ISSN 1870-4069

A Local Image Feature Approach as a Step of a Top-Down RGBD Semantic Segmentation Method

Gerardo Ibarra-Vázquez, Cesar A. Puente-Montejano, José I. Nuñez-Varela

Universidad Autónoma de San Luis Potosí, Facultad de Ingeniería,
San Luis Potosí, Mexico
gerardo.ibarra@alumnos.uaslp.edu.mx, {cesar.puente,jose.nunez}@uaslp.mx

Abstract. Semantic segmentation is a per-pixel class labeling problem, a method to assign a class label from a set of classes to each pixel on an image. Recent works have shown great progress using RGB images. However, indoor environments is still a challenging problem for the state-of-the-art algorithms due to the high variability of the scenarios. Additionally, semantic segmentation architectures based on Convolutional Neural Networks have reported to be vulnerable to adversarial attacks. However, elements of these architectures are similar to hand-crafted features and pipelines used in computer vision. In this paper, we explore the use of local image features by making an analysis and proposing a combination of feature detectors to make a robust classification of these features. This approach is a step of a top-down RGBD semantic segmentation method. Experiments on indoor environments show that the mean classification accuracy of feature descriptors can be improved by up to 3.3% with respect to the performance of a single feature detector. Also, using a balanced dataset an applying a cross-validation technique could improve up to 5.5% of the average of mean accuracy, obtaining better performance than just applying a single feature matching algorithm.

Keywords: local image features, feature detectors, feature descriptor, semantic segmentation.

1 Introduction

Semantic segmentation is a per-pixel class labeling problem, a method to assign a class label from a set of classes to each pixel on a RGB and RGBD (RGB+Depth) image [8]. Computer vision has tackled semantic segmentation on both RGB and RGBD perspectives. RGBD approaches introduced RGBD cameras to assist indoor scene segmentation by increasing the capabilities of getting the shape and spatial information from a depth image [10,11,26,27]. Deep learning has achieved very good results and RGB approaches for semantic segmentation have converged to encoder-decoder architectures based on Convolutional Neural Networks (CNN's) [3,7,17,18]. However, when they have been tested using indoor environments, they have experienced a lack of performance. For example, Badrinarayanan et al. [3] proposed the first encoder-decoder architecture for semantic

pp. 35–47; rec. 2019-06-13; acc. 2019-08-10

segmentation. They reported for an outdoor experiment using Camvid dataset 60.10% of mean Intersection-over-Union (mIoU), an evaluation metric that gives the similarity between the predicted region and the ground-truth. Meanwhile, for indoor environments using the SUN RGBD dataset, they reported 31.84% mIoU. This drop in performance could be explained by the high variability of indoor scenarios. Furthermore, adversarial attacks is a problem recently reported that affect CNN's and encoder-decoder architectures [2,20,31]. Adversarial attacks can be generated through a variety of forms, including making small modifications to the input pixels, using spatial transformations, or by a simple guess and check to find misclassified images.

Hand-crafted representations such as local image features have been widely used in a large variety of computer vision tasks. In specific, they were used before deep learning approaches emerged with excellent results on image classification [23,28]. However, CNN's have outperformed the results made by local image features. Notably local image features and pipelines used in computer vision can be seen as corresponding to layers of a standard CNN. Also, it has been reported in [26] the use of local image feature for an indoor scene segmentation approach using RGBD information. So, the question we address in this paper is whether it is possible to improve the performance of local image features by combining feature detectors and after answering this question build a top-down semantic segmentation method using RGBD images.

2 Local Image Feature Approach

Figure 1 presents an initial proposal of a top-down RGBD semantic segmentation method. Within this proposal, the focus of this work is on the local image feature approach (see Figure 1 yellow rounded box). Local image features are used due to its low computational cost, and robustness to changes on the scale, rotation, viewpoint change, blur and in some cases to lighting conditions. These features are used to influence the perception of the top-down process, recognizing and localizing points over the image. This approach is followed by a region growing segmentation process (currently under development) on the depth image, where the object shapes would be extracted to complete the semantic segmentation method.

