PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Evaluation of algorithms for traffic sign detection

Miguel Lopez-Montiel, Yoshio Rubio, Moisés Sánchez-Adame, Ulises Orozco-Rosas

Miguel Lopez-Montiel, Yoshio Rubio, Moisés Sánchez-Adame, Ulises Orozco-Rosas, "Evaluation of algorithms for traffic sign detection," Proc. SPIE 11136, Optics and Photonics for Information Processing XIII, 111360M (6 September 2019); doi: 10.1117/12.2529709

Event: SPIE Optical Engineering + Applications, 2019, San Diego, California, United States

Evaluation of Algorithms for Traffic Sign Detection

Miguel Lopez-Montiel[a], Yoshio Rubio[a], Moisés Sánchez-Adame[a,b], and Ulises Orozco-Rosas[b,*]

[a]Instituto Politécnico Nacional, CITEDI-IPN, Ave. Instituto Politécnico Nacional No. 1310, Nueva Tijuana, C.P. 22435, Tijuana, Baja California, México
[b]CETYS Universidad, Centro de Innovación y Diseño (CEID), Ave. CETYS Universidad No. 4, Fracc. El Lago, C.P. 22210, Tijuana, Baja California, México

ABSTRACT

Traffic sign detection is a crucial task in autonomous driving systems. Due to its importance, several techniques have been used to solve this problem. In this work, the three more common approaches are evaluated. The first approach uses a model of the traffic sign which is based in color and shape. The second one enhances the image model of the first approach using K-means for color clustering. The last approach uses convolutional neural networks designed for image detection. The LISA Traffic Sign Dataset was used which it was divided into three superclasses: prohibition, mandatory, and warning signs. The evaluation was done using objective metrics used in the state-of-the-art.

Keywords: Detection, Traffic Sign, Machine learning, Computer vision, Deep learning, Autonomous vehicles

1. INTRODUCTION

Road traffic collisions are a major health problem in most countries, mainly in low and middle-income countries. It is the third leading cause of death for people between 5 and 44 years old. The United Nations estimates that, between 2010 and 2020, the number of yearly road deaths will increase from 1.3 to 2.4 million. Furthermore, each year, twenty to fifty million people have some type of non-fatal injury, that can lead to disability.[1]

Advanced Driver Assistance Systems (ADAS) can help to decrease the number of car accidents by automating tasks than enhances security such as pedestrian and obstacle detection, to more advanced tasks that help in fully autonomous driving. An important task in ADAS is Traffic Sign Recognition (TSR) systems, that have been emerging as a necessary tool for the intelligent transportation system. Traffic sign recognition systems consist of three main stages: localization, detection, and classification. In the case of any false alarm in the detection stage, performance will be lower in the classification one: this is since a classifier is not usually trained on false alarms. The stages with most research activity found on literature are Traffic Sign Detection (TSD) and Traffic Sign Classification (TSC), and these works focus on the stage of classification, however, it is well known that this task requires detection stage to do classification.[2,3]

The main task of TSR and TSD systems is to enhance the drivers safety by alerting the driver about different possible types of situations that arise in the road, and informing with the information provided by the traffic signs such as speed limit, warning signs for icy roads, massive road works, or zebra crossing, among other things. The traffic signs that can cover the general tasks for an ADAS can be found in three main superclasses: prohibitory, warning and mandatory, these contain many classes to do the basic tasks of an autonomous vehicle such as speed control, steering control and complete stop.

TSD have many external environment variables that reduce the detection and recognition performance such as illumination variations, scale changes, weather conditions, occlusions, and rotations. Other issue is the state of the traffic sign which involves there are no-identical traffic signs for different countries, that the signs may be partially occluded, faded, damaged or the presence of multiple traffic sign at a time. These different environments and physical conditions of the object generates an interesting research problem, and an area of opportunity to develop models and methods that solve this problem under these conditions.[4–6]

*Further author information:
U. Orozco-Rosas: E-mail: ulises.orozco@cetys.mx

Optics and Photonics for Information Processing XIII, edited by Khan M. Iftekharuddin, Abdul A.S. Awwal, Victor H. Diaz-Ramirez, Andrés Márquez, Proc. of SPIE Vol. 11136 111360M · © 2019 SPIE · CCC code: 0277-786X/19/$21 · doi: 10.1117/12.2529709

The two main features of traffic sign are shape and color, different methods use one or both properties. The most common approach using color feature is found in the segmentation stage, to help locate objects related to these color properties. Some color-based methods used the threshold to segment objects that are found with those color properties. In the detection stage, these methods use shape features of traffic signs to find specific shapes to detect the traffic sign. The most common approaches for these methods are the Hough transform and Radial symmetry voting to detect traffic signs.

Other approaches are learning-based methods, which improve the results of these two methods (color-based and shape-based) using different categories of machine learning such as supervised, unsupervised and reinforcement-based learning. The current trends in machine learning are the use of deep learning models. These models solve computer vision problems with powerful feature extractors and meta-architecture nets, that are trained with a huge amount of data and effectively solve different problems such as TSD.[7–12]

Some recent works are focused on the evaluation of different methods for TSC and TSD. Saadna et. al. in[7] used The German Traffic Sign Detection Benchmark (GTSDB dataset)[13] for TSD with the most recent, efficient and common methods such as color-based, shape-based and learning-based, getting results from 90 to 100%. In the same work, for TSC they evaluated learning methods based on hand-crafted features and deep learning methods getting an accuracy higher than 99%.

In a different work, Arcos et. al. in[10] focused on the comparative evaluation of different deep learning methods with some meta-architectures (Faster R-CNN, R-FCN, SSD, and YOLOV2) combined with various feature extractors (Resnet V1 50, Resnet V1 101, InceptionV2, Inception Resnet V2, Mobilenet V1, and Darknet-19). They used GTSDB for TSD and TSC focused on superclasses based on their shapes and colors: mandatory, prohibitory, and danger. They found that Faster R-CNN Inception Resnet V2 obtains the best mAP (95.77%).

