Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230891

Features extraction for traffic sign recognition

Tianyi Wang[1,6,11], Jianlin Dou[2,7,&], Junyi Wu[3,8], Hang Zhang[4,9], and Yunhua Lu[5,10]

[1]Computer Science, Pennsylvania State University, State College, 16802, United State
[2]College of Arts and Sciences (Computer and information Science), The Ohio State University-Columbus, Columbus, Ohio, 43210, United State
[3]Electrical and Electronic Engineering, Imperial College London, London, SW72BX, United Kingdom
[4]Tandon school of engineering, New York University, New York, 11201, United States
[5]Guangdong Experience High School, Guangzhou, 528306, China

[6]xalyedaniel@gmail.com
[7]pochooge@gmail.com
[8]jw3621@ic.ac.uk
[9]Hz2764@nyu.edu
[10]glorysingry@outlook.com
[11]These authors contributed equally to this work and should be considered co-first authors.

Abstract. Traffic sign recognition (TSR) is the basic technology of the Advanced Driving Assistance System(ADAS) and intelligent automobile, while a highly qualified feature vector plays a key role in TSR. Therefore, the feature extraction of TSR has become active research in the fields of computer vision and intelligent automobiles. Although deep learning features have made a breakthrough in image classification, it is difficult to apply to TSR because of its large scale of training dataset and high space-time complexity of model training. Considering visual characteristics of traffic signs and external factors such as weather, light, and blur in real scenes, an efficient method to extract high-qualified image features is proposed. As a result, the lower-dimension feature can accurately depict the visual feature of TSR due to its powerful descriptive and discriminative ability. In addition, benefiting from a simple feature extraction method and lower time cost, our method is suitable to recognize traffic signs online in real-world applications scenarios. Extensive quantitative experimental results demonstrate the effectiveness and efficiency of our method.

Keywords: Traffic Sign Recognition, Advanced Driving Assistance System, Neural Network, Feature Extraction.

1. Introduction

The TSR function reduces the possibility of drivers not complying with traffic regulations such as stop signs, and avoids illegal left turns or other unintentional traffic violations, thus improving safety. These systems need flexible software platforms to enhance detection algorithms and adjust them according to

DOI: 10.54254/2755-2721/6/20230891

traffic signs in different regions. In the underdeveloped stage of transportation facilities, maps and smart cities, the recognition of traffic signs is also one of the highlights of camera sensors.

Traffic sign recognition system generally includes two parts: detection and recognition. Generally, the detection uses the shape and color features of traffic signs to extract traffic signs from natural scenes. Recognition is to identify the content of the detected traffic signs [1]. Traffic sign recognition is of great significance in standardizing traffic behavior and ensuring safe driving. Traffic signs are usually located in outdoor complex environmental conditions, and the recognition process is easily affected by environmental lighting and direction rotation.

Traffic sign recognition system is an important part of intelligent transportation system and advanced assistant driving systems. Improving the accuracy and real-time of traffic sign detection and recognition algorithm is a key problem to be solved in the process of practical application. The accuracy of the algorithm is a very important factor in the research of traffic sign recognition. The wrong recognition results can not only play a role in assisting driving, but also lead to serious safety accidents. The real-time performance of the algorithm determines whether the research results can be transformed into products with practical application value [2]. In the face of the increasing number of cars, high traffic safety accidents and the realistic pressure of constantly improving the driving intelligence of cars, it is of great significance to carry out research on traffic sign detection and recognition technology aimed at real-time application to increase driving safety.

In recent years, deep learning, especially Convolution Neural Networks (CNN), has achieved great success in object recognition and classification [3]. German Traffic Sign Recognition Benchmark (GSTRB) dataset, including 39,209 training images and 12,630 test images, provides a considerable amount of data for deep learning to train [4], resulting in a high classification accuracy. The CNN-based method with spatial transformation and learned color receives an accuracy score of 99.59. However, those training methods have three obvious drawbacks: complex structure, requirement of a great amount of parameters and time.

