**NAICE Journal of Next Applied Informatics in Computer Engineering**
Vol. 1, No. 2, October 2026, pp. 1~12
ISSN: 3164-0249, DOI:


# Vehicle Image Classification Based on HOG Features with SVM and Random Forest

**Kevin Gerialdi Ekel Hagata[1], I Made Oka Wali Putra[2], Ni Putu Purnia Dewi[3], I Kadek Agus Wahyu Raharja[4]**
[1,2,3,4]Department of Computer Engineering, Universitas Warmadewa, Denpasar, Indonesia

**Article Info**

***Article history:***

Received July 19, 2026
Revised October 29, 2026
Accepted October 30, 2026

***Keywords:***

Vehicle Classification
Support Vector Machine
Random Forest
HOG
Machine Learning

**ABSTRACT**

Automatic vehicle-type classification is essential for Intelligent Transport Systems in Indonesia, where 148.21 million motor vehicles were registered in 2022. This study compares Support Vector Machine (SVM) and Random Forest (RF) for classifying bus, car, motorcycle, and truck images using Histogram of Oriented Gradients (HOG) features. A total of 891 images from the public kendaraan-indonesia dataset were converted to grayscale, resized to 64×64 pixels, and represented by 1,764-dimensional HOG descriptors. The data were split by stratified sampling into training (623), validation (134), and test (134) subsets, and 5-fold cross-validation was performed on the training subset only. SVM achieved a test accuracy of 79.85%, a macro F1-score of 0.80, an AUC of 0.9648, and a cross-validation accuracy of 78.34% (±4.98%), whereas RF achieved 75.37%, 0.75, 0.9265, and 72.72% (±3.66%). SVM was more accurate in all five folds ($p = 0.005$). Most errors involved buses and trucks, and RF was biased toward the truck class. Because the dataset contains augmented copies of 297 source images, absolute accuracies are likely optimistic, but the advantage of SVM was consistent. HOG–SVM is therefore a suitable low-cost baseline for small datasets, although HOG alone cannot reliably separate large vehicles.

***Corresponding Author:***

I Kadek Agus Wahyu Raharja
Department of Computer Engineering, Universitas Warmadewa
Email: raharja.wahyu.agus.kadek@gmail.com

## 1. INTRODUCTION

Reliable information on the types of vehicles using a road is a basic requirement for traffic management, road-maintenance planning, and traffic-law enforcement in Indonesia. The national vehicle fleet is large and dominated by two-wheelers: Statistics Indonesia (BPS) recorded 148.21 million motor vehicles in 2022, consisting of 125.27 million motorcycles, 17.18 million passenger cars, 5.53 million trucks, and 0.24 million buses [1]. The vehicle type matters because these classes place very different demands on road space and pavement, so traffic counts that are not disaggregated by type are of limited use for planning. Image-based vehicle classification is a core component of Intelligent Transport Systems (ITS), enabling automatic monitoring of vehicle types from roadside cameras and supporting traffic-volume counting and traffic-flow analysis [2], [3]. Approaches range from handcrafted visual features such as SURF [4] to deep convolutional neural networks (CNNs) [5], each with distinct advantages and limitations depending on data availability and computational resources.

The practical problem addressed in this study is how to classify vehicle types accurately when only a small, labeled dataset and modest computing resources are available, a situation that is common in local deployments that rely on low-cost cameras and edge devices. Deep CNNs achieve high accuracy for fine-grained vehicle classification, but they require large datasets and considerably longer training time [5]. Baghdadi and Aboutabit reported that HOG-based classifiers outperformed a CNN for four vehicle categories

when the dataset was small, whereas the CNN was superior only on a larger dataset [6]. Classical feature extraction combined with conventional machine learning therefore remains a relevant option under such constraints [7]. The Histogram of Oriented Gradients (HOG) represents object shape through the distribution of local intensity-gradient orientations [8], which suits vehicles whose classes differ mainly in silhouette. HOG features can be paired with a Support Vector Machine (SVM), which seeks the maximum-margin hyperplane separating the classes [9], [10], or with Random Forest (RF), which aggregates the votes of many decision trees built on bootstrap samples and random feature subsets [11].

Table 1 summarizes previous studies that combined HOG or related handcrafted features with SVM, RF, or other classifiers; the reported values are taken directly from the cited articles.

Table 1. Summary of previous studies relevant to the choice of method

| Study | Task and data | Features and classifiers | Reported result |
|---|---|---|---|
| Baghdadi and Aboutabit [6] | Vehicle category (bus, car, motorcycle, truck); 2,182 front-view and 2,000 rear-view images | HOG, LBP, Gabor + SVM, k-NN; CNN | HOG+SVM 99.58% (rear view), HOG+k-NN 99.17% (front view); HOG outperformed CNN on the small dataset; buses mainly confused with trucks |
| El-Sayed and El-Sayed [12] | Vehicle type; two vehicle image datasets | HOG, ACO + k-NN, SVM, RF, Softmax | SVM outperformed all other classifiers, including RF |
| Sugiharto et al. [13] | Traffic signs | HOG + SVM, RF, k-NN | SVM gave competitive accuracy |
| Hameed et al. [14] | Traffic signs; 6,164 images, 58 classes | HOG + SVM, k-NN, decision tree, voting | HOG+SVM 99.8%; at 3-fold validation SVM 100%, k-NN 98.2%, decision tree 98.8% |
| Yogitha et al. [15] | Breast histopathology images | SVM, RF, CNN | SVM and RF remained competitive with CNN on a limited dataset |
| Adugna et al. [16] | Land-cover mapping from satellite images | RF vs. SVM | Relative performance depended on data characteristics |
| Indra et al. [17] | BISINDO sign-language letters (Indonesia) | Shape and color features + RF | RF applicable to small Indonesian image datasets |

Table 1 shows two consistent patterns. First, when HOG is used as the descriptor, SVM is usually the best or among the best classifiers, with reported accuracies of 99.58% for vehicle categories [6] and 99.8% for traffic signs [14], and SVM outperformed RF on vehicle-type images in [12]. Second, the ranking of SVM and RF is not universal and depends on the data [16], and none of the vehicle studies used Indonesian road imagery from a small public dataset. In addition, the errors reported for vehicle categories concentrate on large vehicles, particularly buses confused with trucks [6]. These observations motivate the choice of HOG as the descriptor and of SVM and RF as two classifiers with fundamentally different decision mechanisms, a single maximum-margin boundary versus an ensemble of axis-aligned decision trees, to be compared on Indonesian vehicle images.

