An overview of traffic sign detection methods

Karla Brkić
Department of Electronics, Microelectronics, Computer and Intelligent Systems
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
Email: karla.brkic@fer.hr

***Abstract*—This paper reviews the popular traffic sign detection methods prevalent in recent literature. The methods are divided into three categories: color-based, shape-based, and learning-based. Color-based detection methods from eleven different works are studied and summarized in a table for easy reference. Three shape-based detection methods are presented, and a recent method based on Hough transform is studied in detail. In the section on learning-based detection, we review the Viola Jones detector and the possibility of applying it to traffic sign detection. We conclude with two case studies which show how the presented methods are used to design complex traffic sign detection systems.**

I. Introduction

Recent increases in computing power have brought computer vision to consumer-grade applications. As computers offer more and more processing power, the goal of real-time traffic sign detection and recognition is becoming feasible. Some new models of high class vehicles already come equipped with driver assistance systems which offer automated detection and recognition of certain classes of traffic signs. Traffic sign detection and recognition is also becoming interesting in automated road maintenance. Every road has to be periodically checked for any missing or damaged signs, as such signs pose safety threats. The checks are usually done by driving a car down the road of interest and recording any observed problem by hand. The task of manually checking the state of every traffic sign is long, tedious and prone to human error. By using techniques of computer vision, the task could be automated and therefore carried out more frequently, resulting in greater road safety.

To a person acquainted with recent advances in computer vision, the problem of traffic sign detection and recognition might seem easy to solve. Traffic signs are fairly simple objects with heavily constrained appearances. Just a glance at the well-known PASCAL visual object classes challenge for 2009 indicates that researchers are now solving the problem of detection and classification of complex objects with a lot of intra-class variation, such as bicycles, aeroplanes, chairs or animals (see figure 1). Contemporary detection and classification algorithms will perform really well in detecting and classifying a traffic sign in an image. However, as research comes closer to commercial applications, the constraints of the problem change. In driver assistance systems or road inventory systems, the problem is no longer how to efficiently detect and recognize a traffic sign in a single image, but how to reliably detect it in hundreds of thousands of video frames without any false alarms, often using low-quality cheap sensors available in mass production. To illustrate the problem of false alarms, consider the following: one hour of video shot at 24 frames per second consists of 86400 frames. If we assume that in the video under consideration traffic signs appear every three minutes and typically span through 40 frames, there are a total of 800 frames which contain traffic signs and 85600 frames which do not contain any signs. These 85600 frames without traffic signs will be presented to our detection system. If our system were to make an error of 1 false positive per 10 images, we would still be left with 8560 false alarms in one hour, or *two false alarms every second*, rendering the system completely unusable for any serious application! To make the problem even harder, we cannot expect the vehicle on which a commercial traffic sign detection system will be deployed to be equipped with a very high-resolution camera or other helpful sensors, as the addition of such sensors increases production costs.

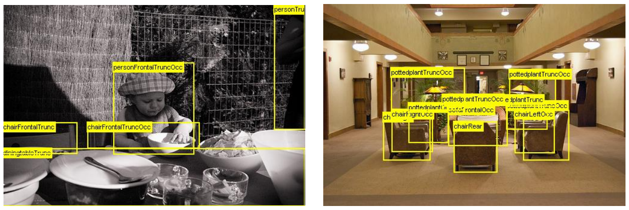


Fig. 1: Labeled examples from the PASCAL visual object classes challenge 2009. [1]

This paper presents an overview of basic traffic sign detection methods. Using the presented methods as commercial stand-alone solutions is impossible, as they fail to provide the required true positive and false positive rates. However, combining the methods has a synergistic effect, so they are commonly used as building blocks of larger detection systems. In this paper, the traffic sign detection methods are divided into three categories: color-based methods, shape-based methods and methods based on machine learning. After introducing the methods, we present two traffic sign detection systems which use them.

A. Traffic sign classes - the Vienna convention

Before investigating common traffic sign detection methods, it is useful to briefly review the data on which these methods operate, i.e. the classes of traffic signs. In 1968, an international treaty aiming to standardize traffic signs across different countries, the so-called Vienna Convention on Road Signs and Signals, was signed [2]. To date there are 52 countries which have signed the treaty, among which 31 are in Europe. The Vienna Convention classifies road signs into seven categories, designated with letters A-H: danger warning signs (A), priority signs (B), prohibitory or restrictive signs (C), mandatory signs (D), information, facilities, or service signs (F), direction, position, or indication signs (G) and additional panels (H). Examples of Croatian traffic signs for each of the categories are shown in figure 2.

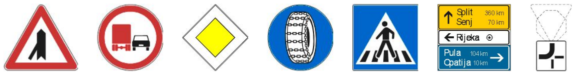
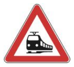
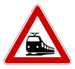
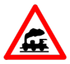
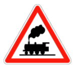
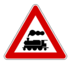
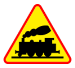
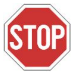
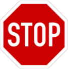
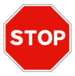
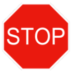
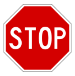
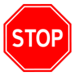


Fig. 2: Examples of traffic signs. From left to right: a danger warning sign, a prohibitory sign, a priority sign, a mandatory sign, an information sign, a direction sign and an additional panel.

