Reducing Area Recognition for Vehicle Model Classification using Car's Front Side

Arjun Sutrisno[1, a], Indrabayu[2*], Intan Sari Areni[3, b], Anugrayani Bustamin[2, c]
Artificial Intelligence and Multimedia Processing Research Group
[1]Postgraduate Student of Electrical Department, Hasanuddin University
[2]Department of Informatics, Hasanuddin University
[3]Department of Electrical Engineering, Hasanuddin University
Makassar, Indonesia 90425
[a]sutrisnoa19d@student.unhas.ac.id, {[b]intan, [c]anugrayani}@unhas.ac.id
*Corresponding Author: indrabayu@unhas.ac.id

***Abstract*—A Car Make and Model Recognition (CMMR) system plays an essential role in Intelligent Transport System (ITS) development. The challenge is identifying the features of a car and simplifying the process of a system. This work presents a system that can handle the challenges. This research aims to classify car models based on global features in the car's front-side view image. The dataset used consists of 5 classes spread into 387 images with 312 train data and 75 test data. The method used in feature extraction is the Bag of ORB Feature (BOF) method, which is a combination of the Oriented and Rotated BRIEF (ORB) feature extraction method and the Bag of Visual Word (BOVW) concept. While at the classification stage, it uses the Support Vector Machine (SVM) method. The results show that the proposed approach can overcome the challenges of CMMR with an F1 score for each class of 96.3%, 91.2%, 87.0%, 81.8%, and 85.7%. In addition, the approach of using the car's front-side view image can also increase the system performance with an average increase of 10% than using the whole car image.**

Keywords—Car Model Classification, Intelligent Transport System (ITS), Bag of ORB Features (BOF), Car's Front Side

I. Introduction

Car Make and Model Recognition (CMMR) is a research topic that makes Intelligent Transport Systems evolve rapidly. Nowadays, CMMR is still of interest to some researchers because it can be used in various fields, including security and industry. CMMR has emerged as a challenging problem in the computer vision domain. Some cars have a similar shape and model, so that is difficult to identify [1]. Therefore, a practical approach is needed to produce a system with high accuracy.

CMMR works based on video or image data, and there are three categories of CMMR techniques, such as features-based, appearance-based, and model-based techniques. The technique has been widely used in recent years, as in [2] with feature-based techniques. This work used the Convolutional Neural Network (CNN) method based on the car's front view image. This approach achieved a classification accuracy of 98.7%. Nevertheless, in this case, it requires many data, around 40.000 image data.

Feature-based techniques are also used in [1] with the Bag of SIFT method. This study uses the car's front view image. Local features in the image are extracted using Scale Invariant Feature Transform (SIFT). The local features are mapped based on the Bag of Visual Word (BOVW) concept. The Support Vector Machine (SVM) method is used to classify car makes and models and produces an accuracy of 89%.

In [3] also use the same technique to recognizing car models. However, in this work, the BRISK (Binary Robust Invariant Scalable Keypoint) method is used based on the car's front view. The approach used is to match the images one by one with the same perspective but different backgrounds. The system created has an accuracy of 96.25%. Meanwhile, The authors in [4] also used this approach using the Oriented and Rotated BRIEF (ORB) method and brute force matching for the classification process. This approach also achieves good performance with an accuracy of 88%. However, it is only efficient when working with a small amount of data as it will take a while to match each image individually.

Besides that, another method also proposed the classification approach with the blob analysis method based on the foreground-background segmentation process. This method works by calculating the pixel area of an object [5]. Unfortunately, this approach is not adequate to be implemented in identifying car models.