Also, Tabernik et. al. in[14] did a comparative evaluation of different deep learning methods, as well as creating his own database DFG traffic-sign dataset for both TSD and TSC. For the comparative, they use the Mask R-CNN using ResNet-50 with some adaptations and data augmentations. The results show 108 of 200 categories with an average precision (AP) of 100%, 60 categories with an AP higher than 90%, and 23 with more than 80% of AP.

Another approach for TSD and TSR are automatic measures used for evaluation of TSD and TSR systems performance. Khalid et. al.in[15] introduced a new methodology to solve the problem of evaluation using the German Traffic Sign Detection Benchmark (GTSDB) and German Traffic Sign Recognition Benchmark (GTSRB) dataset. They used HOG features by employing the SVM-KNN classifier getting 0.96 in the detection stage using F-score measure, and 99.32% in the recognition stage using measure precision.

The aim of this paper is to present an evaluation of three different methods used to solve TSD and compare their performance. The first method employs a model of the image that makes use of the color and shape of the traffic sign to extract it from the background. The second approach enhances the image model of the method using K-means for color clustering. Finally, two convolutional neural networks were evaluated which are designed for image detection. The nets are the Faster R-CNN+ResNet_v1 101 and the SSD+MobileNet v1. For the experiments, we used the LISA Traffic Sign Dataset divided into three superclasses: prohibition, mandatory, and warning signs.

The paper is divided as follows. In Section 2, the three traffic sign detection methods are presented. Section 3 shows the evaluation of the different methods for the TSD problem using the LISA dataset. In Section 4 we discuss the results and give conclusions and describe future work.

2. METHODS FOR TRAFFIC SIGN DETECTION

A TSD system can be divided into four phases: image input, pre-processing, segmentation and detection, see Figure 1. The focus of many model-based methods is the segmentation and detection steps. The segmentation stage is the step that focuses on locating the region of interest (ROI), in this case, the traffic sign. The main problem of the segmentation stage is to correctly identify the ROI from different objects in the scenario. Different preprocessing methods are used to help the segmentation step, with the aim of reducing the unfavorable environmental conditions such as lighting, weather, noise, and blur.

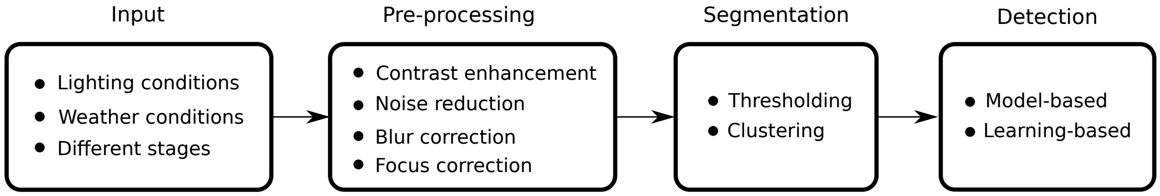


Figure 1. General Traffic Sign Detection system

For the detection step, there are many approaches, the most commonly used are the shape-based and the learning-based methods. Traditionally, shaped-based methods develop an image model using filters or other techniques. The objective is to extract edges and corners that are consistent with the described model. Learning-based models are trained with a set of training images and labels, and they learn through generations in many cases unsupervised.

In this work, the two approaches mentioned above are implemented, and each of them has two variations. For the shape-based approach, a model of the traffic sign is developed using the known color and shape of the signs with two variations for the color segmentation: thresholding and K-means. In the learning-based methods, two deep convolutional neural networks for detection are used.

2.1 Model-based methods

Ideally, the traffic signs are easily distinguished from the background with their two unique features (color and shape). In real scenes, many objects share one of both features of the traffic signs, or even external factor can change the appearance of the traffic signs. To deal with these problems, what is needed is a robust model that generalize the traffic signs based on their color and shape. This can be difficult since there are many types of traffic signs, and within them, many variations may exist.

The model approach is a hybrid method using both color and shape features to acquire the complete properties of the traffic sign. In this case, the first feature detected is the color of the object of interest, which is used in the segmentation stage. The second feature is the shape, which is used in the detection stage.

The proposed methodology is illustrated in Figure 2. The input image is converted to the LAB color space, with contrast enhancement and noise reduction in the preprocessing stage. Then, the image is separated in different LAB channels, and histogram analysis is used to identify the type of signal based on the model of the traffic sign. Using the results of the histogram analysis, the ROI is segmented by color. The output of the segmentation stage is modified by morphological operations, and the signal is detected based on the shape of the objects in the image.

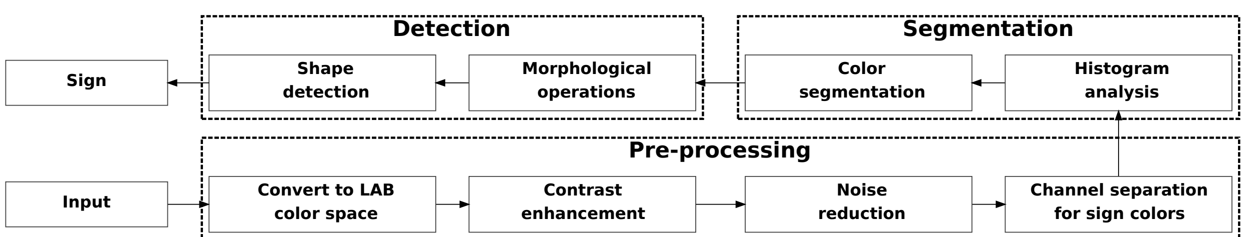


Figure 2. Proposed Traffic Sign Detection method.

2.1.1 Color segmentation using thresholding

The color segmentation method mostly used is thresholding. Where color information about traffic signs is considered, but the different conditions of weather and lighting effects this type of method. To solve these problems color spaces are used considering the property of light, such as HSV, HSI, Ycbcr, YUV, CIECAM97, and LAB. Another way to solve this problem is to use contrast enhancement such as Contrast Limited Adaptive Histogram Equalization (CLAHE) applied to the light channel of these color models or on a grayscale image.[2,16] These types of preprocessing use different approaches to obtain results in scenes with variable lighting conditions and low resolution.

