PAPER • OPEN ACCESS

Traffic sign detection using histogram of oriented gradients and max margin object detection

To cite this article: Reinaldo *et al* 2019 *J. Phys.: Conf. Ser.* **1230** 012098

View the article online for updates and enhancements.

Traffic sign detection using histogram of oriented gradients and max margin object detection

Reinaldo, Natanael Manurung[*], Juara immanuel Simbolon, Christnatalis

Informatics Engineering Study Program, Deparment of Technology and Computer Science, Universitas Prima Indonesia, Medan, Indonesia

*naelmanurng@gmail.com

Abstract. Traffic signs are important markers in two-wheeled and four-wheeled vehicles. However, there is a change in direction or arrangement on the road that cannot be opened on a map which can cause incorrect information, which can cause traffic jams. In this journal the author uses a camera mounted on a car that provides a solution for drivers who issue problems that occur on the road that show directions or arrangements that are not directly updated using the HOG and MMOD methods. HOGs and MMODs are methods that can refute objects well, and move images and will be recognized immediately. The information received can be sent directly to an electronic map so that it can be accessed automatically by the driver's information and assistance and other information finds the right path, so that it can help the driver and can avoid traffic jams.

1.   Introduction

Traffic signs are an important component of driving. By utilizing and knowing the meaning of traffic signs, drivers can reduce the risk of accidents. Sometimes when there is a change in direction of traffic, a new traffic sign will be installed around the road if there is a change in direction of traffic so that road users can find out the changes that have been applied to avoid accidents.

Histogram of Oriented Gradients (HOG) is a common method used for facial recognition problems. The basic idea of this method is how to divide images in several areas and calculate gradients in each area. HOG was introduced in 1986. Since then many researchers have used this method and developed it for better performance and function. HOG with a sliding-window object detector proposed by Dalal and Triggs which divides the detection window into cells that have their own gradient orientation histogram.

Based on the information provided by the HOG we can combine it with the training method, so that it can be trained and it will recognize the object every time it appears in the image

Sharma's research uses HOG combined with neural networks to classify a person's age. HOG is used to detect wrinkles and expressions on a person's face, information obtained will be processed with Principal Component Analysis (PCA) and Support Vector Machine (SVM) to estimate the age of the face. [1] [2] [3] [4] [5] While research conducted by Ali and friends used HOG to detect mangoes that were still in the tree. Using a quadcopter, taking photos of mangoes is carried out and continued with naming objects using Matlab Labeler. The photos used for the training amounted to 600 photos, this is a fairly large number, so the idea arises whether this number can be reduced and the accuracy of the resulting recognition remains high. Experiments on mango encountered problems because sometimes mangoes grew inside and photos taken were blocked by leaves which resulted in a lack of accuracy in the introduction of mango objects. Besides using image recognition found in Matlab, this method cannot be applied in real time. [6] [7] [8] [9] [10]

In this study we used Max Margin Object Detection (MMOD), which is the development of a linear Support Vector Machine. The basic idea is to get HOG information from the area in the image that we want to identify and then we use MMOD to train it, after that we can try to test it with other images containing or not containing the object that has been studied. This application can also visualize the HOG detector for that object.

2.   Methods

2.1.   Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) Histogram of Oriented Gradients (HOG) is one feature extraction method that is used as image processing to detect an object. HOG comes from an assumption that states that an object can be explained by form. To obtain different information, the image will be divided into cells and each cell will be counted as a histogram of oriented gradients. Each pixel in a cell contributes when weighting is done to build a histogram oriented to the calculated gradient values.

