Traffic Sign Detection and Recognition

Luka Maletin, Aleksandar Nikolić, Helena Zečević
Department of Computing and Control Engineering
Faculty of Technical Sciences, University of Novi Sad
Novi Sad, Serbia

***Abstract*—During the last decade the importance of Intelligent Transport Systems and Advanced Driver Assistance Systems has been increasing rapidly. These systems are important because they improve the comfort and safety of driving. One crucial part of these systems is the recognition of traffic signs. The first part of the recognition process is the detection of traffic signs in a picture or a video frame. Traffic signs are designed in a specific way (e.g. certain colors and shapes) and are strategically placed in the environment so that they are easily spotted. Due to these characteristics, color and shape detection are widely used for the detection phase. Both of these methods are discussed in this paper, along with their advantages and disadvantages. Both color and shape detection have proven to be very effective, especially when used in conjunction. After the detection and extraction of blobs containing only the parts of the picture with traffic signs, the recognition phase begins. The purpose of the recognition phase is to determine exactly which sign is present in the blob. Template matching and many different classifiers can be used for the recognition. Here we will discuss recognition using neural networks and support vector machines. The main advantage of neural networks over template matching is that no input transformations are needed, however there is a training overhead. Support vector machines, comparing to neural networks, are less prone to over-fitting and can be trained faster. The detection phase was very successful, although a small dataset was used. Both recognition methods used have proven to be very accurate and achieved decent performance.**

Keywords—traffic sign detection; traffic sign recognition; color segmentation; shape detection; neural networks; SVM

I. Introduction

Traffic sign detection and recognition have a very important role in Advanced Driver Assistance Systems (ADAS) and Intelligent Transport Systems (ITS). Other parts of ADAS and ITS include road lanes detection, obstacles recognition, etc., but traffic signs carry a lot of information and are easily missed by the driver. During the last decade the development of ADAS and ITS has rapidly increased, with a lot of big companies getting involved. These systems aim to increase the comfort of driving and traffic safety. Other applications include sign inventory and highway maintenance [1].

Sign detection is the first step in the recognition of signs and its goal is to answer whether the given picture or video frame contains a traffic sign or not. If a sign or multiple signs are present they are extracted and forwarded to the recognition phase. The two most adequate techniques are color detection and shape detection. Both techniques have their advantages and disadvantages, so different authors apply them in different ways. Some use only color detection or only shape detection, while others use both, either sequentially or in conjunction [2].

In the recognition phase the signs need to be classified. The information whether a sign is, for instance, a speed sign, is not enough. The information that the sign carries (text, picture, arrows, numbers, etc.) needs to be extracted and processed as well. Most approaches used in this phase can be divided into two categories: template-based and classifier-based.

In order for the whole system to be applicable in the real world, both the detection and recognition phases need to be robust and have high performance. Signs need to be detected in different lighting conditions, weather conditions (e.g. rain, snow), geometric orientations, etc. In the detection phase, false-negatives should be at a minimum rate.

In this paper we will address both the detection and recognition phase, the problems faced, and the advantages and disadvantages of different approaches. Since sign recognition is based on machine learning, in section 2 we introduce the basics of this field. Section 3 covers sign detection using color segmentation and shape detection. In section 4 we cover sign recognition using neural networks (NN) and support vector machines (SVM). Results are presented and discussed in section 5. Section 6 provides a summarization of the methods used for detection and recognition.

II. Machine Learning

Machine learning represents a computer's ability to learn automatically based on experience. Experience is gained from e.g. observed data, statistical data, etc. The computer is able to interpret this data in a way that leads to self-improvement. By improving itself, the computer is capable of acquiring domain-specific knowledge in a certain field. Simply put, machine learning is an intelligent way of acquiring knowledge. It is loosely connected to the way humans learn, however it is not exactly the same, because our learning process is still not researched enough and is too complex.

Based on reasoning of the system, there are two kinds of machine learning: inductive and deductive. Inductive learning methods extract rules from big datasets, and then apply them on new datasets. It cannot be guaranteed that these rules will be correct. Deductive learning methods use a set of known rules to produce additional rules that guarantee to be true.

Machine learning is usually used for problems that are associated with artificial intelligence, e.g. image recognition, prediction, medical diagnosis, data mining, etc.