Therefore, the contribution of this paper is the analysis of local image features and how a *combination* of feature detectors can improve the classification accuracy of features descriptors (see Figure 1 yellow rounded box). Local image features can be seen as a two-part process: *detection* and *description* (see Figure 1). Feature detection refers to the process of selecting regions or interest points in an image that have unique content, such as edges, corners, ridges or blobs [25]. These interest points can be used for further processing. Feature description involves computing a descriptor, which is typically done on regions centered around the feature detector. A descriptor is a compact vector representation of a local pixel neighborhood around an interest point. A histogram of the image gradients of a region centered on a point is an example of a descriptor. Thus, this

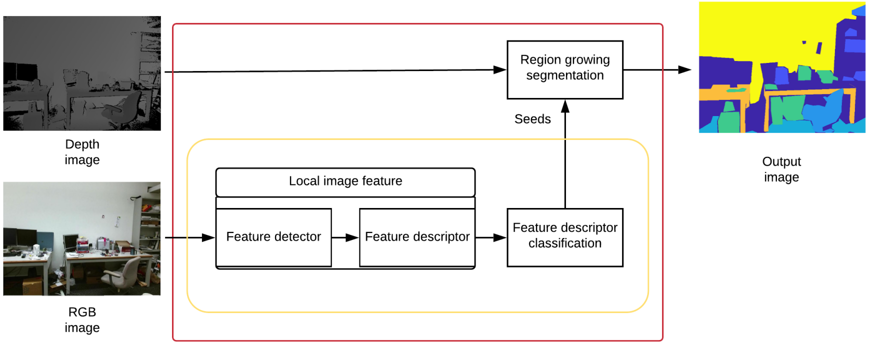


Fig. 1. The semantic segmentation method (red rectangle box), starts by using the local image feature approach by computing the feature detectors on the RGB image and constructing a feature descriptor on each one of the detectors. Afterwards, a classifier predicts a class of the local image features, which in turn becomes the input for the segmentation process (not reported in this paper).

work made an analysis focused on the *combinations* of feature detectors (e.g., regions, points, or corners) which are supposed to have a strong response to a series of filters in either spatial or frequency domains.

2.1 Feature Detectors

Two research works were followed to check their analysis of the current most common feature detectors [21,25]. The first work compares the invariance of feature detectors to the rotation, scale, and affine transformations, as well as some qualities such as repeatability, localization, robustness, and efficiency (see [25] to check such comparison). In terms of invariance, Maximally Stable Extremal Regions (MSER) [6], and Salient Regions (SR) [1], are invariant to all image transformations. The most common used Scale-Invariant Feature Transform (SIFT) [19], Speeded-Up Robust Features (SURF) [4], and Binary Robust Invariant Scalable Keypoints (BRISK) [16] detectors are invariant to rotation and scale. Corner detectors like Harris [13] and Features from Accelerated Segment Test (FAST) [24], are only invariant to rotation. In terms of qualities, MSER has a good performance on all the qualities mentioned above as well as the Harris corner detector. SIFT has a good performance on robustness and localization, while SURF has a good performance on efficiency and localization. Even though SR is invariant to all the transformations, it has poor performance on all the mentioned qualities.

Runtime performance is also taken into consideration. Mikolajczyk et al. [21] analyzed runtime performance and the number of regions for a test image of size 800x600 pixels. The shortest runtime was 0.66 seconds obtained by MSER. The longest was SR with 2013.89 seconds. Harris-Affine achieved the second fastest time with 1.43 seconds. Hessian-Affine have the third fastest run time with 2.73 seconds. Hence, based on the results reviewed above, we decided to use the following four feature detectors:

- **Maximally Stable Extremal Region (MSER)** [6], extracts from an image a number of co-variant regions, called MSERs. An MSER is a stable connected component of some sets of gray-level pixels of the image.
- **Harris Corner** [13], finds corner points using the Harris-Stephens algorithm which considers the differential of the corner score with respect to direction directly.
- **Features from Accelerated Segment Test (FAST)** [24], is a corner detection method that uses a circle of 16 pixels (a Bresenham circle of radius 3), to classify whether a candidate point is actually a corner.
- **Binary Robust Invariant Scalable Keypoints (BRISK)** [16], is a novel scale-space FAST-based detector in combination with a bit-string descriptor obtained from intensity comparisons retrieved by dedicated sampling of each keypoint neighborhood.