In our method, we extracted features from HOG and color histogram based method. After that, we improve those HOG and color histogram-based features by utilizing Principal Component Analysis (PCA), which can help us to achieve the purpose of dimension reduction. We also use the GSTRB dataset, which includes traffic signs in different environments (fog, rain, light levels, shadow, and twilight).

2. Background

Traffic Sign Detection (TSR) contains two main parts, detection and classification. The detection stage selects the Region of Interest (ROI), areas that can help us do classification, in pictures, while the classification stage takes ROI as input and identify the class of the sign in the picture.

2.1. Detection

The two most common ways of doing detection are color base and shape base. Base on these methods, hybrid methods perform both color and shape base detection to reduce error. Under proper conditions, these methods are accurate and efficient. However, Machine learning methods were applied in recent years, and after training and adjustment, it can achieve higher accuracy.

2.1.1. Color base. Color is the most obvious feature, hence easy to extract, but easily influenced by illumination and weather. The color distance method is used in this article to extract certain colors, which calculate the Euclidean distance of pixels' color value in RGB color space [5]. In other articles, Hue-Saturation-Intensity (HSI) Hue-Saturation-Value (HSV) and YCbCr color space were used to off-set the illumination effect on traffic signs, and because these color space each have an independent value that represents brightness, the accuracy is quite decent [6-8]. The authors in this article use white balance to even illumination conditions in different pictures [9].

2.1.2. Shape base. Compared to the color base methods, shape base methods are less sensitive to illumination, hence more stable. Hough transform is used by authors in [10]. It's a mathematical method that detects relations between pixels and checks for certain shapes. A threshold is also used, if the number of points on a line above the threshold is too large, it won't be computed, and time can be saved. In another article, radial symmetry transformation is used [11]. By converting images into greyscale, then calculating gradient, magnitude, and direction for each pixel, we can get two maps respectively for magnitude and direction. Shapes can be detected by polarising maps using thresholds.

2.1.3. Hybrid base. Under some circumstances, object in background with similar color or shapes to traffic sign may contribute to inaccuracy in the detection result, so some authors combined color base and shape base methods to avoid this.

In this article, authors used RGB ratios to perform color segmentation, then Douglas-Peucker algorithm is used to identify shapes, hence reduce interferences caused by similar color in the background [12]. In another article, HOG feature, and color histogram feature were extracted, converted to a one-dimensional vector, then added together [2]. In this case, higher dimensional feature vectors make the detection less likely to went wrong.

2.1.4. Deep learning. Deep learning methods can be very accurate, but they usually requires tons of data, time, and computing power to train. Authors in their article used Support Vector Machine (SVM) to classify the pixels, and achieve decent accuracy, but the process is relatively slow [13]. There are also authors trained two Fully Convolutional Network (FCN) and use them simultaneously, one does detection only, the other one do detection and classification simultaneously [14]. Faster R-CNN is also used, but the result showed that this method is neither fast nor accurate [15].

2.2. Classification

2.2.1. Base on color and shape. color and shape of a traffic sign is enough to get information, we just need to extract them. Color is easy to get, while extracting shapes is more complex. In one article, circularity, a variable that has value 0 for a perfect circle and becomes higher with more non-circular shapes, is calculated [16]. A threshold is then applied to determine the shape. Distance to Border (DtB) is also used together with a SVM to identify the shape [5]. Four DtB vectors were used to represent distances from the sign to four corners of the picture, and then they were classified by SVM.

2.2.2. Base on deep learning. Many Deep learning methods can be used to classify traffic signs without other algorithms. Hence they are really popular. Support Vector Machine (SVM), convolutional neural network (CNN) and radial-based neural networks [17] all have been used, and SVM seems to be the most popular one [17-19].