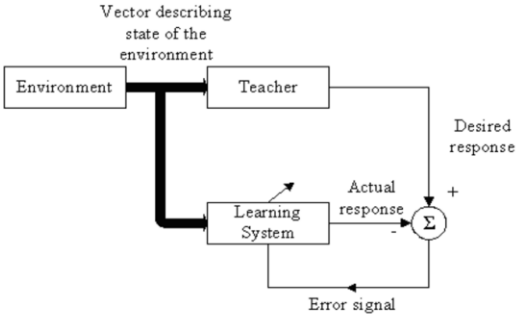

Fig. 1. In supervised learning a "teacher" is used for providing feedback

Another more known classification of machine learning is based on whether the data is labeled or not – supervised and unsupervised learning.

A. *Supervised Learning*

Supervised learning is a machine learning technique in which we try to create a function from the training data. The function should represent a generalization of the problem, and can be used for prediction. The training data are made of pairs of input vectors and desired outputs (i.e. labels). The task of supervised learning is to predict the output values of new vectors based on knowledge acquired from the training dataset.

The visual representation of supervised learning is shown in Fig. 1. Both the teacher and the learning system get input vectors from the environment. The learning system uses the teacher's response to generate an error indicator, which is then used to adjust the system's parameters. This process is repeated iteratively until the system is able to reproduce the teacher's outputs, and is capable of dealing with the environment without the help of the teacher.

B. *Unsupervised Learning*

The second type of machine learning is unsupervised learning. In this method we use a training set in which the outputs are unknown (i.e. unlabeled). The task is to find some similarities between the data samples, in order to uncover some sort of structure within the data.

In this paper we discuss some supervised machine learning techniques to recognize traffic signs in images.

III. Sign Detection

Traffic signs play a very important role in our everyday commute. They provide drivers with very important information. They give warnings, set restrictions, or just give general information about surroundings (crossing signs, distance, etc.). Signs are produced in a way that they are easily distinguished by drivers. They are made with a limited number of specific colors (e.g. red, blue, yellow, green) and shapes (triangle, square, circle, rectangle), because such colors and shapes, when used together, are not normally found in nature [3][4][5]. Traffic signs are made to be easily spotted and humans are able to process information they carry in a matter of moments and with great accuracy [6].

The placement of traffic signs is carefully thought out. They are placed near roads, intersections, crosswalks, etc. They are also always elevated, so that other vehicles don't cover them. This can be very important considering that sometimes photographs (e.g. stickers) of traffic signs can be found on other vehicles or elsewhere. A good example of this would be that vehicles sometimes have speed sign stickers on the rear end. These signs should either not be detected at all, or they should be interpreted in a different way. This could be used as a possible optimization.

Traffic sign detection is usually the first step of the traffic sign recognition process. In this step there is a great risk of not detecting some of the signs in a photo or video frame (false-negatives), which makes this a very important step. The problem with missing some signs in the detection phase is that it can't be fixed in the recognition phase. Signs that go undetected in this step won't be recognized.

Since traffic signs are made with specific and limited number of colors and shapes, most authors tend to go with color and/or shape based detection. Shape detection is a lot computationally-heavier than color detection, especially considering that a lot of sign-like shapes can be found in the background. Sign shapes are more distinguishable when color is taken into account as well. So, using only shape based detection can have worse performance [4]. Color based traffic sign detection has high performance, but using it solely also encounters certain problems. Color is sensitive to lighting, it wears down eventually and it differs on different cameras [3][5]. Some suggest that the best practice would be to use the combination of color and shape detection [3][5]. A lot of authors first use color detection, and then shape detection. The disadvantage of using them sequentially would be that, in case of false-negatives in the color detection stage, it would be impossible to recover them in the shape detection stage. Because of that, [7] suggest that it would be best to use the joint application. In section *A* we will discuss the color based detection, and section *B* describes shape based detection.

A. *Color Detection*

As mentioned above, color based detection isn't a perfect solution. These are the most common problems:

- Signs can get damaged over time, since they are exposed to different weather conditions (snow, rain, sunlight etc.), and even vandalism.
- Sunlight and air pollutants can cause color of traffic signs to wear down over time.
- Color looks different depending on daytime and weather conditions [5]. See Fig. 2 for examples.
- Different illuminations can be present on a single surface (e.g. a shadow covering a part of the sign) [4].
- Non-sign objects of similar color and shape can be found outside (e.g. restaurant and store signs). This can lead to a lot of false-positives which takes a toll on performance [8].

Fig. 2. Different weather conditions can cause problems during the detection phase

- Color thresholds need to be manually tuned, which is a time consuming task [2].