In spite of appearances of traffic signs being strictly prescribed by the Vienna convention, there still exist variations between countries which have signed the treaty. The variations are seemingly irrelevant for a human, but might pose significant challenges for a computer vision algorithm. For an example, see table I, where variations of two traffic signs across six different European countries are shown. Therefore, contemporary traffic sign detection systems are still country-specific.

TABLE I: Two traffic signs, as defined by regulations of six different European countries. Notice the subtle differences that don't present a problem for a human, but might influence the performance of a detection algorithm.

Croatia	Germany	Spain	France	Italy	Poland
					
					

B. Evaluating traffic sign detection methods

Research concerning traffic sign detection is often hard to compare, as different researchers approach the problem with different application areas and constraints in mind. Traffic sign detection methods are inherently dependent on the nature data for which they were developed. Some factors in which the methods differ are the following:

- Input type: videos or static images?
- Scope of the method: is the method applicable for a single traffic sign class or for multiple classes?
- Filming conditions: is the data shot in broad daylight, in nighttime or both? Are there adverse weather conditions such as rain, snow, fog?
- Sensor type: high resolution or low resolution camera, grayscale or color? Multiple cameras? Other sensors?
- Processing requirements: should the signs be detected in realtime or is offline processing acceptable?
- Acceptable true positive and false positive rates: determined by the nature of the problem.

The nature of the problem, the availability of sensors and the target application determine which method to use. For example, color-based detection is pointless if we are working with a grayscale camera. On the other hand, it might be very useful if we are trying to detect traffic signs in high resolution color images taken in broad daylight with a high quality camera. Shape-based detection might not work if we are using a camera with interlacing. Learning-based approaches might be a perfect solution if we have a lot of labeled data, but if no labeled data is available we cannot use them.

II. Color-based detection methods

The prevalent approach in detecting traffic signs based on color is very obvious - one finds the areas of the image which contain the color of interest, using simple thresholding or more advanced image segmentation methods. The resulting areas are then either immediately designated as traffic signs, or passed on to subsequent stages as traffic sign location hypotheses (i.e. regions of interest). The main weakness of such an approach lies in the fact that color tends to be unreliable - depending on the time of day, weather conditions, shadows etc. the illumination of the scene can vary considerably. RGB color space is considered to be very sensitive to illumination, so many researchers choose to carry out the color-based segmentation in other color spaces, such as HSI or L*a*b.

A. Color spaces: a short review

To understand why some color spaces are considered illumination-sensitive and some not, we briefly review the theory of color spaces. A color space is defined using a color model. In general, a color model is an abstract mathematical model which defines how colors can be represented as tuples of numbers. All the valid tuples constitute a color space. Common dimensionality of tuples is three to four. There is a myriad of color spaces which differ in the basic colors used. Some of the most pupular are RGB (red-green-blue) color space, HSI (hue-saturation-intesity) color space, L*a*b (lightness-color opponent dimensions), CMYK (cyan-magenta-yellow-black) color space, CIECAM97 color space.

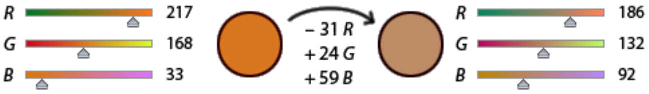


Fig. 3: Obtaining two different shades of orange by mixing red, green and blue components. By just looking at the differences between the channels it would be hard to conclude that the colors are similar.

TABLE II: A summary of color-based traffic sign detection methods from eleven different papers.

