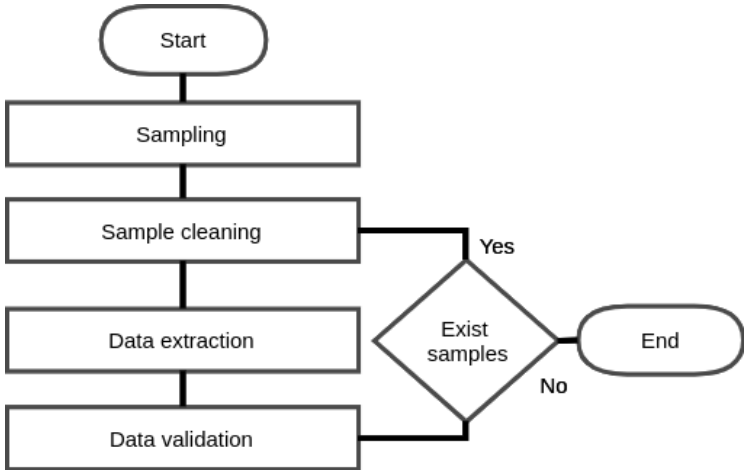


Fig. 1. Flowchart to generate data set.

it was essential to build one. This task will be discussed in this section of the methodology.

The procedure for extracting these features was broken down into four stages: 1) sampling, 2) sample cleaning, 3) data extraction, and 4) data validation. The whole procedure for creating the dataset is depicted in Figure 1.

3.1 Sampling

The first stage involves sampling, which required deploying two devices to retrieve the data set to construct the vehicle speed prediction model based on video sequences. The first one is a video camera located inside a smartphone. We configure the camera at 60 frames per second with Full HD resolution (1920 x 1080 pixels). We stabilize the camera so that the cars do not pass parallel to the camera view, preventing the computation in a 2D plane. The second one is a Bushnell radar (Figure 2) with a +/- 1.6 kilometer per hour accuracy to establish an accurate reference of the vehicle's speed analyzed to form the videos.

3.2 Sample Cleaning

The radar and the tracking system are not linked, then it is essential to combine the speed supplied by the radar with the information received by the system. For this, a visual inspection of the samples taken is required. The first step is to identify the Points in x that are used as a reference for taking them as vehicle entry and exit points. Using our Point of exit as a reference, we will identify the vehicles to which the speed was taken and write down the second it passes.

In addition, we identify the lane in which the vehicles pass to separate them in the three street lanes, we represent with zero the first lane, one the central

Fig. 2. Radar used for sampling.

Table 1. CSV output attributes

Attribute	Description
Output Angle	Angle from the input to the output.
Distance Traveled	Distance traveled in pixels from the input to the output.
Input Area	Area in pixels at the input.
Output Area	Area in pixels at the output.
FPS	Frames per seconds.
Time	Time to pass from the input to the output.
Lane	Street lane where pass the vehicle.
Speed	Speed detected by radar.
Identifier	Identifier to the created image.

lane, and two the last lane, with this we seek to create three datasets based on the work of [4].

This information obtained visually we save in a CSV file, which will have the following three attributes:

- The second the object passes through the exit point.
- The vehicle's speed.
- Lane.

3.3 Data extraction

When the term creation of the dataset is used, it refers to extracting the video's key features. These enable the determination of a vehicle's speed, frequency, and other critical parameters for investigating vehicular flow. The characteristics extracted are listed in Table 1, along with a brief explanation.

The execution of the system is simple; we need the CSV file generated earlier with the speeds and limits that we also identified in the previous step. The system

Fig. 3. Vehicle identified with its respective tracking.

is in charge of reading the video using the OpenCV library, and it examines frame by frame the content. Besides, it identifies the vehicles using the neural network YOLO ([7]), which identifies multiple objects in a single prediction; once it has identified all the vehicles, it draws the box corresponding to each one of them. This process allows the detection and identification of the objects of interest inside each frame.

The vehicles are tracked through a Kalman filter ([11]) to determine their location in the next frame. We choose this filter because it estimates a joint probability distribution over the variables for each timeframe. Besides, it has better performance when is used for linear or linearized processes and measurement systems than particle filter, which is more suitable for nonlinear systems. The system is in charge of preserving all the locations of the vehicles over time. Therefore, we can draw each vehicle's path inside the scenes and calculate the straight line corresponding to each trajectory. Figure 3 shows a detected vehicle in a white box, as well as a pair of lines, one yellow and another red, that correspond to the tracking of the same one and the calculated straight line corresponding to the tracking.

When the system detects a vehicle passing through the first boundary, it saves the frame to join it later, identifying when it passes through the exit point. It is worth noting that this is done for all detected vehicles, but only those that correspond to the second exit point identified in the CSV file are saved.

When the system detects a vehicle leaving the specified point, it records the generated data and speed in a line of a CSV file and an image of the vehicle entering and leaving the limits and an image identification. The technology operates at 30 to 40 frames per second, meaning each minute of video will take two minutes to process in the worst-case scenario.

Fig. 4. Valid system detection.

Fig. 5. Invalid system detection.

3.4 Data cleaning.

As previously stated, the system generates an image for each line of the resulting CSV. This image is an arrangement of two images: the image taken when the vehicle enters the entry point and the image taken when the vehicle exits the exit point; we use this image to verify that the vehicle was taken correctly by the system. In this situation, we consider legitimate pictures to be those that are not too far from the departure point and in which the vehicle has been identified in its whole or in a significant portion of it, as shown in Figure 4.

In the instance of incorrect images, we can observe how the system, in some stages, produces a partial output vehicle as a consequence. Figure 5 depicts an example where the system detects a cropped vehicle image; therefore, we delete it from the resulting CSV file. To remove invalid samples from the resulting CSV file, we must rely on the image identifier. This is found in the image name and is the last attribute of the CSV file. See Table 2.

The vehicle that the system is tracking to create its related information is shown in a white box in Figure 6, while the rest of the identified cars are shown in yellow and will not have a line generated for the resultant CSV.

4 Results

So far, implementing the system has generated a total of 178 videos ranging in length from one to two minutes, of which a total of 29 have been processed,

Fig. 6. Image generated by the system.

Fig. 7. Invalid image to the first lane.

from these 29 videos generated 532 samples, for each of the lanes the number of samples generated is as follows:

- First lane: 8 samples
- Middle lane: 239 samples
- Last lane: 285 samples

The first lane has few samples generated since this lane passes very close to the camera view, which causes a cut in the vehicle detection when passing to the exit point. Figure 7 shows one of these cases.

The following Table 2 shows us five random samples generated by the system, these samples were round to 3 decimals to be shown and the header description is in the Table 1.

We can use the video dataset to supply the system presented in previous sections, recognize, monitor, offer interest for traffic studies, and automatically calculate the vehicle's speed.

5 Conclusions

The system can provide a set of high-quality data. We may run tests with these data using various statistical and artificial intelligence approaches to forecast the speed at which objects occur in a video sequence. In addition, it allows us

Table 2. Example with data generated to the CSV.

Output Angle	Distance Traveled	Input Area	Output Area	FPS	Time	Speed	Lane	Identifier
80.458	714.973	26797	226200	58.063	0.878	45	1	246
79.570	879.673	25800	397794	58.063	0.964	66	1	407
75.300	778.597	25088	289962	60.018	0.600	61	0	6372
77.428	751.122	43960	368145	58.063	0.603	62	1	1380
82.959	718.533	9864	104535	58.063	1.240	57	2	1708

to combine sensors to create additional parameters that we can use to infer the object's speed. Also, it enables sensors integration to generate more parameters.

On the other hand, the generated dataset requires the removal of data that may affect the predictions. Hence, additional work is needed over the dataset by the possibles users. This process is a challenge to generate an acceptable dataset because of the time required to process all of the videos and validate the generated data.

A vital system improvement activity eliminates the need for a person to generate the CSV file with speed and validate that the information generated is correct. We aim to reduce the time required to generate the data set as future work related to the time spent by an operator because currently is relatively high, and the system could autonomously handle this task.

References

1. Burnett, K., Samavi, S., Waslander, S., Barfoot, T., Schoellig, A.: autotrack: A lightweight object detection and tracking system for the sae autodrive challenge, pp. 209–216 (2019) doi: 10.1109/CRV.2019.00036
2. Carro-Pérez, E. H., Ampudia-Rueda, A.: Conductas de riesgo al conducir un automóvil en zonas urbanas del sur de tamaulipas y la ciudad de méxico. CienciaUAT, vol. 13, no. 2, pp. 100–112 (2019)
3. Lee, K.-H.: A study on distance measurement module for driving vehicle velocity estimation in multi-lanes using drones. Applied Sciences, vol. 11, no. 9 (2021) doi: 10.3390/app11093884
4. Moritz Kampelmuhler, M. G. M., Feichtenhofer, C.: Camera-based vehicle velocity estimation from monocular video. Institute of Electrical Measurement and Measurement Signal Processing, (2018)
5. Peng, Y., Liu, X., Shen, C., Huang, H., Zhao, D., Cao, H., Guo, X.: An improved optical flow algorithm based on mask-r-cnn and k-means for velocity calculation. Multidisciplinary Digital Publishing Institute, (6 2019) doi: 10.3390/app9142808
6. Rao, Q., Frtunikj, J.: Deep learning for self-driving cars: Chances and challenges. In: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems. pp. 35–38 (2018)
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016) doi: 10.1109/CVPR.2016.91

8. Singh, S.: Estimating speed, velocity, acceleration and angle using image addition method. International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, pp. 11585–11589 (11 2015) doi: 10.15680/IJIRCCE.2015
9. Vakili, E., Shoaran, M., Sarmadi, M.: Single-camera vehicle speed measurement using the geometry of the imaging system. Multimedia Tools and Applications, vol. 79 (07 2020) doi: 10.1007/s11042-020-08761-5
10. Velázquez Narváez, Y., Zamorano González, B., Ruíz Ramos, L.: Siniestralidad vial en la frontera norte de tamaulipas. enfoque en los procesos administrativos de control. Estudios fronterizos, vol. 18, no. 36, pp. 1–24 (2017)
11. Welch, G., Bishop, G., et al.: An introduction to the kalman filter (1995)
12. Yang, L., Li, M., Song, X., Xiong, Z., Hou, C., Qu, B.: Vehicle speed measurement based on binocular stereovision system. IEEE Access, vol. 7, pp. 106628–106641 (2019) doi: 10.1109/ACCESS.2019.2932120
13. Yang, L., Li, Q., Song, X., Cai, W., Hou, C., Xiong, Z.: An improved stereo matching algorithm for vehicle speed measurement system based on spatial and temporal image fusion. Entropy, vol. 23, no. 7 (2021) doi: 10.3390/e23070866
14. Yang, L., Luo, J., Song, X., Li, M., Wen, P., Xiong, Z.: Robust vehicle speed measurement based on feature information fusion for vehicle multi-characteristic detection. Entropy, vol. 23, pp. 910 (07 2021) doi: 10.3390/e23070910
15. Zaki, P. S., William, M. M., Soliman, B. K., Alexsan, K. G., Khalil, K., El-Moursy, M.: Traffic signs detection and recognition system using deep learning. arXiv preprint arXiv:2003.03256, (2020)

ISSN 1870-4069

Development of an Optimal Model of Space use through AHP Model and Boolean Logic in GIS Context Case study: Kurdistan of Iran–Sanandaj

Kiomars Naimi[1], Mina Soleymani[2]

[1] School of Architecture and Urban Design,
Art University of Isfahan,
Iran

[2] Fine Arts School, University of Tehran,
Iran

K.naimi@aui.ac.ir, mina.soleimany66@gmail.com

Abstract. Sports spaces are one of the most important urban Land-uses to increase the physical and mental health of citizens, it seems necessary to pay attention to the fact that sports spaces should be properly located in the city. The purpose of this study is to provide an optimal model for locating sports facilities in Sanandaj regarding Urban Facilities` improvements. The research method is descriptive and analytical based on field and library observations and the use of analytical tools AHP, Boolean logic, and GIS. In the re-search process, the study area in terms of population density, building density, access, neighborhoods in the city is examined and the AHP model is used to weigh the criteria then using ARC GIS maps are prepared according to the criteria finally, a combination map of the criteria is extracted then using Boolean, two criteria of existing sports spaces and river area are included as a limit for establishing sports uses and a map. The usage of these research methods let us to decrease the uncertainties and achieve much trustable outcomes. The final that represents the best places for sports spaces in this area is obtained. By combining and stacking the maps, it is determined that the places that have a higher population density and a high building density, as well as places that have suitable uses for adjacent to sports spaces and passages that are more suitable for establishing sports spaces. The places that have existing sports use and areas are located in the river area should not be constructed sports facilities, so four locations were selected as the best locations for sports facilities.