This study therefore aims to (i) evaluate and compare SVM and RF with HOG features on a public Indonesian vehicle dataset with four classes (bus, car, motorcycle, and truck); (ii) report a fully specified pipeline with a hold-out test set and cross-validation restricted to the training data; and (iii) analyze why one classifier performs better and which vehicle classes remain difficult. Performance is assessed with accuracy, precision, recall, F1-score, ROC-AUC, and 5-fold cross-validation.

## 2. METHOD

### 2.1. Research Workflow

The research was carried out in seven sequential stages, as illustrated in Figure 1: (1) dataset acquisition, (2) image preprocessing, (3) HOG feature extraction, (4) stratified splitting into training, validation, and hold-out test subsets (70:15:15), (5) 5-fold cross-validation on the training subset, (6) training of the final SVM and RF models on the training subset, and (7) evaluation on the hold-out test subset, followed by comparison and error analysis. Both classifiers used fixed settings, so no stage selected parameters on the basis of the test subset, and the test subset was used only once, in stage (7).

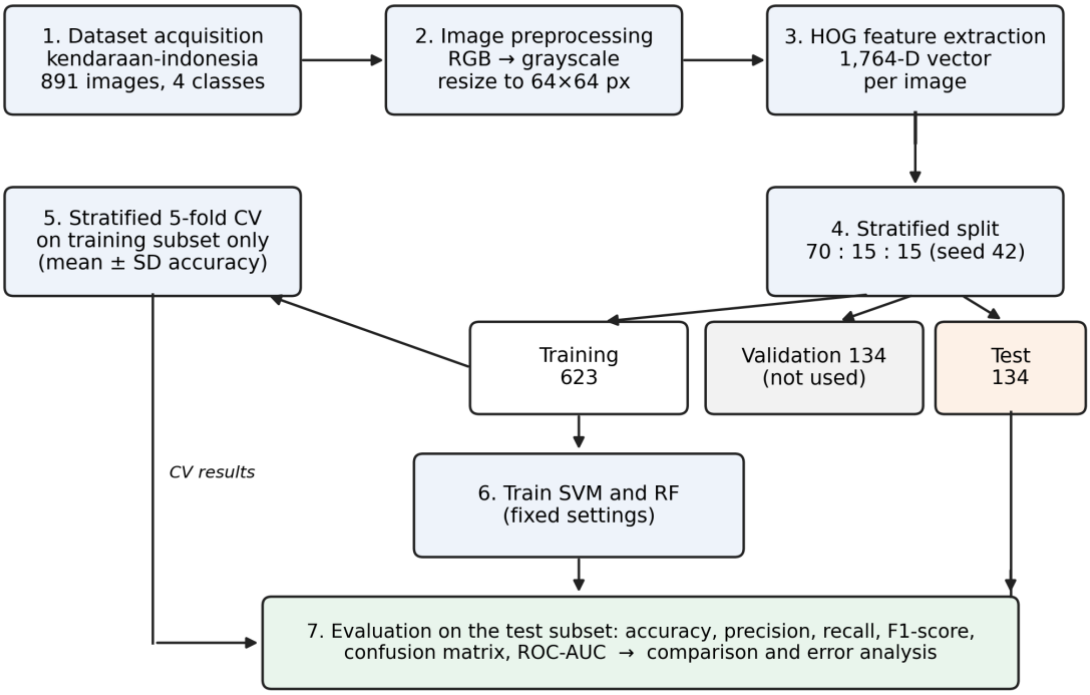



Figure 1. Research workflow

**2.2. Dataset**

The data were taken from *kendaraan-indonesia* (version 1), a public dataset on the Roboflow Universe platform released under a Creative Commons Attribution 4.0 (CC BY 4.0) license. The dataset was selected for three reasons. First, it contains images of vehicles on Indonesian roads, so the appearance of the vehicles, for example local bus and truck body styles, matches the intended application. Second, its four classes, bus, car, motorcycle, and truck, correspond to the main vehicle groups used in Indonesian vehicle statistics [1]. Third, its open license allows the experiment to be reproduced. The exported dataset consists of 1,019 images: 891 images in the training folder, 85 in the validation folder, and 43 in the test folder. According to the export documentation, every image was auto-oriented and stretched to 640×640 pixels, and the training folder was generated from 297 source images, each represented by three augmented versions (50% probability of horizontal flip, random rotation between −15° and +15°, Gaussian blur of 0–1.5 pixels, and salt-and-pepper noise on 0.1% of the pixels).

Only the 891 images of the training folder were used. The validation and test folders are too small to give stable per-class estimates (about 21 and 11 images per class on average), and a sample of several hundred images represents the small-data scenario targeted by this study, in which the number of samples is of the same order as the HOG descriptor length (1,764) and classical classifiers are expected to be competitive with deep learning [6]. The consequences of the augmented copies for the evaluation are discussed in Section 3.8. The 891 images were divided into training, validation, and hold-out test subsets as described in Section 2.5. Sample images of each class are shown in Figure 2.