Based on several previous works' observations, most research on car model classification uses full front view images of the car through full-body scans [2–4]. However, several parts of cars are not considered unique features of the car. Other problems related to car model recognition are diversity and similarity. The diversity problem refers to the car models of one make (manufactures) having different shapes, as shown in Fig. 1. For example, the Toyota Agya model produced in 2013, 2017, and 2020 has a different appearance, especially on the car's front side. Similarly, the Toyota Avanza, Calya, Innova, and Rush models pose the same challenges. Furthermore, the similarity problem arises when different car models have a similar front appearance, as shown in Fig. 2. For example, the Toyota Avanza 2007 and Toyota Innova 2009 models have similar grilles and bumpers. Likewise, on the Agya 2020 and Calya 2016 models.

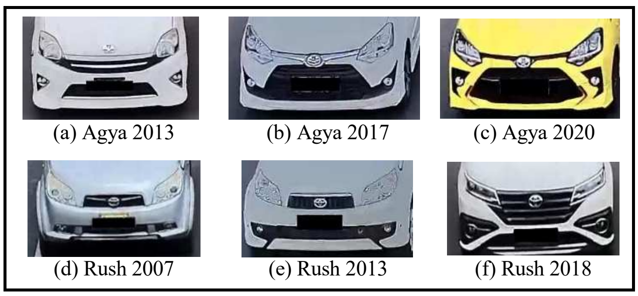

Fig. 1. Diversity problem, (a)-(c) Toyota Agya and (d)-(f) Toyota Rush

To overcome the problems mentioned above, this paper presents:

- A comparison between the proposed approach using a car's front-side view and the previous approach using the car's full front view.
- An evaluation of the proposed method, that is, ORB combined with BOVW to represent global features. Furthermore, it compared with the previous method, that are SIFT and BRISK methods. It aims to

determine the effectiveness of overcoming the problems.

Fig. 2. Similarity problem between (a) Avanza 2007, and (b) Innova 2009

The proposed approach is trained using a machine learning algorithm, namely Support Vector Machine (SVM). This method is one of the most popular and effective methods for classifying. It is proven in some test analyzes, SVM achieves high classification accuracy [6–10].

The paper structure is as follows: Section I shows an introduction, previous research, and the purpose of the research. The dataset and proposed method were described in Section II. Section III discusses the experimental results, and Section IV presents the conclusions.

II. Proposed Method

In this paper, there are several stages for classifying car models. These stages are shown in Fig. 3 and are described in the following section.