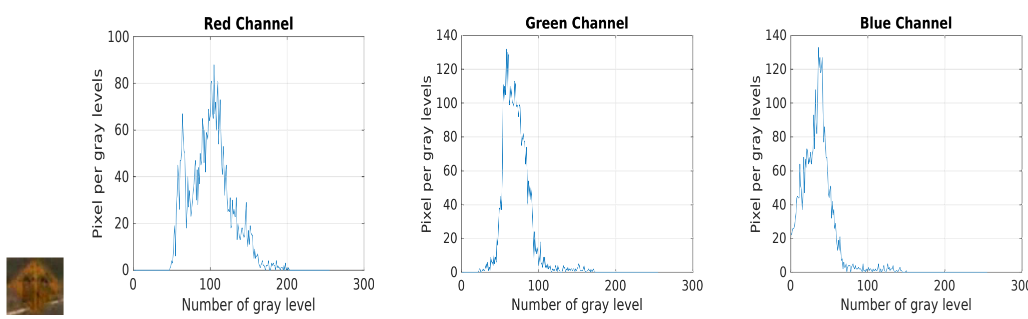
Another issue of this method is to select the optimal threshold. There are different ways to generate this parameter, and one of them is using statistics with histogram analysis such as Qin in,[17] but the histogram's distribution types are different for the range of color values found on the traffic sign, then a distinct adequate range of values could be defined that enclose these features. Good results can be generated, however, with different lighting conditions the results are not the same.

The approach of this work is to consider these two variables, the color of the traffic sign and light conditions of the scene using a histogram analysis with statistics as Qin but for both features color and light distributions. In this environment, there are 12 different cases of lighting, 6 for both sign and stage are shown in Table 1. This work considers 6 cases only because not all of the cases are present in the LISA dataset.

Table 1. Constant value for different lighting conditions

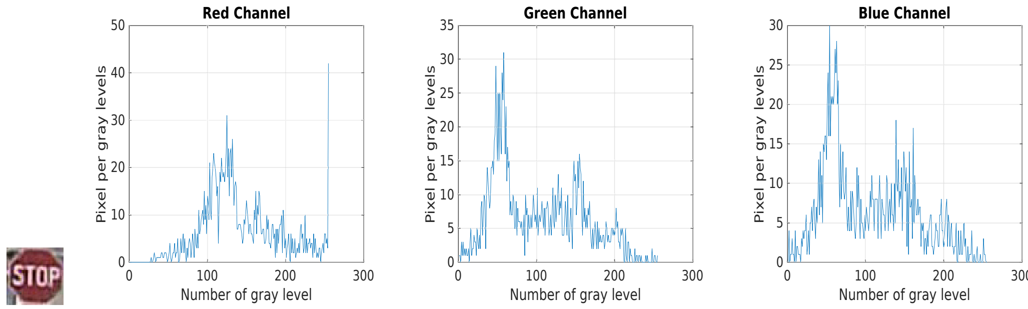
case	Sign	Stage
1	poor	poor
2	poor	normal
3	poor	a lot
4	normal	normal
5	normal	a lot
6	a lot	a lot

The first approach is to use the median of the distribution of the channels in LAB based on the type of signal and lighting. The first case occurs when the traffic sign has poor conditions, see Figure 3.



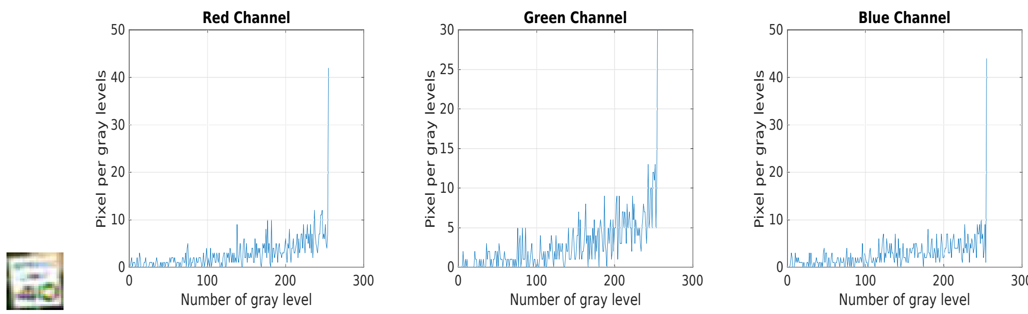
(a) Poor (b) R channel histogram (c) G channel histogram (d) B channel histogram

Figure 3. Distributions of each RGB channel of yellow sign with poor condition.



(a) Normal (b) R channel histogram (c) G channel histogram (d) B channel histogram

Figure 4. Distributions of each RGB channel of red sign with normal condition.



(a) A lot (b) R channel histogram (c) G channel histogram (d) B channel histogram

Figure 5. Distributions of each RGB channel of white sign with high condition.

Different authors use this method changing the formula to approach different categories of traffic signs such as Harshavardhan[18] adjusting the formula focused in red traffic signs.

Our proposed method uses the L channel of color LAB to generate an adjust variable to search a better threshold, the first step is to find a new distribution between the max value of the A or B channel of LAB and then used in Equation 1, and its represented as follows:

$$bthr = mc + (stdc * adj) + (ml * 0.0450) \tag{1}$$

Where

mc = median of channel color (A channel for red sign and B channel for yellow sign)

ml = median of lightning (L channel)

$stdc$ = standard deviation of (A or B channel)

2.1.2 Color segmentation using clustering

The second approach used for color segmentation is K-means. This unsupervised machine learning technique uses the LAB color channels to generate different clusters with the same color properties, to find segments with the objects that meet those properties. Some works such as Yadav in[19] used this method to solve medical segmentation image problems, for this reason, this work tests this method to solving another image segmentation problem.

K-means minimizes $W(C)$ when $D = ||x_i - x_{i'}||^2$

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k}||x_i - x_{i'}||^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k}||x_i - \overline{x}_k||^2 \tag{2}$$

2.1.3 Shape detection

After the stage of segmentation it is necessary to do a post-processing to eliminate the noise and undesired objects using different methods such as the elimination of small and big objects, highlight the characteristics of the components found, preserve square and symmetric objects using some morphological operations such as dilatation and erosion, and to remove more noise with different operations used in[20] by Villalobos.