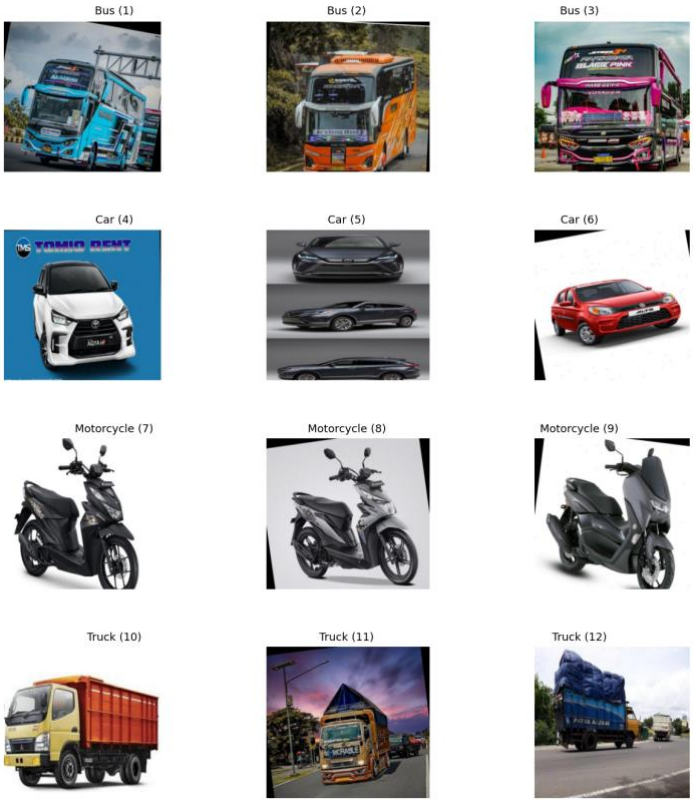



Figure 2. Sample images from each dataset class

Table 2 lists the number of images per class in the experimental data and in each subset. The totals were obtained by counting the images in each class folder, and the subset counts by counting the class labels assigned to each subset after splitting; the test counts equal the support values of the classification report in Table 5. The class names used in the dataset (*bis*, *mobil*, *motor*, and *truk*) are also given because they appear in the confusion matrices.

Table 2. Distribution of images per class

| Class | Dataset label | Total | Training (70%) | Validation (15%) | Test (15%) |
|---|---|---|---|---|---|
| Bus | *bis* | 195 | 136 | 29 | 30 |
| Car | *mobil* | 222 | 155 | 34 | 33 |
| Motorcycle | *motor* | 198 | 139 | 29 | 30 |
| Truck | *truk* | 276 | 193 | 42 | 41 |
| Total | | 891 | 623 | 134 | 134 |

Table 2 shows that the class distribution is moderately imbalanced: truck is the largest class (276 images, 31.0%), whereas bus is the smallest (195 images, 21.9%). Because the split was stratified, each subset preserves these proportions. The imbalance is considered when the per-class results are interpreted in Section 3.

### 2.3. Image Preprocessing

Each image passed through the preprocessing and feature-extraction steps shown in Figure 3. (1) The 640×640-pixel image file was read with OpenCV. (2) The image was converted to grayscale during reading (OpenCV grayscale read mode), because HOG uses only intensity gradients and the vehicle classes differ mainly in shape rather than color; this also reduces the amount of data by a factor of three. The luminance follows $Y = 0.299R + 0.587G + 0.114B$. (3) The grayscale image was resized to 64×64 pixels with bilinear interpolation (the OpenCV default), so that every image produces a HOG vector of identical length; this size is divisible by the 8×8-pixel cell size and keeps the outline of the vehicle while producing a compact descriptor. (4) The resulting 8-bit intensity array (0–255) was passed directly to HOG extraction. No histogram equalization, denoising, or further augmentation was applied, so that differences in performance reflect the classifiers rather than the preprocessing. (5) The HOG vector of each image and its class label were appended to the feature matrix X (891 × 1,764) and the label vector y.

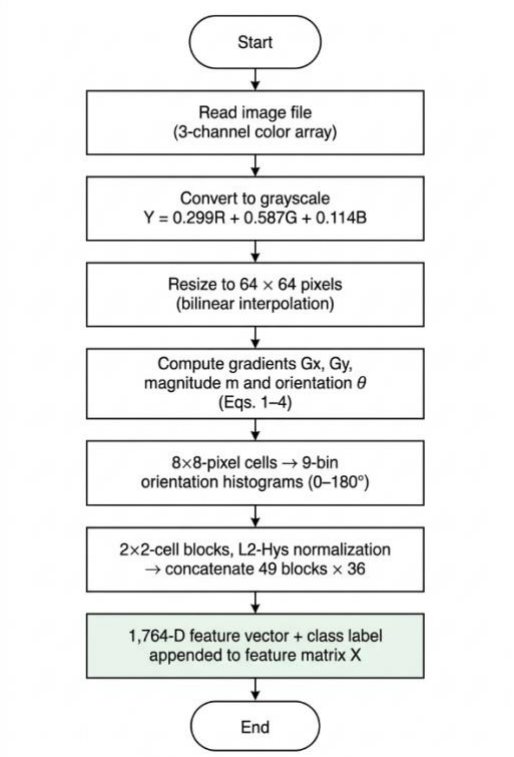



Figure 3. Flowchart of image preprocessing and HOG feature extraction for each image

**2.4. Histogram of Oriented Gradients (HOG) Feature Extraction**

Feature extraction was performed using the HOG method, which represents object shape through the distribution of pixel intensity gradient orientations [8]. For each pixel at coordinates (x, y), the horizontal and vertical gradients are computed using Equations (1) and (2).

$$G_x(x,y) = I(x+1,y) - I(x-1,y) \quad (1)$$

$$G_y(x,y) = I(x,y+1) - I(x,y-1) \quad (2)$$

where I(x, y) denote the pixel intensity at coordinates (x, y). The gradient magnitude and orientation are then computed using Equations (3) and (4).

$$m(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (3)$$

$$\theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (4)$$

The orientation is computed with the four-quadrant arctangent and mapped to the unsigned range 0°–180°. As a real calculation example, one pixel at coordinates (x = 32, y = 32) was taken from a bus-class image after grayscale conversion and resizing to 64×64 pixels. The neighboring intensities were I(x−1, y) = 138, I(x+1, y) = 111, I(x, y−1) = 165, and I(x, y+1) = 162. Equations (1) and (2) give $G_x$ = 111 − 138 = −27 and $G_y$ = 162 − 165 = −3. Equation (3) gives a magnitude of √(729 + 9) = 27.17, and Equation (4) gives arctan(−3/−27) = −173.66°, which corresponds to 6.34° in the unsigned 0°–180° representation. This pixel therefore adds a vote of 27.17 to the first orientation bin (0°–20°) of its cell histogram.