Authors	Color space used	Type	Color of interest	Formulas
Benallal and Meunier [3]	RGB	thresholding	red	If $R_i > G_i$ and $R_i - G_i \geq \delta_{RG}$ and $R_i - B_i \geq \delta_{RB}$ then pixel p_i is RED.
Estevez and Kehtarnavaz [4]	RGB	thresholding	red	$\text{REDNESS} = R - \max(G, B) - a \times \text{abs}(G - B)$, thresholded at zero to obtain the red pixels.
Varun et al. [5]	RGB	thresholding	red	$1.5R > G + B$
Broggi et al. [6]	RGB	enhancement	red	$\alpha_{\min} \times G < R < \alpha_{\max} \times G$ and $\beta_{\min} \times B < R < \beta_{\max} \times B$ and $\gamma_{\min} \times B < G < \gamma_{\max} \times B$
Ruta et al. [7]	RGB	enhancement	red, blue, yellow	$R >$ theshold and $f_R(x) = \max(0, \min(x_R - x_G, x_R - x_B)/s)$
Kuo and Lin [8]	HSI	thresholding	red	$H \geq 0$ and $H < 0.111\pi$ or $H \geq 1.8\pi$ and $H < 2\pi$, $0.1 < S \leq 1$,$0.12 < I < 0.8$
Piccioli et al. [9]	HSI	thresholding	red	$-30^\circ < H < 30^\circ$, minimum saturation of 20%
Paclik et al. [10]	HSV	thresholding	any	Set $V = \sup(R, G, B)$, $S = 256\frac{V - \inf(R,G,B)}{V}$. If $V = R$ then $H = \frac{G-B}{V - \inf(R,G,B)}$ else if $V = G$ then $H = 2 + \frac{B-R}{V - \inf(R,G,B)}$ else if $V = B$ then $H = 4 + \frac{R-G}{V - \inf(R,G,B)}$. Finally threshold H to obtain target color, black and white are found by thresholding S and I.
Fang et al. [11]	HSI	similarity measure	any	Calculate the hue h of the candidate pixel. Denote $(h_1, ..., h_q)$, a set of hue values which a particular color of a traffic sign can take (precomputed), normally distributed with variance σ^2. The output is the degree of similarity $z = \max_{k=1,\ldots,q}(z_k)$ where $z_k = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(h-h_k)^2}{2\sigma^2})$.
Escalera et al. [12]	HSI	thresholding	red	Set $H(i) = 255\frac{i_{\min} - i}{i_{\min}}$ if $0 \leq i \leq i_{\min}$; $H(i) = 0$ if $i_{\min} \leq i \leq i_{\max}$, $H(i) = 255\frac{i - i_{\max}}{i_{\max}}$ if $i_{\max} \leq i \leq 255$. Then set $S(i) = i$ if $0 \leq i \leq i_{\min}$; $S(i) = 255$ if $i_{\min} \leq i \leq 255$. Multiply the two images and upper bound the results at 255.
Gao et al. [13]	CIECAM97	thresholding	red	(a) average daylight conditions: hue 393-423, chroma 5795; (b) sunny day: hue 375-411, chroma 31-43; (c) cloudy day: hue 370-413, chroma 25-45; (d) rainy day: hue 345-405, chroma 30-50

In the RGB color model, colors are specified as mixtures of red, green and blue components. Figure 3[1] illustrates how two different shades of orange can be obtained by mixing red, green and blue. The differences between RGB components for the first and the second color are -31, +24 and +59. The RGB color model is unintuitive from a human standpoint - a human might expect to vary just one parameter, namely illumination, to obtain the second color from the first. It would be hard for a human to guess the changes in R, G and B necessary for the required change in color. Similarly, it is hard for a computer to learn that these two colors are similar based purely on the distances between the numerical values of their R, G and B.

Several color models were designed to address this problem. In the mid-1970s researchers in computer graphics developed the HSL and HSV color models, which rearrange the RGB color space in cylindrical coordinates so that the resulting representation is closer to human visual perception. A very similar model is HSI, commonly used in computer vision. HSL, HSV and HSI differ only in the definition of the third component – L stands for lightness, V for value and I for intensity. The first two components are hue and saturation. In the HS* cylinder, the angle around the central vertical axis corresponds to hue, the radius to saturation and the height to lightness or value or intensity. In the HS* representation, the components of two similar colors are numerically much closer, which is why it is said to be less sensitive to illumination.

B. Color-based segmentation

The color of a traffic sign should be easily distinguishable from the colors of the environment. After all, traffic signs are specifically designed with this requirement in mind. In order to find the sign of a target color, one segments the image based on that color. Image segmentation is a process which assigns a label to each pixel of an image so that the pixels with the same labels share similar visual characteristics. The simplest method of image segmentation is thresholding: every pixel with a value above a certain threshold is marked with the appropriate label. Various authors have experimented with color thresholding, especially in the 1990s. High detection rates were reported, but the experiments were usually done on small testing sets. For example, simple thresholding formulas (see table II) are used by Varun et al. [5] and Kuo and Lin

[1] Image adapted from http://en.wikipedia.org/wiki/File:Unintuitive-rgb.png

[8]. External factors such as illumination changes, shadows, adverse weather conditions can greatly impact the success of color-based detection techniques. This significantly reduces the potential of color thresholding as a stand-alone solution for detection. In recent research, color thresholding commonly finds is purpose as a preprocessing step to extract regions of interest [14], [15].

Influences of daily illumination changes are recognized by Benallal and Meunier [3]. They present an interesting experiment in which they observe the color of a red STOP through 24 hours. They show that the red color component is dominant between approximately 6.30 am and 9 pm. During that time, the differences δ_{RG} and δ_{RB} between the red color component and the green and blue components remain high, the red color having a value of approximately 85 above the green and the blue components. Based on this experiment, they propose formulas for color segmentation intended to correctly segment the red, green and blue signs (see table II).

Estevez and Kehtarnavaz [4] present an algorithm for detecting and recognizing a small subset of traffic signs which contain red components. The first stage of their algorithm is color segmentation, used to localize red edge areas. The formula for the segmentation, given in table II, relies on a tunable parameter α which can be tuned to varying sensitivities based on intensity levels, in order to avoid illumination sensitivity. Average intensity values are obtained by sparsely sampling the top line of the image, usually corresponding to the sky. From these values one can speculate about the weather conditions and choose the proper value of α. The exact values of α chosen are not given in the paper.

Broggi et al. [6] propose a way of overcoming the color dependency on the light source. The default way to determine the light source color is to find a white object in the scene and compute the difference between the image white and theoretical white (RGB values 255, 255, 255). In road sequences one cannot count on having a white reference point, but the road is usually gray. Broggi et al. therefore find a piece of road (it is unclear whether this is an automated procedure or it needs to be done by hand) and estimate the light source color by assuming that the road should be gray. They then perform chromatic equalization, similar to gamma correction but with the linearization of gamma function.