After post-processing and removing all unwanted objects information is passed to the detection stage which is based on geometric shape finding the diameter of the object using the next Equation: 3

$$D = \begin{cases} W, & W > H \\ H, & \text{other case} \end{cases} \tag{3}$$

Where:

$D =$ Diameter of the object.

$W =$ Width of the object.

$H =$ Height of the object.

Calculate the area of the geometric figure as follows: if it is a stop sign then use Equation 4 because the segmentation resembles a circle. If it is a warning sign then use Equation 6 and if it is mandatory sign then use Equation 5

$$area = \frac{\pi * diameter^2}{4} \tag{4}$$

$$area = W * H \tag{5}$$

$$area = \frac{W * H}{2} \tag{6}$$

The next step is to obtain the area of the objects to be compared to obtain a difference. Follow the next procedure:

$$Dif = PA - AO \tag{7}$$

Where

$Dif =$ Difference

$ACE =$ Perfect area of the outer geometric figure.

$AO =$ Area of the object

After this step the ratio is obtained using the following code:

$$R = Dif - ACE \tag{8}$$

Where R is the ratio or proportion of the difference between the area of the object and the area of a circle, diamond or rectangle. Taking into account this value it is sought to find the small ratio that approaches these geometric figures to find the high signal used by Igor in.[21]

2.2 Deep Learning detection

Convolutional neural networks (CNNs) have become one of the most popular techniques for classification and object detection. Since the introduction of Alexnet in the ILSVRC-2012,[22] different models have been proposed of deep convolutional neural networks, outperforming traditional models. In this work, we selected two meta-architectures combined with two feature selectors and trained them with the LISA Dataset.[23] The selections of the nets and features are based on the evaluation of performed by Arcos-García,[10] the idea was to select the nets with the best precision and the fastest. The analysis of Arcos-García on the German Traffic Sign Database found that the best performance in the mean average precision was obtained by the meta-architecture Faster R-CNN using the Resnet V1 101 feature extractor. and the fastest net with competitive precision was the meta-architecture SSD with the Mobilenet V1 feature extractor. The complete description of the nets can be found in.[10]

2.2.1 Faster-RCNN

Faster R-CNN was proposed by Ren[24] as an improvement of SPPnet[25] and Fast R-CNN,[26] in speed and accuracy. SPPnet uses a layer called *spatial pyramid pooling* (SPP) and Selective Search (SS) which eliminates the need for a fixed-size input image. The downside of SPPnet is the computational cost of the SPP layer. Faster R-CNN overcomes the computation bottleneck replacing the SS with a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network.

RPN is a fully convolutional neural network that simultaneously predicts objects bounds and objectless scores. Multiple region proposals are predicted by the RPN, and the redundant regions are eliminated using a non-maximal suppression algorithm. The top-ranked proposal regions are passed to the detection network, which in Fast R-CNN.

For the experiments, the number of regions proposals was set to 30, the SDG momentum set to 0.9, the training batch to 1 (due to memory size), the number of steps was set to 50000, the learning rate set to 0.0003 with a reduction factor of ten and a step decrement of 33%.

2.2.2 SSD

The Single Shot Multibox Detector (SSD)[27] summarizes all the computations in a single network, making it easy to train and to integrate to systems that need a detection component. SSD eliminates bounding box proposal generation and feature resampling stages, it discretizes the output space of bounding boxes into a set of default boxes, with different aspect ratios and scales. When predicting, the network generates a default box over the object and adjust the box to better match the shape of the object.

The early layers of SSD are based on standard architecture for high-quality classification, in this specific implementation, we used the suggested by the original authors (VGC-16). The auxiliary structure is added to produce multi-scale feature maps for detection, which is comprised of convolutional features layers. The layers decrease in size progressively, allowing predictions of detections at multiple scales.

In this experiment, the SSD model is trained using RMSprop, with the momentum of 0.9 and a batch size of 32. The learning rate was set to 0.004 with an exponential decay factor for every 80000 iterations. The input size images were resized to a fixed shape of 300 pixels for width and height.

2.2.3 Resnet V1 101

Residual Nets (Resnets)[28] reformulate the layers as learning residual functions with reference to the layer inputs. Rather than skipping a certain number of stacked layers to directly fit a desired underlying mapping, Resnets let the layers fit a residual mapping. To use a Resnet as a feature extractor for Faster R-CNN, the network is split into two steps. The Faster R-CNN performs the extraction of the RPN features and the Resnet extract the box classifier features. The feature extractors are built with four residual blocks. The first three residual blocks extract RPN features, and the last one is used for prediction region proposals and to extract the box classifier features.

2.2.4 MobileNet v1

MobileNets were proposed by Howard et. al.[29] for mobile and embedded vision applications. This type of nets is based on a streamlined architecture that used depthwise separable convolutions with the aim of building lightweight nets, reducing the number of parameters and computational cost. To use MobileNet with the SSD as a feature extractor, the feature extraction of regions proposals is omitted in the SSD. The architecture of MobileNet is modified by adding auxiliary convolutional feature maps at multiple steps, and four additional convolutional layers are appended.

2.3 LISA Traffic Sign Detection Dataset

The LISA Traffic Sign Detection Dataset[23] is a set of videos that have annotated frames of the US traffic signs. The images are stored in grayscale and 8-bit color. For the experiments, only color images were used. An example of the LISA Dataset can be seen in Figure 6.

Figure 6. Samples of prohibitory, mandatory and warning super classes traffic signs with different lighting conditions.

The subset used of LISA has 2857 labeled color images, with 3232 traffic signs and 47 classes. These 47 classes can be divided by appearance and function into three superclasses: prohibition, mandatory, and warning. The division is useful to reduce the computational training cost and to improve accuracy in classes with a low number of examples. The list of classes and superclasses is shown in Table 2. A further classification stage can be implemented over the detected superclass in order to detect the specific traffic sign that belonging to it.

Table 2. Table of classes and super classes of the LISA dataset.