Even with all of these drawbacks, high performance of color based detection is the reason that this is the most used technique.

The first part of color based detection is color segmentation. Since the RGB color space is dependent on illumination, a lot of authors first transform the image from RGB to an illumination independent color space before performing color segmentation [3][4][5]. Most commonly used are YUV (Y – brightness, U and V – color components) and HSV/HSI (Hue, Saturation, Value/Intensity) color spaces. These transformations need to be linear, otherwise they would be too computationally expensive.

Color segmentation is a process of “marking” regions of interest in an image, for instance a red-colored border of a speed sign. The segmentation process usually returns a binary mask where possible signs are white-colored and the background is black [3][4][8]. Depending on the color space used, different authors use different methods of creating a binary mask. On Fig. 3 are shown three examples of a binary mask gotten as a result of color segmentation. The first image is a binary mask that’s marking all the possible signs, based on the inner white area of the circular signs. The second and third image represent the binary masks marking all red and blue pixels of the possible signs, respectively.

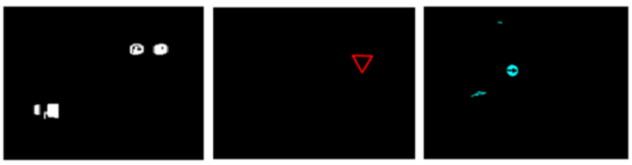

Fig. 3. Binary masks used to detect traffic sign candidates

Color segmentation often results in a lot of false-positives, which is why the number of potential candidates needs to be reduced. This is usually done with shape detection. Some authors perform simpler calculations using shape and geometric properties of traffic signs instead of more complicated shape detection methods, such as pattern matching [8].

B. Shape Detection

Shape based detection of traffic signs is a method that can be used solely, or paired with color based detection. Shape detection can be sufficient when it comes to sign detection since shapes are not affected by lighting in any way. Also, weather conditions and time of day don’t affect shapes as they affect colors. Another thing in support of shape detection is the fact that traffic sign colors are not standardized in countries around the world. This means that systems based on color detection would need to be tuned manually for different datasets (i.e. in different countries) [5]. Also, in case of signs that are placed very close by, color segmentation may detect those signs as one. In this case, additional shape detection is assurance that these signs will be correctly classified in the recognition stage [4]. Shape detection also has its downsides. Sign-like shapes can be found in outdoor photos, such as billboards, restaurant signs, windows, etc. Signs can also be occluded by other objects. They can be physically damaged, turned upside down, or rotated sideways.

Many authors use shape detection in combination with color detection [3][4][8]. Using shape detection after color segmentation may result in the loss of some good sign candidates, in case of falsely not detecting them. Color segmentation, considering that color is very dependent on lighting, can falsely reject a sign as a non-sign object. This is the reason some authors suggest using the joint treatment of color and shape detection [7].

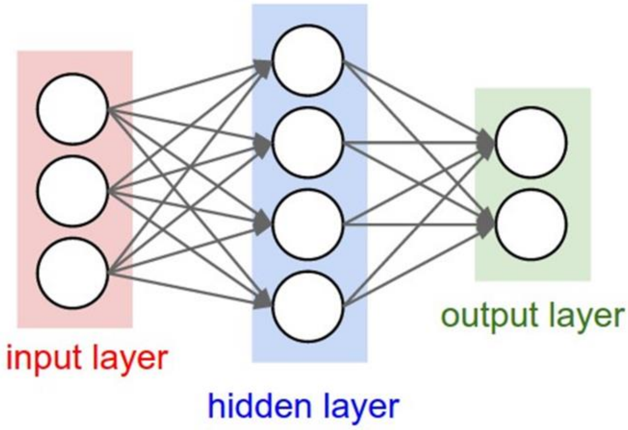

Fig. 4. Basic architecture of a neural network – input neuron values are propagated to the output via weighted synapses

Fig. 5. An example of a synthetic data set used for training a neural network

There are a lot of possibilities of using shape detection when it comes to choosing the algorithm. Some authors use Hough transform for detecting circles [3], some perform calculations based on geometric properties like height and width of traffic sign shapes [8], others go with pattern matching [4]. Regardless of the possible problems with color and shape detection and the methods chosen for both, authors have had very good results [4]. Detection stage was successful in over 95% of the cases [3][8].