2.3. State-of-the-art

Instead of using one of two properties (color and shape) to select ROI, State-of-the-art used a different method [2]. It extracts both HOG feature and color histogram feature, then combines them, uses principle component analysis (PCA) to reduce dimensions of feature vectors to speed up the process, and finally, use SVM to classify Traffic signs. A combination of both color and shape information can increase the accuracy of the model, and without using Deep-learning in the detection part, it doesn't require too much data and training time, hence more efficient.

3. Approach

Overarchingly, using the deep learning feature is the most ideal approach to pro- cess and recognize images since it generates a greater accuracy in object detection, image classification, Simultaneous Localization and Mapping (SLAM), and semantic segmentation compared to traditional image processing techniques. However, it is not always the most significant approach, especially for traffic sign

DOI: 10.54254/2755-2721/6/20230891

recognition. Compared to deep learning, traditional feature extraction methods could conduct a timesaving and robust process that may perform better on TSR [2]. In this project, color histograms and HOG are used to extract the color and shape features of images of different traffic signs. PCA is used to reduce the dimension to reduce overfitting and improve algorithm performances. The result is put into the SVM classifier for recognition.

3.1. HOG method

For shape features, we mainly use the Histogram of Oriented Gradient (HOG) features. It is a feature descriptor that uses the distribution of directions of gradients to generate a histogram. By combining the blocks of gradients divided from the graying images, the HOG features can be generated [20]. The main method is described below. Firstly, an input image will be converted to a three-dimensional grayscale image with x-axis, y-axis, and z-axis. Then, the image will be normalized using Gamma correction method, which can adjust the contrast ratio and to minimize the effect caused by shadow and illuminations [21]. The magnitude and the direction of each pixel are calculated to obtain the edge and the image is divided into small cells composed of a matrix of pixel. After combining cells into several blocks, the HOG descriptor of such areas can be generated. The overall HOG will be constructed by adding the sub descriptors together [22]. Figure 1 shows the result of feature extraction done by HOG.

3.2. Color histogram based method

For color based method, we use color-histogram-based method, and both RGB and HSV color space are included. The color histogram of an image can provide a solid result of how different colors are reveled and how many pixels are in these colors [23]. In RGB scale, images are divided into three channels which are red, green, and blue.The HSV color space can be represented as a three-dimensional hexcone, where the central vertical axis represents intensity which takes a value between 0 and 255 [24]. Basically, within the hexcone model, colors are set in range of 360 degrees where red starts from 0 degree, green's from 120 degrees and blue starts from 240 degrees [25]. By examining the hue of the colors in the cube around the axis, it generates a characterization of hue, saturation and brightness. A histogram is built by dividing the color space into small bins and counting the number of pixels in each of the bins [2]. The following graph shows the color extraction done by Color histogram for the same image in Figure 2.

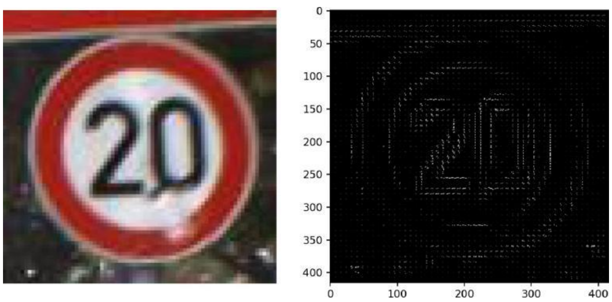

Figure 1. Feature extraction done by HOG.

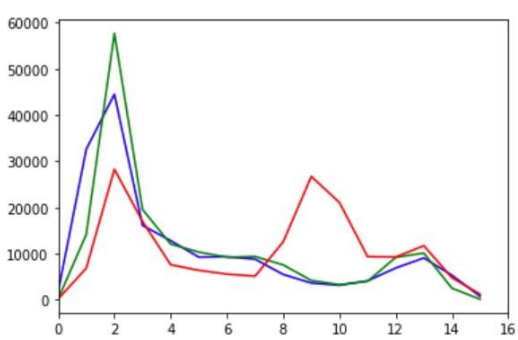

Figure 2. Color extraction done by Color histogram.