Keywords: Sustainable city, urban space, socio-spatial development, GIS, Sanandaj.

1 Introduction

Rapid population growth and disproportionate physical development in large cities have created complex and un-solvable problems. Urban development in previous decades has been such that it has led to an imbalance in how urban land is used, turning villages into cities and small towns into large cities, while most of these transformations and changes without Planning has been done and has not been commensurate with the

needs of society. Improving this situation has made the responsibility of urban planners heavier and has required them to respond (thoughtfully) to inconsistencies (Fazelnia et al., 2010). Urban spaces are one of the most important parts of the city. These spaces are located in the framework of settlements and habitats, can be distributed in different scales in the city and create landmarks in the city. They are a function of human goals and collective activities, urban complexes are a tool to promote the spirit of collective thinking, cooperation, understanding and sincere communication and create a safe, comfortable and identity environment. Therefore, paying attention to the required spaces that guarantee the health of the body and soul of citizens, seems necessary in the planning and structure of the city (Nejati, 2006: 7).

Challenges and the Necessity of Modeling: Sports facilities are also among the public service centers that for their great importance cannot be left to the mechanism of the market economy. It should be noted that every year, many sports venues are built in different parts of the country, which according to studies conducted by the organizations in charge, it was determined that their location is based on traditional methods. It seems that in some of these constructions or giving licenses for establishing the important points of the correct location, some attention has not been paid, which may reduce the optimal efficiency of these spaces or create problems for the city and citizens (Razavi Et al., 2009). Undoubtedly, the optimal and successful management and implementation of physical education and sports programs require the provision of a set of conditions and facilities. One of the most important conditions is the creation, development and optimal use of sports facilities providing the necessary facilities for easy access of sports enthusiasts to these spaces (Razavi, et al., 2007). How to build a city and design urban environments and how to access the natural environment, can be considered as an important motivating factor or a major barrier to physical activity and active living.

There are other barriers to physical activity in community settings where people are educated, or work, play and live. Many people who suffer from the negative effects of obesity and chronic diseases are those who experience poverty and social exclusion. The necessary condition to ensure equality and inclusion of measures related to the promotion of physical activity and active life is to assess the needs and participation of all citizens in various areas of daily life (Edward and Taurus, 2008: 34). Health is an important driver of economic development and productivity. Promoting active life in cities can have many economic and social benefits in addition to ensuring the health of citizens. Active living is a way of life that incorporates physical activity into everyday life. This physical activity can include walking and cycling for recreation and movement, playing and exercising in parks, or any other form of recreation. Physical activity is one of the most important measures that people of any age can take to improve their health (Pate, 2009, 2). The purpose of this study is to use the capabilities of Advanced Geographic In-formation System (GIS) software to find suitable locations for the construction of sports centers in Sanandaj.

Literature Review: The background of the present study has shown that relevant scientific resources regarding the organization of sports facilities have been scarce and most of the research conducted has been related to the location and development of sports facilities. Attention to the combined approach of capabilities with GIS, models and techniques that can be used in solving urban problems and especially the optimal location of urban uses has been considered by scientific and executive circles.

Table 1. Resources related to the location of sports spaces.

2020	Armana Sabiha, and Behnaz aminzadeh.	Evaluation of Physical Urban Management Strategies in Major Crisis Management Policies in Tehran.	In this research the Significant Urban Factors and the amount of level of responsiveness have been evaluated.
2020	Farid Karimipour, A javidaneh.	An Approach for Automatic Matching for Descriptive Addresses, Journal of Geomatics	This article proposes a solution for automatic matching of descriptive addresses.
2018	Farid Karimipour, RA Abaspour, and S faze	Investigation the Capabilities of Geo-Simulation Based Approaches to Urban Development Modeling.	In this research the Agent-based Modeling and the Cellular automata have been studied to examine their performance in the field of Urban Forecasting and Physical Development.
2018	Kiomars Naimi, Freydon Babaii Aghdam	City and Spatial Justice; The Analysis of The Distribution of Urban Public Services In The 22 Areas of City Sanandaj	Spatial and environmental justice means the fair distribution of resources to achieve a balanced society, and one of the approaches is social justice. The measurement of spatial justice in the distribution of urban public services is Superior goal and important for urban planners, and is the of needs urban management.
2017	Hassan Karimi, Bobak Karimi	Geospatial Data Science Techniques and Applications	This published book highlights the unique characteristics of geospatial data, demonstrate the need to different approaches and techniques for obtaining new knowledge from raw geospatial data, and present select state-of-the-art geospatial data science techniques and how they are applied to various geoscience problems.
2016	Miguel Felix Mata River and et al	Environmental GIS to identify municipalities with high potential of biogas production in Mexico	This research proposes a methodology to make a system that performs geographic analyses to show potential landfill locations, and the calculation of potential to support the municipalities in building a landfill that has the necessary infrastructure to capture, treat and take advantage by the MSW.
2016	Shahrevar Rostai, Kiomars Naimi, Salman Mahmodi	A Spatial Analysis of Educational Inequalities and Its Role in Urban Social Sustainability the Spatial Statistical Methods (a Case Study of Saqqez)	Urban sustainability depends on relative welfare, citizen participation and social awareness for all community members.
2010	California Land Use Institute	Advantages of locating sports complexes in case of good design	Policymakers must be committed to addressing the key challenges of designing leisure centers and sports complexes in the public interest.
2010	Isfahan Municipality	Spatial analysis and location of sports venues using GIS	Relying on the results of this study, in addition to knowing the location of their sports venues in the area, sports managers can proceed with the construction of different types of venues with a much higher confidence factor.
2009	Mohammad Taghi Rahnamaei and Leila Aghaei	The Role of Municipalities in the Development of Sports Spaces for Citizens' Leisure Time (Case Study: District 6 of Tehran Municipality)	The results of this study showed that the municipality, due to its continuous and close relationship with consulting engineers and other decision-making elements and preparers of documents and urban plans and detailed plans, has a special place in the development of recreational and sports spaces (parks and sports spaces).
2000	Chapin team	Political Economy Location of Sports Facilities: A Review	The location of a sports center in urban areas is always tied to three categories of decision-making factors: technical factors such as site characteristics, economic factors such as land costs, and political factors such as economic development plans.

Therefore, it will be necessary for urban planners to pay attention to the proper organization of sports venues. Some of the major sources related to the research topic are listed in Table 1.

2 Research Methods

This research has been done in two descriptive-analytical sections. In the descriptive section, using library-documentary studies in order to provide an optimal model for the location of sports facilities in the study area, criteria appropriate to the purpose of the research were selected and then the mentioned descriptive data were collected. Effective factors that have been studied in order to provide an optimal model for the location of sports spaces. These are access to the communication network, population density, building density, suitable neighborhoods. In the analytical section, in order to perform location operations, standard maps obtained from the previous steps were provided to integrate the standardized layers, using two methods, **Boolean Logic Model** and **AHP**: In the following, the results of each of these methods and their combination are presented:

Location: It is an activity to select a suitable place for special use, which analyzes the capabilities and capabilities of an area in terms of the existence of suitable and sufficient land and also its relationship with specific land uses and urban land uses (Salehi, 2004).

Sports centers: All spaces where there are sports, physical activities and sports and recreational movements for all people in a community can be considered as sports venues (Atshan, 1997).

Analytic Hierarchy Process (AHP): This process incorporates various options in decision making and allows the analysis of sensitivity to criteria and sub-criteria. In addition, it is based on the pairwise comparison, which facilitates judgments and calculations and shows the degree of consistency and incompatibility of the decision, which is one of the advantages of this technique in multi-criteria decision making. The process is designed to conform to and proceed with the human mind and nature. This process is a set of personal judgments (decisions) and personal evaluations in a logical way so that it can be said that the technique, on the one hand, depends on personal perceptions and experience, to form and plan a hierarchical problem. On the other hand, it is related to logic, understanding and experience for final decision and judgment (Ghodsipour, 2006).

Models for the Combination of Maps. Boolean region model. In this model, an input map is generated for each factor. A value of one indicates the appropriateness and a value of zero indicates the inadequacy of its spatial position. It is a pixel. If the integration of the maps is done using the AND operator, the pixels containing the value 1 in the output map indicate locations that meet the criteria for the intended application, and if the input maps are used with the OR operator, an output map identifies that one or more criteria apply.

Index Overlay Model: The index overlap model can be performed in two ways. In both methods, weight is first assigned to all effective factors based on relative importance and according to expert opinions. These weights are determined as positive integers or real numbers in a given interval. In the first method, the invoice input maps are binary, as in the Boolean method. In this method, each factor map has a single weighting factor and, for the combination of other maps, is multiplied only by its own weighting factor. The importance of the different classes in an invoice map is considered the same in the

first method. The second method has more flexibility than the first method. In this method, in addition to being assigned to each of the weight input maps, it is also assigned to each of the classes and spatial units in each invoice map based on the relative importance and weight expert opinions. In other words, the different classes on a single map have different weights.

Fuzzy Logic Model: The fuzzy logic is the developed region of Boolean. In fuzzy logic, the membership rate of an element in a set is defined by a value in the range of one (full membership) to zero (no full membership). Types of fuzzy operators include Fuzzy AND operator, Fuzzy OR operator, Fuzzy Algebraic product, and Fuzzy Algebraic Sum operator (Qahroudi Tali, 2012).

Case Study: The city of Sanandaj in recent years with the prosperity of gambling and land grabbing, lack of proper control and management and the formation of various land and housing cooperatives has witnessed an explosive development, inconsistent and contrary to the basic principles of urban planning. At the same time, due to the increasing attractiveness of urban life and the growing poverty of the villagers, migration to the city has developed into an insecure and problematic face called marginalized neighborhoods. According to the latest information of the consultant, more than one-third of the population and area of the city is allocated to these contexts.

Total Areas in Sanandaj:

- The area of Sanandaj City is 6.6886 hectares.
- The built-up area of the city is 2187/8 hectares.
- The empty tissue area is 7703/1,500 hectares.

The Population of Sanandaj City in 2021:

- According to the Statistics Center of Iran, its population in the population and housing population of 2021 is 417177 people.
- The population growth rate is 1.9 percent per year.

Densities:

- The gross density of Sanandaj is now 87.42 people per hectare.
- The net density of Sanandaj is currently 321.87 people per hectare.
- The net density of the city in 437,1364 people was per hectare (Sanandaj Govern).

3 Results and Discussion

In this section, the practical steps performed to provide the optimal model for locating sports spaces in Sanandaj are presented: At first, suitable and compatible indicators for the construction of sports spaces were extracted from the sources and due to the availability of this information, the following indicators were considered in this research. In the first stage, in order to find the most suitable places for establishment

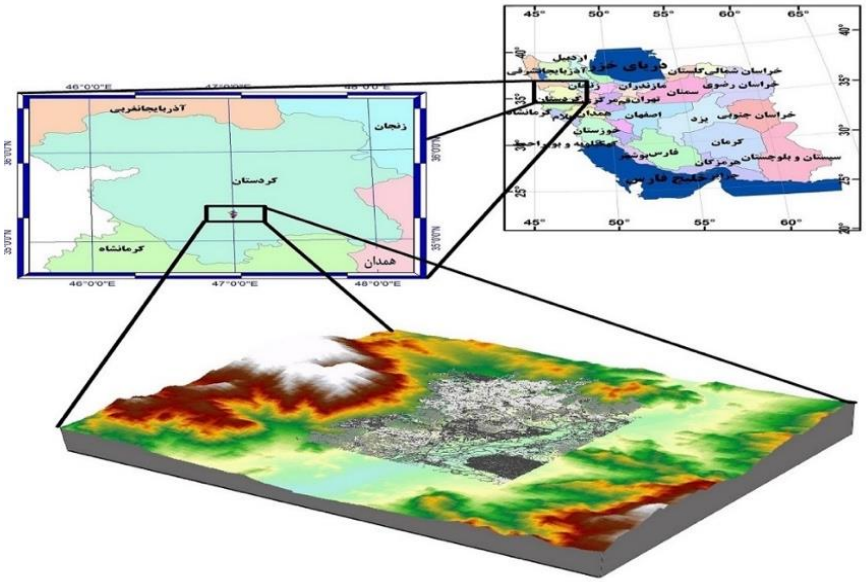


Fig. 1. Case study.

Table 2. Matrix of pairwise comparisons of indicators.

Indicators	Population Density	Building Density	Accessibility	Proximity
Population Density	1	2	3	5
Building Density	1/2	1	3	4
Accessibility	1/3	1/3	1	3
Proximity	1/5	1/4	1/3	1

sports spaces the AHP model is used. Criteria selected for the location of sports spaces in the city are.