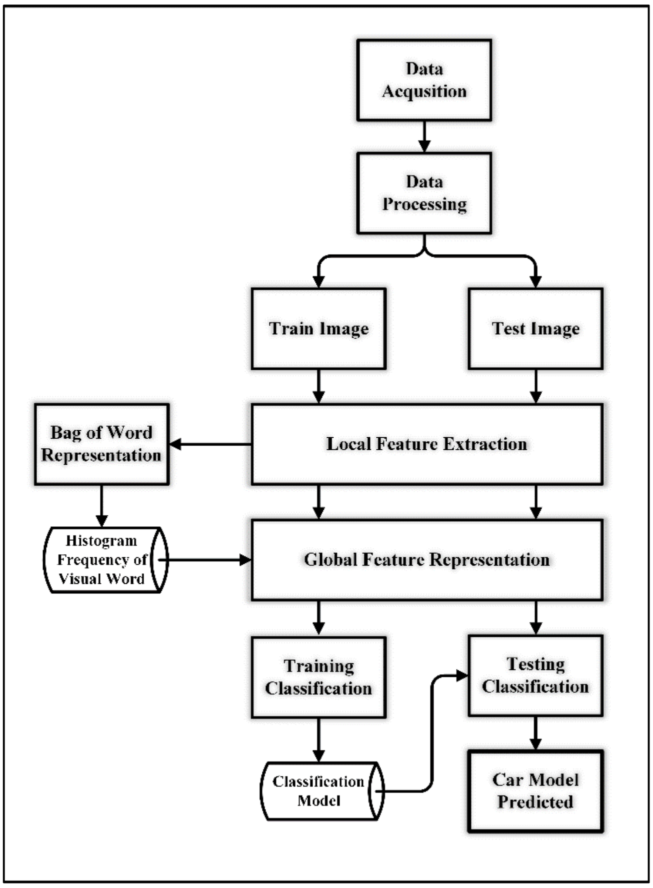

Fig. 3. Block diagram of the proposed system

A. Data Acquisition

This research used an image dataset derived from video data containing vehicles on the highway. Video data were collected from the pedestrian bridge using a Vivotek IP camera with 1920 x 1080 pixels. The videos collected are 100 videos with the mp4 file format, and each video has a duration of 1 minute with a frame rate of 30 and 60 fps (frames per second).

B. Data Processing

The video is processed to produce a collection of car images used to classify car models. This work uses the car's front-side view image because this part is a feature that can distinguish each car model and streamline the feature extraction process. An example of the car's full front view is shown in Fig. 4(a), and the car's front-side view is shown in Fig. 4(b). The data processing step is shown in Fig. 5 and described as follows.

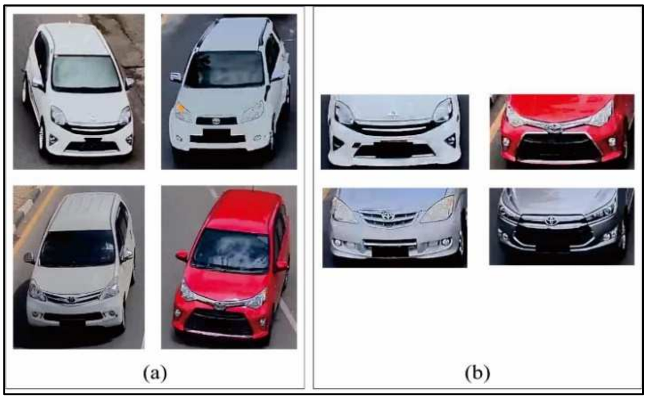

Fig. 4. Example of car images; (a) Full front view; (b) Front-side view

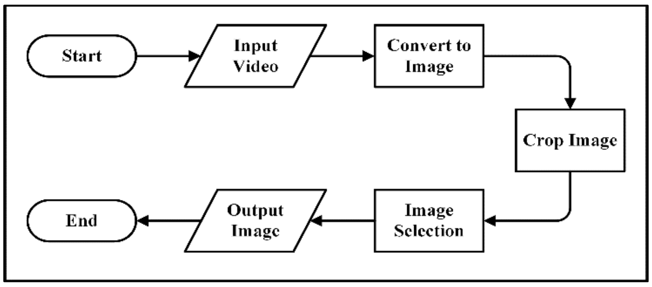

Fig. 5. Flowchart of data preprocessing.

Each video is converted into images. The front side of the car is cropped in each image. An image annotation technique is used by marking the coordinate points on the car's front side. Each of these coordinates is stored in an XML (Extensible Markup Language) file. The image will be cropped based on the coordinates in the file. Then, all the crop images are grouped based on each make and model of the car.

The number of images on each car make and model varies so that one brand and five models of the car are selected based on the highest number of images. The brand is Toyota, and the models are Agya, Avanza, Calya, Innova, and Rush. The total number of images is 387, which are described in Table I.

TABLE I. DETAILED DESCRIPTION OF CAR IMAGES DATA

Make	Toyota					Total
Model	*Agya*	*Avanza*	*Calya*	*Innova*	*Rush*	
Train	64	86	57	51	54	312
Test	13	28	12	11	11	75
Total	77	114	69	62	65	387

The image will be split into two parts, 312 images for the training step and 75 images for the testing step. Each image is converted into grayscale before the feature extraction step to simplify and reduce the calculation's complexity.

C. Local Feature Extraction

The car model classification is based on the global features that will represent each image. However, the global feature is determined from the local feature in an image extracted using the ORB method in this research. Rublee et al. [11] introduce this method as an efficient alternative and improvement of the Features from Accelerated Segment Test (FAST), an algorithm for corner detection.