3.3. PCA method (dimension reduction)

After getting the features applying color histogram and HOG, we find that the dimensions of the features are too high. In that case, it is difficult to compute the raw data. Therefore, we decide to use the Principle Component Analysis (PCA) to reduce the dimension. The principle of how PCA works is that it helps find out the correlation between the feature extracted and image itself. Therefore, by reducing a tremendous amount of irrelevant and repeating errors, it can efficiently decrease the difficulty of computing and reduce the run time [26]. Moreover, in this project, We use a nonlinear principle components analysis. As shown in figure 3 below, kernel PCA can resolve more complicated data patterns by projecting the data into a higher dimension. By increasing the higher exponential of the data's function, kernel PCA helps create an efficient transformation to make the highest exponential of the original function become linear using the kernel function [27]. The methods are explained in the function below. In the projected features, the eigenvalues and eigenvectors can simply be written as

$$V_k = \sum_{i=1}^{N} ak_i \, \phi(x_i) \tag{1}$$

The resulting kernel principle components can be generated using

$$y_k(X) = \phi(X)^T V_k = \sum_{i=1}^{N} a_k \, iK(X, X_I) \tag{2}$$

The polynomial matrix can be written as

$$k(x, y) = \phi(x_i)^T \phi(x_j) \tag{3}$$

The projection of features can be operated by the projection operator.

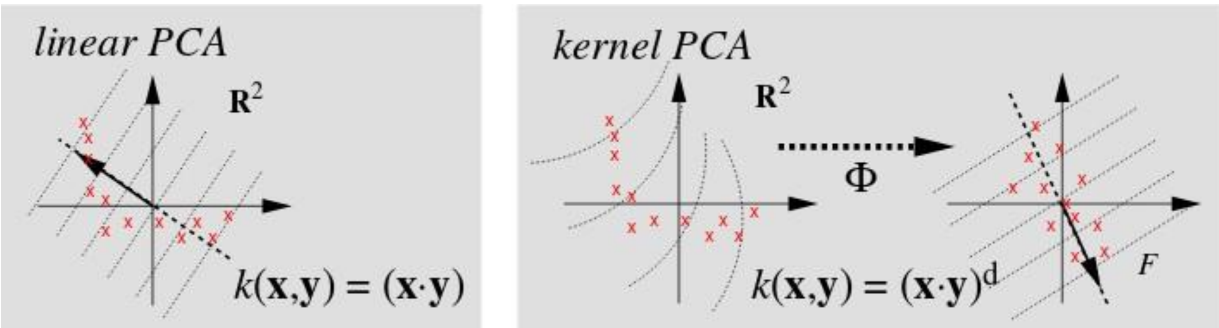

Figure 3. Difference between PCA and kernel PCA.

3.4. SVM classifier

$$P_m\phi(X) = \sum_{k=1}^{m} y_k(X) V_k \tag{4}$$

In this project, we use the support vector machine(SVM) as the classifier. SVM classifier can change the dot production operation to the kernel function, which sufficiently distinguishes and eliminate the similarity of the distribution [18]. In general, the radial basis function and polynomial function are the popular choices of functions. For radial basis function, it can reflect the complexity of the model.