IV. Sign Recognition

When it comes to sign recognition, the input should be a sign image, usually passed on from the detection phase. That image is cropped from the original input image, and its dimensions should be only as large as the dimensions of the sign itself. The input images also need to be scaled to a specific dimension, so that all images can be used as inputs in the same way. It should be mentioned that not all traffic sign recognition techniques require a separate detection phase. Recognition can be achieved by using solely genetic algorithms or simulated annealing for instance [9].

In the recognition phase many different techniques can be used, such as template matching, clustering classifiers, nearest neighbor classifiers, Laplace kernel classifiers, fuzzy classifiers and Bayesian networks. However, the most widely used are neural networks [4][5][9][10], and there has also been a significant amount of SVM [12][13] usage.

A. Recognition Using Neural Networks

The basic architecture of a neural network is shown in Fig. 4. The values from the input neurons are forwarded to the next layer via synapses of certain weights. Each neuron sums up all its inputs from the previous layer, and passes them to the activation function. The activation function converts the neuron's input to its output, and its purpose is to introduce nonlinearity and squash the value to a certain range. During training of the neural network back-propagation is used to minimize the error of the outputs. The error on the output is propagated backwards to neurons, which alter their synapses' weights depending on how much impact they had on the total error. The impact of a weight is determined by calculating the derivative of the total error in respect to the weight.

Neural networks are very suitable for real-time recognition of traffic signs. Comparing to template matching, which is also widely used, neural networks don't have to transform the input images, or move the template in order to find a match [5]. The main disadvantages of using neural networks are the risk of over-fitting (excessive adapting to a specific training dataset), and the training overhead, which also means that adding a new sign usually requires the whole network to be retrained.

The architecture of the network can vary greatly. In [4] a specific neural network was trained for each sign category. The number of input neurons for each network was equal to the number of pixels (2500 input neurons for a 50x50 input image size). The number of output neurons was corresponding to the number of signs in the specific category. Each output neuron had a value from 0 to 1 which represented the probability that the input image is the sign related to that output. One hidden layer was used. In [10] a more sophisticated method was used. Only 63 input neurons were used: 30 inputs from a vertical histogram, 30 inputs from a horizontal histogram and normalized average maximum pixel values for red, green, and blue.

One of the problems when using neural networks for recognition is the lack of a proper dataset. Most studies used smaller datasets limited to several different traffic signs and their variations. A proper dataset would also need samples of varying qualities (e.g. new signs vs old, rusty, etc.) and weather conditions (sunrise, sunset, fog, rain, snow, etc.). Because of that, some researchers [4] use synthetic datasets for training neural nets. A synthetic dataset is created by applying various image filters and geometric transformations to a real photograph of a sign, as shown in Fig. 5. One study that had a satisfactory dataset is [11]. The images were acquired from three different countries (Czech Republic, Spain and the United Kingdom) and consisted of 1500 road scene images and over 3500 images of individual traffic signs. The images were shot in various illumination and meteorological conditions.

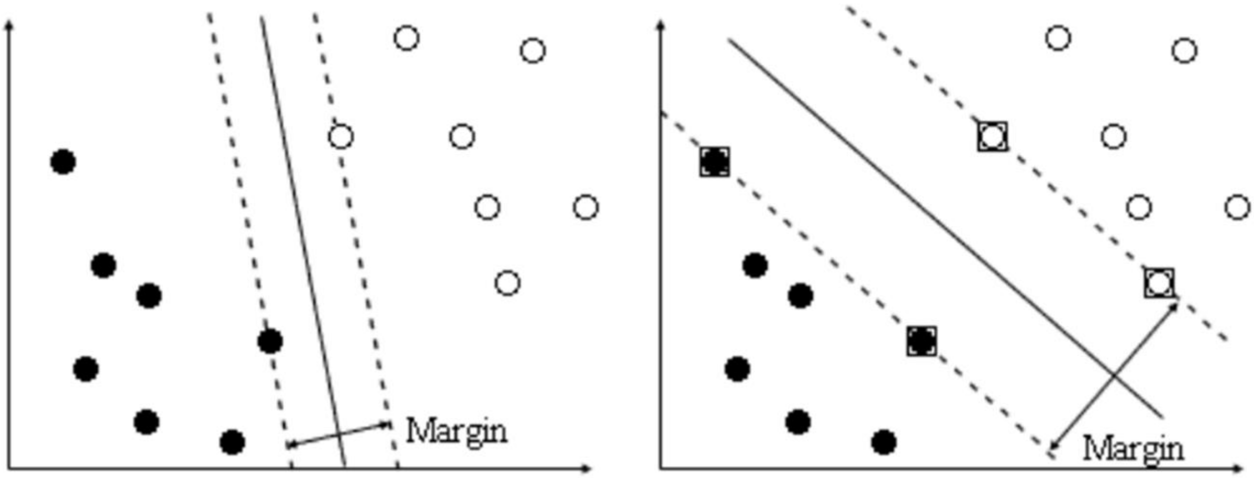

Fig. 6. Different margins used for the same data – wider margins are better at generalizing a given problem

For the implementation of the NN there are many libraries available. In [4] the Lightweight Neural Network++ (LWNN++) was used, and in [10] the Intel Open Source Computer Vision Library (OpenCV).