The magnitude and orientation of all pixels in a cell are accumulated into a magnitude-weighted orientation histogram, and the histograms are normalized over overlapping blocks of cells [8]. The configuration used in this study, which follows the standard configuration of Dalal and Triggs, is: 9 orientation bins of 20° over 0°–180°; cells of 8×8 pixels, giving 8 × 8 = 64 cells per image; blocks of 2×2 cells moved with a stride of one cell, giving 7 × 7 = 49 blocks; and L2-Hys block normalization (L2 normalization, clipping at 0.2, and renormalization). Each block contributes 2 × 2 × 9 = 36 values, so the final descriptor has 49 × 36 = 1,764 features. The descriptor was computed with the hog function of scikit-image; the complete configuration is listed in Table 3.

### 2.5. Data Splitting and Validation Protocol

The 891 feature vectors were divided in a 70:15:15 ratio by a two-stage stratified split with a fixed random seed (random_state = 42): first 70% of the data were assigned to the training subset (623 vectors), and the remaining 30% were then divided equally into a validation subset (134 vectors) and a hold-out test subset (134 vectors). Stratification preserves the class proportions in every subset (Table 2). Because both classifiers used fixed, predefined settings (Table 3) and no hyperparameter search was performed, the validation subset was not needed for model selection and was not used in training or evaluation. The final SVM and RF models were trained once on the 623 training vectors and evaluated once on the 134 test vectors; all test-set results in Section 3 (Tables 4 and 5 and Figures 4 and 5) were computed on these same 134 images. Since no parameter was chosen based on the test subset, the test results are independent of model selection.

Model stability was assessed by stratified 5-fold cross-validation (shuffle = True, random_state = 42) performed only on the training subset [18]; the validation and test subsets were not involved. The 623 training vectors were divided into five folds of 124–125 vectors with the same class proportions. In each round, a new model was trained on four folds (498–499 vectors) and its accuracy was measured on the remaining fold, so every training vector was used for validation exactly once. The mean and standard deviation of the five fold accuracies are reported.

### 2.6. Classification Algorithms and Implementation

Two supervised classifiers were compared. SVM was used with a linear kernel; for two classes it finds the weight vector w and bias b that minimize $\frac{1}{2}\|w\|^2 + C\Sigma\xi_i$ subject to $y_i(w \cdot x_i + b) \geq 1 - \xi_i$, where $\xi_i$ are slack variables and C controls the trade-off between margin width and training errors [9], [10]. The four-class problem was handled with the one-vs-one scheme of the scikit-learn SVC implementation, and class probabilities for the ROC analysis were obtained with Platt scaling (probability = True). RF was built with 100 decision trees, each grown on a bootstrap sample of the training data and split using the Gini impurity on a random subset of $\sqrt{1{,}764} \approx 42$ features at each node; the final class is the majority vote of the trees [11]. Table 3 lists the complete configuration and the software environment.

Table 3. HOG, classifier, and software configuration

| Component | Parameter | Value |
|---|---|---|
| Preprocessing | Color conversion / image size | Grayscale / 64 × 64 pixels (bilinear) |
| HOG | orientations | 9 (unsigned, 0°–180°) |
| | pixels_per_cell | 8 × 8 |
| | cells_per_block | 2 × 2 |
| | block_norm | L2-Hys (scikit-image default) |
| | Descriptor length | 1,764 |
| SVM | kernel / C | linear / 1.0 (default) |
| | Multi-class scheme / probability | one-vs-one / Platt scaling (probability = True) |
| | Feature scaling / random_state | None (HOG blocks are already normalized) / 42 |
| Random Forest | n_estimators / criterion | 100 / Gini (default) |
| | max_features / max_depth | √n_features (≈42) / None (defaults) |
| | bootstrap / random_state | True / 42 |
| Data split | train / validation / test | 70% / 15% / 15% (623 / 134 / 134), stratified, random_state = 42 |
| Cross-validation | StratifiedKFold | n_splits = 5, shuffle = True, random_state = 42, training subset only |
| Software | Environment and libraries | Google Colaboratory; Python 3.13, scikit-learn 1.6.1, scikit-image 0.25.2, OpenCV 5.0.0 |

### 2.7. Model Evaluation

All metrics were computed from the confusion matrix of each model on the hold-out test subset. For each class c, $TP_c$ is the number of test images of class c predicted as c, $FP_c$ is the number of images of other classes predicted as c, and $FN_c$ is the number of images of class c predicted as another class. Accuracy, precision, recall, and F1-score were computed with Equations (5)–(8), where N is the number of test images and K = 4 is the number of classes; the macro average is the unweighted mean of the per-class values.

$$Accuracy = \frac{1}{N}\sum_{c=1}^{K} TP_c \quad (5)$$

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \quad (6)$$

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \quad (7)$$

$$F1_c = \frac{2 \times Precision_c \times Recall_c}{Precision_c + Recall_c} \quad (8)$$

The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) were used to measure the ability of each model to rank the correct class above the others [19]. The labels were binarized in one-vs-rest form, and the curve was micro-averaged by pooling the 134 × 4 = 536 binary decisions using the predicted class probabilities. Cross-validation accuracy (Section 2.5) was used to measure the stability of each model across different partitions of the training data, because it provides a more reliable estimate than a single train–test split [18].