Density: In this model, places with high population density will have more priority in order to establish sports use. Figure 2 shows the population density of Sanandaj.

Building density: As the population grows in the region, access to land and open spaces for different uses decreases and building density increases in the regions. In this model, neighborhoods with high building density will have a higher priority for sports use. Figure 3 shows the building density of Sanandaj.

Access: In this model, passages with a width of 15 to 20 meters have the most opportunity for the establishment of sports use, so the places that are placed in different classes according to the distance from these passages. Figure 4 shows the access map of Sanandaj City.

Table 3. Relative weight of indicators.

Indicators	Population Density	Building Density	Accessibility	Proximity
Relative Weight	0.461	0.310	0.156	0.073

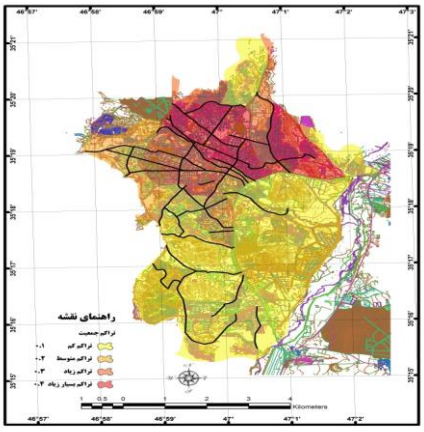


Fig. 2. Population density map.

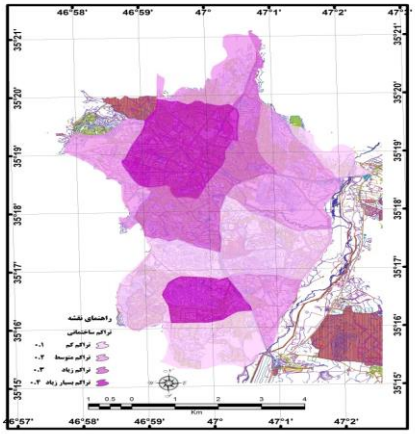


Fig. **3** Building density map.

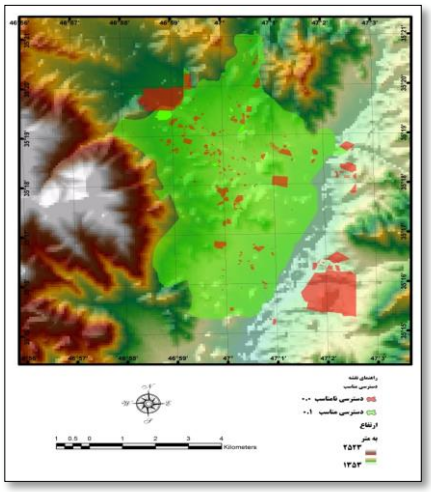


Fig. 4. Access map.

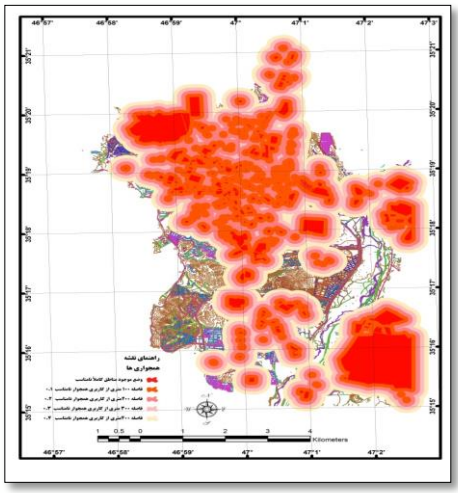


Fig. 5. Neighborhood map.

Proximity: In this section, military, medical, health, and workshop workshops are not suitable neighborhoods for sports use. As a result, all places are classified in terms of distance from this type of use. Figure 5 shows the map of the neighborhoods of Sanandaj.

Then, in order to apply the AHP model, the matrix pairwise comparisons of the indices were formed as follows and we obtained the relative weight of the indices

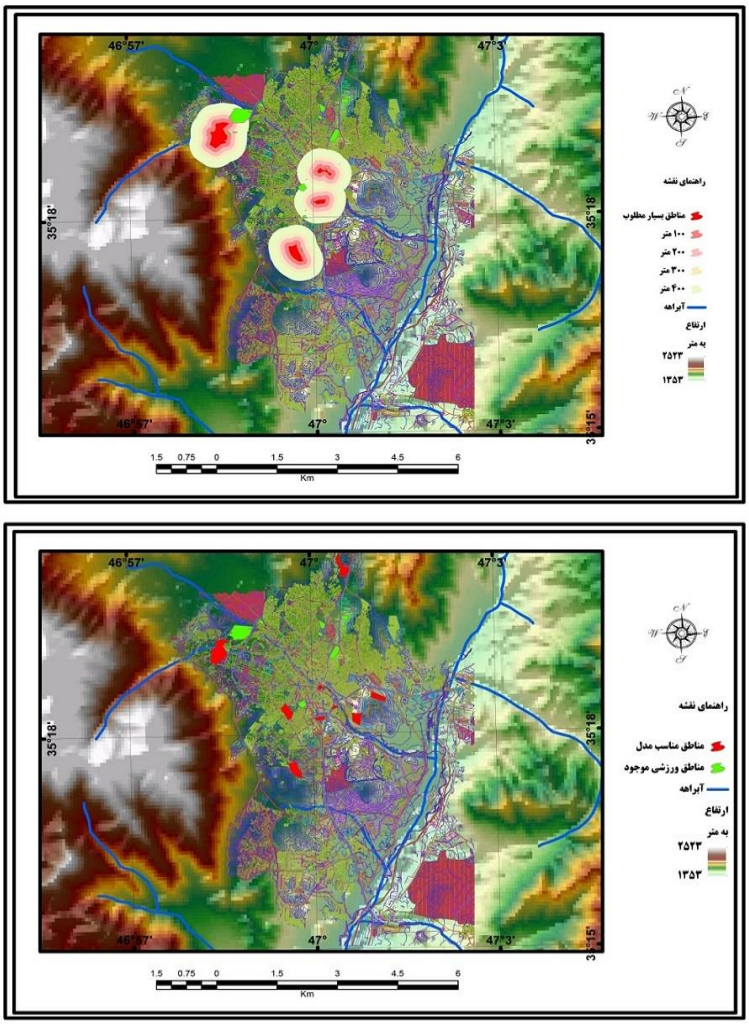


Fig. 6. Final map based on AHP model.

(Tables 2 and 3). Then, with the help of GIS, all the layers related to these criteria have been prepared and the relevant layers have been placed on top of each other according to their relative weight, and the most suitable places for the establishment of sports spaces have been obtained.

In the next step, the limitations that exist in the location of land uses should be considered that according to the characteristics of the study area, the two factors of existing sports spaces and the river area are considered as restrictions for the establishment of sports spaces.

Existing sports facilities: In areas covered by existing sports facilities, embryos should not be located. As a result, all places that have existing sports use have been identified. Figure 7: The map of existing sports areas in Sanandaj is shown.

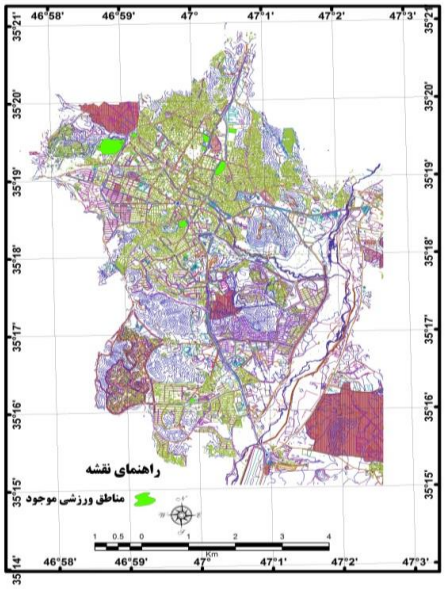


Fig. 7. Map of existing sports areas

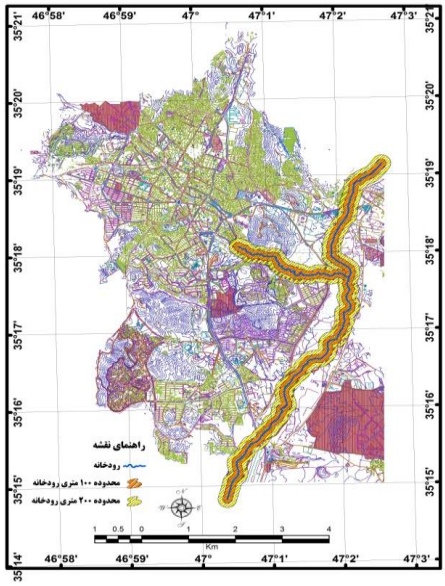


Fig. 8. River area map

River area: It has been assumed that there are restrictions on the establishment of sports uses in the river area.

Then, using GIS, information layers related to these criteria were created and also scored according to Boolean logic. Then, the most suitable places (final map based on the AHP model) and restrictions were placed on top of each other and options were

obtained to establish sports spaces. Figure 9: The map shows the most suitable options for establishing sports facilities in the city of Sanandaj.

4 Conclusions

Sports facilities are one of the most important urban uses to increase the physical and mental health of citizens, therefore, it is necessary to pay attention to the fact that sports facilities should be properly located in the city because to its significant role in as-assessment of any urban situation and criteria, so as to identify the best places for Sanandaj sports spaces were studied according to the criteria of population density, building density, access, neighborhoods and existing sports spaces, river area and using AHP model and Boolean logic in GIS environment. It was found that places that have a higher population density and have a high building density, as well as places that have suitable uses for adjacent to sports spaces and have passages that are more suitable for establishing sports spaces, and places that have existing sports use and areas that they are located in the river area, sports facilities should not be built, so four urban areas were selected as the best areas for the establishment of sports facilities. Therefore, paying attention to these factors and augment their responsiveness plays vital role in attracting individuals to healthy and sport activities in a proper urban texture. Considering these urban roles help urban managers and planners in raising the urban flexibilities.

References

1. Naimi, K., Aghdam, B.: City and Spatial Justice; The Analysis of The Distribution of Urban Public Services in the 22 Areas of City Sanandaj. Geographical Planning of Space Quarterly Journal, 7(23), pp. 173–186 (2018)
2. Roostei, S., Naimi, K., Mahmmodi, S.: A Spatial Analysis of Educational Inequalities and Its Role in Urban Social Sustainability the Spatial Statistical Methods (a Case Study of Saqqez). Social Development and Welfare Planning, 7(26), pp. 61–92 (2016)
3. Soleymani, M.: Evaluation of Physical Urban Management Strategies in Major Crisis Management Policies in Tehran. ICTH, Elsevier (2020)
4. Mata River, M.F.: Environmental GIS To Identify Municipalities with High Potential of Biogas Production in Mexico. In: IEEE (2016)
5. Karimipour, F.: Investigation of the Capabilities of Geo-Simulation Based Approaches to Urban Development Modeling. Geospatial Engineering Journal, 9(4) (2018)
6. Karimipour, F., Javidaneh, A.: An Approach for Automatic Matching for Descriptive Addresses. Journal of Geomatics Science and Technology, 9(4) (2020)
7. Karimi, H., Bobak, K.: Geospatial Data Science Techniques and Applications. CRC Press (2017)
8. Nejati, H.: Sports Spaces and Urban Design Active Urban Spaces. In: Enthusiastic Citizens, The First National Conference on Sports and Urbanism, Tehran (2006)
9. Fazelnia, G., Kiani, A., Rastegar, M.: Optimal Location of Sports Spaces in Zanjan Using the Series Analysis Model. Journal of Research and Urban Planning, 1 (2010)
10. Razavi, S., Hossein, M., Kolsoom, E.: Spatial Analysis of Sports Spaces in Amol City Using Geographic Information System (GIS). Journal of Sports Management and Motor Behavior, 5(10) (2009)

11. Razavi, S., Mohammad, H., Rahmani, M.: Strategies of New Technologies of Reference Location Information Systems (GIS). In: Comprehensive and Integrated Management of Sports Facilities and Potentials Abstract Proceedings of Mazandaran Sports Challenge and Opportunities Conference (2007)
12. Edward, P., Taurus, A.: Towards Healthy Cities: The Role of Local Governments in Promoting Physical Activity and Active Living in Urban Areas. Translated by: Dehghan Manshadi, Mehdi, Shahidi Publications (2008)
13. Salehi, R.: Spatial Organization of Educational Places in Zanjan Using GIS. Master Thesis in Urban Planning, Faculty of Geography, University of Tehran (2004)
14. Ghodsipour, S.H.: Hierarchical Analysis Process. Amir Kabir University of Tehran Press, Fifth Edition (2006)
15. Atshan, F.: Sports Places and Building. Textbook of the Deputy of Cultural Education of the Physical Education Organization (1997)
16. Ghahroudi, T., Manijeh, B.F.: Introduction to Geographic Information Systems (Geography). Payame Noor University, Azar (2012)
17. Pate, R.R.: A National Physical Activity Plan for the United States. Journal of Physical Activity and Health, 6 (2009)

ISSN 1870-4069

DIP-Toolbox: A Matlab Toolbox for Automated Image Band Registration and RGB Composition from Multispectral Images Obtained Using UAVs

Gabriela Rabelo Andrade [1,2], Fernanda Coelho Pizani[2,3], Daniel Henrique Carneiro Salim[2], Patrícia Corrêa Fonseca[1,2], Camila Costa de Amorim[2,3]

[1] Instituto Teia, R. dos Timbiras, Brazil

[2] P&D Aneel/Cemig GT-607, Brazil

[3] Federal University of Minas Gerais, Brazil

gabrielarabelo@gmail.com

Abstract. This work presents the DIP-Toolbox - an open toolbox, developed in Matlab for automated processing of multispectral images. In this paper, we introduce two tools - DIP-align and DIP-RGB, that perform automatic registration of misaligned bands of multispectral images and generate RGB compositions. The aim of the toolbox is to present a quick and easy tool for processing multispectral images that could be easily embedded in image processing workflows or other Matlab applications. The samples used in the development and testing of the toolbox have been obtained using UAVs flying at different altitudes, in the margins of a freshwater reservoir. We developed and calibrated the tools using images containing different proportions of water, land and vegetation.