In the ORB algorithm, the keypoint feature is detected using the FAST method. The orientation of the FAST angle is calculated by referring to the centroid intensity using the moment (*m*) defined in (1).

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (1)$$

$$c = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \quad (2)$$

$$\theta = atan2(m_{01}, m_{10}) \quad (3)$$

Where *m* is moment of patch *p* and *q*. *I* is denoted as the intensity of *x* and *y*, which are the pixel coordinates of the keypoint. When *m* has been determined, the centroid (*c*) can be calculated using (2). Then, the vector is obtained from the center of the angle θ to the *c* using (3). Where *atan2* is the quadrant-aware version of *arctan*. Equation (3) is a simplification for calculating the orientation of the patch.

The keypoint feature contains some information. That is point, size, angle, response, octaves, and class id. Based on this information, ORB uses Binary Robust Independent Elementary Features (BRIEF) as a keypoint descriptor by increasing the invariant rotation.

The entire dataset is used in this process and produces a matrix with $n * 32$ dimensions for each image, where *n* is the number of detected keypoints and *32* is the fixed-length vector of the keypoint description. Thus, each image will produce a different *n*, and this matrix becomes a local feature on an object.

D. Bag of Visual Word Representation

The next step is to represent the image using the BOVW concept. Its concept was adopted from the Bag of Word (BOW) concept for Natural Language Processing (NLP). This method was developed to be used in image classification based on the descriptor keypoint as the image's local feature [12].

Local Features extracted from all training images using ORB and combined into the feature space called a dictionary, as illustrated in Fig. 6(a). Next, local features are mapped to obtain discrete visual vocabulary. In this case, the K-Means clustering algorithm is used to map visual vocabulary with a certain number of clusters, as illustrated in Fig. 6(b). This algorithm will produce a center in each cluster called visual word, as shown in Fig. 6(c). The number of features in each cluster will be represented in a frequency of visual word histogram and will be used to distinguish each image. An illustration of a visual word frequency histogram is shown in Fig. 7. The x-axis represents the index of the visual word, and the y-axis represents the number of features in each visual word. This histogram will be a reference in determining global features.

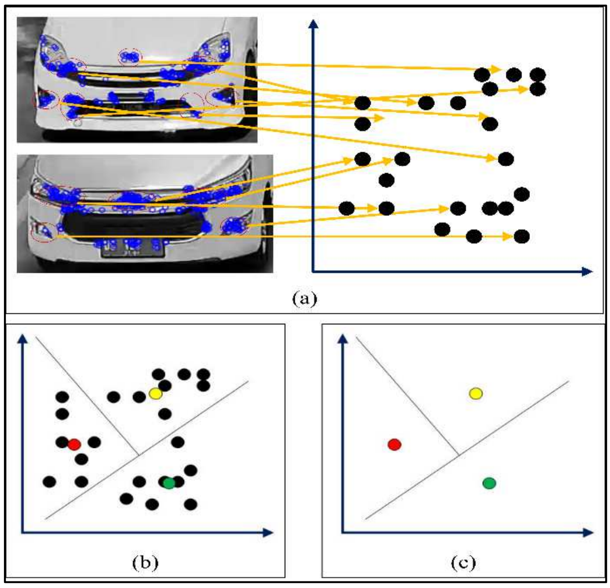

Fig. 6. Illustration of bag of visual word steps, (a) build a dictionary, (b) visual vocabulary clustering, (c) bag of visual word.