Prohibitory	Mandatory	Warning
stop	schoolSpeedLimit25	pedestrianCrossing
yield	speedLimit15	signalAhead
yieldAhead	speedLimit25	addedLane
noRightTurn	speedLimit30	merge
doNotPass	speedLimit35	stopAhead
noLeftTurn	speedLimit40	school
doNotenter	speedLimit45	dip
	speedLimit50	intersection
	speedLimit55	turnRight
	speedLimit65	curveLeft
	speedLimitUrdbl	curveRight
	keepRight	slow
	rightLaneMustTurn	laneEnds
	zoneAhead25	rampSpeedAdvisory20
	zoneAhead45	rampSpeedAdvisory35
	truckSpeedLimit55	rampSpeedAdvisory40
		rampSpeedAdvisory45
		rampSpeedAdvisory50
		rampSpeedAdvisoryUrdbl
		roundabout
		thruMergeLeft
		thruMergeRight
		thruTrafficMergeLeft
		turnLeft

3. RESULTS AND DISCUSSION

For the experiments, both deep nets were trained using the LISA Traffic Sign Dataset.[23] For the validation step, 30% of the database was used, and the rest was used for training. The nets use transferred learning of pre-trained nets on the COCO dataset and were trained with the same hyperparameters as in[10] (except for a number of steps). Both models were trained on an Nvidia GeForce GTX TITAN X GPU using the Tensorflow Object Detection API.

3.1 Evaluation metrics

The metrics used to evaluate the deep learning models are the precision, recall, the interpolated Average Precision (AP), and the average intersection over union. To calculate the precision and recall we use Equations 9 and 10, where TP indicates true positives, FP false positives, and FN false negatives.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

The interpolated AP tracks the precision/recall curve, and it is calculated by setting the precision for recall r to the maximum precision obtained for any recall $r' \geq r$, see Equation 11, where $p(r')$ is the measured precision at recall r'. The AP is calculated as the area under the precision/recall curve, and by approximation is the sup of the precision at every k where the recall changes, multiplied by the change in recall $\Delta r(k)$, see Equation 12.

$$p(r) = \max_{r':r' \geq r} p(r') \tag{11}$$

$$AP = \sum_{k=1}^{N} p(k) \Delta r(k) \tag{12}$$

The intersection over union (IoU) measures the similarity between the detected bounding box and the ground truth. This metric is used to indicate if detection is valid or not using a threshold. The prediction is correct if the IoU is greater than 0.5. Equation 13 describes the calculation of the IoU, where G is the ground truth box and D is the detected bounding box.

$$IoU = \frac{|G \cap D|}{|G \cup D|} = \frac{|G \cap D|}{|G| + |D| - |G \cap D|} \tag{13}$$

3.2 Analysis

As mentioned in subsection 3.1, for the model-based methods the average IoU, Avg. IoU >0,5 and the Recall was calculated. The result of these metrics is indicated in Table 3 K-means had better results on average.

Not all works take these kinds of superclasses and these metrics for these methods but with the ones obtained by Quin[17] it could be said that it is a new approach to solve this problem.

Table 3. Model based methods average IoU and recall

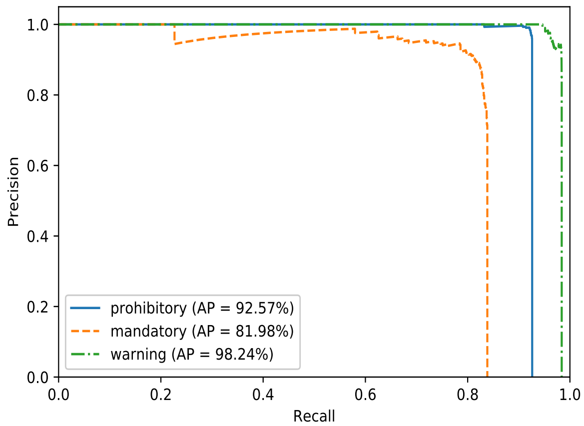
Model-based	Class	Avg. IoU	Iou >0.5	Recall
Thresholding	Prohibitory	**0.3048**	**0.7654**	**0.3787**
	Mandatory	N/A	N/A	N/A
	Warning	0.1389	0.6883	0.1574
K-means	Prohibitory	0.2957	0.7479	0.3774
	Mandatory	**0.0290**	**0.829**	**0.0344**
	Warning	**0.2368**	**0.7229**	**0.3092**

As indicated in subsection 3.1, for the learning-based methods the average IoU and the AP were calculated. The result of this metrics is indicated in Table 4, and the precision/recall curves for both nets are shown in Figures 7(a) and 7(b). The Faster R-CNN + ResNet V1 101 net had a higher performance than the SSD + MobileNet V1 in all the metrics, except for the Precision for the mandatory class. On average, the Faster R-CNN had a 3% higher performance for the average IoU, Precision, and Recall, and had a 6% higher AP. The obtained results are consistent with the ones obtained by Arcos,[10] with higher performance for Faster R-CNN but to a lesser degree. This can be due to the lower resolution and quality of the LISA database compared to the GTSDB.

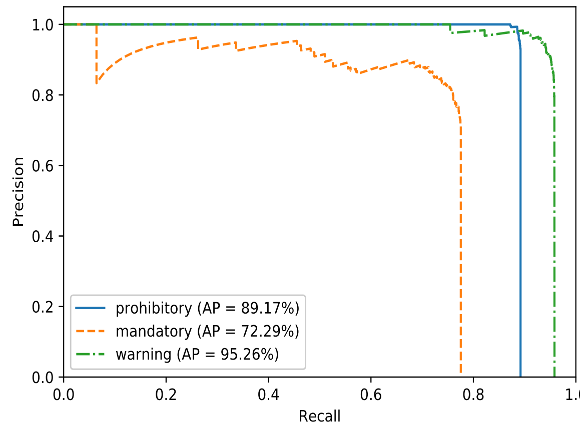
Table 4. Learning-based methods average IoU, precision, recall, and average precision for learning based methods.