Keywords: Multispectral images, image registration, image processing, matlab toolbox, UAV.

1 Introduction

UAV multispectral sensors have been used for many purposes and applied on different scales. Studies including [1, 2] have stressed that the capacity to capture data at exceptional spatial and temporal resolutions increases the range of solutions and opportunities for environmental research and remote sensing applications. Following, [1] compared to traditional airborne systems and satellites, UAVs offer high flexibility and versatility, and can be rapidly adapted to changing flight plans and/or schedules. Moreover, the quality and quantity of spatial data gathered in this fashion lead the decision-making process to be faster and more accurate. This is a great advantage for

quick-responses to time-crucial applications, e. g., water pollution, invasive macrophytes (aquatic plants) and algae bloom monitoring [3, 4].

Multispectral images captured by drones or other autonomous devices are usually composed of several bands, or scenes, each captured in a by a sensor and covering the same area. These scenes can be combined into RGB compositions or processed to generate indexes such as Normalized Difference Vegetation Index (NDVI), Visible Atmospherically Resistant Index (VARI), Green Atmospherically Resistant Index (GARI), Surface Algal Bloom Index (SABI), among many others [4-7]. Because they are captured by physically separate sensors, the bands of these images are offset relative to one another and need to be aligned before the generation of multiband compositions. The misalignment between bands generally varies according to the physical distance between the sensors and the distance between the camera and the photographed objects [8-10].

Processing images from either hyper or multispectral cameras coupled with drones can often be time consuming. When considering the time reduction for each stage essential in obtaining the final product, the benefits become clear, since the time reduction also culminates in cost reduction. Currently, the mechanisms used for the precise alignment of the spectral bands can be rudimentary and/or often rely on manual techniques or in the use of tools that are specific for each camera model or brand, when these exist. This scenario can be quite limiting for certain camera models or when batch processing is required. The automation of this step can make the processing more viable, resulting in a gain for the community that needs to deal with a large volume of images.

When processing images composed entirely of land, the correct alignment and orientation of the models occurs by obtaining precise coordinates where the connection between the support points registered in the field and the control information stored by the equipment is determined. This process is called phototriangulation or aero triangulation [11]. However, in images composed of various surfaces or composed entirely of water, this procedure can be conflicting and should be avoided due to the difficulty in detecting marked points that can be identified on the image [12].

A long drone flight (around 30 minutes) at 120m (height), with overlap and sidelap values of 80 and 70%, using a camera such as the MicaSense Altum can result in approximately 600 multispectral images, composed of 3600 files. It can be of great use to have the ability to quickly visualize RGB compositions of a certain area, without the need of advanced software, this could even be achieved whilst in the field to check areas which could merit further imaging.

DIP Toolbox (Available at: <github.com/gabrielarabelo/DIP/>) was created in order to provide tools for initial processing of images obtained by multispectral cameras, especially those commonly used in Drones. Toolbox was developed in Objective-C language in Matlab software and is freely available on the GitHub repository. A step-by-step video can also be seen on <youtu.be/9N0O5Os6J78>.

The registration of the images is done using the intensity-based image registration technique and the optimization parameters were adjusted from training images obtained in mixed areas on the edge of a lake, containing different proportions of water, land and vegetation. The optimized parameters were set as the program's default, but these can be changed by the user to adapt to any particular image set.

In this article, we will present the DIP_align tool, part of the Toolbox, which performs the following actions:

- Import all bands.
- Perform intensity-based image registration for all the bands.
- Crop image edges where not all bands overlap.
- Save files of the aligned bands.
- Generates RGB compositions from the selected bands using the encapsulated tool DIP_RGB.
- Enhances RGB compositions with the use of histogram correction, haze elimination and gamma correction tools.
- Save enhanced RGB composition files.

The main techniques used in the tool will be presented over the next sections.

2 Materials and Methods

2.1 Sample Images

The camera we use (MicaSense Altum) has 6 sensors which capture 6 16-bit monochromatic images of the same region, in different bands of the spectrum. For each shot, 6 files will generate - each with the information captured by a band (see Fig. 1).

The images used for calibration were obtained near the Três Marias Reservoir, in Brazil (-18.604158, -45.233450), using a multirotor NuvemUAV Spectral. The samples have been obtained during two field campaigns, and along eight days with variating atmospheric conditions.

The flights varied in duration (12 to 30 minutes), and in height (80m and 200m). After these two field campaigns we acquired over 9,000 images (over 56,000 files). From this set, we selected 50 samples, containing diverse features (land, trees, ground, water) in different proportions (varying from water only to land only), among other images such as those in which our boat could be seen or those with features of interest, such as aquatic plants or fish farms (see Fig. 2).

State of the Art Technique

In order to perform a comparison with state-of-the-art techniques, in this section we utilized one of the most traditional software in the field to register the bands of one of the images.

When processing is performed in ArcGIS on raw images, the method can be performed using the Georeferencing tool (ArcGIS v.10.5).

Initially, it is necessary to identify some reference points between the images and add control points that will guide the alignment. When choosing which layer to align, the points must be selected, one by one, in the first image and related according to the reference of the second image. After this process it is necessary to save the updates.

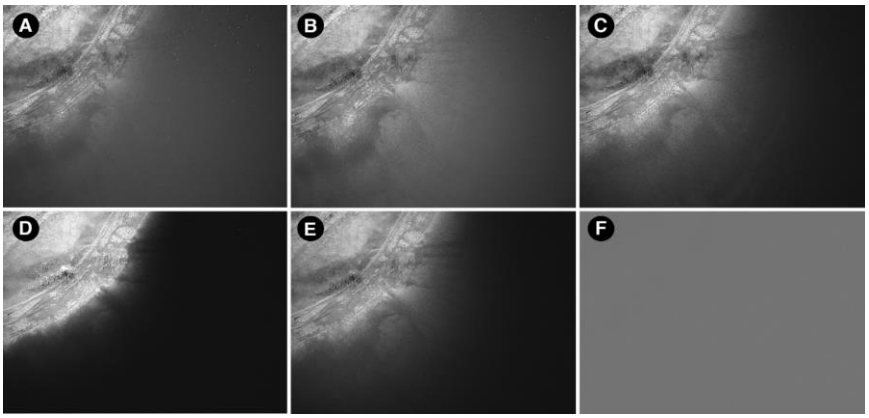


Fig. 1. Six bands of the same image captured by the camera MicaSense Altum, taken in a flight at a height of 120m. (A) Blue (475 nm center, 32 nm bandwidth), (B) Green (560 nm center, 27 nm bandwidth), (C) Red (668 nm center, 14 nm bandwidth), (D) Red Edge (717 nm center, 12 nm bandwidth), (E) Near-IR (842 nm center, 57 nm bandwidth), (F) LWIR thermal infrared 8-14um.

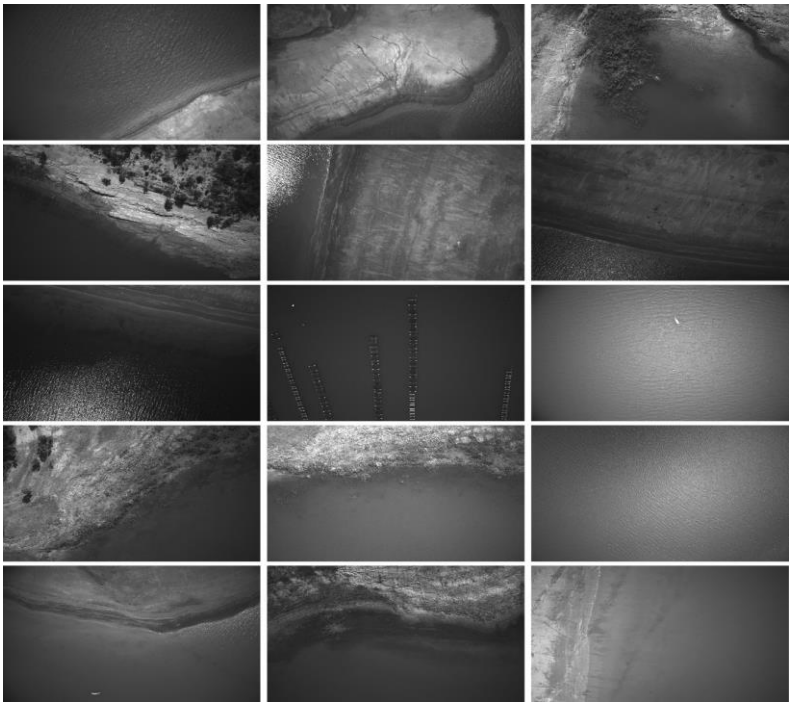


Fig. 2. Selected bands of some of the samples used in the calibration step.

Aligned, it is possible to perform the RGB composition using the tool available in *ArcToolbox* called *Composite Bands* and, thus, obtain the colored composition (see **¡Error! No se encuentra el origen de la referencia.**).

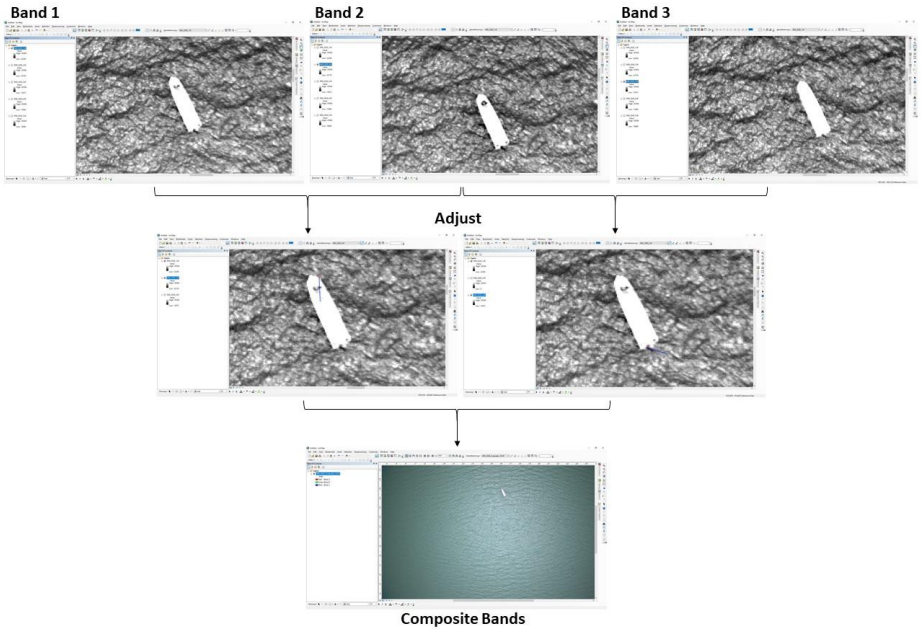


Fig. 3. Image Registration process using ArcGIS and the Georeferencing tool (ArcGIS v.10.5).

2.2 Band Register Alignment

The optimization methods and parameters were defined after a sequence of tests with images obtained in different areas, under different atmospheric conditions and through flights at different altitudes (see Fig. 2).