## 3. RESULTS AND DISCUSSION

### 3.1. Overall Performance Comparison

The results in this section were obtained by evaluating the final SVM and RF models, each trained once on the 623 training vectors, on the 134 images of the hold-out test subset; the cross-validation values come from the protocol in Section 2.5. Table 4 summarizes the main metrics. Accuracy is the number of correctly classified test images divided by 134, that is, the sum of the diagonal of the confusion matrices in Figure 4 (107/134 for SVM and 101/134 for RF). The macro-averaged precision, recall, and F1-score are the unweighted means of the per-class values in Table 5, the AUC is the area under the ROC curves in Figure 5, and the cross-validation values are the mean ± standard deviation of the five fold accuracies in Table 6 and Figure 6.

Table 4. Performance comparison of the SVM and Random Forest models

| Evaluation metric | SVM | Random Forest |
|---|---|---|
| Accuracy on the test subset (correct/134) | 79.85% (107) | 75.37% (101) |
| Precision (macro average) | 0.80 | 0.77 |
| Recall (macro average) | 0.80 | 0.74 |
| F1-score (macro average) | 0.80 | 0.75 |
| AUC | 0.9648 | 0.9265 |
| 5-fold CV accuracy on the training subset (mean ± SD) | 78.34% ± 4.98% | 72.72% ± 3.66% |

SVM obtained higher values than RF for all aggregate metrics in Table 4. The difference in test accuracy corresponds to six images (107 versus 101). With 134 test images, the 95% confidence interval of each accuracy (normal approximation) is about ±6.8 percentage points for SVM (73.1%–86.6%) and ±7.3 points for RF (68.1%–82.7%), so the two intervals overlap and the single test split alone cannot establish which model is better. The cross-validation results give a clearer picture: the mean accuracy of SVM exceeds that of RF by 5.62 percentage points, and SVM was more accurate than RF in each of the five folds (Section 3.5). The largest differences in Table 4 are in macro recall (0.80 versus 0.74) and F1-score (0.80 versus 0.75), and Section 3.2 shows that they are caused mainly by the bus class.

### 3.2. Per-Class Performance

Table 5 reports the precision, recall, and F1-score of each class. These values were computed from the confusion matrices in Figure 4 using Equations (6)–(8). For example, the SVM precision for bus (0.68) is the 19 correctly predicted buses divided by the 28 test images predicted as bus (19 + 1 + 0 + 8), and its recall (0.63) is 19 divided by the 30 actual buses. The Support column gives the number of test images of each class (Table 2).

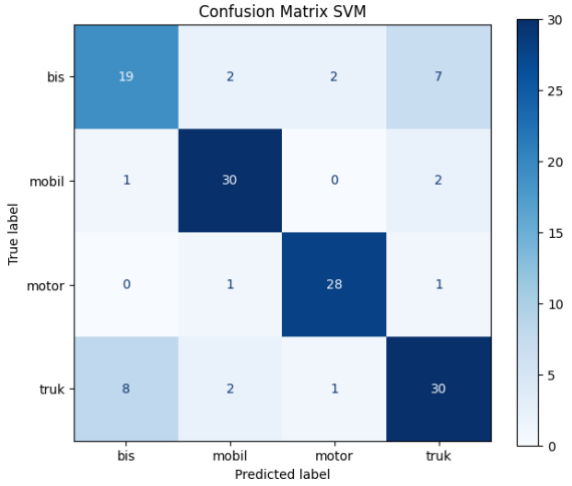
Table 5. Per-class classification report of the SVM and Random Forest models on the test subset

| Class | Precision (SVM) | Recall (SVM) | F1-score (SVM) | Precision (RF) | Recall (RF) | F1-score (RF) | Support |
|---|---|---|---|---|---|---|---|
| Bus | 0.68 | 0.63 | 0.66 | 0.78 | 0.47 | 0.58 | 30 |
| Car | 0.86 | 0.91 | 0.88 | 0.81 | 0.79 | 0.80 | 33 |
| Motorcycle | 0.90 | 0.93 | 0.92 | 0.84 | 0.87 | 0.85 | 30 |
| Truck | 0.75 | 0.73 | 0.74 | 0.66 | 0.85 | 0.74 | 41 |

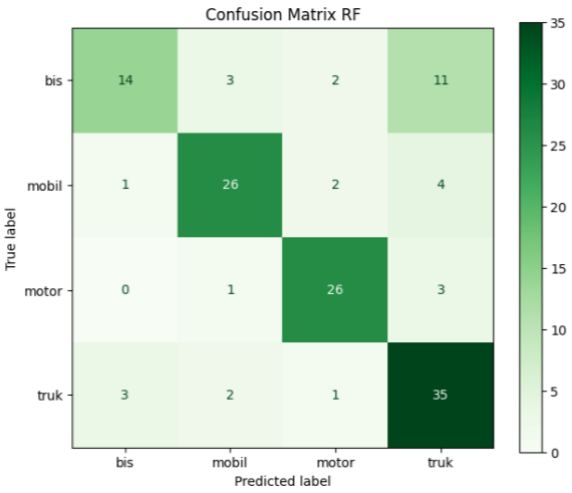
The motorcycle class achieved the best performance in both models (F1-score of 0.92 for SVM and 0.85 for RF). A motorcycle has a narrow, upright silhouette with two wheels and usually a rider, which produces a gradient pattern very different from the boxy outlines of the other three classes. The two models differ most on the bus class: RF recognized only 14 of the 30 buses (recall of 0.47), whereas SVM recognized 19 (recall of 0.63). The truck results of RF show the counterpart of this weakness: a high recall (0.85) combined with the lowest precision in the table (0.66), which means that RF assigns many non-truck images, mostly buses, to the truck class. SVM obtained a higher or equal F1-score for every class, although RF had a higher precision for bus (0.78 versus 0.68) because it predicted the bus class rarely, and the truck F1-score of the two models was the same (0.74).

### 3.3. Confusion Matrix Analysis

Figure 4 shows the confusion matrices obtained by comparing the predicted label of each of the 134 test images with its true label. Rows correspond to the true classes and columns to the predicted classes; the diagonal cells contain the correctly classified images, and the class names follow the dataset labels (*bis* = bus, *mobil* = car, *motor* = motorcycle, *truk* = truck).