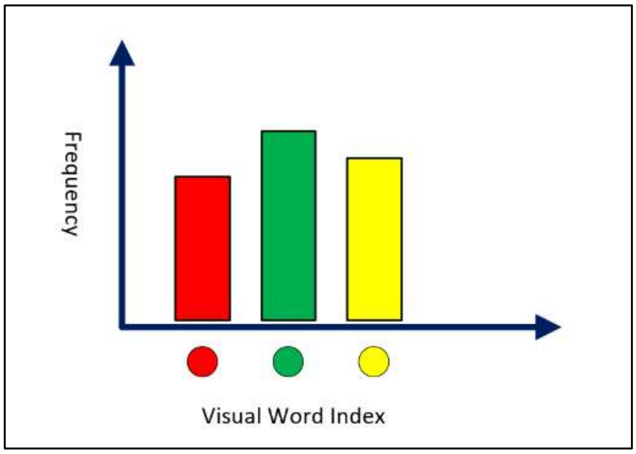

Fig. 7. Illustration of a visual word frequency histogram

The cluster size was varied between 100 until 1000 with 100 intervals to achieve optimal results. If the cluster size is too small (<100), the visual word will be limited to representing all the essential features of the images, which can reduce classification performance. Meanwhile, if the cluster size is too large (>100), it will be an overfitting of the training model. Large size also requires more processing time.

E. Global Feature Representation

The global feature is determined by referring to the frequency of the visual word histogram, as shown in Fig 7. Local features in the entire image data (training and testing image) will be mapped to the closest visual word in the BOVW. This process uses K-Means again by calculating each feature vector's distance to the closest visual word in the BOVW constructed. Thus, each image can be represented as a frequency histogram of the features contained in the image. This frequency histogram becomes a global feature in each image and is sent to the classifier at the training and testing stages.

F. Classification

In this study, the SVM classification method will be adopted. In SVM, there are two important parameters in the classification process. First, the regularization parameter, which is denoted as *C*. The value of *C* must be a positive number (*C* > 0) and the value used is 0.1, 0.15, 0.2, 0.25, 0.3, and 0.5. Second, the kernel parameter is the kernel type used in this algorithm. This algorithm has several kernel type choices: linear, Radial Basis Function (RBF), polynomial, and sigmoid. The best value for each parameter is searched on SVM to obtain optimal results. This process is also known as tuning hyper-parameter, which uses the grid search method [13].

Generally, in the classification process, there are two stages, namely training and testing. The input to this process is the global features that have been obtained in the previous stage. The training stage aims to look for patterns in features so that each class can be distinguished. The training process result is stored as a training model and used as a reference for further testing.

G. Evaluation Technique

The system is tested to evaluate the performance of the proposed approach. There are two test scenarios carried out to analyze the effect of the proposed approach. The proposed system is in the second scenario. The scenarios are detailed in Table II. In addition, the proposed method is compared with other feature extraction methods such as SIFT and BRISK. The method will also be combined with BOVW and use the SVM classifier. It aims to prove the superiority of the proposed method.

TABLE II. DETAILED DESCRIPTION OF TESTING SCENARIO

Scenario	Data	Method	Classifier
1	Full Front View	ORB + BOVW	SVM
2 (Proposed System)	Front-side View		

In this study, the evaluation metric used is F1 score, considering an unbalanced class distribution [14]. The calculation of F1 score is based on the harmonic mean of precision and recall values calculated using the following formulas.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1 = 2 * \left(\frac{Precision*Recall}{Precision+Recall}\right) \tag{6}$$

Where *TP*, *FP*, and *FN* are obtained from a confusion matrix with the following rules:

- True Positive (*TP*): positive data that is correctly predicted as the positive class,
- False Positive (*FP*): negative data that is incorrectly predicted to be positive class,
- False Negative (*FN*): positive data that is incorrectly predicted to be negative class.

III. RESULTS AND DISCUSSION

In this section, the results of testing and evaluating system performance are discussed to determine the efficiency and effectiveness of the proposed approach. This car model classification system was built using the Python programming language and tested using the car images data. As described in Table I, there are 75 test images used with five classes: Agya (*K1*), Avanza (*K2*), Calya (*K3*), Innova (*K4*), dan Rush (*K5*).