The initial process for building a HOG is to calculate horizontal and vertical gradient values. Here are the calculation formulas:

$$dx = I(x + 1, y) - I(x - 1, y)$$

$$dy = I(x, y + 1) - I(x, y - 1)$$

Information:

dx = vertical image gradient value

dy = horizontal image gradient value

$I(x, y)$ = pixel value in row x and column y

The first HOG feature was proposed to help humans detect objects and then become a very popular feature for detecting objects. When extracting the HOG feature, the gradient orientation is usually quantized into a garbage histogram and each garbage has an orientation distance. Images are divided into overlapping blocks and in each block, histograms of oriented gradients fall into each bin are calculated and then normalized to overcome variations in illumination. The features of all blocks and then are combined together to form a feature description of the entire image. Figure 1 illustrates the classic HOG feature extraction process. Because of its toughness to variations in illumination and invariance to local geometric and photometric transformations, many scene text recognition works use HOGs to recognize text in scenes. Besides that, the HOG captures the orientation of only isolated pixels, while the spatial information of pixels is also ignored. [8]

(a) Sample Image   (b) Gradient Orientation

(c) Histogram and Vectorization
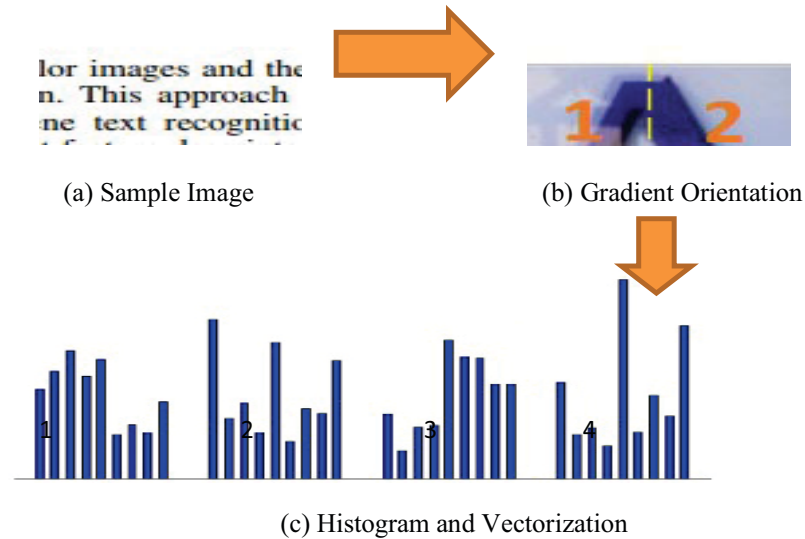
Figure 1. llustration of HOG feature extraction: (a) shows a character sample which is divided into 4 blocks (the blocks overlay with neighboring blocks in implementation). (b) shows the corresponding gradient orientation of each block. (c) shows the histogram of gradient orientation and concatenated one after another to form a HOG feature vector.

The first step in calculating the HOG is the calculation of the gradient value. The most common method is to apply 1-D centered, discrete derivative mask in one or both horizontal and vertical directions. Specifically, this method requires color filtering or image intensity data with the following kernel filter : [2]

$$D_x = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

$$D_x = \left| \begin{matrix} \\ _- \end{matrix} \right.$$

Next, computing the x and y derivatives using a convolution operation with:

$$I_x = I \times$$

$$I_y = I \times$$

The magnitude of the gradient:

$$|G| = \sqrt{I_x^2 + }$$

The orientation of the gradient:

$$\theta = \tan^-$$

Each pixel within cells provided directional vote for oriented based histogram channel based on the values found in the gradient computation. The cells can be rectangular or radial in shape with histogram channels points in 0 to 180 degrees or 0 to 360 degrees, depending on the gradient is signed or unsigned.

The strength of the gradient must be normalized locally. Cells must be grouped together in larger connected blocks. Therefore, HOG descriptors are the main vector component. Overlapping blocks determine the final descriptor. The descriptor block is represented by a rectangular type and a circular type. The last way is to normalize the block and execute the size of similarity between the leading vectors. [2]

The HOG outperformed almost all other features because of its security against variations in illumination and invariance to local geometric and photometric transformations. Another important reason is that HOG has the ability to encode and match strong radiographs in character. In fact, the HOG-based approach obtained the best performance in both the ICDAR2003 and SVT datasets when combined with deep neural networks as trained with a large amount of training data (up to 40 million). [7]