(a)




(b)

Figure 4. Confusion matrices of the test subset: (a) SVM model and (b) Random Forest model

SVM classified 107 images correctly and made 27 errors: 8 trucks predicted as buses, 7 buses predicted as trucks, and 12 errors scattered over other class pairs, none with more than two images (for example, 2 buses predicted as cars and 2 cars predicted as trucks). RF classified 101 images correctly and made 33 errors, of which 11 were buses predicted as trucks, 4 were cars predicted as trucks, 3 were motorcycles predicted as trucks, and 3 were trucks predicted as buses; the remaining 12 errors were scattered. Confusions between bus and truck therefore account for 15 of the 27 SVM errors (56%) and 14 of the 33 RF errors (42%), making them the dominant error type, whereas confusions between bus and car account for only 3 SVM errors and 4 RF errors. The two models, however, err differently. The SVM errors between bus and truck are nearly symmetric (7 versus 8), which indicates genuine visual ambiguity between the two classes. The RF errors are strongly one-directional: 18 of its 33 errors (55%) are images of other classes predicted as trucks, and RF predicted the truck class 53 times for 41 actual trucks, compared with 40 times for SVM. This asymmetry indicates a systematic bias of RF toward the largest class rather than ambiguity alone.

### 3.4. Class Discrimination Analysis (ROC-AUC)

The ROC curves in Figure 5 were obtained from the class probabilities predicted by each model for the 134 test images, following the procedure in Section 2.7.

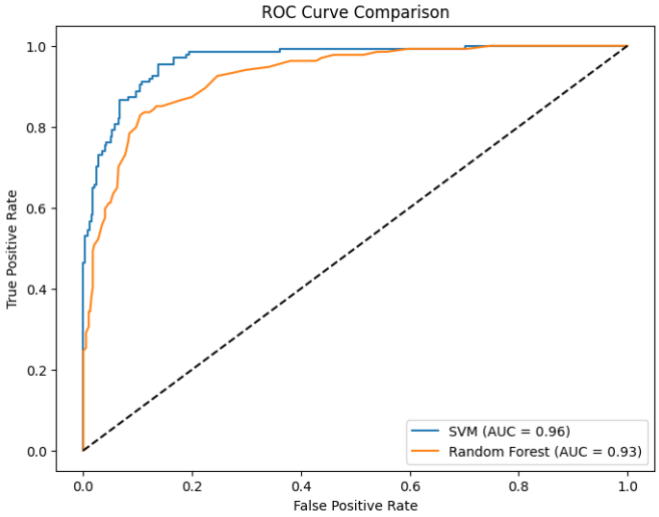



Figure 5. ROC curves of the SVM and Random Forest models on the test subset

The AUC reached 0.9648 for SVM and 0.9265 for RF. Both values are well above 0.9, which indicates good ranking ability, and SVM remained superior. The AUC values are considerably higher than the accuracies (0.80 and 0.75) because the AUC evaluates whether the correct class receives a higher probability than the incorrect ones, not whether it receives the highest probability. For RF, this suggests that many buses predicted as trucks still received a relatively high bus probability, so the bias identified in Section 3.3 could be partly reduced by class weighting or by adjusting the decision thresholds.

### 3.5. Model Stability Analysis (Cross-Validation)

Figure 6 summarizes the 5-fold cross-validation results obtained on the training subset with the protocol described in Section 2.5; the bars show the accuracy of each fold, and the horizontal lines and shaded bands show the mean ± standard deviation of each model.

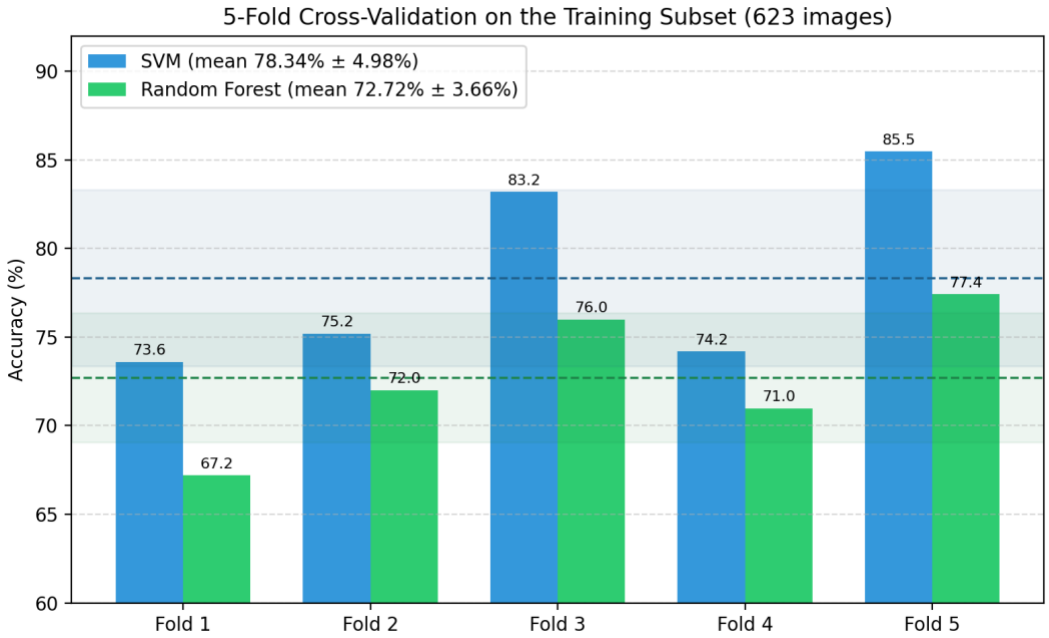



Figure 6. Results of 5-fold cross-validation on the training subset

Table 6 lists the accuracy of each fold. Each value is the number of correctly classified images in the validation fold divided by the fold size (125 images in folds 1–3 and 124 in folds 4–5); for example, the SVM accuracy in fold 1 is 92/125 = 73.60%.