2.2 Feature Descriptors

A descriptor is a compact vector representation of a local pixel neighborhood around an interest point or a region. This vector can be constructed using a feature detector as an input to build a representation of the surrounding pixels. Speeded Up Robust Features (SURF) [4] has a detector and a descriptor part. This paper uses the SURF descriptor in a combination of the feature detectors listed above[1]. SURF descriptor is extracted constructing a square region centered around the interest point (given by a feature detector), and oriented along an assigned orientation. The size of this region is $20s$, where s is the scale in which the interest point was found from a scale-space extrema detection. The orientation is computed with a sliding orientation window that detects the dominant orientation of a Gaussian weighted Haar wavelet.

The following procedure describes how the SURF descriptor is calculated from each one of the four detectors used in this work. A SURF descriptor is computed from MSER using a circle representing the feature with an area proportional to the MSER ellipse area. This area is needed to approximate a scale to construct the descriptor and is computed in terms of the ellipse's axes. The scale value must be greater or equal to 1.6 as is needed by the SURF descriptor [5]. Therefore, the MSER ellipse area is saturated to 1.6. The SURF descriptor orientation uses the MSER orientation directly. Since Harris, FAST and BRISK, generate interest points rather than regions, the minimum scale value of 1.6 was used to construct the square region on which the descriptor would be extracted. Since these interest points do not have an orientation assigned, thus the upright orientation was chosen.

[1] The SURF descriptor was compared against *Binary Robust Invariant Scalable Keypoint* (BRISK), *Fast Retina Keypoint* and *KAZE* descriptors in an experiment not reported in this paper. SURF descriptor showed the best results.

2.3 Classification Algorithms

Classification algorithms are needed to predict a class for each feature descriptor on the local image feature approach. In these experiments four different machine learning classifiers were used and are described next:

- **Probabilistic Neural Network (PNN)**: Is a two-layer network where the first layer computes the distance from the input vector to the training vectors of each class and the second layer produces an output vector of probabilities with the sums of the contributions of each class. The maximum value of these probabilities is chosen as the predicted class. The Euclidean distance computed from the center point of the training vectors is approximated applying a radial basis function using a sigma value.
- **Support Vector Machines (SVM)**: Builds a model that assigns new examples to one category or another. In this paper, a multi-class model for SVM is used that utilizes an Error-Correcting Output Codes (ECOC) model. ECOC reduces the problem of classification with three or more classes to a set of binary classifiers. Additionally, it uses $K(K–1)/2$ binary SVM models using one versus one coding design, where K is the number of classes.
- **Deep Neural Networks (DNN)**: Is an artificial neural network of multiple processing layers that learns representations of data with multiple levels of abstraction.
- **Feed-Forward Neural Network (FFNN)**: Is an artificial neural network composed typically of three layers: an input layer, a hidden layer and an output layer. A specific implementation of this algorithm is used for classification of local image features for semantic segmentation in [26]. It is used as the appearance model for the unary potential function of a Conditional Random Field (CRF) algorithm.

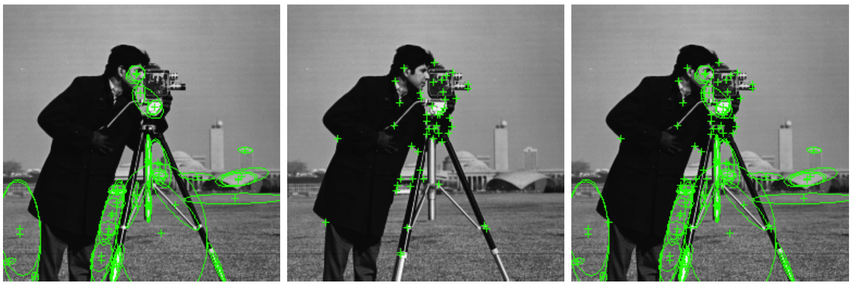
2.4 Dataset

A state-of-the-art dataset was chosen to test the combination of feature detectors and the classification algorithms. The SUN RGBD dataset [29] from Princeton University was selected. This dataset contains 10,335 RGBD images of indoor scenarios such as bedroom, furniture store, office, among others. Four different sensors (Intel RealSense, Microsoft Kinect v1 and v2, and Asus Xtion), were used to capture the images. The dataset contains annotations in 2D and 3D for both, objects and rooms. The dataset is composed of the NYU depth v2 [27], Berkeley B3DO [14], and SUN3D [30] datasets. Also, it provides benchmarks on six important tasks: Scene categorization, semantic segmentation, object detection, object orientation, and room layout estimation.