Below is an example of a 1D histogram whose circuit provides a feature vector. Let L be the intensity (grayscale) function that describes the image to be analyzed. The image is divided into cells of size N × N pixels (as in Fig. 3a) and Orientation θx, y of the gradient in each pixel is calculated (Figure 3b, c) with the following conditions: [9]

$$\theta_{x,y} = \tan^{-1} \frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}$$

Successively, the orientations $\theta j i$ $i = 1 \dots N2$, i.e. belonging to the same cell j are quantized and accumulated into a M-bins histogram (Figure. 3d, e). where the results finally get the picture below this.
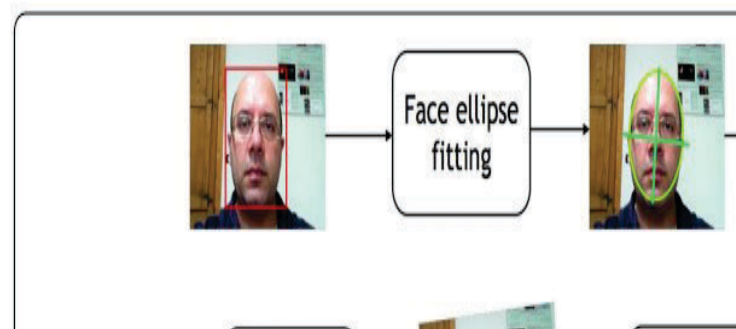
Figure 2. Detect face images

Face registration: the detected face is fitted in an ellipse used to rotate the face in a perfectly vertical position; successively eyes are detected and used to scale the image and to cro€p the area of interest
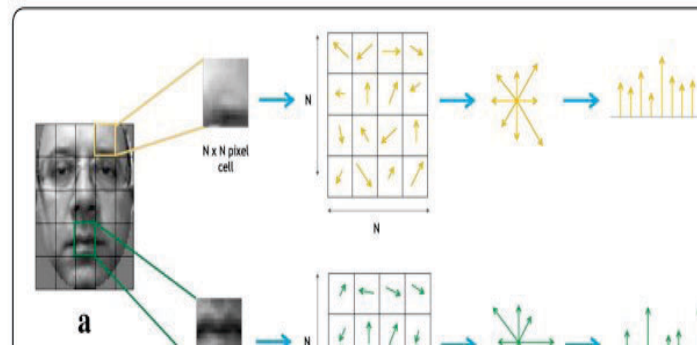
Figure 3. HOG extracts the face image

HOG features extraction process: image is divided in cells of size N × N pixels. The orientation of all pixels is computed and accumulated in an M-bins histogram of orientations. Finally, all cell histograms are concatenated in order to construct the final features vector. The example reports a cell size of 4 pixels and 8 orientation bins for the cell histograms.

2.2.   Max Margin Object Detection

In the following case, we will use r to show the rectangular area of an image. In addition, r displays the set of all rectangular areas scanned by the object detection system [3]. To combine the general practice of non-maximum emphasis, we define labeling valid images as part of r so that each labeling element "does not overlap" with each other. We use the following popular definitions "no overlap": rectangles r1 and r2 do not overlap if the ratio of the junction area to the total enclosed area is less than 0.5, as follows:

$$\frac{Area(r_1 \cap r_2)}{Area(r_1 \cap r_2)} < 0.5.$$

Finally we use y to show a collection of all valid labels. Then, given the x sign and the function of the scoring window is f, and we can define the object detection procedure as

$$y^* = \frac{argmax}{y \in Y} \sum_{r \in y} f(\mathrm{x}, \mathrm{r}).$$

Now we use MMOD to consider the function of scaling the right window on its parameters. In particular we use functions from the form

$$f(x, r) - \langle \omega, \emptyset(x, r) \rangle$$

Where ϕ extracts into the vector feature of the location of the r sliding window in figure x, and w is the parameter vector. If the number of window scores is denoted for a set of rectangles, y, such as F (x, y), then Equation (2)

$$y^* = \frac{argmax}{y \in Y} \sum_{r \in y} f(\mathrm{x}, \mathrm{r}).$$

The next step is to find the parameter vector w which leads to the least detection error, after that it is given a randomly selected pair of images and labels (xi, yi) x x y, and make the score for the correct labeling of xi be greater than the score for all labeling that is wrong. Therefore,

$$\mathrm{F}(x_i, y_i) > \max_{y \neq y_i} F \quad (x_i, y)$$

3.   Testing dan Result

First step is training the object so it can be detected on image. This experiment uses 4 images that consist no u-turn sign. Indonesia is a country that applied left driving system so a u-turn sign is look like this:

Figure 4. No u-turn sign for left driving countries

Figure 5 represented images that we used for training:

**Figure 5.** No u-turn signs for training dataset.

As shown on figure 5, the size of sign and images have different size so we need to tell which area on these images is no u-turn sign. Next step is mark area that represent object that we want to be found on testing images so MMOD will can be trained with those information.

Figure 6. Images with no u-turn marked

The learned HOG detector is looked like this:
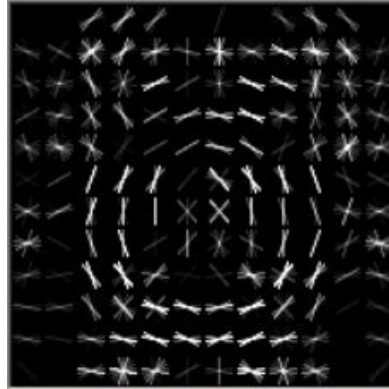
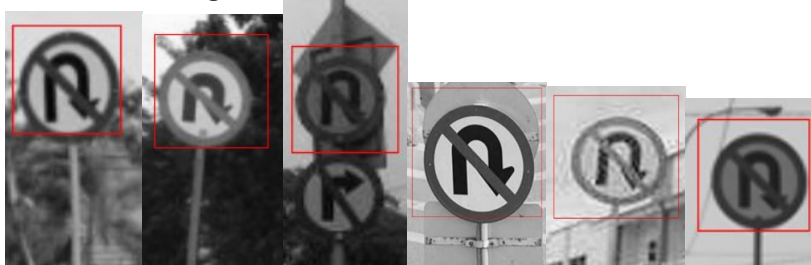**Figure 7.** Learned HOG detector for no u-turn

Now we try to detect no u-turn signs in several images. The images is varied from size and resolution so we can try if it could detect images with different resolution from training images.

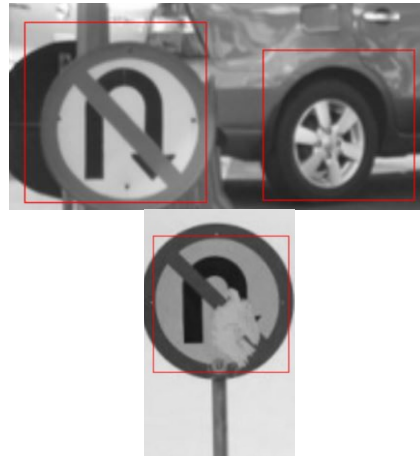**Figure 8.** Images that unsuccessful identified

Figure 9. Images that successful identified

Figure 10. Images that successful identified (continued)

Figure 11. Images that have false alarm

Based on the results that is shown from figure above, we can tell that this method is good to learn and identified the object based on its training. Unfortunately there is unsuccessful identified images and false alarm. On figure 11 it gives a false alarm on tire and damaged no u-turn sign (it is counted as different object because about 20% part is missing from that sign).

For future research we will try to find what is the cause for this false alarm and unsuccessful identified case.

From the research conducted. The researcher found the results made in the test results table:

Table 1. Test Results

Object	Condition Process	Results
Signs are not allowed to turn back	The object is captured by the application during the day with a distance of 4M from the object	Objects can be detected and notifications not to play back
Signs are forbidden to turn right	The object is captured by the application during the day with a distance of 4M from the object	Objects can be detected and notifications not to turn right
Car tires	The object is captured by the application in the morning with a distance of 5M from the object	Objects can be detected but the alarm sounds because the object does not include traffic signs
Signs that are damaged or not in good condition	Objects are captured in the application during the afternoon with a distance of 5M from the object	The object can be detected but the alarm sounds because the object is damaged
Signs are not allowed to turn back	The object is captured during cloudy weather in the afternoon	Object not detected

Objects that are captured and identified and seen when they are taken and objects are trained using the program and the results of the HOG process are obtained.