Methods	Class	Avg. IoU	Precision	Recall	AP
Faster R-CNN + ResNet V1 101	Prohibitory	**0.8224**	**0.9406**	**0.9262**	**0.9257**
	Mandatory	**0.8113**	0.7143	**0.8380**	**0.8193**
	Warning	**0.8284**	**0.8238**	**0.9838**	**0.9823**
SSD + MobileNet V1	Prohibitory	0.8147	0.9290	0.8920	0.8917
	Mandatory	0.7568	**0.7192**	0.7747	0.7229
	Warning	0.7999	0.7816	0.9581	0.9523

Comparing the two model approaches with the learning-based methodologies, the later ones are far superior in terms of precision and average IoU.



(a) Faster R-CNN + ResNet v1 101 curve.



(b) SSD + MobileNet v1 curve.

Figure 7. Precision-recall curves of learning methods.

3.3 Detection in the LISA dataset

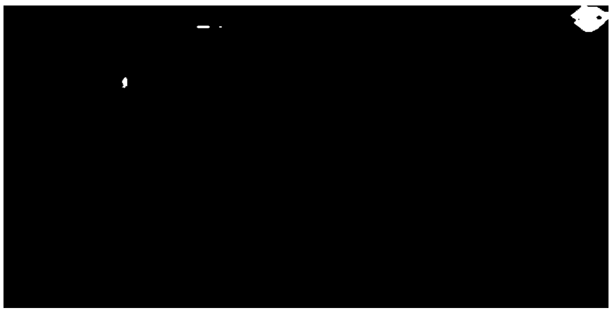
Figures 8 to 12 shows an example of the different stages of the thresholding method of detection for the three superclasses. The preprocessing stage helps to get a better contrast between the colors and help the segmentation stage. The red and the yellow sign outstand from the background, and they are easy to segment, see Figures 12(c) and 8(c). The problem arises when segmenting the white sign (mandatory class), since in the image there are many white objects, see Figure 10(c). The detection stage eliminates irregular shapes and those that are not under the desired size, which helps to correctly detect the desired traffic sign under the presence of many artifacts.

The K-means based method, see Figures 9 to 13, has a higher fidelity on the segmentation step than the proposed threshold method. The downside is the presence of more artifacts in the image, which is treated in the shape detection stage.

Figures 14(a) to 15(b) illustrate the output after using the deep learning methods for the detection of the traffic signs. For most of these cases, the detection certainty was higher than 99%, with bounding boxes that surrounds the sign in an adequate shape. As Figures 15(a) and 15(b) indicate, the deep learning methods can effectively detect more than one sign in a scene, which is different form the proposed model-based methods that only detect one sign at a time.

(a) Original image

(b) Pre-processing of sample using Contrast enhancement and median blur



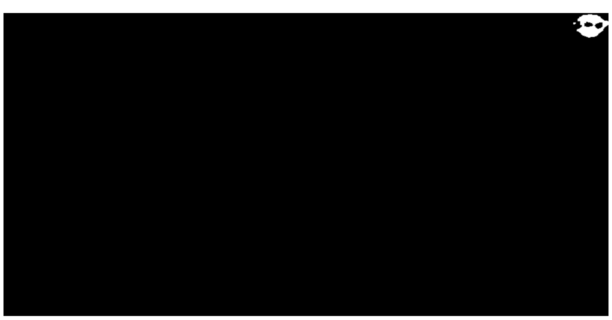
(c) Segmentation image

(d) ROI image

Figure 8. Output of Thresholding method for prohibitory class.

(a) Original sample of prohibitory

(b) Pre-processing of sample using Contrast enhancement and median blur



(c) Segmentation image

(d) ROI image

Figure 9. Output of K-means method for prohibitory class.

(a) Original image

(b) Pre-processing of sample using Contrast enhancement and median blur

(c) Segmentation image

(d) ROI image

Figure 10. Output of Thresholding method for mandatory class.

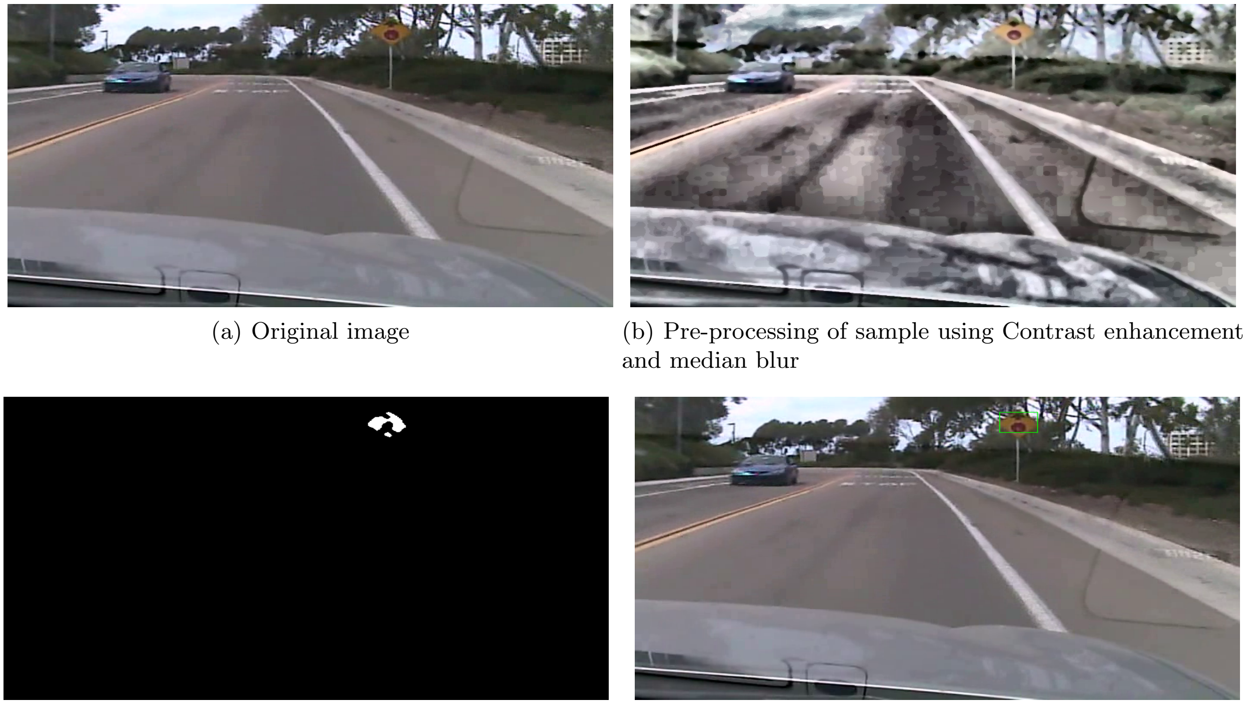
(a) Original image

(b) Pre-processing of sample using Contrast enhancement and median blur

(c) Segmentation image

(d) ROI image

Figure 11. Output of K-means method for mandatory class.