Furthermore, to complement the experiment of local image features and check the behaviour of the classification algorithms to balanced data. We constructed a balanced set of images from the RGBD Object Dataset [15]. This dataset contains 300 common everyday objects from multiple view angles, totaling 250,000 RGBD images organized into 51 categories. The objects are commonly found in indoor environments, such as homes and offices. As each object category has different numbers of objects, we selected specific categories and build a balanced dataset.

3 Experiments and Results

The main idea presented in this paper is the combination of feature detectors to improve the classification accuracy of feature descriptors. An example is shown in Figure 2, where it is observed a combination of two feature detectors, MSER and FAST, on a test image. Different feature detectors are observed on the same object, e.g. the tripod which contains detectors of both types. This could allow having different starting points for a segmentation algorithm.



MSER FAST MSER+FAST

Fig. 2. Example showing the combination of two feature detectors (MSER and FAST) using the "cameraman" image.

In terms of the dataset, SUN RGBD provides two ground truth sets for all images, one using 37 classes and another one using 6,590 classes. Handa et al. [12] proposed the ground truth of semantic segmentation for a variety of datasets to standardize benchmarks between datasets. They proposed 13 classes for the SUN RGBD dataset (see Table 1). Rather than using the classes as defined by the SUN RGBD dataset, we followed the classes defined by Handa et al. For our experiments we manually selected a subset of 200 RGBD images for training and 93 images for testing from the SUN RGBD dataset (see Figure 1). All selected images belong to an *office* scenario.

Table 1. 13 classes defined by Handa et al. for the SUN RGBD dataset scenarios [12].

Number	1	2	3	4	5	6	7	8	9	10	11	12	13
Name	Bed	Books	Ceiling	Chair	Floor	Furniture	Objects	Picture	Sofa	Table	TV	Wall	Window

For our experiments, we used a computer with an Intel Xeon E5-1620 processor and 48GB of RAM. The hyperparameters for the classification algorithms were set as follows:

– **PNN**: $\sigma = 0.025$.
– **SVM**: One vs one coding, linear kernel function, kernel scale of 1, polynomial order of 3, and iterations limit of 1×10^6.
– **DNN**: An input layer, 3 fully connected layers of 1000, 500, 50 neurons and a softmax output layer. It was trained using stochastic gradient descent with momentum, an initial learning rate of 0.01, a maximum of 20 epochs and mini-batch size of 250.
– **FFNN**: An input layer, a fully connected layer of 1000 neurons and a softmax output layer. It was trained using stochastic gradient descent with momentum, initial learning rate of 0.01, maximum epochs of 20, and mini batch size of 250.

Table 2. List of feature detectors combinations and the mean accuracy over the 13 classes for each classification algorithm.

Experiment		Mean accuracy			
Feature detectors combinations	Descriptor	PNN	SVM	DNN	FFNN
MSER	SURF	21.5%	28.5%	27.8%	26.7%
MSER + Harris	SURF	21.5%	30.2%	29.4%	28.7%
MSER + FAST	SURF	21.3%	**30.6%**	29.4%	29.9%
MSER + BRISK	SURF	NA	30.4%	**30.2%**	28.6%
MSER + Harris + FAST + BRISK	SURF	NA	**30.6%**	30.1%	**30.0%**