Therefore, it can be used for most of the models with appropriate parameters [28]. However, it usually requires a large amount of memory which is expensive to compute. The polynomial kernel function can speed up the computing process with a little loss of accuracy compared with radial basis function [29]. At this time, we choose the polynomial kernel since it matches the purpose of efficiency. The polynomial kernel with power 2 is calculated in formula 1 below:

$$K(x, y) = \sum_{i=1}^{n} (x_i y_i + c)^2 = \sum_{i=1}^{n} (x_i^2)(y_i^2) + \sum_{i=2}^{n} \sum_{j=1}^{i-1} (\sqrt{2}\, x_i x_j)(\sqrt{2}\, y_i y_j) + \sum_{i+1}^{n} (\sqrt{2c}\, x_i)(\sqrt{2c} y_i) + c^2 \tag{5}$$

DOI: 10.54254/2755-2721/6/20230891

4. Experimental results

4.1. Data-set description

The data-set we applied to our research is called German Traffic Sign Recognition Benchmark (GTSRB). The training data-set is composed of totally 42 different class folders. The average count of Images in each class folder is around 250 above. Also, inside the testing data-set, no specific classes folder well arranges these images. 155 images belonging to different classes were mixed. This briefly describes the situation of our picked GTSRB data-set.

4.2. Experiments

Our experiment was conducted in the CoLab virtual environment. Our team selected 3779 traffic sign images from the first 9 different classes for training purposes. The average count of Images for training purposes from each class is 200 above. For testing purposes, our team randomly selected 43 images among images in the testing data-set in the range of the first 9 different classes. The algorithm applied in the experiment is Support Virtual Machine (SVM) [18]. Our team picked the polynomial kernel to train the model. Also setting the degree of polynomial kernel equal to 3 and c value equal to 120. Before training the model, our team proposed some hypotheses to extract the useful feature information hidden in images according to the steps mentioned in the research [2].

4.2.1. Experiment1: Training RGB&HOG features. We first extract the RGB color-histogram as a vector concatenate to the extracted HOG feature vector and use the combined feature vector of each image to train our model by using the polynomial kernel without the dimension being reduced. The test results on 43 test images are as follows (see Figure 4):

Table 1. Experimental result of testing RGB&HOG features.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
RGB&HOG	72.09%	70.88%	74.0%	72.0%	108s

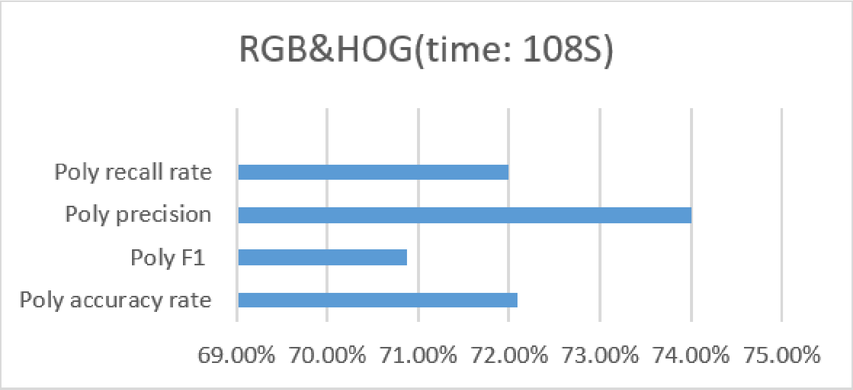

Figure 4. Result visualization of testing RGB&HOG features.

4.2.2. Analysis of Experiment1. As table 1 shown, testing on RGB color histogram plus extracted HOG feature vector did not reach up to 90% accuracy. Our team thinks training data in each given class is still not larger enough to detect a robust and distinct feature pattern to classify the test images; that's why the accuracy rate is sort of worse.

4.2.3. Experiment2: Training HSV&RGB&HOG Features. We then extract the HSV color-histogram as a vector concatenate to the RGB color-histogram and the extracted HOG feature vector, then use this combined feature vector of each image to train our model by using the polynomial kernel without the dimension being reduced, the test results on 43 test images are as follows (see Figure 5)

Table 2. Experimental result of testing HSV&RGB&HOG features.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
HSV&RGB&HOG	72.09%	70.88%	74.0%	72.0%	117s

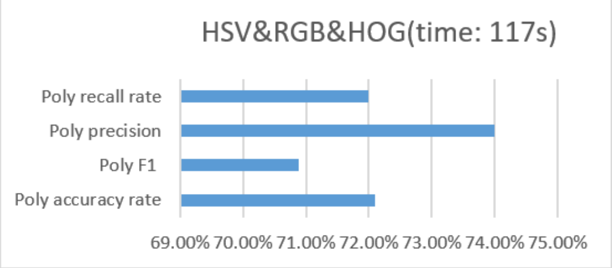

Figure 5. Result visualization of testing HSV&RGB&HOG features.