Ruta et al. [7] use color-based segmentation as a starting stage in traffic sign recognition. They first segment the image based on fixed thresholds (which are not listed in the paper), and then enhance the obtained colors using formulas shown in table II.

Escalera et al. [12] present an approach for detecting red in HSI color space. The input image is first converted from RGB to HSI. For each pixel, values of hue and saturation are re-calculated so that the range of saturated red hues is emphasized. This is done by using a lookup table described in table II. The authors assume that the values of hue and saturation are scaled to the range of 0 to 255. The resulting hue and saturation are then multiplied and the result is upper bounded by 255. Thus the response image is obtained. The authors state that the values are multiplied so that the two components can correct each other - if one component is wrong, the assumption is that the other one will not be wrong.

Fang et al. [11] classify colors based on their similarity with pre-stored hues. The idea is that the hues in which a traffic sign appears are stored in advance, and the color label is calculated as the similarity measure with all available hues, so that the classification which is most similar is chosen.

Paclik et al. [10] present approximative formulas for converting RGB to HSI. The desired color is then obtained by choosing an appropriate threshold for hue, while black and white are found by thresholding the saturation and intensity components.

Gao et al. [13] use the CIECAM97 color model. The images are first transformed from RGB to CIE XYZ values, and then to LCH (Lightness, Chroma, Hue) space using the CIECAM97 model. The authors state that the lightness values are similar for red and blue signs and the background, so only hue and chroma measures are used in the segmentation. Authors consider four distinct cases: average daylight viewing conditions, as well as conditions during sunny, cloudy and rainy weather. Using the acceptable ranges, sign candidates are segmented using a quad-tree approach, meaning that the image is recursively divided into quadrants until all elements are homogeneous or the predefined grain size is reached.

For an another view on traffic sign detection by color, see a review paper by Nguwi and Kouzani [16], in which the color-based detection methods are divided into seven categories.

III. Shape-based detection

Several approaches for shape-based detection of traffic signs are recurrent in literature. Probably the most common approach is using some form of Hough transform. Approaches based on corner detection followed by reasoning or approaches based on simple template matching are also popular.

Generalized Hough transform is a technique for finding arbitrary shapes in an image. The basic idea is that, using an edge image, each pixel of the edge image votes for where the object center would be if that pixel were at the object boudary. The technique originated early in the history of computer vision. It was extended and modified numerous times and there are many variants. Here we will present work by Loy and Barnes, as it was intended specifically for traffic sign detection and was used independently in several detection systems. Loy and Barnes [17] propose a general regular polygon detector and use it to detect traffic signs. The detector is based on their fast radial symmetry transform, and the overall approach is similar to Hough transform. First the gradient magnitude image is built from the original image. The gradient magnitude image is then thresholded so that the points with low magnitudes, which are unlikely to correspond to edges, are eliminated. Each remaining pixel then votes for the possible positions of the center of a regular polygon. One pixel casts its vote at multiple locations distributed along a line which is perpendicular to the gradient of the pixel and whose distance to the pixel is equal to the expected radius of the regular polygon (see figure 4). Notice that there are actually two lines which satisfy these requirements, one in the direction of the gradient and the other in the opposite direction. Both can be used if we don't know in advance whether signs will be lighter or darker than the background. The length of the voting line is bounded by the expected radius of the regular polygon. The votes towards the end of the line have negative weights, to minimize the influence of straight lines in an image which are too long to be polygon edges. The resulting vote image is labeled O_r.

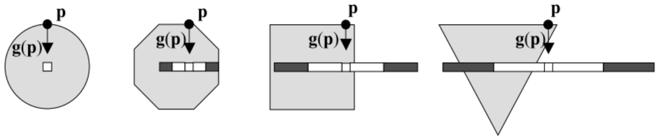


Fig. 4: Locations on which a pixel votes for the object center. The parts of the line which are black indicate negative votes. Image from [17].

In addition to the vote image, another image called equiangular image is built. The proposed procedure favors equiangular polygons by utilizing the following property: if the gradient angles of edge pixels of an n-sided regular polygon are multiplied by n, the resulting angles will be equal (see figure 5). For instance, cosider an equiangular triangle for which we sample one gradient angle value at each side. Suppose that we obtain gradient values of $73°$, $193°$ and $313°$. The gradients are spaced at $360°/n = 120°$. Then $73° \times 3 = 219°$ and $193° \times 3 = 579°$, $579° - 360° = 219°$. Similarly $313° \times 3 = 939°$, $939° - 2 \times 360° = 219°$. For each pixel which voted for the polygon center, a unit vector is constructed. The slope of the unit vector is made equal to the gradient angle of the pixel multiplied by the number of sides of the sought regular polygon. The pixel then again casts its vote on locations determined by the voting line, except that this time the vote takes the form of the constructed unit vector. The votes are cast in a new image called the equiangular image. Each point in this image represents a vector which is the sum of all contributing votes. The votes coming from edges of equiangular polygons will have the same slope, so the magnitudes of vote vectors in equiangular polygon centroids should be the largest.