To align the bands obtained by different sensors, we used the intensity-based image registration technique, implemented through the function *imregister*, introduced in R2012a. The function *imregister* uses the optimizer, a metric, and a transformation type to find the search and find the best fit for each two images. The best results were achieved with the transformation of type 'rigid' and the optimizer and metric configures according to the following parameters:

```
InitialRadius = 0.0002

Epsilon = 0.0000015

GrowthFactor = 1,002

MaximumIterations = 500
```

The alignment is done two by two, from a band of reference. The best results for our calibration dataset were obtained using Band 2 (Green, 560 nm center and 27 nm bandwidth) as a reference for the alignment of the other bands (see Fig. 4).

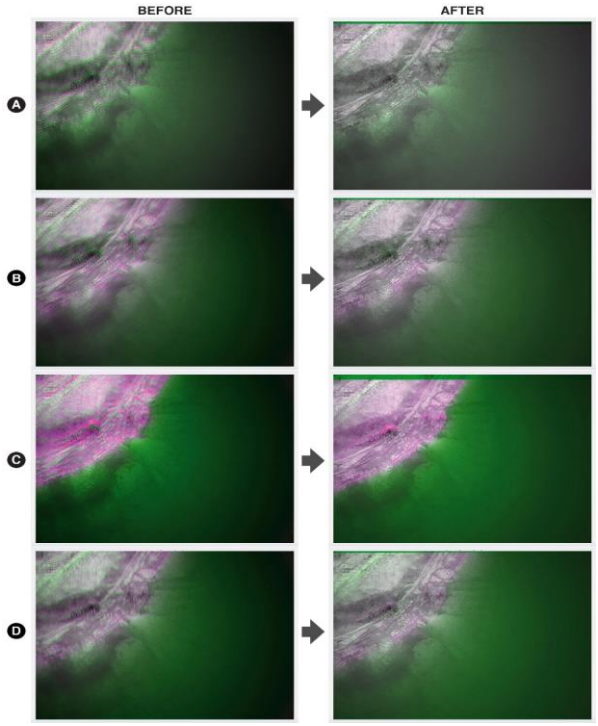


Fig. 4. Alignment of each band according to the reference band (band 2). (A) Bands 1 and 2. (B) Bands 3 and 2. (C) Bands 4 and 2. (D) Bands 5 and 2. As for MicaSense Altum the sixth band has a size much smaller than the others (160x120px, whereas the other bands are 2064x1544px), the alignment is not necessary.

Figure Fig**. 5** shows the example of an RGB image (generated from bands 3-2-1) before and after the alignment. As a last step after all the bands have been aligned, the program will crop the edges of the image that are missing one or more bands.

2.3 Image Color Enhancement

For each RGB composition, the program will plot and save extra images with enhanced colors. The color enhancement usually provides a better visualization of the spectral features, and it is especially useful for multispectral images. At its current stage, the program applies three different color enhancement transformations, coupled in two presets:

- Haze & Gamma Adjustment: this preset performs a combination of reducing Atmospheric Haze and Gamma correction (see Fig. 6A).
- Stretching Contrast Limits: this preset computes the lower and upper limits of each band and uses the values obtained for contrast stretching the RGB image (see Fig. **6**B).

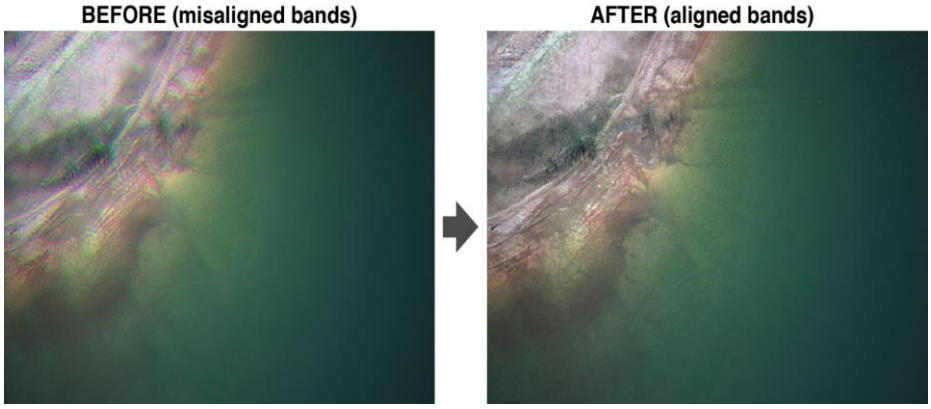


Fig. 5. RGB composition using bands 3-2-1 before and after the alignment of the bands.

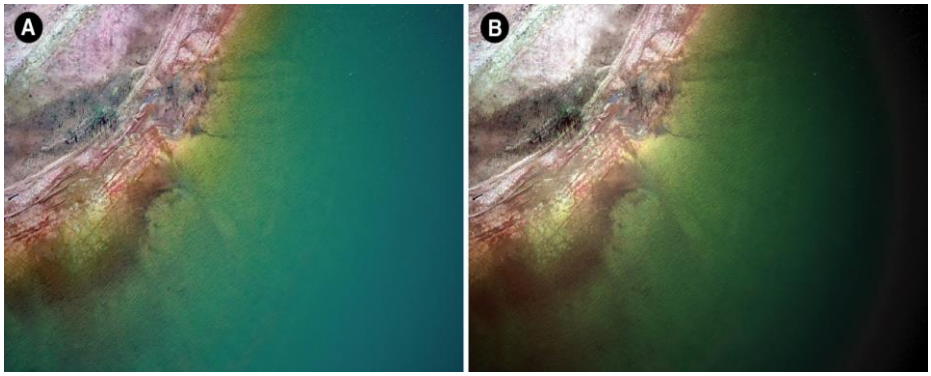


Fig. 6. Enhanced RGB (3-2-1) compositions using (A) Default Haze reduction and Gamma correction and (B) Stretching Contrast Limits.

2.4 Custom RGB Compositions

Once the bands have been aligned, the program proceeds to generate custom RGB images, composed from bands other than the corresponding Red, Green and Blue (see Fig. 7).

The band combinations can be inputted before calling the program function or after the initial RGB processing there will be a prompt for further custom combinations. For each composition, the program will automatically apply the color enhancement methods described in the previous section and save both images.

Custom RGB compositions can be useful for highlighting certain objects or features in images that may appear similar on the visible spectrum. In Fig. 8, a default RGB (3-2-1) composition is compared to a custom RGB composition (4-2-1). The latter highlights the contrast between the aquatic plants and the water.

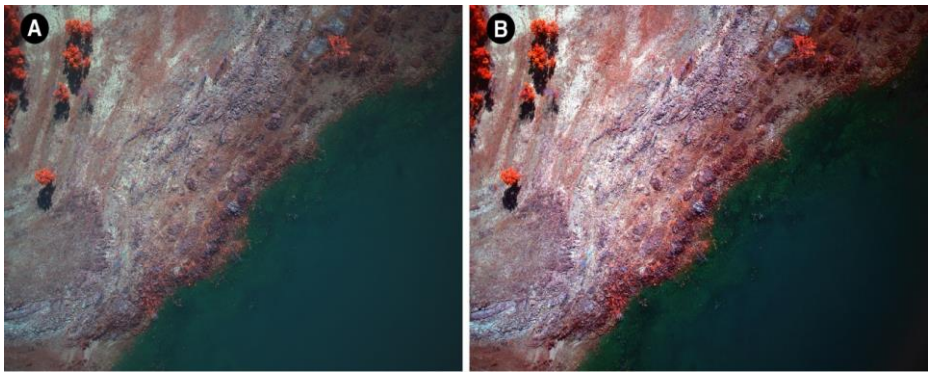


Fig. 7. Two RGB (4-2-1) compositions. (A) RGB (4-2-1) before enhancement. (B) Enhanced RGB (4-2-1) after using Stretching Contrast Limits.

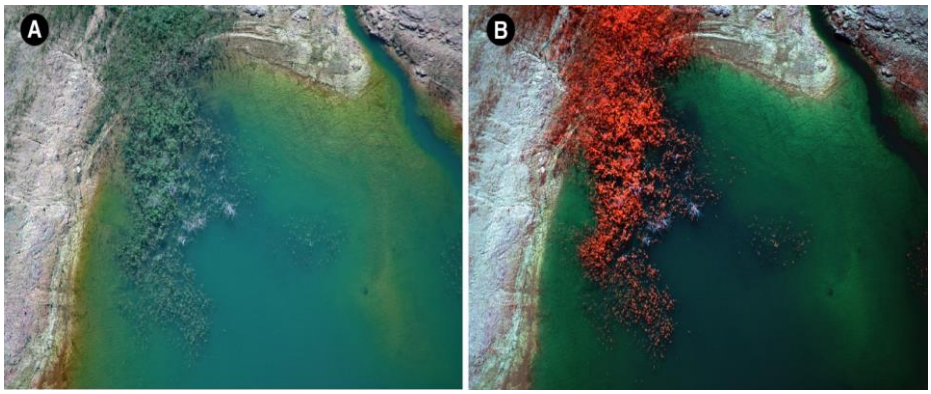


Fig. 8. Two RGB compositions from the same multispectral image. (A) Visible RGB (3-2-1) enhanced composition. (B) RGB (4-2-1) enhanced composition highlighting the presence of aquatic plants.

2.5 User Commands, Function Calling and Program Routine

The DIP_align tool can be used by calling the DIP_align function. This function has 1 optional input (parameters), which is a variable of type struct containing information about the camera, number of image bands, optimization parameters for image registration, and for image enhancement:

```
DIP_align
DIP_align(parameters)
```

The optional variables of the current version are presented in Table 1.

Each parameter presented in Table 1 should be declared as a field of the input variable parameters, as in the following example:

Table 1. Optional Input Parameters for DIP_align.

Parameter	Type	Description
nband	number (double)	Number of bands captured by the camera
camera	string	Camera model
band_specs	cell array	Cell array containing the specification of each band of the camera (automatically generated for Altum and RedEdge)
RGB_bands	3-column vector	Band sequence for traditional RGB composition. Default is 321.
customRGB	3-column vector or matrix	Band sequence for RGB prompt compositions that will be generated without user
customMode	logical	Opens a dialog box so the user can enter custom RGB compositions
InitialRadius	number (double)	Optimization parameter for *imregister*
Epsilon	number (double)	Optimization parameter for *imregister*
GrowthFactor	number (double)	Optimization parameter for *imregister*
MaximumIterations	number (double)	Optimization parameter for *imregister*
TransformType	string	Optimization parameter for *imregister*
ref_band_align	number (double)	Band to be taken as reference for the image registration
scale	number (double)	Factor to scale images (useful for images that are too big or for using orthomosaics)
skip_bands	vector (double)	Bands to be ignored during the image alignment (to be used with bands which the size is too different from the other bands)
haze_adj	number (double)	Number between 0 and 1 as input for *imreducehaze*
haze_adj_method	string	Method for *imreducehaze*
gamma_adj	number (double)	Number between 0 and 1 as input for gamma correction using *imadjust*

```
parameters              = struct;
parameters.camera       = 'altum';
parameters.customRGB  = [4 5 2; 4 2 1];
parameters.customMode = true;
```

After calling the function *DIP_align*, or *DIP_align(parameters)*, the program will start to run.

The user will first be prompted to select one band of the multispectral image to be aligned. Selecting only one band is enough and the program will automatically identify and import the other band files (as long as the file names follow the pattern *"_<band-number>"*).

The user will then be prompted to select a folder to save the output files.

The user will then be prompted to enter a custom name prefix for the output images that will be exported.

After that, the program will perform the image registration alignment and the program will plot the following:

- RGB image before the image registration.
- Before and After of the bands of the image being aligned.
- RGB image after the image registration.

The program will then plot and save high-resolution images of the RGB Compositions in the following sequence: Regular RGB; Haze & Gamma Adjusted RGB; Stretch Limits Adjusted RGB.

After that, in case the user selected the Custom Mode, the user will be prompted to enter any band combinations to generate other custom RGB compositions.

Finally, all the high-resolution images will be saved in the output folder.

3 Results and Conclusion

After the calibration step, the program managed to successfully align 40 of the 50 multispectral images of the sample set (see Fig. 9) using the default image registry optimization parameters of the program. However, in 10 of the sample images, one or more bands did not align correctly; these 10 images did not have any land in them. In all of the cases, the alignment of the band 4 (Red-Edge) was the most persistent issue. All 10 issues were resolved by either adjusting the *InitialRadius* to 0.00025 and or, changing the Reference Band (*ref_band_align*) to Band 5.

Although all of the image samples for this work were aerial images obtained with UAVs, it is very likely that the tools presented would work for other types of images obtained with multispectral cameras.

The computational toolbox resulting from this work is fully automated, which means it requires very little or no programming background, but still allows custom parameters, which means that it can be adapted to different areas and purposes. As the program relies on the visual information of the images it works with different camera models and manufacturers with consistent results. In the current version, the program requires a Matlab license to run but the program itself is open access.