B. Recognition Using Support Vector Machines

SVM is a supervised learning method that classifies data by constructing a hyperplane (a subspace of one dimension less than the observed space). The maximum margin of the hyperplane is defined by the support vectors. In Fig. 6 we can see two ways for the hyperplane to classify data. The first one is over-fitted and will achieve high accuracy while training, but will misclassify new data. A larger margin is better at generalizing the given problem. Apart from being a very fast and highly accurate classification method, unlike many other classifiers (including neural networks) SVM are less prone to over-fitting. Although SVM are primarily used for classifying data into two classes, a multiclass classification can be implemented by training many binary SVM.

For binary classification, the training data are labeled $\{x_i, y_i\}$, where $i = 1\ldots k$, $y_i \in \{-1, 1\}$, $x_i \in \{R^d\}$. Vector x_i is the feature vector, the values y_i are "1" for the first class and "-1" for the second class, d is the dimension of the vector, and k is the number of training vectors. The points lying on the hyperplane $\{w, b\}$ satisfy $x \cdot w^T + b = 0$ [12]. If the training data are linearly separable, they satisfy:

$$y_i (x_i \cdot w^T + b) - 1 \geq 0 \ \forall i \qquad (1)$$

The simplest classifier that can be used is a maximum margin classifier, which can only be applied to linearly separable datasets. For satisfying real-world problems where the data are not linearly separable in the feature space, a soft margin classifier is used. Soft margin classifiers tolerate noise and outliers. They are extended from a maximum margin classifier by introducing a positive slack variable as in (2), where ξ_i, $i = 1\ldots k$ [14]. In that case, some data are allowed to be misclassified. Fig. 7 shows an example of a soft margin classifier.

$$y_i (x_i \cdot w^T + b) - 1 \geq \ \xi_i \ \forall i \qquad (2)$$

One of the main advantages of SVM is their ability to classify data that cannot be linearly classified in their current dimension. It accomplishes that by using a kernel method, also known as "kernel trick". Instead of mapping the feature vector to a higher dimension, the kernel trick allows us to do computations in higher dimensions while keeping the feature vector unchanged. By doing so, a lot of computational time is saved. An illustration of the kernel trick is shown in Fig. 8.

In the following equations x and z represent feature vectors.

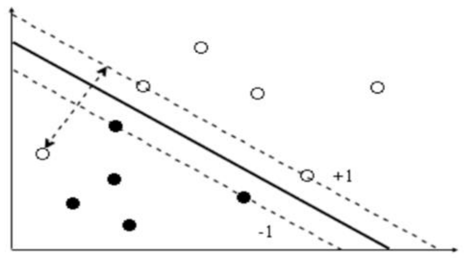

Fig. 7. An example of a soft margin classifier – outliers are tolerated

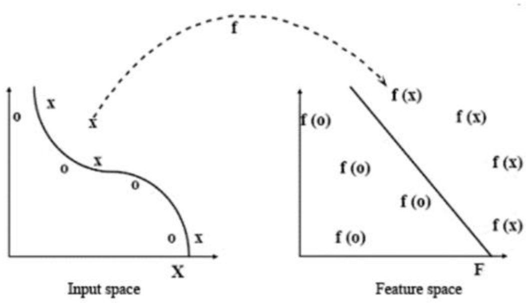

Fig. 8. Applying a "kernel trick" allows us to do calculations in a higher dimension, without changing the feature vector

Some of the most commonly used kernels [14] are:

- Linear kernel – the simplest kernel used when data can be linearly separated.

$$K(x, z) = \langle x, z \rangle \qquad (3)$$

- Polynomial kernel – used for modelling nonlinear data. Variable d is the polynomial degree.

$$K(x, z) = \langle x, z \rangle^{d} \qquad (4)$$

- Gaussian Radial Basis Function – most commonly used for SVM. Value σ is a free variable.