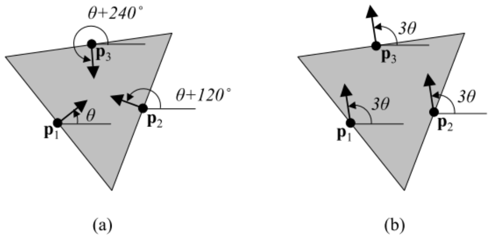


Fig. 5: Multiplying the gradient angles of a triangle by 3. The resulting angles are equal. Image from [17].

Finally, the vote image and the norm of the equiangular image are combined to produce the overall response. Computational complexity of this method is $O(Nkl)$, where l is the maximum length of the voting line, N is the number of pixels in an image and k is the number of radii being considered. The main weakness of the approach is that the radius of the sought polygon should be known in advance, which is not always easy to accomplish. This can be solved by trying out multiple radii, but it might be too expensive in terms of processing time.

Another interesting approach in finding shapes of interest is to use a corner detector and then hypothesise about the locations of regular polygons by observing the relations between corners. Paulo and Correia [18] detect triangular and rectangular signs by first applying the Harris corner detector to a region of interest, and then searching for existence of corners in six predefined control areas of the region. The shape is determined based on the configuration of the control regions in which corners are found. Control areas are shown in figure 6.

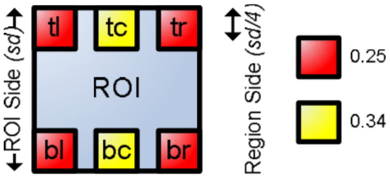


Fig. 6: Predefined control areas from [18]. The shape is determined by the existence of corners in the control areas.

Gavrila [19] uses distance transform based template matching for shape detection. First, edges in the original image are found. Second, a distance transform (DT) image is built (see figure 7). A DT image is an image in which each pixel represents the distance to the nearest edge. To find the shape of interest, the basic idea is to match a template (for instance, a regular triangle) against the DT image. In order to find the optimal match, the template is rotated, scaled and translated. One might consider attempting to match the template with the raw edge image instead, but by matching with DT image the resulting similarity measure is much smoother. In Gavrila's extension of this basic idea, the edges are differented by orientation, so that separate DT images are computed for distinct edge orientation intervals and templates are separated into parts based on orientations of their edges. The overall match measure is a sum of match measures between DT images and templates of specific orientations. Gavrila also uses a template hierarchy, with the idea that similar templates are grouped into prototypes, and, once the prototype has been found, the process finds the best template within the prototype. This saves computation costs.

IV. Detection based on machine learning

In the approaches outlined above, the prior knowledge of the problem (the expected color and shape of a traffic sign) is manually encoded into the solution. However, this knowledge could also be discovered using machine learning.

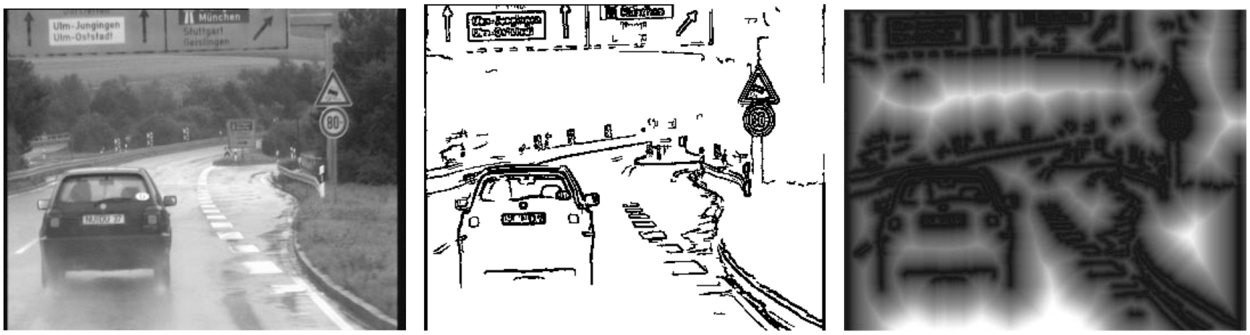


Fig. 7: Building the distance transform image. From left to right: the original image, the edge image and the distance transform image. The template for which the DT image is searched is a simple triangle. Images from [19].

Research by Viola and Jones [20] presented a significant milestone in computer vision. Viola and Jones developed an algorithm capable of detecting objects very reliably and in real time. The detector is trained using a set of positive and negative examples. While originaly intended for face detection, various other researchers have succesfully applied the detector to a lot of other object classes. Among others, traffic signs were successfully detected.

The detector of Viola and Jones is an attentional cascade of boosted Haar-like classifiers. It combines two concepts: (i) AdaBoost and (ii) Haar-like classifiers. Haar-like classifiers are built using simple rectangular features which represent differences of sums of specific pixels in an image. Each feature is paired with a threshold, and the decision of the so-built classifier is determined by comparing the value of the feature with the threshold. Four feature types used in the original paper are shown in figure 8. Viola and Jones propose a very fast method of computation for such features which utilizes the so-called integral image. The value of each feature can be computed in less then ten array references.