Experiment results and the list of feature detector combinations are shown on Table 2. It is seen that there was no improvement using the PNN algorithm with any feature detector combinations. The mean accuracy obtained over the 13 classes is 21.5% for MSER and MSER + Harris combinations. Whereas for the MSER + FAST combination the mean accuracy is 21.3%. It should be mentioned that it was not possible to process MSER + BRISK and the combination of all detectors because the training data was too large and it was not possible to process using the PNN algorithm. On the other hand, SVM shows an improvement for all feature detectors combinations. MSER + Harris obtained 30.2% of mean accuracy, MSER + BRISK 30.4%, MSER + FAST and the combination of all feature detectors obtained 30.6%. This means an improvement of up to 2.1% from 28.5% of the MSER detector. DNN also obtained an improvement up to 2.4%, and its best result was 30.2% using MSER + BRISK. MSER + Harris + FAST + BRISK combination obtained 30.1%. MSER + Harris and MSER + FAST improved to 29.4% from the 27.8% of the MSER detector. For the case of FFNN, the result for the combinations was the following: MSER + Harris 28.7%, MSER + BRISK 28.6%, MSER + FAST 29.9% and MSER + Harris + FAST + BRISK 30.0% of mean accuracy. The best result of FFNN showed an improvement of 3.3% from the obtained 26.7% of the MSER detector.

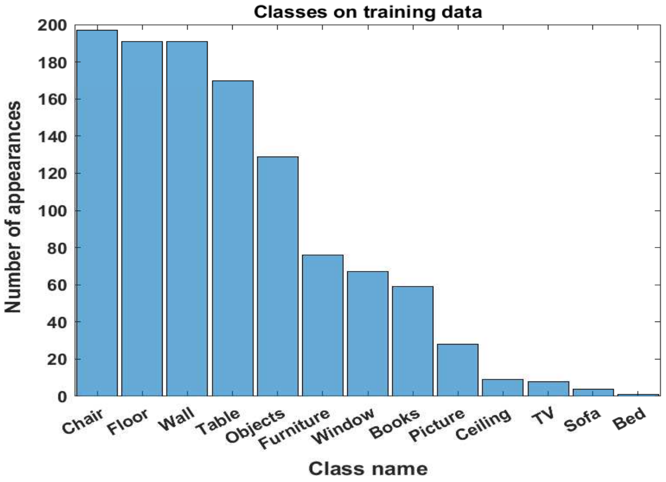


Fig. 3. Histogram of the number of appearances of classes in the training data.

In general, all algorithms presented a low performance. Thus, further analysis of the dataset showed that the training data was unbalanced which it causes the low accuracy of the four classification algorithms. Figure 3 presents the histogram of classes for the selected scenario (*office*), where the objects with most appearances in the training images are *chair*, *floor*, *wall* and *table*. However, other classes are not as recurrent, hence they are more difficult to classify.

Therefore, we defined a dataset for the second experiment choosing five object categories (food bag, food box, notebook, kleenex, and instant noodles) from the RGB-D Object Dataset. From these categories, five objects were selected and 40 images were taken from each object. A total of 1000 images were obtained which were split into five groups of 200 images each (40 images from each object category). We applied a cross-validation technique with four groups for training and one for testing. The same experiment settings listed above were used for this experiment. Additionally, we made a comparison with a feature matching algorithm[22] because we wanted to compare with a traditional local image features technique for object recognition. It consists of a priority search on hierarchical k-means trees for approximate nearest neighbor search in high-dimensional spaces. It is an efficient method for clustering and matching features in large datasets.

Table 3 shows the results for the experiment using a balanced dataset. It is shown that combining feature detectors can significantly increase the average value of the mean classification accuracy of feature descriptors in some cases up to 5.5% (MSER+BRISK and FFNN case). Also, the best configuration for this experiment was MSER+BRISK detectors obtaining better performance with all the algorithms. It improved for PNN 3.1% of the average value of the mean acurracy, SVM increased 3.04% using the same combination, and DNN 2.03%. The feature matching was the algorithm with the lowest performance. The low performance can be explained by the inability of the algorithm to generalize a class from the feature descriptors. It only focuses on find similarities with the training descriptors. Just in one case the computer could not process the test dataset because it got out of memory, it was using PNN and the sum of all the detectors.

Table 3. List of feature detectors combinations, the average value and the standard deviation using the cross-validation technique of the mean accuracy over the five classes for each classification algorithm.