$$K(x, z) = \exp(-\frac{\| x - z \|^2}{2\sigma^2}) \qquad (5)$$

- Sigmoid function – most commonly used for neural networks. It provides a good balance between linear and nonlinear behavior. An SVM using the sigmoid function is equivalent to a two-layered feed-forward neural network. Variables γ and r are kernel parameters.

$$K(x, z) = \tanh(\gamma\langle x, z \rangle + r) \qquad (6)$$

In order to perform classification of traffic signs using SVM, it is preferred that the signs are first classified by shape. If shape detection was used in the sign detection phase, the results can be passed on to the recognition phase. Otherwise, SVM can be used for classifying sign blobs by shape. In [13] a histogram of oriented gradients (HOG) feature vectors were used as input vectors for the shape detection SVM. HOG represent the occurrence of gradient orientations in the image. The authors claim that the use of HOG method is very suitable due to the nature of traffic signs (strong geometric shapes and high-contrast edges). In [12] the input vectors were distance to borders (DtBs), which represent the distances from the external edge of the blob to its bounding box. Fig. 9 shows these distances for a triangular shape, where D1, D2, D3, and D4 are the left, right, upper, and bottom DtBs, respectively.

After classifying the blobs by shape, another set of SVM is

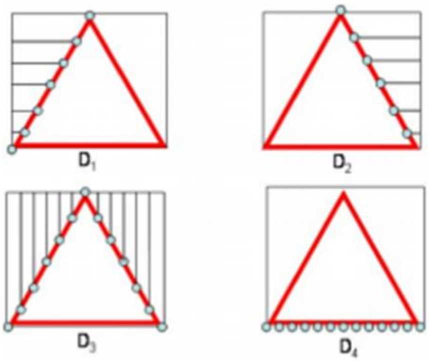

Fig. 9. Distance to borders method used for determining shapes

Fig. 10. To reduce the feature vector of SVM, only the points of interest are used (white pixels)

used for the recognition phase. Since the data are not linearly separable, Gaussian kernels were used in [12]. In order to recognize every sign, different one-versus-all SVM were trained. The amount of training samples per class was between 20 and 100, but only some of them define a decision hyperplane as support vectors.

In [12] 31 x 31 grayscale blobs were used so each picture was normalized to those dimensions. In order to reduce the feature vector, only the pixels that are part of the sign itself were used. These pixels of interest (PoI) can be extracted using DtBs from the shape detection phase. The PoI for a triangular and circular sign are shown in Fig. 10.

SVM solve some of the biggest problems of other recognition methods, like over-fitting. They also have their disadvantages. One of the biggest disadvantage of SVM lies in the choice of the kernel. If a wrong kernel is chosen it can actually lead to over-fitting. Some advice on selecting the right kernel is presented in [14]. Another problem of SVM is choosing the right values for the kernel parameters, like σ in the Gaussian kernel, and the slack variable. Optimal results for these parameters can be found in [12].

V. Results

In this section we will present some authors' results using the methods we researched.

A. *Results of Color and Shape Detection*

We have compared the results presented in several different papers, and we will present those from [3] and [8]. We chose these results as they both use color segmentation as the first step and shape detection as the second. However, these authors have chosen a different method for both of these steps. We will present a short description and results of both regarding speed signs.

Authors in [8] have concentrated on the red boundary surrounding the sign. They performed color segmentation in RGB color space, using two criteria. Both criteria concentrate on the red component of RGB. Color segmentation returns a binary mask where all red regions, or regions detected as a possible red, are marked. The number of marked regions is decreased by labeling connected regions as one. The next step is based on shape detection, which reduces the number of potential candidates. These authors perform simple calculations regarding geometric properties of signs, such as the ratio of height to width, the number of black pixels inside the sign, etc.