The developed system is tested by finding optimal parameters on BOVW and SVM. Based on the experimental result using the proposed approach, the best system performance with the cluster size is 700 on BOVW, 0.1 and linear for *C* and *kernel* on SVM. Thus, the values of these parameters are used in all experiments. Table III shows the confusion matrix obtained from the system with the proposed approach and is used to measure the system's performance using F1 score. Figure 8 shows the evaluation of the F1 score for each class using the proposed method and compared with the features extraction methods in previous research.

TABLE III. CONFUSION MATRIX FROM THE PROPOSED APPROACH

		Predict Class				
		K1	***K2***	***K3***	***K4***	***K5***
True Class	***K1***	13	0	0	0	0
	K2	1	26	0	1	0
	K3	0	0	10	2	0
	K4	0	2	0	9	0
	K5	0	1	1	0	9

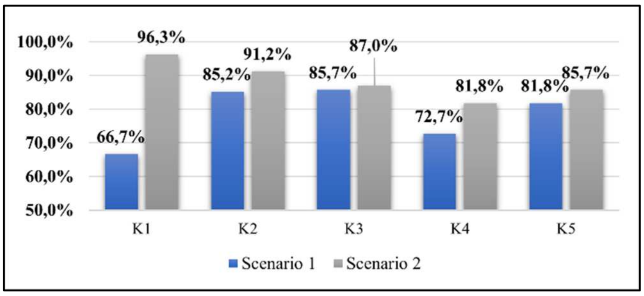

Fig. 8. Performance comparison of two scenarios based on each class

Basically, limiting the area on the car will affect the process of determining the car's features. This area limitation makes the feature extraction process running optimally because feature determination will focus more on the area that contains features. Fig. 8 proves that using the car's front-side view image can improve the performance of the car model classification system. Based on the calculation of the increase in system performance, the average increase is 10%.

System performance is also presented by measuring the F1 score in each class compared with different feature extraction methods, as presented graphically in Fig. 9. The highest F1 score of 96.3% was obtained in *K1* with the proposed approach, which has a significant effect than the two other methods. In addition, the proposed method also performs better on *K2*, *K3*, and *K5* than the other two methods, with F1 scores of 91.2%, 87.0%, and 85.7%, respectively. However, this did not occur in *K4*, where SIFT got a higher F1 score of 87.0%.

The problems of CMMR, as described in Section I, were resolved by applying the proposed method. For example, the diversity, as shown in Fig. 1, was overcome. The system can classify each class with good performance. The similarity problem between *K1* and *K3*, as shown in Fig. 2, was also overcome. However, the problem between *K2* and *K4* did not have a significant effect. There are still two images on *K4* that are predicted to be class *K2*. In addition, the lack of train data in this class makes the system with the proposed method not work optimally to recognize patterns in that class. This effect is seen in *K2*, which has more data than *K4* so that the system can recognize patterns in that class.

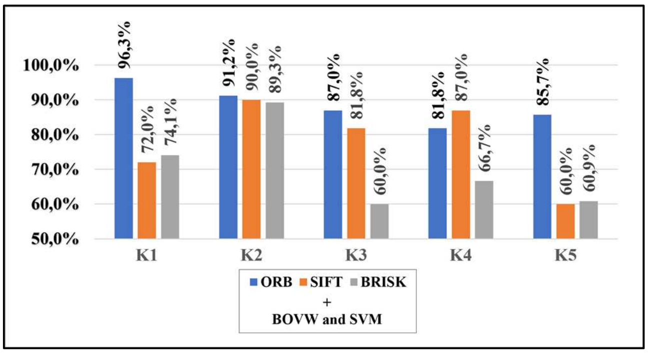

Fig. 9. F1 score comparison for each class with different feature extraction methods

IV. Conclusion

In this paper, a car model classification system based on global features is proposed with the car's front-side view image. The proposed method is ORB used to extract local features in the image and BOVW to represent global features. The combination of these two methods is then called the Bag of ORB Feature (BOF). The proposed approach using the car's front-side view image can improve the system performance with an average increase of 10%. Compared to other feature extraction methods, the proposed method works better on *K1*, *K2*, *K3*, and *K5* with F1 score of 96.3%, 91.2%, 87.0%, and 85.7%, respectively. While in *K4*, the SIFT method obtained a higher F1 score of 87.0% than the proposed method.