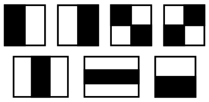


Fig. 8: Haar-like features used in training the detector.

AdaBoost is a technique for combining a number of weak classifiers into a strong one. It has been proven to converge to the optimal solution with a sufficient number of weak classifiers. AdaBoost assigns weights to weak classifiers based on their quality, and the resulting strong classifier is a linear combination of weak classifiers with the appropriate weights.

Viola and Jones group multiple strong classifiers constructed by AdaBoost into an attentional cascade, which enables faster processing. The strong classifier in the first stage of the cascade is chosen so that it discards a number of false positives, while preserving almost all true positives of the training set. For example, the OpenCV implementation defaults to the minimum hit rate of 0.995 and maximum false positive rate of 0.5 per cascade stage. Subsequent stages of the cascade follow the same numerical limitations. Each stage is trained so that the false positives of the previous stage are labeled as negatives and added to the training set. Hence, subsequent stages are trained to correct the errors of previous stages, while preserving the high true positive rates. Using the cascade enables faster processing, as obvious false positives are discarded early on.

The process of the detection is carried out by sliding a detection window across the image. Within the window, the response of the cascade is calculated. After completing one pass over the image, the size of the detection window is increased by some factor (OpenCV defaults to 1.2, meaning that the scale of the window will be increased by 20%). The window size is increased until some predefined size is reached. Increasing the detection window by a smaller percentage yields better detection rates, but increases the total processing time.

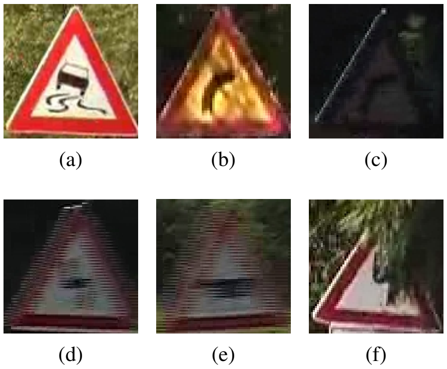


Fig. 9: Examples of sign images used for training the Viola-Jones detector in [21]: (a) a normal sign, (b) shadow, (c) color inconsistency, (d) interlacing, (e) motion blur and interlacing, (f) occlusion.

In our work [21] we experimented with using the Viola-Jones detector for triangular traffic sign detection. The detector was trained using about 1000 images of relatively poor quality. Some especially problematic images are shown in figure 9. The obtained detector achieved a very high true positive rate (ranging from 90% to 96%, depending on the training set and the configuration of the detector). A part of the trained cascade is shown in figure 10. The first few Haar-like features superimposed on traffic sign images are shown in figure 11. Notice how the Haar-like features capture some natural properties of the traffic sign, such as the bottom edge, almost perfectly. In our experiments [21], [22], we obseved two main weaknesses of the Viola-Jones detector: (i) the requirement of a large number of training images and (ii) high false positive rates. Nevertheless, our research also indicates that the Viola-Jones detector is robust w.r.t. noisy and low-quality training data (cf. figure 9, where six real training images are shown).

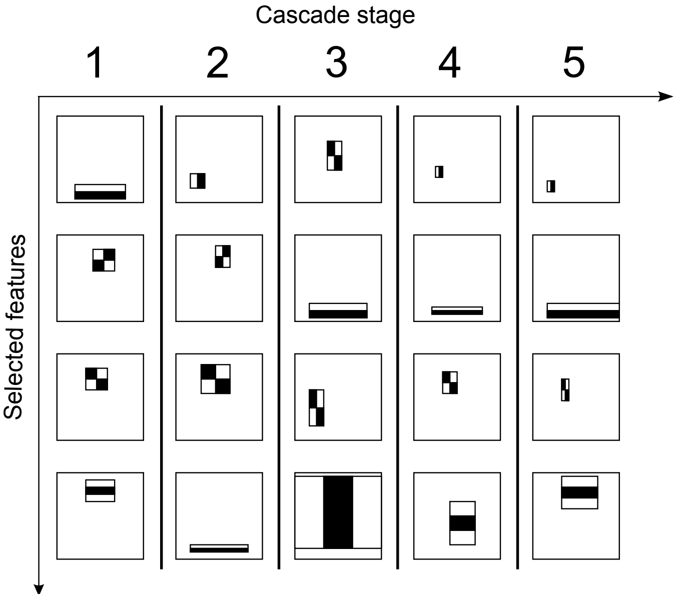


Fig. 10: Top four features of the first five stages of the Viola-Jones detector cascade in [21]. The stages are distributed on the horizontal axis, while the first four features used by each stage classifier are shown on the vertical axis.