4 Future Work

We are currently working on the improvement and expansion of the Toolbox and soon it will come equipped with tools for obtaining RGB compositions and some of the most common algorithms (such as NDVI, VARI, GARI, SABI) from multispectral orthomosaics, alongside other Tools. The tools presented will also be updated in order to automatically display band information from images obtained with other multispectral cameras and brands available on the market.

Acknowledgments. Our deepest gratitude to CEMIG for funding our research through The Research and Development Project P&D Aneel-Cemig GT-607, and allowing us the freedom to create this toolbox alongside our other works. Our warm thanks to Emma Jones for proofreading this paper.

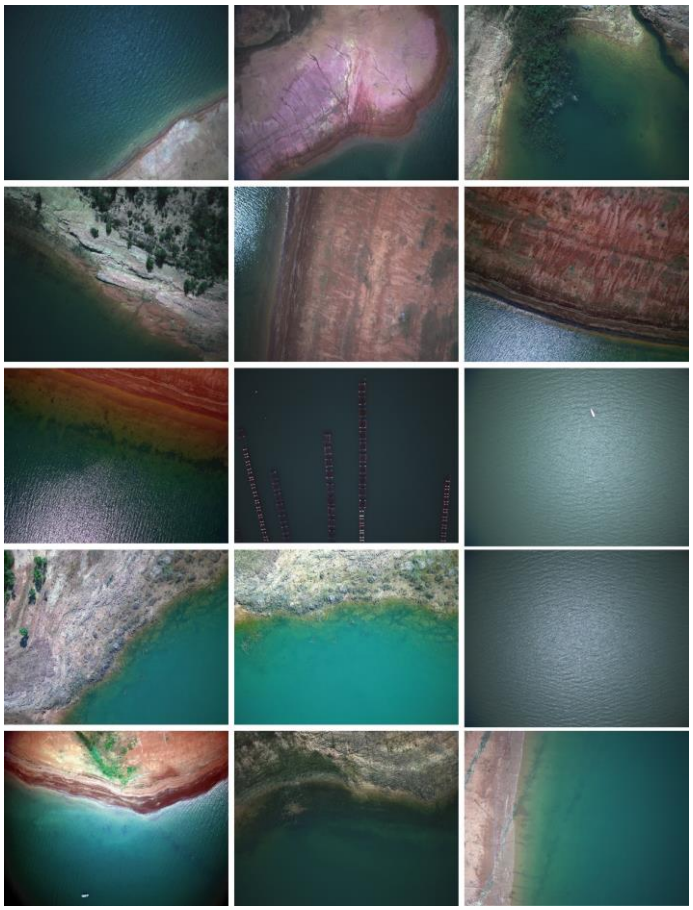


Fig. 9. A selection of some of the RGB (3-2-1) compositions obtained from the samples used for calibrating and testing the program.

References

1. Pajares, G.: Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). Photogrammetric Engineering & Remote Sensing. 81, pp. 281–329 (2015)
2. Yao, H., Qin, R., Chen, X.: Unmanned Aerial Vehicle for Remote Sensing Applications - A Review. Remote Sensing, 11, pp. 1443 (2019)
3. Flynn, K.F., Chapra, S.C.: Remote Sensing of Submerged Aquatic Vegetation in a Shallow Non-Turbid River Using an Unmanned Aerial Vehicle. Remote Sensing, 6, pp. 12815–12836 (2014)
4. Castro, C.C., Gómez, J.A.D., Martín, J.D.: An UAV and Satellite Multispectral Data Approach to Monitor Water Quality in Small Reservoirs. Remote Sensing, 12, 1514 (2020)
5. Eng, L.S., Ismail, R., Hashim, W.: The use of VARI, GLI, and VIGREEN Formulas in Detecting Vegetation in Aerial Images. pp. 1–10 (2019)
6. Rouse, J.W., Haas, R.H., Schell, J.A.: Monitoring Vegetation Systems in the Great Plains with ERTS, 351, 309 (1974)

7. Alawadi, F.: Detection of surface algal blooms using the newly developed algorithm surface algal bloom index (SABI). Presented at the Proceedings of the SPIE October (2010)
8. Jhan, J.-P., Rau, J.-Y., Huang, C.-Y.: Band-to-band registration and ortho-rectification of multilens/multispectral imagery: A case study of MiniMCA-12 acquired by a fixed-wing UAS. ISPRS Journal of Photogrammetry and Remote Sensing, 114, pp. 66–77 (2016)
9. Banerjee, B.P., Raval, S.A., Cullen, P.J.: Alignment of UAV-Hyperspectral Bands Using Keypoint Descriptors in a Spectrally Complex Environment. Remote Sensing Letters, 9, pp. 524–533 (2018)
10. Hassanpour, M., Dadras Javan, F., Azizi, A.: Band to Band Registration of Multi-spectral Aerial Imagery - Relief Displacement and Miss-Registration Error. ISPRS International Archives of the Photogrammetry, 4218, pp. 467–474 (2019)
11. Coelho, L., Brito, J.N.: Fotogrametria digital (2007)
12. Gülch, E.: Automatic Control Point Measurement. In: Fritsch, D. and Hobbie, D. (eds.) Photogrammetric Week, 95, pp. 185–196 (2001)

ISSN 1870-4069

Microalgae Cell Counting and Identification via Artificial Intelligence Techniques: An Interdisciplinary Approach

V. A. González-Huitrón[2], A.E. Rodríguez-Mata[1], L. Miranda [2], D.M. Arias[3], Lisbel Bárzaga-Martell[4], H. Rodríguez-Rangel[2,*]

[1] Tecnológico Nacional de México/IT de Chihuahua, Mexico

[2] Tecnológico Nacional de México/IT de Culiacan, Mexico

[3] Universidad Nacional Autónoma de México, Instituto de Energías Renovables Mexico

[4] Universidad de Chile, Departamento de Ingeniería Eléctrica, Chile

[*]Corresponding author: hrodriguez@itculiacan.edu.mx

Abstract. Artificial intelligence is currently being applied to many industrial processes. However, there is little work on the relationship between artificial intelligence and biotechnology areas oriented towards the cultivation of seaweeds, especially in the production of *Chlorella Vulgaris*. This microalgae is a genus of unicellular green algae of the phylum Chlorophyta. It is spherical in shape, measuring 2 to 10 micrometers in diameter, and has no flagellum, which has many applications in the food and pharmaceutical industry. This is why it is essential to experiment with these cells and at the same time develop technologies or tools that allow us to carry out projects or experiments in an easier way in order to speed up the results. This work proposes the development of software capable of speeding up and optimizing one of the many tasks carried out with these cells through artificial intelligence and its pattern detection methods based on autoencoders to count the cells present in a certain number of samples. An identification process of more than 95 % more effective in prediction and identification is achieved.

Keywords: Autoencoder, graph theory, pattern detection, microalgae.

1 Introduction

Microalgae are autotrophic photosynthetic microorganisms with a high diversity group. More than 40,000 species of eukaryotic microalgae have been explored

for their potential in food processing as they are a source of fatty acids and vitamins, in wastewater treatment, biofuels, and the production of high-end pharmacological products. *Chlorella Vulgaris* is a single-celled green microalgae present in some freshwater rivers of Mexico; with an important photosynthetic contribution in the waterways of Sinaloa [13]. This microalga is a source of proteins, omega-3 polyunsaturated fatty acids, polysaccharides, vitamins, and minerals, which provide several biological, industrial and pharmacological properties, many of which have been shown to be crucial in supplying human dietary deficiencies and providing an alternative for the treatment of some diseases. In fact, medical studies have shown that its use improves hyperlipidemia and hyperglycemia and protects against oxidative stress. Therefore, its cultivation and large-scale production, as well as research on its optimization, continues in an improvement process [17, 19, 18, 14].

One of the main issues related to microalgae optimizing production is an efficient estimation of the algae population. Microalgae counting is crucial to maintain control of biomass growth, prevent culture contamination by other species and other unfavorable conditions, and determine the level of lipids, carotenoids, proteins, and other substances of interest. Moreover, microalgae are also generally used as a bioindicator that provides information on water quality, so cell count is of great importance in the mass cultivation of Chorella. Therefore, specialized monitoring of microalgae is essential to up-scale *Chlorella Vulgaris* cultivation. The concentration of microalgae cells is the main and most used parameter in the cultivation of microalgae and ecological monitoring since it allows us to measure the optimal conditions of growth and growth and increase productivity while predicting any cell damage in real-time [3, 20].

Off-line techniques of microalgae counting include manual counting using a Neubauer counting chamber and expert judgment, adding high uncertainty to the measurement. The traditional detection process is cumbersome, laborious, and time-consuming, and it is impossible to analyze the concentration of algal cells quickly [7]. The automatic identification of microalgae samples is, therefore, a technical need to be solved since this would reduce decision-making times, in addition to the problem of reducing the impact on water quality. The identification and quantification of microalgae in water samples is made manually and at intervals in the Neabauer chambers, which is a slow and complex process.

The cell measurement and decision-making process can be optimized, leading to more reliable measurements using artificial intelligence techniques. Computer vision is a broad area of science that relies on techniques from various fields, such as pattern recognition, artificial intelligence, statistics, and machine learning. Automatic visual classification of objects is based on the geometric similarity of objects in several objects. It is the technical basis for implementing intelligent data and object recognition, including the classification and counting microalgae cells. The classifiers are designed manually under the knowledge and adherence to the experience of the expert trainer.

To improve cell counting, several remarkable works have studied the classification and cell quantification through microscope images, that is, from a

visual sample of the amplification of a photograph obtained from a microscope, automatic quantification can be made using as reference the Neubahuer counting chamber, and with this infer the total concentration in growth phase in a river or bioreactor [15]. Based on general handcrafted features, some classifiers have given good results in the implementation of computer vision techniques using neural networks for the classification of marine and freshwater microalgae and diatom cells [16, 2, 12, 1].

For instance, Luo and Gao [19] used artificial neural networks (ANN) with Fourier spectrum to identify diatoms, achieving 94% accuracy. Mosleh et al. (Ref 5.) used Fourier spectrum with principal component analysis (PCA) and ANN to classify microalgae *Navicula*, *Scenedesmus* and cyanobacteria *Microcystis Oscillatoria* and *Chroococcus* in river water, reaching 93% accuracy. Other work includes Cerbin et al. [3], who employed ANN to identify *Scenedesmus obliquus*, achieving 90% accuracy. Furthermore, recently, Giraldo-Zuluaga et al. ([20]) achieved accuracies of 98.63% and 97.32% with support vector machine and ANN, respectively, alternative automatic algae counting of *Scenedesmus* sp. However, there are not a study of quantification and classification of cells of the microalgae *Chlorella Vulgaris* through these models to the best of our knowledge [8, 9].

Autoencoders are a sort of artificial intelligence that functions as a filter, learning to rebuild the picture to be classed to calculate a percentage of similarity with the item to be discovered [21]. This type of artificial vision technique are implemented to identify cells need to collect data and identify patterns with the help of scientific data obtained by colloquial scientific methods. With this, the expert knowledge, to provide training of autoencoders [6].

In this work, we prepared an experimental technical methodology where we propose a system of counting and classification of cells of the microalgae *Chlorella Vulgaris* present in the rivers of Sinaloa, Mexico. We have made the extraction and isolation of the microalga *Chlorella Vulgaris*, as well as its production of controlled and automated rectangular photobioreactors in the laboratory, to implement an artificial vision and deep learning techniques for quantification and classification of *Chlorella Vulgaris* cells giving with this the design of a biomass sensor of the microalga through the use of the images obtained in a microscope.

For this, It was necessary to develop software that can detect the area of interest and, in turn, can perform the detection of the cells to count them subsequently. For this purpose, it was necessary to use image processing techniques, such as the Hough transform to detect straight lines and detect circumferences, edge detection, and artificial intelligence, such as autoencoders. The development of the project consists of 3 main parts, detection of cells or possible cells in the image, development and training of the auto-encoder that processes the possible cells found to determine whether or not it is a cell and the development of the user interface.

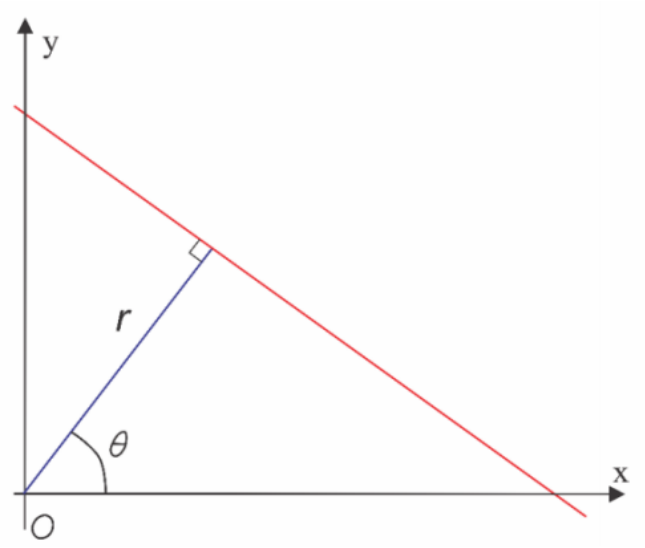


Fig. 1. Hough's T. Line Detection Graph.