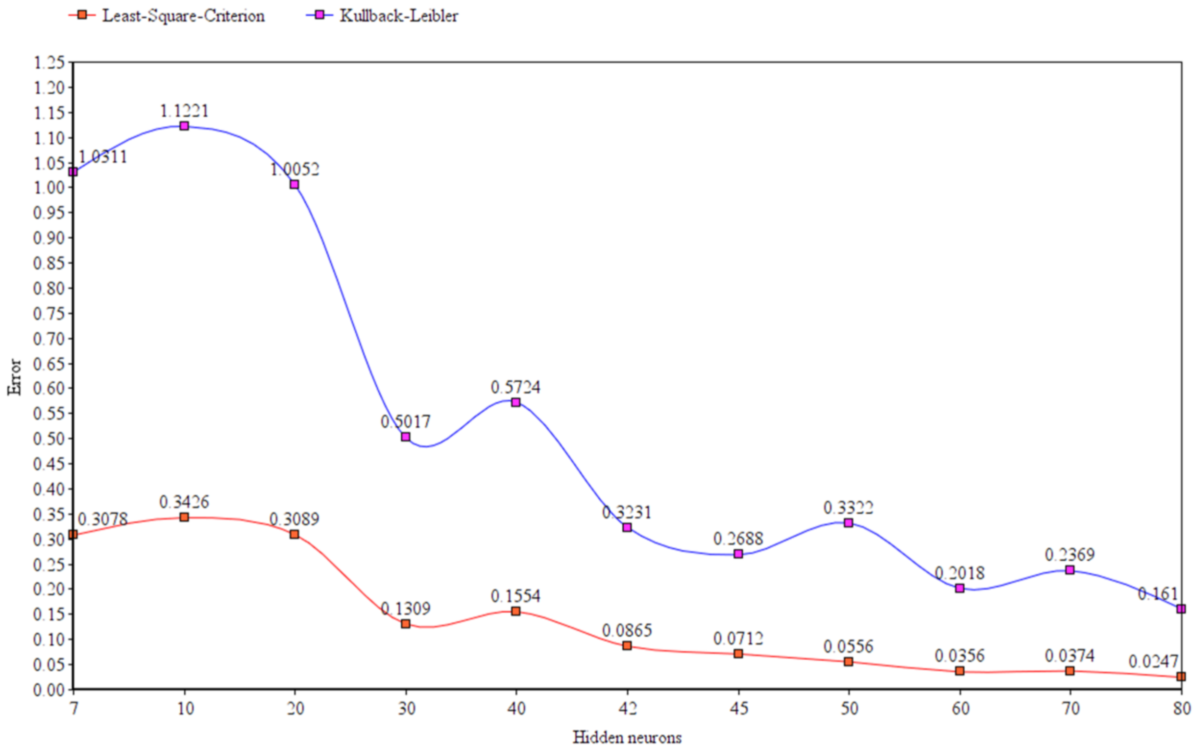

Fig. 11. Errors for different number of neurons in hidden layer – optimal results start with 42 neurons [10]

Authors in [3] used a YUV color space. They also acknowledge that these signs have a distinctive red boundary. However, they claim that the red color has a low intensity value. This is why they perform color segmentation concentrating on the inner white region of a speed sign. For each sign candidate they calculate aspect ratios and accept only candidates within certain thresholds. Since segmentation is sensitive to lighting they perform this step several times with minor changes to thresholds. After this, they move on to shape detection. They perform Hough transform in order to detect which of these candidates are actually circles.

In [8] the total number of speed and warning signs was 172, and only 4 of them weren't detected, which means the success rate was 97.67%. In [3] out of 71 signs tested, 2 weren't detected, a success rate of 97.18%.

These results show that these methods can be very successful, though we should keep in mind that a very small dataset was used in both cases.

B. Results of Neural Network based recognition

Here we will discuss the results of [10]. The input neurons used were:

1) 30 inputs from a vertical histogram
2) 30 inputs from a horizontal histogram
3) normalized average maximum pixel values for red, green, and blue

Different numbers of hidden layer neurons were tested, and for each test they measured the performance of the net using Least-Square-Criterion and the Kullback-Leibler distance. The test results are shown in Fig. 11. The optimal results start with 42 neurons in the hidden layer.

The achieved average processing time was 37.27 milliseconds per frame.

C. Results of SVM based recognition

In [14] a Canon MVX30i video camcorder was mounted on

TABLE I. TEST RESULTS FROM [14] SHOWING A HIGH SUCCESS RATE OF DETECTION AND RECOGNITION USING SVM

Sequence	**1 (day)**	**2 (day)**	**3 (day)**	**4 (rain)**	**5 (night)**
Number of images	746	1774	860	995	798
Number of traffic signs	21	21	20	25	17
Detection of traffic signs	218	237	227	285	127
Noisy candidate blobs	601	985	728	622	434
False alarms	0	3	4	8	7
Confused recognition	4	4	4	2	7

the front of the windshield for the recoding of the test sequences. While the sequences were recorded the car was moving at a usual driving speed. The size of each image was 720 x 576 pixels and the time between each frame was 0.2 s.