4.2.4. Analysis of Experiment2. As table 2 shown, testing on HSV color-histogram plus RGB color-histogram plus HOG feature vector presents the same result as table 1 shown. So, we think RGB color space is already good enough to describe the color information since there is no boost in accuracy rate by applying both HSV and RGB color histogram as two additional feature vectors.

4.2.5. Experiment3: Training HSV&RGB features. We also tried only color spaces to train the model. Meaning we only concatenate vectors of extracted RGB color-histogram and HSV color-histogram of each image without dimension being reduced, the test results on 43 test images are as follows (see Figure 6):

Table 3. Experimental result of testing HSV&RGB features.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
HSV&RGB	74.42%	72.87%	76.0%	74.0%	6s

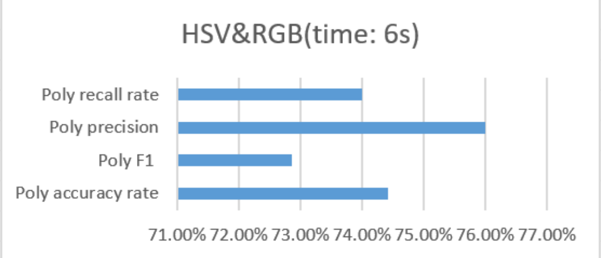

Figure 6. Result visualization of testing HSV&RGB features.

4.2.6. Analysis of Experiment3. As table 3 shown, testing on HSV color-histogram plus RGB color-histogram feature vector seems to increase a 2% accuracy rate compared with results from both table1 and table2. Also, the training time was drastically shrunk. So, we think the extracted HOG feature vector must be the only obstacle to lowering the training efficiency and testing correctness. That makes sense, when trying to extract the HOG feature, our team sets the cell size to 16 long, and the number of bins to 8. Under these given values, some specific feature patterns have already gone. For example, the feature pattern of numbers two and three became so ambiguous to recognize. That's a trade-off, if our team makes the individual feature pattern distinct and recognizable, the cell size parameter will be increasingly smaller. The smaller the cell size is, the larger the feature vector will be. Currently, the feature vector size of HOG is below 20,500. We have tried if we change the cell size to 8 long, the feature vector size will be 90,000 above. Therefore, the time used for training model increased.

DOI: 10.54254/2755-2721/6/20230891

4.2.7. Experiment4: Testing Differences After Reducing Feature Dimension(PCA) [30]. We also tried the PCA dimension reduction algorithm on the input feature matrix used for training purposes. Vertically comparing the previous results without dimension reduced, the testing results are now as follows (see Figure 7, Figure 8, Figure 9):

Table 4. Experimental result of testing RGB&HOG features after applying PCA.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
RGB&HOG	27.91%	27.45%	30.0%	28.0%	16s

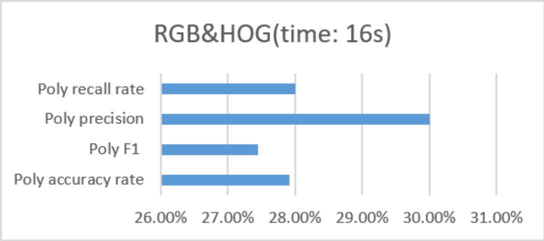

Figure 7. Result visualization of testing RGB&HOG features after applying PCA.