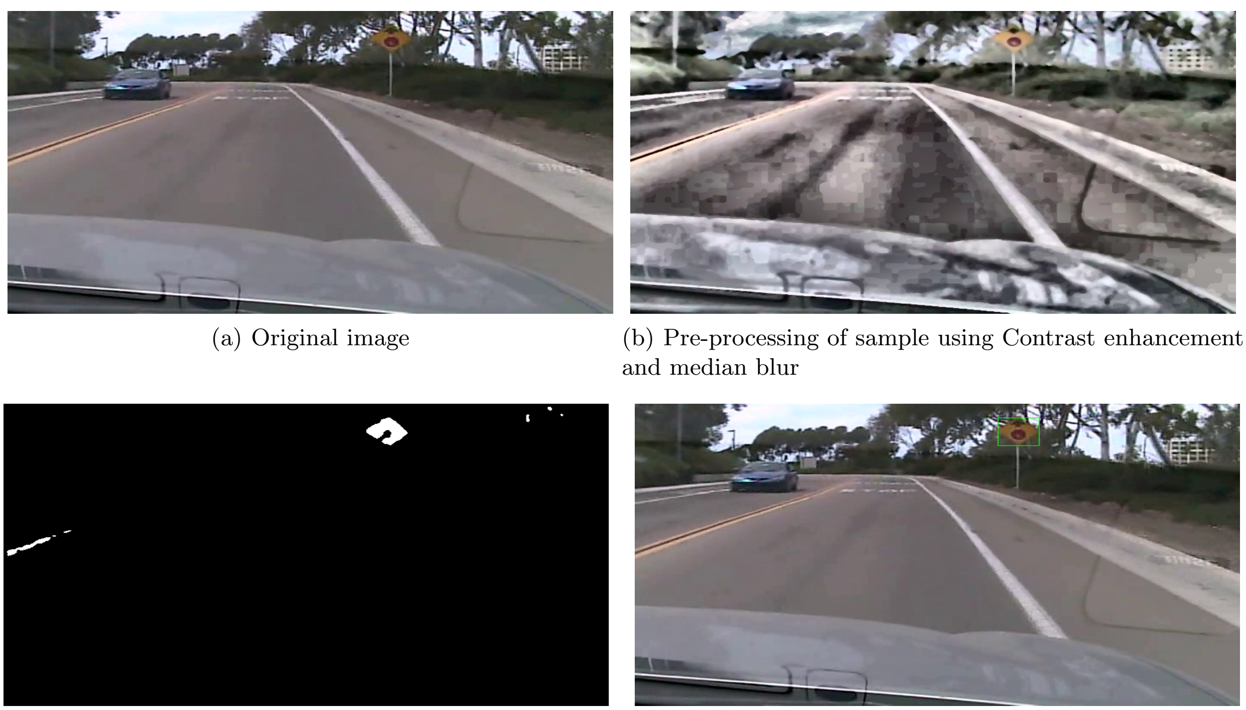
(a) Original image

(b) Pre-processing of sample using Contrast enhancement and median blur

(c) Segmentation image

(d) ROI image

Figure 12. Output of Thresholding method for warning class.



(a) Original image

(b) Pre-processing of sample using Contrast enhancement and median blur

(c) Segmentation image

(d) ROI image

Figure 13. Output of K-means method for warning class.

(a) Output of Faster R-CNN + ResNet v1 101 methods (b) Output of SSD + MobileNet v1 method

Figure 14. Output of Deep Learning methods for prohibitory class.

(a) Output of Faster R-CNN + ResNet v1 101 methods (b) Output of SSD + MobileNet v1 method

Figure 15. Output of Deep Learning methods for warning class.

(a) Output of Faster R-CNN + ResNet v1 101 methods (b) Output of SSD + MobileNet v1 method

Figure 16. Output of Deep Learning methods for mandatory class.

4. CONCLUSIONS AND FUTURE WORK

In this paper, comparative evaluation between two ways of solving the traffic sign detection problem is presented, in which two model-based methods and two Deep Learning methods are used. Three superclasses that encompass different traffic sign classes of each superclass were implemented as well. All of which bear different color and shape features. Both methods apply a preprocessing stage to highlight the color features, improving the segmentation stage performance. The main issue of the Thresholding method is finding the appropriate threshold value so that the segmentation stage of the ROI between the signal color and background is done. In this paper, finding the appropriate value of the threshold for the segmentation stage based in color is the focal point. This is achieved using a histogram statistical analysis, where two lighting variables are considered, one for the traffic sign and the other for the scenario.

Comparing the results obtained by the implemented model-based methods, (Thresholding and K-means) it can be observed that the Thresholding method presented a higher performance on all the metrics on the prohibitory class, however, the K-means method presented a higher performance on the remaining mandatory and warning classes. LAB color space was used for both methods to find the RGB colors of each signal, nonetheless, in the Thresholding method the mandatory class is not considered, due to the lighting channel being used as an adjustment variable for the threshold, meanwhile, in the K-means method, an adjustment variable is not used. For the learning-based methods, Faster R-CNN + ResNet V1 101 was the one with a higher performance on all the metrics on all the classes, except for the precision metric on the mandatory class, in which the SSD + MobileNet V1 method presented better performance. The results obtained are consistent with the ones obtained by Arcos, with a higher performance on the Faster R-CNN method, but in a lesser degree. This can be due to the lower quality and resolution of the LISA dataset in comparison to GTSDB. For future works, tests can be running on the K-means method to obtain a better group color selector so that the color of the desired traffic sign is obtained. For the Thresholding method, different adjustment variables for the threshold can be looked for, this way, the threshold can be adapted to find the mandatory class while using the A and B channels. For the learning-based methods, it is required to evaluate the different meta-architectures and feature extractors to find the most optimal system for traffic sign detection.