2 Mathematical Background

2.1 Hough Transform

The Hough transform is an algorithm used in pattern recognition of an image, which allows finding shapes such as circles, lines, among others, within the image. The simplest version consists of finding lines, but depending on the image and the problem, it can be modified to see other types of shapes. The operation mode is statistical, and according to the points you have, you must find out the possible lines in which the issue can be achieved employing an operation applied to each line in a particular range.

Line detection. The Hough transform uses in its operation a parametric representation of geometric form, i.e., if it has a straight line, it would be represented as a line geometric parametric representation, i.e., if we have a straight line, this would be represented with the parameters *rho* and *theta* (Eq. (1)), where *rho* is the distance between the line and the origin, and *theta* is the angle of the vector from the origin to the nearest point, see Fig. 1. Employing the parameterization, the equation of the line could be written as follows:

$$\rho = x \cos\theta + y \sin\theta, \tag{1}$$

One of the characteristics that the Hough transform has is that if they are represented in a Cartesian plane, the line would be represented by a Cartesian plane, the line would be represented by the coordinates of the (ρ,θ), and the point

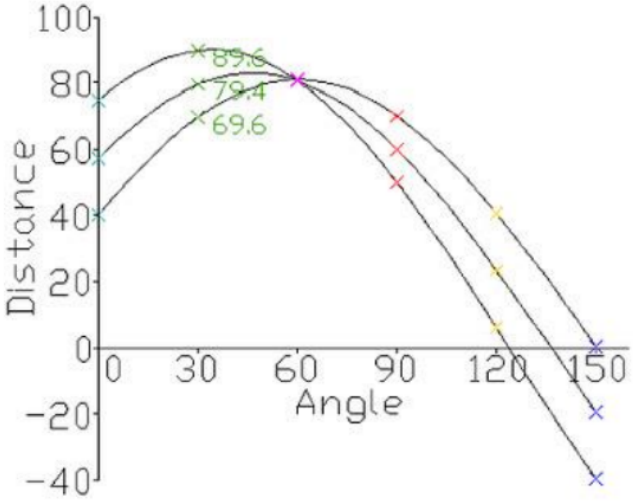


Fig. 2. Point intersection.

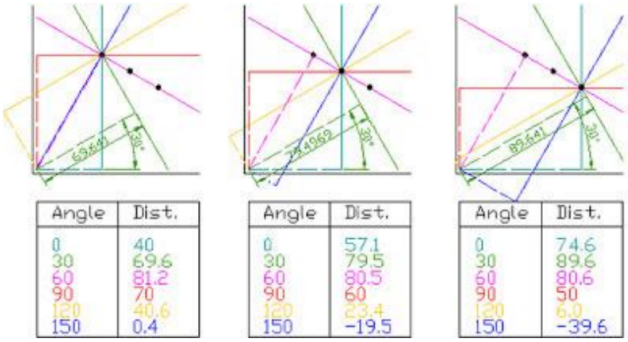


Fig. 3. Angles and distances of each line from origin.

would be represented as a sine function. Therefore, if you have two points, they would be symbolized by two sinusoidal symbols by means of two sinusoidal out of phase alpha degrees according to the coordinates of the points. Coordinates of the issues, but if the two points share the same straight line, the two sinusoids will end up crossing every 180 degrees since the sinusoidal function represents the set of infinite consecutive lines that pass through the point, as shown in Figs 2 and 3.

Circumference detection As previously mentioned, the Hough transform is not restricted only to line detection, although it is commonly used for that

purpose. Circumferences can be detected by applying the equation (2):

$$(x-a)^2 + (y-b)^2 = r^2. \tag{2}$$

Three parameters are necessary to describe a circumference:

- Axes of the center of the circumference (a,b).
- Radio (r).

To find circles using the Hough transform, an accumulator with three dimensions (a, b, r) is needed. Once this procedure is completed, the highest values in the accumulator are searched for and the radius and center of the circle are obtained. If the radius were known beforehand, only a two-dimensional accumulator would be needed.

2.2 Edge Detection by Canny's Method

The Canny algorithm is an operator developed by John F. Canny in 1986 that uses a multi-stage algorithm to highly detect the edges of objects contained in images. Canny's goal is to find the optimal edge detection algorithm. An optimal edge detector fulfills the following:

- Good detection: the algorithm should mark as many real numbers on the edges of the image as possible.
- Good location - the marker edges should be as close as possible to the edge of the actual image.
- Minimal response - The edge of an image should only be marked once, and whenever possible, image noise should not create false edges.

Canny use the calculus of variations to satisfy these requirements - a technique that finds the function that optimizes a given process. The optimal operation in Canny's algorithm is described by the sum of four exponential terms, but can be approximated by the first derivative of a Gaussian to satisfy these requirements; Canny uses the calculus of variations - a technique that finds the function that optimizes a given functional. The optimal operation in Canny's algorithm is described by the sum of four exponential terms but can be approximated by the first derivative of a Gaussian.

Noise reduction Canny's edge detection algorithm uses a filter based on the first derivative of a Gaussian. Since it is susceptible to noise present in raw image data, the original image is transformed with a Gaussian filter. The result is an image that is slightly blurred concerning the original version. This new image is not affected by a single pixel of noise to any significant degree.

Example of a 5x5 Gaussian filter:

$$B = \frac{1}{159}\left(\begin{bmatrix} 2 & 4 & 8 & 2 & 3 \\ 4 & 9 & 7 & 7 & 9 \\ 5 & 5 & 6 & 5 & 3 \\ 4 & 1 & 6 & 1 & 7 \\ 2 & 3 & 8 & 4 & 7 \end{bmatrix}\right) * A. \tag{3}$$

Finding the gradient intensity of the image The edge of an image can point in different directions, so Canny's algorithm uses four filters to detect horizontal, vertical, and diagonal blurred image edges. The edge detection operator returns a value for the first derivative in the horizontal direction (Gy) and the vertical direction (Gx). From this, the edge gradient and direction can be determined:

$$G = \sqrt{G_x^2 + G_y^2}, \tag{4}$$

$$\theta = \arctan \frac{G_y}{G_x}. \tag{5}$$

3 Methodology

Microalgae were grown at the laboratory level under controlled conditions. Samples were taken from the lower basin of the Culiacán river since it is the area with the most downslope and the lowest flow velocity. Samples were taken in a $3L$ Vandorn bottle, and the samples were preserved at 4^{o}C until they were analyzed in the laboratory *Chlorella Vulgaris* was identified, and by means of a Pasteur pipette, it was manually isolated in the culture medium. This operation was repeated until the algal biomass that reproduced corresponded only to *Chlorella Vulgaris*. The culture was grown in Suoeka medium [11] in a cylindrical photobioreactor with a 3-liter Airlift configuration (see figure 4). biomass, pH, ORP, dissolved solids, conductivity, and culture parameters were measured with LabGenius multiparametric equipment. After 15 days, it was possible to take samples in the laboratory and obtain Neubauer chamber photographs through microscope photography.

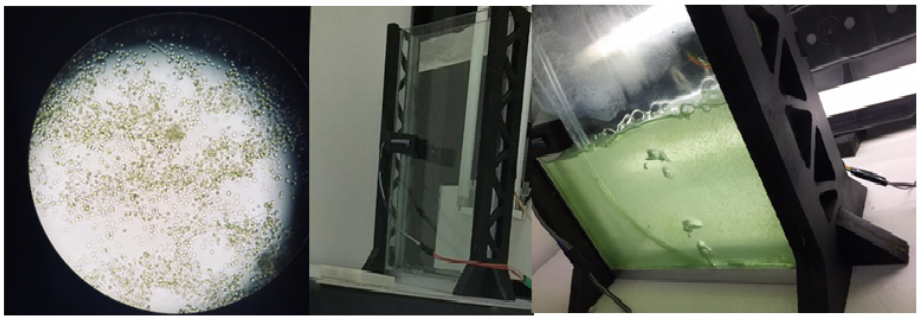


Fig. 4. Isolet *Chlorella Vulgaris.* and the photobioreactor used

Autoencoders are neural networks that aim to generate new data by first compressing the input into a latent variable space and then reconstructing the output based on the acquired information. This type of network consists of two parts: encoders and decoders.

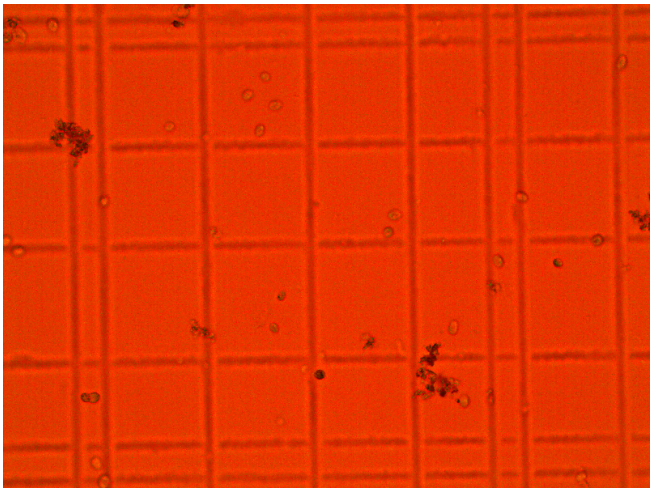


Fig. 5. Image to be processed through a microscope, for cell counting with a Neubauer camera.

4 Results

4.1 Image Segmentation and Detection of Possible Cells

Remarkable results were obtained in the detection of live *Chlorella Vulgaris* cells. It is essential to know the type of image to be processed in order to be able to make the detections correctly. In this case, we have the images as in Figure 5. To perform the analysis, we isolated and delimited the image to our area of interest, using the image processing methods of Canny and edge detection and the Hough Transform with the help of artificial vision techniques [5].

It is worth mentioning that for the Hough transform to work, it is necessary to pre-process the images to detect the edges of the objects so that the Hough transform function can perform its straight line detection. Figure 6 shows an example of how edge detection is done.

Figure 7 shows the example of line detection which helps us to determine the area of interest, as well as to do the cropping and segmentation of the image. To eliminate these parts, we generate four binary masks made from the detection of the lines at the boundary of the area of interest.

Once the multiplication of our binary mask by the original image is done, we will obtain something like in Figure 8 and then proceed to detect our viable cells employing the Hough Transform again, repeating all the previous process, but, now to notice circumferences. [10].

We are applying the Canny edge detection method as shown in Figure 9.Ṫo finish this process, we use the Hough transform for circumferences to detect viable cells within our image, cut out small images of these feasible cells, and then pass them to our neural network to determine if it is a cell or not. Figure 9

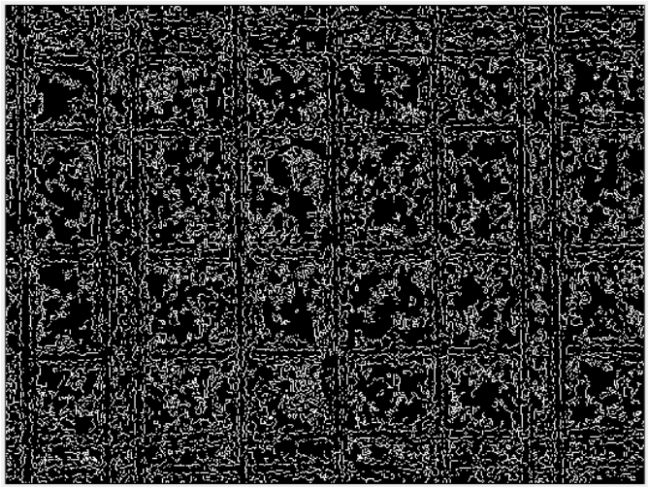


Fig. 6. Image with detected edges.