Table 5. Experimental result of testing HSV&RGB&HOG features after applying PCA.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
HSV&RGB&HOG	27.91%	27.45%	30.0%	28.0%	25s

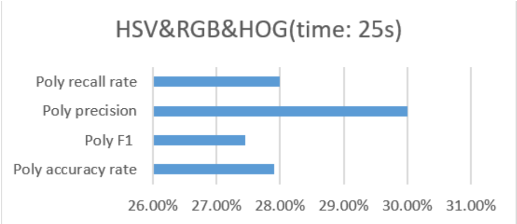

Figure 8. Result visualization of testing HSV&RGB&HOG features after applying PCA.

Table 6. Experimental result of testing HSV&RGB features after applying PCA.

IMAGE FEATURE	Poly_accuracy_rate	Poly_F1	Poly_precision	Poly_recall_rate	Time spending
HSV&RGB	27.91%	27.45%	30.0%	28.0%	2s

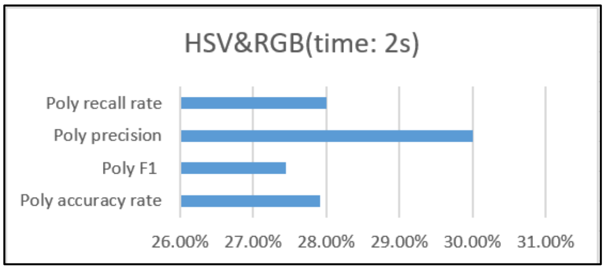

Figure 9. Result visualization of testing HSV&RGB features after applying PCA.

4.2.8. Analysis of Experiment 4. After applying the PCA dimensional reduction algorithm on the input feature matrix used for training model to reduce the roughly 20,000 long feature dimensions to 43 long, as table 4,5,6 shown, the accuracy rate seems to drop drastically compared with the accuracy rate presented

DOI: 10.54254/2755-2721/6/20230891

in table 1,2,3. That makes sense since after one dimension is reduced, the dimensional reduced matrix will lose a part of original feature information. For 20,000 long feature dimensions reduce to 43 long, the dimensional reduced matrix might lose a large amount of feature information. Thus, by using dimensional reduced data to train the model and test, the accuracy rate should be pretty low.

5. Conclusion

In this paper, we use the German GSTRB dataset and extract the main features from HOG and color histograms. Then, Principal Component Analysis (PCA) is used to reduce the dimensionality of the features in HOG and color histograms to perform effective traffic sign detection and recognition from both color and shape. Finally, the SVM classifier is used for classification. This method greatly improves the accuracy and real-time performance of traffic sign detection and recognition. Our main contributions are as follows:

1. We demonstrate that the method using both color histogram and HOG is feasible.
2. We demonstrate that the PCA dimensionality reduction method can improve computation time and accuracy.

The fly in the ointment is that our method is only validated to a certain extent on the GTSRB dataset. We are missing the performance comparison of traffic sign detection and recognition in other countries. Our method may not work well for all traffic sign recognition.

In future research, we will continue to improve our traffic sign detection and recognition method to maximize its accuracy and effectiveness, and to increase its speed. We will also try to conduct experiments on traffic recognition datasets in other countries so that our method can continue to be applicable in more situations.

Acknowledgement

Tianyi Wang and Jianlin Dou contributed equally to this work and should be considered co-first authors.

References

[1] C. J. C. S. Ardianto and H. M. Hang, "Real-time traffic sign recognition using color segmentation and svm," in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 1–5, 2017.

[2] W. Li, H. Song, and P. Wang, "Finely crafted features for traffic sign recognition," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, pp. 159–170, 2022.

[3] I. S. A. Krizhevsky and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25, Vol.25*, pp. 1097–1150, 2012.

[4] J. S. J. Stallkamp, M. Schlipsing and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

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

[6] Y.-Y. Nguwi and A. Z. Kouzani, "Automatic road sign recognition using neural networks," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 3955–3962, IEEE, 2006.