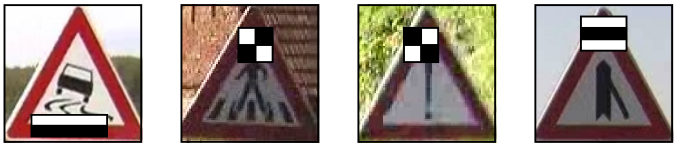


Fig. 11: The first four features of cascade stage 1 superimposed on real traffic sign images. [21]

Several researchers used the Viola-Jones detector for traffic sign detection. Chen and Sieh [23] use the Viola-Jones detector to detect circular signs. Baro and Vitria [24] detect signs using the method of Viola and Jones augmented by a feature set proposed by Lienhart and Maydt [25]. They further eliminate false positives by using fast radial symmetry transform and principal component analysis. Their results are used by Escalera and Radeva [26]. Timofte et al. [15] use the Viola-Jones detector to detect six different sign classes.

In their work on traffic sign detection, Bahlmann et al. proposed extending each Haar-like feature with a parameter representing in which color channel (or a combination of channels) the value of the feature should be calculated. The features are then boosted as in the original algorithm of Viola and Jones. In selecting the best feature, the boosting algorithm automatically selects the best color channel too, eliminating the need for manual thresholds. The experiments indicated that grayscale and color-aware features result in similar false negative rates – 1.4 % and 1.6 %, respectively. The main advantage of color-aware features is the reduction of false positive rate, which is 0.03 %, ten times less than the recorded false positive rate of 0.3 % for grayscale features.

V. Subsequent stages: tracking and recognition

Real-life traffic sign detection systems generally also include a recognition and tracking step (if processing videos). For recognition researchers usually apply standard machine learning techniques, such as neural networks, support vector machines, LDA and similar. In tracking, there are two dominant approaches: KLT tracker and particle filters.

Detailed study of recognition and tracking techniques is outside the scope of this paper. Recognition, tracking and detection are interwoven - tracking helps verify both detection and recognition hypothesis (as more samples are obtained as the time passes), while recognition reduces false positives in detection. If a sign candidate cannot be classified, then it was probably not a sign at all.

VI. Case studies

The detectors outlined in the previous sections (color-based, shape-based, learning-based) are used as building blocks of contemporary systems for traffic sign detection. In this section we will review two such systems, one by Ruta et al. [14] and the other by Timofte et al. [15].

Ruta et al. [14] present a system for detection and recognition of a large number of different traffic sign classes. They address the necessity for large training sets inherent in machine learning approaches by using artificial template images of traffic signs. The signs are detected in several steps. First the image is thresholded and colors of interest are enhanced (for details, see section II of this paper). Shapes of interest are then found using the regular polygon detector similar to the one described by Loy et al. [17]. The extracted sign candidates are tracked using Kalman filter. The classification step uses template matching based on distance transforms (DTs). Unlike the approach described by Gavrila [19], where DT was calculated for binarized sign image (section III), here DTs are computed for each color of the template. Furthermore, the candidate shape is not directly matched to the DT image. Ruta et al. propose a method for selection of regions of interest in a template, i.e. the regions which most discriminate the target sign class from all other known classes. Such regions are found by maximizing the sum of dissimilarities between the template of the target sign and all other available templates. In order to enhance the result, the evidence for a sign candidate is accumulated over time. Each observation is weighted, so that the observations belonging to a longer chain are more relevant. The idea behind this approach is that when a traffic sign is first detected it is too small for the classification to work reliably, so later appearances of the sign are more valuable. Experimental results indicate rates of correctly detected and recognized traffic signs ranging from 79.4 % for circular signs with red rim to 97.3 % for blue circular signs.

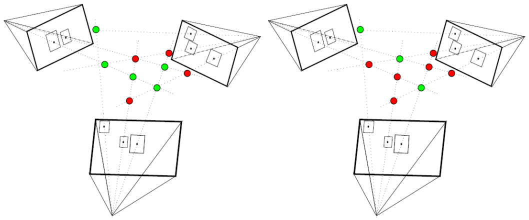


Fig. 12: Pruning the 3D hypotheses using the MDL principle. The shorter set of hypotheses (right) is preferred over the longer one (left). Image from [15].

Timofte et al. [15] present a complex system for traffic sign detection and recognition. For the data acquisition they use a van with eight high-resolution cameras, two on the each side of the van. The detection is carried out in two phases: (i) single-view phase and (ii) multi-view phase. In the single-view phase, the image is first thresholded to find the colors of interest. Also, a tranformation similar to generalized Hough transform is used for finding the shapes of interest. This step is very fast, and yields very few false negatives. To verify that the extracted candidates are traffic signs, the Viola-Jones detector is ran on the obtained bounding boxes. For additional verification they employ an SVM classifier which operates on normalized RGB channels, pyramids of HOGs and discriminative Haar-like features selected by AdaBoost. To recognize the class of the traffic sign, six one-vs-all SVMs are used, each corresponding to one class of traffic signs (triangular, triangular upside-down, circular with red rim, circular blue, rectangular, diamond-shaped). In the multi-view phase, first the data collected during the single-view phase is integrated into hypotheses. Every pair of detections taken from different views is considered. The pair is checked for geometrical consistency (the position of the hypothesis is backprojected to 2D and checked against the image candidates) and visual consistency (pairs of detections with the same basic shape are favored). Next, the set of all hypotheses is pruned using the minimum description length (MDL) principle. The idea is to find the smallest possible set of 3D hypotheses which matches the known camera positions and calibrations and is supported by detection evidence. For an illustration, see figure 12. In the end, the set of 2D observations forming a 3D hypothesis is classified by an SVM classifier. The majority of votes determines the final type assigned to the hypothesis, i.e. the exact type of the traffic sign.

VII. Conclusion

In this paper, we have presented traffic sign detection methods which are often used as building blocks of complex detection systems. The methods were divided into color-based, shape-based and learning-based. We have shown how the outlined methods are used in two state-of-the-art traffic sign detection systems.

We think that the complexity of traffic sign detection systems will diminish in the future, as technology advances. With the advancement of technology, high quality sensors will become cheaper and more available in mass production. If in the future every car is equipped with a high resolution color camera, a GPS receiver and an odometer, an infrared camera and other sensors, the problem of traffic sign detection will be infinitely more simple then it is now. However, the advancement will probably proceed slowly, because of the persistent need to minimize production costs.

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results,” http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[2] Inland transport comitee, *Convention on road signs and signals*. Economic comission for Europe, 1968.

[3] M. Benallal and J. Meunier, “Real-time color segmentation of road signs,” *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 3, pp. 1823–1826 vol.3, May 2003.

[4] L. Estevez and N. Kehtarnavaz, “A real-time histographic approach to road sign recognition,” *Image Analysis and Interpretation, 1996., Proceedings of the IEEE Southwest Symposium on*, pp. 95–100, Apr 1996.

[5] S. Varun, S. Singh, R. S. Kunte, R. D. S. Samuel, and B. Philip, “A road traffic signal recognition system based on template matching employing tree classifier,” in *ICCIMA ’07: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 360–365.

[6] A. Broggi, P. Cerri, P. Medici, P. Porta, and G. Ghisio, “Real time road signs recognition,” *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 981–986, June 2007.

[7] A. Ruta, Y. Li, and X. Liu, “Detection, tracking and recognition of traffic signs from video input,” Oct. 2008, pp. 55 –60.

[8] W.-J. Kuo and C.-C. Lin, “Two-stage road sign detection and recognition,” *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 1427–1430, July 2007.

[9] G. Piccioli, E. D. Micheli, P. Parodi, and M. Campani, “Robust method for road sign detection and recognition,” *Image and Vision Computing*, vol. 14, no. 3, pp. 209 – 223, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/B6V09-3VVCMCX-4/2/0f2793e7828195ecb68735a80a9ef904

[10] P. Paclík, J. Novovičová, P. Pudil, and P. Somol, “Road sign classification using laplace kernel classifier,” *Pattern Recogn. Lett.*, vol. 21, no. 13-14, pp. 1165–1173, 2000.

[11] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, “Road-sign detection and tracking,” vol. 52, no. 5, pp. 1329–1341, Sep. 2003.

[12] A. D. L. Escalera, J. M. A. Armingol, and M. Mata, “Traffic sign recognition and analysis for intelligent vehicles,” *Image and Vision Computing*, vol. 21, pp. 247–258, 2003.

[13] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, “Recognition of traffic signs based on their colour and shape features extracted using human vision models,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 675–685, 2006.

[14] A. Ruta, Y. Li, and X. Liu, “Real-time traffic sign recognition from video by class-specific discriminative features,” vol. 43, no. 1, pp. 416–430, 2010.

[15] R. Timofte, K. Zimmermann, and L. Van Gool, “Multi-view traffic sign detection, recognition, and 3d localisation,” Snowbird, Utah, 2009, pp. 69–76.

[16] Y.-Y. Nguwi and A. Z. Kouzani, “Detection and classification of road signs in natural environments,” *Neural Comput. Appl.*, vol. 17, no. 3, pp. 265–289, 2008.

[17] G. Loy, "Fast shape-based road sign detection for a driver assistance system," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 2004, pp. 70–75.

[18] C. Paulo and P. Correia, "Automatic detection and classification of traffic signs," in *Image Analysis for Multimedia Interactive Services, 2007. WIAMIS '07. Eighth International Workshop on*, June 2007.

[19] D. Gavrila, "Traffic sign recognition revisited," in *DAGM-Symposium*, 1999, pp. 86–93.

[20] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[21] K. Brkić, A. Pinz, and S. Šegvić, "Traffic sign detection as a component of an automated traffic infrastructure inventory system," Stainz, Austria, May 2009.

[22] K. Brkić, S. Šegvić, Z. Kalafatić, I. Sikirić, and A. Pinz, "Generative modeling of spatio-temporal traffic sign trajectories," held in conjuction with CVPR2010, San Francisco, California, Jun. 2010.

[23] S.-Y. Chen and J.-W. Hsieh, "Boosted road sign detection and recognition," *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 7, pp. 3823–3826, July 2008.

[24] X. Baro and J. Vitria, "Fast traffic sign detection on greyscale images," *Recent Advances in Artificial Intelligence Research and Development*, pp. 69–76, October 2004.

[25] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *IEEE ICIP 2002*, 2002, pp. 900–903.

[26] S. Escalera and P. Radeva, "Fast greyscale road sign model matching and recognition," *Recent Advances in Artificial Intelligence Research and Development*, pp. 69–76, 2004.