The processing time for each frame was 1.77 s on a 2.2 GHz Pentium 4-M. Each sequence was recorded over a 4 km road.

In Table 1 the results for 5 sequences are shown. Sequences 1, 2 and 3 are recorded during a sunny day, while 4 corresponds to a rainy day, and sequence 5 was recorded during nighttime.

The reason for confused recognition can be contributed to the long distance between the sign and the camera, or poor lighting in nighttime the sequence.

An improvement to the system can be made by a rule that would discard a recognition if a sign is not recognized in two sequential frames of the sequence. We can conclude from the results that false alarms do not appear in the sequence multiple times, so the previous rule will reject those recognitions.

The system was also tested with synthetic occlusions on the signs. In that case, the probabilities of the recognition were 93.24%, 67.85%, and 44.90% for the small, medium, and large occlusions, respectively. Based on these results we can conclude that recognition preforms the worst when the occlusions are in the middle of the sign and when the size of the occlusions is increased.

VI. CONCLUSION

In this paper we discussed the importance of traffic sign detection and recognition in Advanced Driver Assistance Systems and Intelligent Transport Systems and its ability to improve safety and comfort of drivers and their everyday lives. Also, we presented some of the biggest obstacles in solving this problem. We introduced the basics of machine learning and some of the most used methods for the detection and recognition phases.

Color and shape detection are two methods used in the detection phase. It was concluded that because each one has its advantages and disadvantages, and they are not mutually exclusive, they are best used in conjunction.

In the recognition phase two of the most popular methods in the field were used: neural networks and support vector machines. Neural networks have proven to be very good in solving these types of problems, even though there are some clear disadvantages. SVM, although a fairly new method for solving this problem, is shown to be very capable in solving it and even resolving some of the biggest problems of neural networks.

This paper shows the most used methods in the industry, and is meant to be a starting point for researchers who want to learn more about some of these methods. A good next step would be to explore some other methods that have proven to be effective, e.g. probabilistic neural networks.

REFERENCES

[1] De la Escalera, Arturo, J. Ma Armingol, and Mario Mata. "Traffic sign recognition and analysis for intelligent vehicles." Image and vision computing 21.3 (2003): 247-258.

[2] Bahlmann, Claus, et al. "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information." Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE, 2005.

[3] Miura, Jun, et al. "An active vision system for on-line traffic sign recognition." IEICE TRANSACTIONS on Information and Systems 85.11 (2002): 1784-1792.

[4] Broggi, Alberto, et al. "Real time road signs recognition." Intelligent Vehicles Symposium, 2007 IEEE. IEEE, 2007.

[5] Fleyeh, Hasan, and Mark Dougherty. "Road and traffic sign detection and recognition." Proceedings of the 16th Mini-EURO Conference and 10th Meeting of EWGT. 2005.

[6] Stallkamp, Johannes, et al. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition." Neural networks 32 (2012): 323-332.

[7] Fang, Chiung-Yao, Sei-Wang Chen, and Chiou-Shann Fuh. "Road-sign detection and tracking." IEEE transactions on vehicular technology 52.5 (2003): 1329-1341.

[8] Andrey, Vavilin, and Kang Hyun Jo. "Automatic detection and recognition of traffic signs using geometric structure analysis." SICE-ICASE, 2006. International Joint Conference. IEEE, 2006.

[9] de la Escalera, Arturo, et al. "Neural traffic sign recognition for autonomous vehicles." Industrial Electronics, Control and Instrumentation, 1994. IECON'94., 20th International Conference on. Vol. 2. IEEE, 1994.

[10] Lorsakul, Auranuch, and Jackrit Suthakorn. "Traffic sign recognition using neural network on opencv: Toward intelligent vehicle/driver assistance system." 4th International Conference on Ubiquitous Robots and Ambient Intelligence. 2007.

[11] Prieto, Miguel S., and Alastair R. Allen. "Using self-organising maps in the detection and recognition of road signs." Image and Vision Computing 27.6 (2009): 673-683.

[12] Maldonado-Bascon, Saturnino, et al. "Road-sign detection and recognition based on support vector machines." IEEE transactions on intelligent transportation systems 8.2 (2007): 264-278.

[13] Greenhalgh, Jack, and Majid Mirmehdi. "Real-time detection and recognition of road traffic signs." IEEE Transactions on Intelligent Transportation Systems 13.4 (2012): 1498-1506.

[14] Shi, Min. "Road and Traffic Signs Recognition using Vector Machines." (2006).