Table 6. Accuracy of each cross-validation fold on the training subset

| Model | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean ± SD |
|---|---|---|---|---|---|---|
| SVM | 73.60% | 75.20% | 83.20% | 74.19% | 85.48% | 78.34% ± 4.98% |
| Random Forest | 67.20% | 72.00% | 76.00% | 70.97% | 77.42% | 72.72% ± 3.66% |
| Difference (SVM − RF) | 6.40 | 3.20 | 7.20 | 3.23 | 8.06 | 5.62 |

SVM achieved a mean cross-validation accuracy of 78.34% with a standard deviation of 4.98%, while RF achieved 72.72% with a standard deviation of 3.66%. The standard deviation measures how much the accuracy changes when the model is trained and validated on different partitions of the data; with folds of about 125 images, 4.98% and 3.66% correspond to roughly 6 and 5 images per fold, which is the variation expected from a small sample. These values have three implications. First, both models are moderately sensitive to the partition; the somewhat larger standard deviation of SVM comes mainly from folds 3 and 5, in which its accuracy exceeded 83%, and with only five folds the difference between the two standard deviations (1.32 percentage points) is not meaningful, so this study does not claim that either model is more stable. Second, the mean ± 2 SD ranges (68.4%–88.3% for SVM and 65.4%–80.0% for RF) indicate how much a single split can deviate from the average; the test accuracies (79.85% and 75.37%) lie within these ranges, 1.51 and 2.65 points above the respective cross-validation means. Third, although the fold-to-fold variation is large, the ranking of the models is not: SVM was more accurate than RF in all five folds, with differences between 3.20 and 8.06 points (Table 6). A paired t-test over the five folds gives $t(4) = 5.53$ and $p = 0.005$, so the advantage of SVM on the training data is statistically significant at the 1% level. As a supplementary check, an unshuffled stratified 5-fold cross-validation over all 891 vectors gave similar means (79.46% ± 1.88% for SVM and 75.20% ± 3.85% for RF).

### 3.6. Why SVM Outperforms Random Forest on This Dataset

The difference between the two classifiers can be explained by how each one uses the HOG descriptor. First, the data are high-dimensional relative to their size: each image is represented by 1,764 features, while the training subset contains 623 images. In this regime, a linear decision boundary with margin maximization and regularization is well suited, because it combines all features in a single weighted sum while the margin limits overfitting [9], [10]. The shape cues that separate the vehicle classes, such as long horizontal roof and window edges or the vertical boundary between the cab and the cargo box of a truck, are distributed over many weakly informative gradient bins, and a weighted sum can accumulate this evidence.

Second, RF splits on one feature at a time. At each node only about 42 randomly selected features are examined, and each tree is grown on a bootstrap sample of a few hundred images, so each split is based on a weak, noisy cue and the axis-aligned thresholds approximate an oblique class boundary only coarsely [11]. HOG features are also strongly correlated, because each cell histogram appears in up to four overlapping blocks, which reduces the diversity that RF gains from random feature selection. Third, where the bus and truck feature distributions overlap, the leaves of the trees are dominated by the more frequent truck class, so the majority vote favors truck; this is consistent with the 53 truck predictions, the bus recall of 0.47, and the known tendency of classifiers to perform worse on minority classes [20]. The SVM boundary, in contrast, is determined by the support vectors near the margin and is less affected by class frequency. These explanations agree with the studies in Table 1, in which SVM outperformed RF and other classifiers on HOG features [6], [12], [14], although RF can be superior on other data types [16].

### 3.7. Error Analysis

The dominant confusion between buses and trucks has a visual explanation. Both are large, box-shaped vehicles with flat side panels and similar front structures (windshield, grille, and headlights), a similarity that was also identified as the main source of error by Baghdadi and Aboutabit [6]. In addition, the preprocessing removes several cues that distinguish the two classes: at 64×64 pixels a row of passenger windows is reduced to a few pixels, the grayscale conversion removes the color and livery of buses, and the HOG descriptor of both classes is dominated by long horizontal and vertical edges. Because the source images were stretched to a square format before export, the characteristic length-to-height ratio of buses and trucks is also partly lost. Variations in viewing angle and lighting, together with the rotation and blur introduced by the augmentation, further weaken the edges captured by HOG. These findings show that the remaining errors are caused mainly by the limited information in the feature representation rather than by the classifiers alone.

### 3.8. Limitations

The conclusions of this study are bounded by its experimental conditions. The most important limitation concerns the dataset: the 891 images are three augmented versions of 297 source images, and the random split was performed on images rather than on source images. Augmented versions of the same source image can therefore appear in both the training and the test subsets, and in different cross-validation folds. Because such versions differ only by flipping, small rotations, blur, and noise, this overlap is likely to make the absolute accuracies of both models optimistic. It affects SVM and RF in the same way, so the relative comparison remains informative, but the accuracy values should not be read as the performance expected on new, independent vehicle images. In addition, the dataset comes from a single source, the classes are moderately imbalanced, and the hold-out test subset contains 134 images, which gives confidence intervals of about ±7 percentage points. Only one HOG configuration and fixed classifier settings were evaluated, without hyperparameter optimization, and no other baselines such as k-NN or a lightweight CNN were included. The superiority of SVM reported here should therefore be understood as holding for this dataset, feature configuration, model configuration, and evaluation protocol.