Experiment		Results									
Feature detectors combinations	Descriptor	PNN		SVM		DNN		FFNN		MATCH	
		avg	std	avg	avg	avg	std	avg	std	avg	std
MSER	SURF	41.80%	7.47%	40.94%	7.42%	38.74%	4.80%	38.96%	7.41%	7.40%	0.9%
MSER + Harris	SURF	34.19%	6.57%	33.61%	6.76%	32.61%	6.77%	33.23%	5.61%	**8.64%**	1.33%
MSER + FAST	SURF	38.40%	6.45%	39.74%	9.10%	37.56%	8.38%	39.27%	8.56%	7.44%	1.20%
MSER + BRISK	SURF	**44.94%**	5.40%	**43.98%**	7.77%	**40.77%**	5.86%	**44.46%**	7.93%	5.11%	0.63%
MSER + Harris + FAST + BRISK	SURF	NA	NA	37.68%	8.52%	37.44%	6.81%	38.96%	8.06%	6.99%	1.11%

Table 4. Time results for each algorithm on the training and testing stages. The average processing time is shown using the cross-correlation technique.

Experiment		Time results (seconds)									
Feature detectors combinations	Descriptor	PNN		SVM		DNN		FFNN		MATCH	
		train	test	train	test	train	test	train	test	train	test
MSER	SURF	0.214	172.3	102.5	**0.102**	157.7	0.343	**61.90**	0.141	0.022	1.61
MSER + Harris	SURF	0.105	747.9	498.2	**0.110**	306.5	0.644	**118.7**	0.258	0.028	3.87
MSER + FAST	SURF	0.107	433.0	379.5	**0.101**	272.1	0.567	**105.1**	0.229	0.029	3.27
MSER + BRISK	SURF	0.117	806.6	508.1	**0.120**	336.37	0.706	**134.7**	0.268	0.034	4.00
MSER + Harris + FAST + BRISK	SURF	0.194	NA	1726.4	**0.205**	600.9	1.249	**232.6**	0.506	0.056	15.42

Table 4 shows the average time performance for training and testing stages over the cross-validation technique using an Intel Xeon E5-1620 processor and 48GB of RAM. Dismissing PNN, which its architecture build the structure of the classifier by simply adjusting some of its parameters during the training process and the feature matching algorithm because we only measure the process of the storage of the training feature descriptors, the best training time performance is FFNN for all the experiments compared to SVM and DNN. It has 40% less time in the training process in the worst case (with MSER and SVM). Although SVM does not have the best training time, it has the best time performance for the testing stage. It has an average of 0.1276 seconds classifying features descriptors of 200 images (an average of 0.638 milliseconds per image). Regardless of the increment of features, it only increases approximately 202% of its testing time on the worst case. DNN, FNN, and Feature matching increased their testing time approximately 381%, 358%, and 957% respectively when the number of feature detectors becomes larger.

4 Conclusions and Future Work

In this paper, we proposed a combination of feature detectors for feature descriptor classification as a step of a top-down RGBD semantic segmentation method. In the first experiment, local image features were used with four classification algorithms: a Probabilistic Neural Network (PNN), a Support Vector Machine (SVM), a Deep Neural Network (DNN), and a Feed Forward Neural Network (FFNN). This showed that the combination of feature detectors could improve performance on the classification of feature detectors on indoor environments. In particular, mean accuracy over 13 classes could be improved up to 3.3% in comparison to the FFNN implemented in [26]. This experiment showed that the combination of feature detectors of Maximally Stable Extremal Region (MSER) and Binary Robust Invariant Scalable Keypoints (BRISK) improved about 2.4% of mean accuracy using DNN. Support Vector Machine improved its performance by 2.1% using Maximally Stable Extremal Region and Features from Accelerated Segment Test (FAST) detectors, also using the combination of the four detectors (MSER + FAST + BRISK and Harris detector). Probabilistic Neural Network was the only one that could not improve its performance using the combination of feature detectors.

In the second experiment, the same four classification algorithms were compared with a feature matching algorithm. A balanced dataset was defined along with a cross-validation technique. It is showed that using MSER + BRISK combination could improve the average value of the mean classification accuracy by up to 5.5% using the FFNN implementation. The best performance was obtained by PNN with 44.94%, while the worst case was the feature matching algorithm with 5.11% using MSER + BRISK. However, taking into consideration the time performances, SVM has the best time performance for testing data. It takes 0.120 seconds for classifying features detectors of 200 images using the MSER + BRISK combination, approximately 0.6 milliseconds per image with the third best average value of mean classification accuracy of 43.98%. FFNN obtained the best training time with 134.7 seconds and the second testing time of 0.268 seconds with the second best result of 44.46% of the average value of the mean accuracy.