4.   Conclusions

HOG with MMOD is easy to use and implement because it only needs a few sample images and we can make our own object detectors based on those images. The process is also fast enough to run in real-time so that it can be implemented on a smart car system for quick updates about changes in traffic direction carried out on the same day.

References

[1]   Hoffstein J., Lieman D., Pipher J., Silverman J. "NTRU: A Public Key Cryptosystem", NTRU
          Cryptosystems, Inc. (www.ntru.com).
[2]   Sharma.,A.K "Estimation of Age Group using Histogram of Oriented Gradients and Neural
          Network", International Journal of Engineering Research and Development, vol. 13, Issue
          11, p.43-49, 2017
[3]   Ali.,N.M, Karis.,M.S, Sobran.,N.M.M,Bahar.,M.B, O.K. Ken, M.M. Ibrahim, and N.F. Johan,
          "Detection of Multiple Mangoes using Histogram of Oriented Gradient Technique in Aerial
          Monitoring", ARPN Journal of Engineering and Applied Sciences, vol. 12, no. 8, p.2730-
          2736, 2017
[4]   King DE., "Max Margin Object Detection", arXiv:1502.00046v1 [cs.CV]
[5]   Dalal N., and Triggs B, "Histogram of Oriented Gradient for Human Detection", IEEE
          Computer Society Conference on Computer Vision and Pattern Recognition, p.886-893,
          2005
[6]   Greenhalgh J. and Mirmehdi M, "Real-Time Detection and Recognition of Road Traffic Signs",
          IEEE Transactions   on Intelligent Transportation System, vol. 13, Issue 4, p.1498-1506,
          2012
[7]   Creusen I.M.,Wijnoven R.G.J.,Herbschleb E., and P.H.N. de With, "Color Exploitation in
          HOG-Based Traffic Sign Detection", IEEE International Conference on Image Processing,
          2010
[8]   Shangxuan Tian, Ujjwal Bhattacharya , Shijian Lu Bolan Su, Qingqing Wang, Xiaohua Wei, ,
          Yue Lu, and Chew Lim Tan, "Multilingual scene character recognition with co-occurrence
          of histogram of oriented gradients", Pattern Recognition 51 (2016) 125–134
[9]   Pierluigi Carcagnì, Marco Del Coco, Marco Leo and Cosimo Distante,"Facial expression
          recognition and histograms of oriented gradients: a comprehensive study", Carcagnì et al.
          SpringerPlus (2015) 4:645, DOI 10.1186/s40064-015-1427-3
[10]   Siogkas G.,Dermatas E. 2006. 1st Initial, Detection, Tracking and Classification of Traffic
          Signs in Adverse Conditions. IEEE MELECON 2006. 537-540.
[11]   Mohd Ali, N., Mohd Sobran, N., Md Ghazaly, M., Ab Shukor, SA. and Tuani, AF. 2014.
          Traffic Sign Detection and Classification for Driver Assistant System (DAS). 277-283
[12]   Zakir, U., Leonce, NJ., and Edirisinghe, EA. 2010. Traffic Sign Segmentation based on Color
          Spaces: A Comparative Study. In the Proceedings of the 11th International Conference on
          Computer Graphics and Imaging, Innabruck, Austria, 2010.
[13]   Mohd Ali, N., Md Rahid, NK. and Mohd Mustafah, Y. 2013. Performance Comparison between
          RGB and HSV Color Segmentations for Traffic Sign Detection. Applied Mechanics and
          Materials. 393: 550-555
[14]   Benallal M and Meunier J, 2003. Real Time Color Segmentation of Traffic Signs. IEEE
          CCECE 2003 Canadian Conference on Electrical and Computer Engineering.