In the future, this research can be developed by exploring dimensionality reduction techniques to reduce features or select features that affect the classification process. In addition, large and suitable datasets with more car models can be utilized in this research area.

Acknowledgment

This research is supported by the LPPM Universitas Hasanuddin via PDUPT Ristekdikti Grant 2021 and *Artificial Intelligent Multimedia Processing* (AIMP) Research Group.

References

[1] M. A. Manzoor and Y. Morgan, "Vehicle Make and Model classification system using bag of SIFT features," 2017, doi: 10.1109/CCWC.2017.7868475.

[2] Y. Ren and S. Lan, "Vehicle make and model recognition based on convolutional neural networks," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2016, pp. 692–695, doi: 10.1109/ICSESS.2016.7883162.

[3] M. Huzaifa and I. S. Suwardi, "Car model recognition from frontal image using BRISK," *ICECOS 2017 - Proceeding 2017 Int. Conf. Electr. Eng. Comput. Sci. Sustain. Cult. Herit. Towar. Smart Environ. Better Futur.*, pp. 104–108, 2017, doi: 10.1109/ICECOS.2017.8167114.

[4] M. Awaludin and V. Yasin, "Application of Oriented Fast and Rotated BRIEF (ORB) and BruteForce Hamming in Library OpenCV for Classification," *J. Inf. Syst. Applied, Manag. Accounting, Reserarch*, vol. 4, no. 3, pp. 51–59, 2020.

[5] Indrabayu, Basri, A. Achmad, I. Nurtanio, and F. Mayasari, "Blob modification in counting vehicles using gaussian mixture models under heavy traffic," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 16, pp. 7157–7163, 2015.

[6] S. Lin, C. Zhao, and X. Qi, "Comparative analysis of several feature extraction methods in vehicle brand recognition," 2016, doi: 10.1109/ICSensT.2016.7796337.

[7] G. Tong, H. Chen, Y. Li, and K. Zheng, "Traffic sign recognition based on SVM and convolutional neural network," in *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications, ICIEA 2017*, 2018, vol. 2018-Febru, pp. 2066–2071, doi: 10.1109/ICIEA.2017.8283178.

[8] R. Rani, R. Kumar, and A. P. Singh, "Implementation of ORB and Object Classification using KNN and SVM Classifiers," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 3, pp. 280–285, 2019, doi: 10.26438/ijcse/v7i3.280285.

[9] M. Ismail, I. Amirullah, and I. Areni, "Support Vector Machine Method to Reduce the Execution Time of Vehicle Plate Recognition System," *EPI Int. J. Eng.*, vol. 1, pp. 69–75, 2018, doi: 10.25042/epi-ije.022018.11.

[10] W. Tsai, S. Lo, C. Su, and M. Sheu, "Vehicle Detection Algorithm Based on Modified Gradient Oriented Histogram Feature," *Adv. Intell. Inf. Hiding Multimed. Signal Process.*, pp. 127–134, 2017, doi: 10.1007/978-3-319-50212-0.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

[12] S. O'Hara and B. A. Draper, "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval," no. July, pp. 1–25, 2011, [Online]. Available: http://arxiv.org/abs/1101.3354.

[13] B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," *2019 2nd Int. Conf. Adv. Comput. Commun. Paradig. ICACCP 2019*, pp. 1–8, 2019, doi: 10.1109/ICACCP.2019.8882943.

[14] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756.