## 4. CONCLUSIONS AND SUGGESTIONS

### 4.1. Conclusion

This study compared a linear SVM and a Random Forest using HOG features for four-class vehicle classification on the *kendaraan-indonesia* dataset. Three findings emerge. First, under the dataset, HOG settings, model configurations, and evaluation protocol used here, SVM performed better than RF (mean cross-validation accuracy of 78.34% versus 72.72%; test accuracy of 79.85% versus 75.37%), and it was more accurate in every cross-validation fold ($p = 0.005$). This indicates that high-dimensional, correlated HOG descriptors are better exploited by a single maximum-margin linear boundary than by an ensemble of axis-aligned trees, so HOG–SVM is a reasonable low-cost baseline for vehicle-type classification when only a few hundred labeled images are available. Second, the absolute accuracies are probably optimistic because augmented versions of the same source images can appear in both training and test data, and the test difference corresponds to only six images; the advantage of SVM is therefore consistent in direction, but its exact size on independent images remains to be confirmed. Third, the main weakness of both models is the confusion between buses and trucks, and RF is biased toward the truck class. Because the separation of these heavy-vehicle classes is important for road-load and traffic monitoring, a system based on this pipeline can be used to separate motorcycles and cars from large vehicles, but it requires additional information to distinguish buses from trucks reliably.

### 4.2. Suggestions

Future work should address the bus–truck confusion and the limitations described in Section 3.8. The first step is to split the data by source image, or to evaluate the trained models on the dataset's separate validation and test folders, which are not augmented, so that the test images are fully independent of the training images. Further recommended steps are: using a higher input resolution or aspect-ratio-preserving resizing; combining HOG with color or texture descriptors; applying class weighting or balanced sampling for RF; optimizing hyperparameters such as the SVM regularization parameter and kernel and the number and depth of RF trees with nested cross-validation; and enlarging the dataset with additional Indonesian road images [21]. Comparisons with k-NN and lightweight CNN architectures are also recommended, keeping in mind that CNNs require larger datasets than the HOG-based approach used in this study [5], [6].

## ACKNOWLEDGMENTS

This research was successfully completed with the support of various parties. The authors would like to thank the lecturer of the Machine Learning course, for his guidance and direction throughout this research, as well as the Department of Computer Engineering, Universitas Warmadewa, for providing the facilities that supported the completion of this research.

## REFERENCES

[1] Badan Pusat Statistik, “Jumlah Kendaraan Bermotor Menurut Jenis Kendaraan (unit), 2022,” BPS-Statistics Indonesia, Jakarta, 2023. [Online]. Available: https://www.bps.go.id

[2] S. H. Tan, J. H. Chuah, C.-O. Chow, J. Kanesan, and H. Y. Leong, “Artificial intelligent systems for vehicle classification: A survey,” *Eng. Appl. Artif. Intell.*, vol. 129, p. 107497, Mar. 2024, doi: 10.1016/j.engappai.2023.107497.

[3] L. Fredianelli, S. Carpita, M. Bernardini, L. G. Del Pizzo, F. Brocchi, F. Bianco, and G. Licitra, “Traffic flow detection using camera images and machine learning methods in ITS for noise map and action plan optimization,” *Sensors*, vol. 22, no. 5, p. 1929, 2022, doi: 10.3390/s22051929.

[4] J. Trivedi, M. S. Devi, and B. Solanki, “Step towards intelligent transportation system with vehicle classification and recognition using speeded-up robust features,” *Archives for Technical Sciences*, vol. 1, no. 28, pp. 39–56, Jun. 2023, doi: 10.59456/afts.2023.1528.039J.

[5] D. ul Khairi, F. Ayaz, N. Saeed, K. Ahsan, and S. Z. Ali, "Analysis of deep convolutional neural network models for the fine-grained classification of vehicles," *Future Transportation*, vol. 3, no. 1, pp. 133–149, Jan. 2023, doi: 10.3390/futuretransp3010009.
[6] S. Baghdadi and N. Aboutabit, "View-independent vehicle category classification system," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 756–771, 2021, doi: 10.14569/IJACSA.2021.0120786.
[7] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.
[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, pp. 886–893, doi: 10.1109/CVPR.2005.177.
[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
[10] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.
[11] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
[12] R. S. El-Sayed and M. N. El-Sayed, "Classification of vehicles' types using histogram oriented gradients: comparative study and modification," *IAES Int. J. Artif. Intell.*, vol. 9, no. 4, pp. 700–712, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp700-712.
[13] A. Sugiharto *et al.*, "Comparison of SVM, random forest and KNN classification by using HOG on traffic sign detection," in *2022 6th International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, Sep. 2022, pp. 60–65, doi: 10.1109/ICICoS56336.2022.9930588.
[14] A. A. Hameed, N. A. A. Khalid, and A. I. Khaleel, "Ensemble-based traffic sign recognition using SVM, K-NN, and decision tree classifiers for ADAS dataset," *Ingénierie des Systèmes d'Information*, vol. 30, no. 7, pp. 1731–1738, Jul. 2025, doi: 10.18280/isi.300706.
[15] Yogitha, R. P, M. M, N. G. Sabhahit, and M. Ainain, "Breast histopathology image classification: A comparative evaluation of SVM, random forest and CNN," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, IEEE, Jun. 2023, pp. 1–6, doi: 10.1109/CONIT59222.2023.10205567.
[16] T. Adugna, W. Xu, and J. Fan, "Comparison of random forest and support vector machine classifiers for regional land cover mapping using coarse resolution FY-3C images," *Remote Sens.*, vol. 14, no. 3, p. 574, Jan. 2022, doi: 10.3390/rs14030574.
[17] D. Indra, L. N. Hayati, M. A. Daris, I. As'ad, and U. Mansyur, "Penerapan metode random forest dalam klasifikasi huruf BISINDO dengan menggunakan ekstraksi fitur warna dan bentuk," *Komputika: Jurnal Sistem Komputer*, vol. 13, no. 1, pp. 29–40, Mar. 2024, doi: 10.34010/komputika.v13i1.10363.
[18] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1586–1594, Aug. 2020, doi: 10.1109/TKDE.2019.2912815.
[19] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[20] K. Sharma, J. Arora, P. Kherwa, and Z. Alansari, "Majority clustering for imbalanced image classification," *PeerJ Comput. Sci.*, vol. 11, p. e2891, Jun. 2025, doi: 10.7717/peerj-cs.2891.
[21] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.