In conclusion, it was observed with these experiments that the best combination of feature detectors could not be defined for the first experiment. Only the sum of all detectors obtained two best performances. However, the second experiment showed that the MSER + BRISK was the best combination for the classification algorithms, except for the feature matching algorithm. The explanation on the MSER + BRISK is that unlike the other detectors (Harris and FAST), BRISK is invariant to rotation and scaling, so it is more robust. SVM was the algorithm with the best average performance for all the combinations of detectors in both experiments followed by DNN, FFNN, and PNN. The structure of one versus other classes helped SVM to classify better the feature descriptors. We noted that, in general, classification of feature descriptors was not high. One reason could be the generalization problem of local image features. It should be mentioned that we have tackled a part of the problem of semantic segmentation from a specific environment (indoor scenes). Garcia et al. [9] reported that the best result found for SUN RGBD Dataset was 48.10% of mean Intersection-over-Union (mIoU) obtained by Z. Li et al.[17]. So, it is still a challenging problem. As future work, we will analyze the generalization problem in terms of the local image features by exploring several neural network approaches. Furthermore, new combinations of feature detectors and feature descriptors will be made to have a broader perspective on the improvement and robustness of the proposed approach. Finally, this classification will be used for the problem of the semantic segmentation process as it is shown in Figure 1.

References

1. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1597–1604. IEEE (2009)
2. Arnab, A., Miksik, O., Torr, P.H.: On the robustness of semantic segmentation models to adversarial attacks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 888–897 (2018)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12), 2481–2495 (2017)
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision. pp. 404–417. Springer (2006)
5. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV library. " O'Reilly Media, Inc." (2008)
6. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (mser) tracking. In: null. pp. 553–560. IEEE (2006)
7. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2650–2658 (2015)
8. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857 (2017)
9. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing 70, 41–65 (2018)
10. Gupta, S., Arbeláez, P., Girshick, R., Malik, J.: Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. International Journal of Computer Vision 112(2), 133–149 (2015)

11. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: European Conference on Computer Vision. pp. 345–360. Springer (2014)
12. Handa, A., Pătrăucean, V., Stent, S., Cipolla, R.: Scenenet: an annotated model generator for indoor scene understanding. In: IEEE International Conference on Robotics and automation (ICRA) (2016)
13. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey vision conference. vol. 15, pp. 10–5244. Citeseer (1988)
14. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3d object dataset: Putting the kinect to work. In: Consumer Depth Cameras for Computer Vision. pp. 141–165. Springer (2013)
15. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1817–1824. IEEE (2011)
16. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: IEEE International Conference on Computer Vision. pp. 2548–2555. IEEE (2011)
17. Li, Z., Gan, Y., Liang, X., Yu, Y., Cheng, H., Lin, L.: Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In: European Conference on Computer Vision. pp. 541–557. Springer (2016)
18. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440 (2015)
19. Lowe, D.G.: Object recognition from local scale-invariant features. In: The proceedings of the seventh IEEE international conference on Computer vision. vol. 2, pp. 1150–1157. IEEE (1999)
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
21. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision 65(1-2), 43–72 (2005)
22. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1) 2(331-340), 2 (2009)
23. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: European conference on computer vision. pp. 143–156. Springer (2010)
24. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision. pp. 430–443. Springer (2006)
25. Salahat, E., Qasaimeh, M.: Recent advances in features extraction and description algorithms: A comprehensive survey. In: IEEE International Conference on Industrial Technology. pp. 1059–1063. IEEE (2017)
26. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: IEEE International Conference on Computer Vision Workshops. pp. 601–608. IEEE (2011)
27. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. Computer Vision–ECCV 2012 pp. 746–760 (2012)
28. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep fisher networks for large-scale image classification. In: Advances in neural information processing systems. pp. 163–171 (2013)

29. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 567–576 (2015)
30. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1625–1632 (2013)
31. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1369–1378 (2017)