[7] N. B. Romdhane, H. Mliki, and M. Hammami, "An improved traffic signs recognition and tracking method for driver assistance system," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–6, IEEE, 2016.

[8] Y. R. Fatmehsari, A. Ghahari, and R. A. Zoroofi, "Gabor wavelet for road sign detection and recognition using a hybrid classifier," in *2010 International Conference on Multimedia Computing and Information Technology (MCIT)*, pp. 25–28, IEEE, 2010.

[9] Z. Zheng, H. Zhang, B. Wang, and Z. Gao, "Robust traffic sign recognition and tracking for advanced driver assistance systems," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 704– 709, IEEE, 2012.

[10] M. A. Garcia-Garrido, M. A. Sotelo, and E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions," in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 811–816, IEEE, 2006.

[11] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," in *IEEE Intelligent Vehicles Symposium, 2004*, pp. 566–571, IEEE, 2004.

[12] Z. Zheng, H. Zhang, B. Wang, and Z. Gao, "Robust traffic sign recognition and tracking for advanced driver assistance systems," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 704– 709, IEEE, 2012.

[13] H. Gómez-Moreno, S. Maldonado-Bascón, P. Gil-Jiménez, and S. Lafuente- Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 917–930, 2010.

[14] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2110–2118, 2016.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[16] H. Supreeth and C. M. Patil, "An approach towards efficient detection and recognition of traffic signs in videos using neural networks," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 456–459, IEEE, 2016.

[17] M. M. Lau, K. H. Lim, and A. A. Gopalai, "Malaysia traffic sign recognition with convolutional neural network," in *2015 IEEE international conference on digital signal processing (DSP)*, pp. 1006–1010, IEEE, 2015.

[18] C. Liu, F. Chang, and Z. Chen, "High performance traffic sign recognition based on sparse representation and svm classification," in *2014 10th International Conference on Natural Computation (ICNC)*, pp. 108–112, IEEE, 2014.

[19] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017.

[20] M. Xing, M. Chunyang, W. Yan, W. Xiaolong, and C. Xuetao, "Traffic sign detection and recognition using color standardization and zernike moments," in *2016 Chinese Control and Decision Conference (CCDC)*, pp. 5195–5198, IEEE, 2016.

[21] P. Carcagnì, M. Del Coco, M. Leo, and C. Distante, "Facial expression recognition and histograms of oriented gradients: a comprehensive study," *SpringerPlus*, vol. 4, p. 645, 11 2015.

[22] S.-H. Lee, M. Bang, K.-H. Jung, and K. Yi, "An efficient selection of hog feature for svm classification of vehicle," in *2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1–2, 2015.

[23] W. J.-F. Y. H.-Y. Wang, Xiang-Yang, *Robust image retrieval based on color histogram of local feature regions*.

[24] K. Zhang, Z. Fang, J. Liu, and M. Tan, "An adaptive way to detect the racket of the table tennis robot based on hsv and rgb," vol. 2015, pp. 5936– 5940, 09 2015.

[25] J. Gordon, I. Abramov, and H. Chan, "Describing color appearance: Hue and saturation scaling," *Perception psychophysics*, vol. 56, pp. 27–41, 08 1994.

[26] I. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, p. 20150202, 04 2016.

[27] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," 2012.

DOI: 10.54254/2755-2721/6/20230891

[28] A. Elen, C. Közkurt, and S. Baş, "An adaptive gaussian kernel for support vector machine," *ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING*, vol. 47, pp. 10579–10588, 03 2022.
[29] Y. Goldberg and M. Elhadad, "splitSVM: Fast, space-efficient, non-heuristic, polynomial kernel computation for NLP applications," in *Proceedings of ACL-08: HLT, Short Papers*, (Columbus, Ohio), pp. 237–240, Association for Computational Linguistics, June 2008.
[30] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.