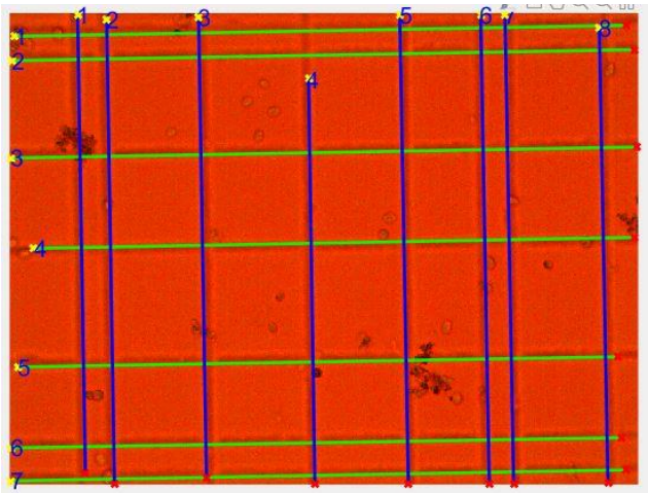


Fig. 7. Image with detected lines.

shows the result of the possible circumferences found in the image, and Figure 10 shows the cut out of an objective cell that will be sent to the neural network to be processed and analyzed.

4.2 Autoencoder development and training

The development of our auto-encoder was based on one made by the developers of Keras [4]; it was only modified and adjusted in terms of information retention

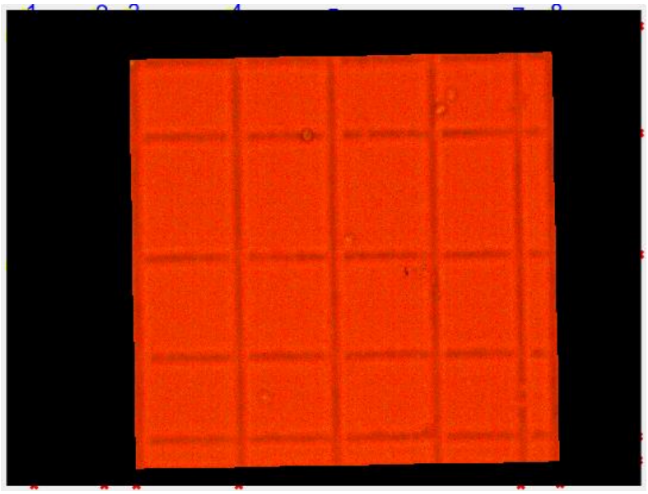


Fig. 8. Image with region of interest detected.

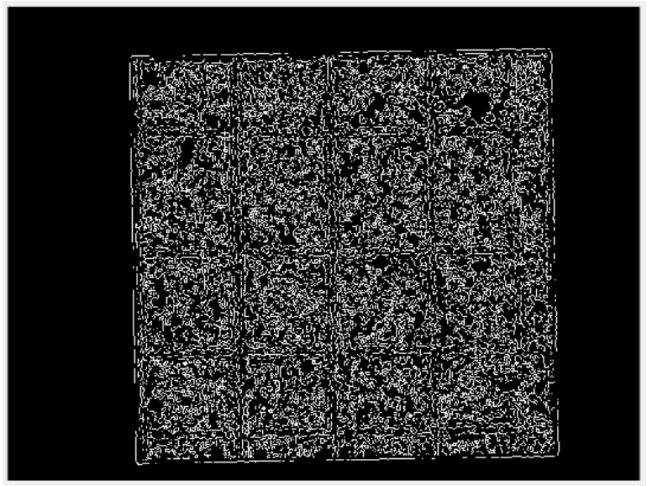


Fig. 9. Image of the area of interest with Canny.

parameters, number of training epochs, etc.

To train the neural network, it was necessary to rely on expert knowledge in cell detection to provide the auto-encoder with accurate information for training. Figure 9 is the type of image that was provided to the network with a cell for training.

It was necessary to have a database of authentic positive and accurate negative images so that the auto-encoder could learn to differentiate between the two. The true-positive images were used for training, while the true-negative

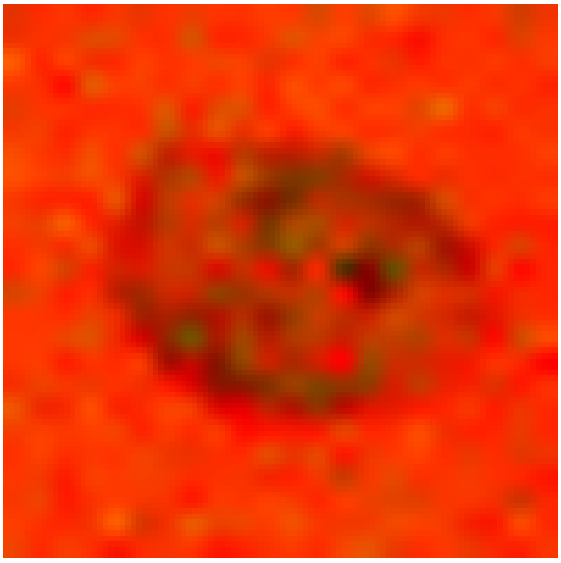


Fig. 10. Image of a real cell with digital zoom.

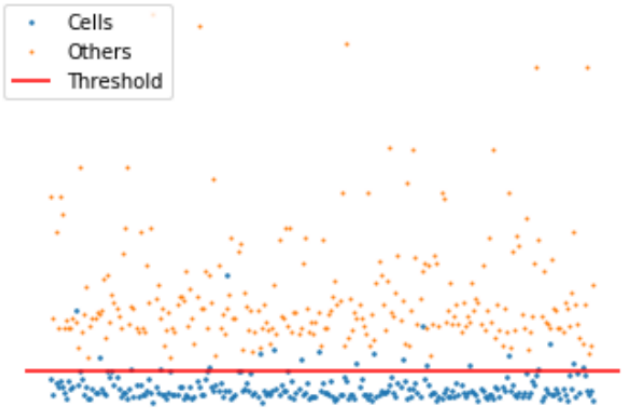


Fig. 11. Neural network prediction results.

images were used to test the network to see if it could detect. These tests yielded very favorable results.Considering that out of a database of 499 images (with 251 negative and 248 positive images), the neural network could recognise 100% of negative images as true negative and 95.76% of positive images as true positive say that the network is highly effective in detection.

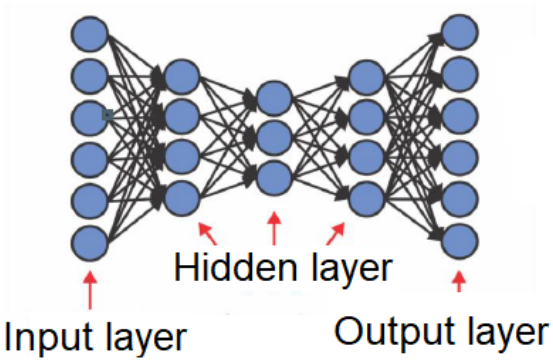


Fig. 12. Autoencoder Conceptual structure.

5 Conclusions

Based on the results obtained, it can be shown that the application of artificial intelligence for various tasks such as optimising time and processes, as well as automating tasks in which only repetitive and little changing work is needed. In this case the development of this platform makes automation possible, which in turn also improves the working conditions for researchers in the field of biology (microalgae research). It could be quantitatively demonstrated that the autocoder can estimate and detect *Chlorella Vulgaris* cells with high accuracy.

6 Futures Works

We are currently developing and designing the user interface, and the expert can give us the guidelines on how it should be managed and planned. This interface is being developed with the MATLAB AppDesigner application.

Acknowledgments. This project belongs to a project of the Tecnologico Nacional de Mexico *"Estimacion en la recuperacion de nutrientes para produccion de microalgas monitoreadas a traves de observadores de alta ganancia. Un enfoque multidiclinario"* with code 1229221-P, in the year 2021-2022.

References

1. Baek, S.-S., Pyo, J., Pachepsky, Y., Park, Y., Ligaray, M., Ahn, C.-Y., Kim, Y.-H., Chun, J. A., Cho, K. H.: Identification and enumeration of cyanobacteria species using a deep neural network. Ecological Indicators, vol. 115, pp. 106395 (2020)
2. Bueno, G., Deniz, O., Pedraza, A., Ruiz-Santaquiteria, J., Salido, J., Cristóbal, G., Borrego-Ramos, M., Blanco, S.: Automated diatom classification (part a): handcrafted feature approaches. Applied Sciences, vol. 7, no. 8, pp. 753 (2017)

3. Cerbin, S., Nowakowski, K., Dach, J., Pilarski, K., Boniecki, P., Przybyl, J., Lewicki, A.: Possibilities of neural image analysis implementation in monitoring of microalgae production as a substrate for biogas plant. In: Fourth International Conference on Digital Image Processing (ICDIP 2012). vol. 8334, pp. 83342A. International Society for Optics and Photonics (2012)
4. Chollet, F.: Building autoencoders in keras. The Keras Blog, vol. 14 (2016)
5. Fokkinga, M.: The hough transform. Journal of functional programming, vol. 21, no. 2, pp. 129 (2011)
6. Hatipoglu, N., Bilgin, G.: Cell segmentation in histopathological images with deep learning algorithms by utilizing spatial relationships. Medical & biological engineering & computing, vol. 55, no. 10, pp. 1829–1848 (2017)
7. Liu, J.-Y., Zeng, L.-H., Ren, Z.-H.: The application of spectroscopy technology in the monitoring of microalgae cells concentration. Applied Spectroscopy Reviews, pp. 1–22 (2020)
8. Luo, Q., Gao, Y., Luo, J., Chen, C., Liang, J., Yang, C.: Automatic identification of diatoms with circular shape using texture analysis, (2011)
9. Mosleh, M. A., Manssor, H., Malek, S., Milow, P., Salleh, A.: A preliminary study on automated freshwater algae recognition and classification system. In: BMC bioinformatics. vol. 13, pp. 1–13. BioMed Central (2012)
10. Olijve, L. L., Oude Vrielink, A. S., Voets, I. K.: A simple and quantitative method to evaluate ice recrystallization kinetics using the circle hough transform algorithm. Crystal Growth & Design, vol. 16, no. 8, pp. 4190–4195 (2016)
11. Ortiz-Moreno, M. L., Cortés-Castillo, C. E., Sánchez-Villarraga, J., Padilla, J., Otero-Paternina, A. M.: Evaluación del crecimiento de la microalga chlorella sorokiniana en diferentes medios de cultivo en condiciones autotróficas y mixotróficas. Orinoquia, vol. 16, no. 1, pp. 11–20 (2012)
12. Pedraza, A., Bueno, G., Deniz, O., Cristóbal, G., Blanco, S., Borrego-Ramos, M.: Automated diatom classification (part b): a deep learning approach. Applied Sciences, vol. 7, no. 5, pp. 460 (2017)
13. PÉREZ-BRAVO, S. G., MENDOZA-MARTÍNEZ, A. M., CASTAÑEDA-CHÁVEZ, M. d. R., AGUILERA-VÁZQUEZ, L.: Bioenergía a partir de microalgas en méxico,
14. Ru, I. T. K., Sung, Y. Y., Jusoh, M., Wahid, M. E. A., Nagappan, T.: Chlorella vulgaris: A perspective on its potential for combining high biomass with high value bioproducts. Applied Phycology, vol. 1, no. 1, pp. 2–11 (2020)
15. Ruiz-Santaquiteria, J., Bueno, G., Deniz, O., Vallez, N., Cristobal, G.: Semantic versus instance segmentation in microscopic algae detection. Engineering Applications of Artificial Intelligence, vol. 87, pp. 103271 (2020)
16. Schulze, K., Tillich, U. M., Dandekar, T., Frohme, M.: Planktovision-an automated analysis system for the identification of phytoplankton. BMC bioinformatics, vol. 14, no. 1, pp. 1–10 (2013)
17. Shah, M., Mahfuzur, R., Liang, Y., Cheng, J. J., Daroch, M.: Astaxanthin-producing green microalga haematococcus pluvialis: from single cell to high value commercial products. Frontiers in plant science, vol. 7, pp. 531 (2016)
18. Sharma, Y. C., Singh, B., Korstad, J.: A critical review on recent methods used for economically viable and eco-friendly development of microalgae as a potential feedstock for synthesis of biodiesel. Green chemistry, vol. 13, no. 11, pp. 2993–3006 (2011)

19. Sudhakar, M., Kumar, B. R., Mathimani, T., Arunkumar, K.: A review on bioenergy and bioactive compounds from microalgae and macroalgae-sustainable energy perspective. Journal of Cleaner Production, vol. 228, pp. 1320–1333 (2019)
20. Tarno, H., Qi, H., Endoh, R., Kobayashi, M., Goto, H., Futai, K.: Types of frass produced by the ambrosia beetle platypus quercivorus during gallery construction, and host suitability of five tree species for the beetle. Journal of Forest Research, vol. 16, no. 1, pp. 68–75 (2011)
21. Wajeed, M. A., Sreenivasulu, V.: Image based tumor cells identification using convolutional neural network and auto encoders. Traitement du Signal, vol. 36, no. 5, pp. 445–453 (2019)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación