Traffic Sign Recognition

[1,] Mr. Chetan J. Shelke, [2,]Dr. Pravin Karde
[1,]Department Of Information Technology P. R. Patil College of Engg. & Tech
[2,]Department Of Information Technology.
Govt Polytechnic Amravati

Abstract

Traffic sign recognition is a difficult task if aim is at detecting and recognizing signs in images captured from unfavorable environments. Complex background, weather, shadow, and other lighting-related problems may make it difficult to detect and recognize signs in the rural as well as the urban areas. Two major problems exist in the whole detection process. Road signs are frequently occluded partially by other vehicles. Many objects are present in traffic scenes which make the sign detection hard (pedestrians, other vehicles, buildings and billboards may confuse the detection system by patterns similar to that of road signs). Color information from traffic scene images is affected by varying illumination caused by weather conditions, time (day night) and shadowing (buildings)

1. Introduction

Traffic sign recognition is important for driver assistant systems, automatic vehicles, and inventory purposes. The best algorithm will be the one that yields the best global results throughout the whole recognition process, which comprises three stages: 1) segmentation; 2) detection; and 3) recognition. Researchers have developed vision-based techniques for traffic monitoring, traffic-related parameter estimation, driver monitoring, and intelligent vehicles, etc. [1]. Traffic sign recognition (TSR) is an important basic function of intelligent vehicles [2], and TSR problems have attracted attention of many research groups since more than ten years ago. Traffic sign recognition is part of the general case of Pattern Recognition. Major problem in pattern recognition is the difficulty of constructing characteristic patterns (templates). This is because of the large variety of the features being searched in the images, such as people faces, cars, etc. On the contrary, traffic signs a) are made with vivid and specific colors so as to attract the driver's attention and to be distinguished from the environment b) are of specific geometrical shapes (triangle, rectangle, circle - ellipse) and c) for each sign there is a specific template. It is therefore rather easy to develop an algorithm in such a way that the computer has "a priori knowledge" of the objects being searched in the image.

2. Related Work

Traffic signs are normally classified according to their color and shape and should be designed and positioned in such a way that they can easily be noticed while driving. Inventory systems must take advantage of these characteristics. However, various questions need to be taken into account in traffic sign-recognition system. For example, the object's appearance in an image depends on several aspects, such as outdoor lighting condition. In addition, deterioration of a traffic sign due to aging or vandalism affects its appearance, whereas the type of sheeting material used to make traffic signs may also cause variations. These problems particularly affect the segmentation step [3], which is usually the first stage in high-level detection and recognition systems. Segmentation can be carried out using color information or structural information. Many segmentation methods have been reported in the literature since the advent of digital image processing. Detection and recognition are two major steps for determining types of traffic signs [4]. Detection refers to the task of locating the traffic signs in given images. It is common to call the region in a given image that potentially contains the image of a traffic sign the region of interests (ROI). Taking advantages of the special characteristics of traffic signs, TSR systems typically rely on the color and geometric information in the images to detect the ROIs. Hence, color segmentation is common to most TSR systems, so are edge detection [5] and corner detection techniques [6]. After identifying the ROIs, we extract features of the ROIs, and classify the ROIs using the extracted feature values. Researchers have explored several techniques for classifying the ideographs, including artificial neural networks (ANNs) [7], template matching [8], chain code [9], and matching pursuit methods [10]. Detection and recognition of traffic signs become very challenging in a noisy environment. Traffic signs may be physically rotated or damaged for different reasons. View angles from the car-mounted cameras to the traffic signs may lead to artificially rotated and distorted

images. External objects, such as tree leaves, may occlude the traffic signs, and background conditions may make it difficult to detec ..0traffic signs. Bad weather conditions may have a detrimental effect on the quality of the images.
The traffic sign-recognition system which was described in detail in [11] consists of four stages as shown in Figure 1.

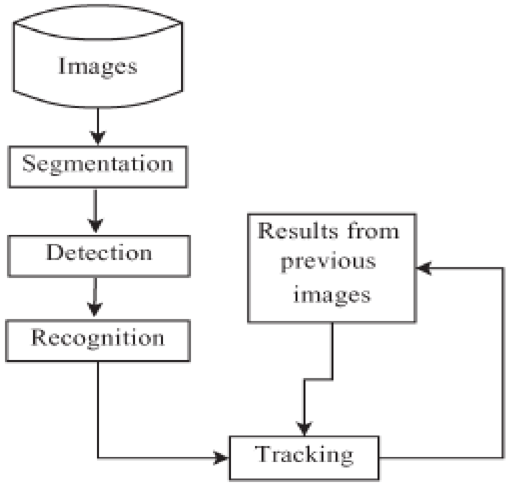

Fig. 1: Traffic sign recognition system.

Segmentation: This stage extracts objects from the background, which are, in this case, traffic signs using color information.

Detection: Here, potential traffic signs are located through shape classification.
Recognition: Traffic sign identification is effected using SVMs.
Tracking: This stage grouped multiple recognitions of the same traffic sign.

3. Region Of Interest (Roi) Detection

Transportation engineers design traffic signs such that people can recognize them easily by using distinct colors and shapes for the signs. Many countries use triangles and circles for signs that carry warning and forbidding messages, respectively. These signs have thick and red borders for visibility from apart. Hence, we may use color and shape information for detecting traffic signs.

4. Color Segmentation

Identifying what pixels of the images are red is a special instance of the *color segmentation* problems. This task is not easy because images captured by cameras are affected by a variety of factors, and the "red" pixels as perceived by human may not be encoded by the same pixel values all the time.Assuming no directly blocking objects, lighting conditions affect the quality of the color information the most. Weather conditions certainly are the most influential factor. Nearby buildings or objects, such as trees, may also affect quality of the color information because of their shadows. It is easy to obtain very dark images, e.g., the middle image in Figure 2, when we are driving in the direction of the sun.

Fig. 2: Selected "hard" traffic signs. The left sign did not face the camera directly, and had a red background. The middle picture was taken in the dusk. The signs in the rightmost image were in the shadow.

As a consequence, "red" pixels can be embodied in a range of values. Hence, it is attempted to define the range for the red color. We can convert the original image to a new image using a pre-selected formula. Let *Ri*, *Gi*, and *Bi* be the red, green, and blue component of a given pixel in the original image. We encode the pixels of the new image by *Ro*, *Go*, and *Bo*. Based on results of a few experiments, we found that the following conversion most effective: $Ro = \max(0, (Ri - Gi) +$

($Ri - Bi$)), $Go = 0$, and $Bo = 0$. After the color segmentation step, only pixels whose original red components dominate the other two components can have a nonzero red component in the new image most of the time.

5. Region Of Interests

Then the red pixels are grouped into separate objects, apply the *Laplacian of Gaussian* (LoG) edge detector to this new image, and use the 8-connected neighborhood principle for determining what pixels constitute a connected object. We consider any red pixels that are among the 8 immediate neighbors of another red pixel *connected*. After grouping the red pixels, we screen the object based on four features to determine what objects may contain traffic signs. These features are areas, height to width ratios, positions, and detected corners of the objects. According to the government's decrees for traffic sign designs, all traffic signs must have standard sizes. Using camera, which is set at a selected resolution, to take pictures of warning signs from 100 meter apart, the captured image will occupy 5x4 pixels. Due to this observation, we ignore objects that contain less than 40 red pixels. We choose to use this threshold because it provided a good balance between recall and precision when we applied the *Detection* procedure to the training data. Two other reasons support our ignoring these small objects. Even if the discarded objects were traffic signs, it would be very difficult to recognize them correctly. Moreover, if they are really traffic signs that are important to our journey, they would get closer and become bigger, and will be detected shortly. The decrees also allow us to use shapes of the bounding boxes of the objects to filter the objects. Traffic signs have specific shapes, so heights and widths of their bounding boxes must also have special ratios. The ratios may be distorted due to such reasons as damaged signs and viewing angles. Nevertheless, we can still use an interval of ratios for determining whether objects contain traffic signs. Positions of the objects in the captured images play a similar role as the decrees. Except driving on rolling hills, we normally see traffic signs above a certain horizon. Due to this physical constraint and the fact that there are no rolling hills in Taiwan, we assume that images of traffic signs must appear in a certain area in the captured image, and use this constraint for filtering objects in images. We divide the bounding boxes of the objects into nine equal regions, and check whether we can detect corners in selected regions. The leftmost image in Figure 3 illustrates one of these patterns by the blue checks. More patterns are specified in the following *Detection* procedure. If none of the patterns is satisfied, chances are very low that the object could contain a triangular sign. Using this principle, system detected the rightmost four signs in Figure 3.

Fig. 3: Using Corners for identifying triangular borders.

Procedure *Detection* (Input: an image of 640x480 pixels; Output: an ROI)

Steps:

1 Color segmentation
2 Detect edges with the LoG edge detector.
3 Remove objects with less than 40 red pixels.
4 Mark the bounding boxes of the objects.
5 Remove objects whose highest red pixel locates below row 310 of the original images, setting the origin (0, 0) of the coordinate system to the upper-left corner.
6 Remove objects with height/width ratios not in the range [0.7, 1.3].
7 Check existence of the corners of each object.
a. Find the red pixel with the smallest row number. When there are many such pixels, choose the pixel with the smallest column number.
b. Find the red pixels with the smallest and the largest column numbers. If there are multiple choices, choose those with the largest row numbers.
c. Mark locations of these three pixels in the imaginary nine equal regions, setting their corresponding containing regions by 1.
d. Remove the object if these pixels do not form any of the patterns listed aside.
8. For each surviving bounding box, extract the corresponding rectangular area from the original image and save it into the ROI list.

Figure 4 illustrates how we detect a triangular sign with the *Detection* procedure. Notice that the sing in (f) is not exactly upright. The tree trunks and red sign behind the sign made our algorithm unable to extract the complete red border. All objects detected by *Detection* are very likely to contain a triangular traffic sign. They will be used as input to the recognition component after the preprocessing step.

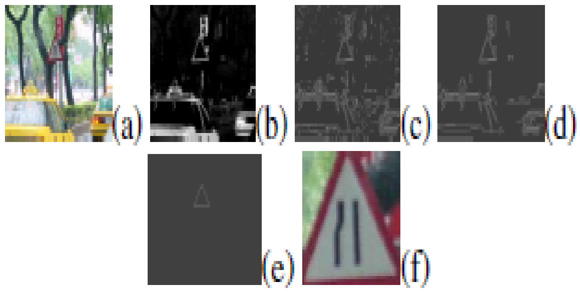

Fig. 4: An illustration of detection steps: (a) the original image; (b) result of color segmentation; (c) result of edge detection; (d) result of removing small objects; (e) results of filtering objects by step 7; (f) the enlarged image of the detected sign.

6. Preprocessing

Procedure *Preprocessing* (Input: an ROI object list; Output: an object list)

Steps:

For each object in the ROI list, do the following:

1. Normalize the object to the standard size 80x70.s
2. Extract the rectangle of 32x30 pixels from (25, 30).
3. Remove remaining red pixels.
4. Convert the object to a gray-level image.

As the first step of the preprocessing, we normalize all objects to the 80x70 standard size. After a simple analysis of the 45 standard triangular signs, we found that the ideographs appear in a specific region in the normalized images. As shown in Figure 5(a), we can extract the ideographs from a particular rectangular area in the image. We extract the ideograph from a pre-selected area of 32x30 pixels from the normalized image. The coordinates of the upper left corner of the extracted rectangle is (25, 30). Notice that, although we have attempted to choose the rectangular area such that it may accommodate distorted and rotated signs, the extracted image may not include all the original ideographs all the time. Figure 5(b) shows that the bottom of the ideograph was truncated. Similarly, the extracted area may contain noisy information. After extracting the rectangular area that might contain the ideograph, we remove red pixels in the extract. We use a more stringent standard for defining "red." Let R, G, and B be the red, green, and blue component of a pixel. A pixel is red if R>20, (RB)> 20, and (R-G)>20. After removing the red pixels, we convert the result into a gray-level image. We adjust pixel values based on the average luminance to increase contrast of the image.We compute the YIQ values of each pixel from its RGB values, set their gray levels to their luminance values, and compute the average gray levels of all pixels. Let the average be α. We invert the colors of the pixels by deducting the amount of ($\alpha-100$) from the gray levels of all pixels. Then, pixels whose remaining gray levels are smaller than 70 are set to 0, and others are set to 255. However, if using 70 as the threshold gives us less than 10 pixels with value 255 or 10 pixels with value 0, we apply another slightly more complex method. We calculate the average gray level of the pixel values, and use this average, λ, as the cutting point for assigning pixel values in the gray-level image. Pixels whose gray levels are less than λ are set to 0, and others are set to 255. Figure 4(c) shows such a gray-level image.

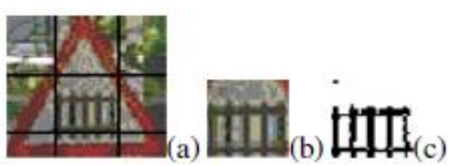

Fig. 5: Preprocessing Steps

7. TRAFFIC SIGN RECOGNITION

After the preprocessing procedure, each object becomes a rectangle of 32x30 pixels. We can use these raw data as features for recognition. In addition, we employ the discrete cosine transform (DCT) and the singular value decomposition (SVD) procedures for extracting the invariant features of the ideographs. DCT is one of the popular methods for decomposing a signal to a sequence of components and for coding images. We concatenate rows of a given object, generated at step 5 in *Preprocessing*, into a chain, and apply the one-dimension DCT over the chain, and use the first 105 coefficients as the feature values. We apply singular value decomposition to the matrices of the objects that are obtained at step 4 in the *Preprocessing* procedure for extracting features of the objects. Let UΣVT be the singular value decomposition of the matrix that encodes a given object. We employ the diagonal values in Σ as the feature values of the given object. Since the original matrix is 32x30, we obtain 30 feature values from Σ.

8. Conclusion and Future Directions

Implementation of the algorithm in test images showed that it is very effective in the sign location phase. There is a slight weakness in the some phase, in cases of color similarity between signs and other areas of the image. It is sensitive in light condition changes during the image acquisition, because of the effect they have in the color thresholds used in the regions of interest segmentation step. The use of proper thresholds is very important as it affects in a great deal the success of the sign detection and it's final recognition. Based in the experience acquired from the tests, the aspects which should be further researched and be improved in the future are:

1. Recognition of signs of more complex shape.
2. Recognition of two (or more) signs in the same region of interest.
3. Increase of the speed of the algorithm by improving the source code and again, by possible changes in its structure.
4. Increase of the robustness of the algorithm in light condition changes.
5. Merging of the rectangle and triangle-ellipse detection process.

REFERENCES

[1] Hilario Gómez-Moreno, *Member, IEEE*, Saturnino Maldonado-Bascón, *Senior Member, IEEE*, Pedro Gil-Jiménez, and Sergio Lafuente-Arroyo, "Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition", *IEEE Trans. On Intelligent transportation system Apr. 2009*.

[2] Hsiu-Ming Yang, Chao-Lin Liu, Kun-Hao Liu, and Shang-Ming Huang, "Traffic Sign Recognition in Disturbing Environments", *N. Zhong et al. (Eds.): ISMIS 2003, LNAI 2871, pp. 252–261, 2003*.

[3] Matthias M¨uller, Axel Braun, Joachim Gerlach, Wolfgang Rosenstiel, Dennis Nienh¨user, J. Marius Z¨ollner, Oliver Bringmann, "Design of an Automotive Traffic Sign Recognition System Targeting a Multi-Core SoC Implementation", *978-3-9810801-6-2/DATE10 © 2010 EDAA*.

[4] N. Barnes, A. Zelinsky, and L. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 322–332, Jun. 2008.

[5] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno, and F. J. Acevedo-Rodriguez, "Traffic sign recognition system for inventory purposes," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 4–6, 2008, pp. 590–595.

[6] C. Nunn, A. Kummert, and S. Muller-Schneiders, "A novel region of interest selection approach for traffic sign recognition based on 3D modelling," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 4–6, 2008, pp. 654–659.

[7] C. F. Paulo and P. L. Correia, "Traffic sign recognition based on pictogram contours," in *Proc. 9th WIAMIS*, May 7–9, 2008, pp. 67–70.

[8] B. Cyganek, "Road signs recognition by the scale-space template matching in the log-polar domain," in *Proc. 3rd Iberian Conf. Pattern Recog. Image Anal.*, vol. 4477, *Lecture Notes in Computer Science*, 2007, pp. 330–337.

[9] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Beijing, China, Jul. 2007, pp. 1427–1430.

[10] G. K. Siogkas and E. S. Dermatas, "Detection, tracking and classification of road signs in adverse conditions," in *Proc. IEEE MELECON*, Málaga, Spain, 2006, pp. 537–540.

[11] P. Gil-Jiménez, S. Maldonado-Bascón, H. Gómez-Moreno, S. Lafuente- Arroyo, and F. López-Ferreras, "Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies," *Signal Process., vol. 88, no. 12, pp. 2943–2955, Dec. 2008.*

AUTHORS PROFILE

.

Chetan J Shelke is Asst Professor in P.R.Patil College of Engg.He did his M.E (information technology) in 2011 at Prof. Ram Meghe Institute Of Technology & Research, Badnera.He did his B.E(Computer Science & Engg) from H.V.P.M College of Engg & Tech Amravati in 2007.