ACKNOWLEDGMENTS

This work was supported by Instituto Politécnico Nacional, the Coordinación Institucional de Investigación of CETYS Universidad, and by the Mexican National Council of Science and Technology (CONACYT, Mexico).

REFERENCES

[1] United Nations Road Safety Colaboration, "Global plan for the decade of action for road safety 2011-2020," (Sep 2016).

[2] Nandi, D., Saif, A., Paul, P., Md. Zubair, K., and Ahmed Shubho, S., "Traffic sign detection based on color segmentation of obscure image candidates: A comprehensive study," *International Journal of Modern Education and Computer Science* **10**, 35–46 (06 2018).

[3] Qu, Y., Yang, S., Wu, W., and Lin, L., "Hierarchical traffic sign recognition," in [*Advances in Multimedia Information Processing - PCM 2016*], Chen, E., Gong, Y., and Tie, Y., eds., 200–209, Springer International Publishing, Cham (2016).

[4] Kukkala, V. K., Tunnell, J., Pasricha, S., and Bradley, T., "Advanced driver-assistance systems: A path toward autonomous vehicles," *IEEE Consumer Electronics Magazine* **7**, 18–25 (Sep. 2018).

[5] Wali, S. B., Abdullah, M. A., Hannan, M. A., Hussain, A., Samad, S. A., Ker, P. J., and Mansor, M. B., "Vision-based traffic sign detection and recognition systems: Current trends and challenges," *Sensors* **19**(9) (2019).

[6] K, S. and Kumar, A., "Detection and recognition of road traffic signs - a survey," *International Journal of Computer Applications* **160**, 1–5 (02 2017).

[7] Saadna, Y. and Behloul, A., "An overview of traffic sign detection and classification methods," *International Journal of Multimedia Information Retrieval* **6**, 193–210 (Sep 2017).

[8] Liu, C., Li, S., Chang, F., and Wang, Y., "Machine vision based traffic sign detection methods: Review, analyses and perspectives," *IEEE Access* **7**, 86578–86596 (2019).

[9] Lillo-Castellano, J., Mora-Jiménez, I., Figuera-Pozuelo, C., and Rojo-Álvarez, J., "Traffic sign segmentation and classification using statistical learning methods," *Neurocomputing* **153**, 286 – 299 (2015).

[10] Arcos-García, A., Álvarez-García, J. A., and Soria-Morillo, L. M., "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing* **316**, 332 – 344 (2018).

[11] Gómez-Moreno, H., Maldonado-Bascón, S., Gil-Jiménez, P., and Lafuente-Arroyo, S., "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Transactions on Intelligent Transportation Systems* **11**, 917–930 (2010).

[12] Yang, Y., Luo, H., Xu, H., and Wu, F., "Towards real-time traffic sign detection and classification," *IEEE Transactions on Intelligent Transportation Systems* **17**, 2022–2031 (July 2016).

[13] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C., "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in [*The 2013 International Joint Conference on Neural Networks (IJCNN)*], 1–8 (Aug 2013).

[14] Tabernik, D. and Skoaj, D., "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Transactions on Intelligent Transportation Systems* , 1–14 (2019).

[15] Khalid, S., Muhammad, N., and Sharif, M., "Automatic measurement of the traffic sign with digital segmentation and recognition," *IET Intelligent Transport Systems* **13**(2), 269–279 (2019).

[16] Yadav, G., Maheshwari, S., and Agarwal, A., "Contrast limited adaptive histogram equalization based enhancement for real time video system," in [*2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*], 2392–2397 (Sep. 2014).

[17] Qin, F., Fang, B., and Zhao, H., "Traffic sign segmentation and recognition in scene images," in [*2010 Chinese Conference on Pattern Recognition (CCPR)*], 1–5 (Oct 2010).

[18] Harshavardhan, S., Madhavi, V., and Tejaswi, S., "Road and traffic sign detection using color segmentation," in [*Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems (MCCS 2017)*], Nath, V. and Mandal, J. K., eds., 189–200, Springer Singapore, Singapore (2019).

[19] Yadav, H., Bansal, P., and KumarSunkaria, R., "Color dependent k-means clustering for color image segmentation of colored medical images," in [*2015 1st International Conference on Next Generation Computing Technologies (NGCT)*], 858–862 (Sep. 2015).

[20] Terol-Villalobos, I. R., "Morphological image enhancement and segmentation," *Advances in Imaging and Electron Physics* **118**, 207 – 273, Elsevier (2001).

[21] Litus, I., Kravchenko, N., and Bochkarev, S., "Irregular objects. shape detection and characteristic sizes," *CEUR Workshop Proceedings* **1814**, 28–35 (2017).

[22] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in Neural Information Processing Systems 25*], Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., eds., 1097–1105, Curran Associates, Inc. (2012).

[23] Mogelmose, A., Trivedi, M. M., and Moeslund, T. B., "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems* **13**, 1484–1497 (Dec 2012).

[24] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 1137–1149 (June 2017).

[25] Hen, K., Zhang, X., Ren, S., and Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition," in [*Lecture Notes in Computer Science*], *Computer Vision-ECCV 2014* **8691**, 1027–1035, Springer, Cham (2014).

[26] Girshick, R., "Fast r-cnn," in [*2015 IEEE International Conference on Computer Vision (ICCV)*], 1440–1448 (Dec 2015).

[27] Wei, L., Dragomir, A., Dumitru, E., Christian, S., Scott, Reed Cheng-Yang, F., and Alexander C., B., "Ssd: Single shot multibox detector," in [*Computer Vision ECCV 2016*], **9905**, Springer, Cham (2016).

[28] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 770–778 (June 2016).

[29] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR* **abs/1704.04861** (2017).