

PCANet-Based Convolutional Neural Network Architecture For a Vehicle Model Recognition System

Foo Chong Soon, *Student Member, IEEE*, Hui Ying Khaw, *Student Member, IEEE*,

Joon Huang Chuah[✉], *Senior Member, IEEE*, and Jeevan Kanesan

Abstract—Vehicle model recognition plays a crucial role in intelligent transportation systems. Most of the existing vehicle model recognition methods focus on locating a large global feature or extracting more than one local subordinate-level feature from a vehicle image. In this paper, we propose the principal component analysis network-based convolutional neural network (PCNN) and pinpoint only one discriminative local feature of a vehicle, which is the vehicle headlamp, for vehicle model recognition. The proposed model eliminates the need for locating and segmenting the headlamp precisely. In particular, PCNN ascertains the effectiveness of both principal component analysis and CNN in extracting hierarchical features from a vehicle headlamp image and also reducing the computational complexity of the traditional CNN system. To further enhance the training procedure while still keeping the discriminative property of the network, the fully connected layer is updated by backpropagation optimized with stochastic gradient descent. The proposed method is validated using a data set that comprises 13300 training images and 2660 testing images, respectively. The model is robust against various distortions. Experiments show that PCNN outperforms state-of-the-art techniques with an average accuracy of 99.51% over 38 vehicle makes and models using the PLUS data set. In addition, the effectiveness of the proposed method is also validated using the public CompCars data set, achieving 89.83% accuracy over 357 vehicle models.

Index Terms—Principal component analysis, convolutional neural network, backpropagation, optimization, stochastic gradient descent, vehicle model recognition.

I. INTRODUCTION

VEHICLE model recognition is an important task in the intelligent transportation system (ITS) which includes intelligent parking system, traffic flow analysis, traffic surveillance, and electronic toll collection. In addition, vehicle model recognition is crucial in vehicle identification, considering vehicle license plate and vehicle logo are often being illegally replaced in criminal cases. Hence, vehicle license plate location [1]–[3] and vehicle model recognition are equally

vital in providing vehicle ownership information to combat illicit activities. In addition, after knowing the model of a vehicle, its physical dimension or its type information, i.e. Sedan, Microvan, Sports Utility Vehicle (SUV), etc, can also be determined. Subsequently, the vehicle model with its type information can aid in the electronic toll collection system that charges based on vehicle type, hence optimize the traffic flow eventually. Despite the importance of vehicle model recognition in the ITS field, most of the previous vehicle recognition research focuses on vehicle license plate recognition [4], vehicle make recognition [5], vehicle type classification [6], and 3D model vehicle type classification [7], [8].

Besides the aforementioned traditional topics which have been extensively investigated, vehicle make and model recognition (VMMR) is also a challenging task and has high potential for improvement. Pearce and Pears [9] utilized Square-Mapped Gradients (SMG) or Locally Normalized Harris Strength (LNHS) as a global feature extractor onto image quadrants to classify vehicle model with Naive Bayes classifier. Although having achieved a classification accuracy of 96%, their method is still limited in complicated environment, i.e. challenging viewpoints and shadowed condition. Petrovic and Cootes [10] obtained an accuracy of 93% over 77 vehicle models by extracting features from normalized structure of frontal view vehicle images. Psyllos *et al.* [11] proposed a framework based on Scale Invariant Feature Transforms (SIFT) keypoints to extract invariant features for VMMR. However, their framework is unable to provide specific vehicle model for each vehicle image. Hsieh *et al.* [12] adopted a symmetric Speeded-Up Robust Features (SURF) as feature extractor to detect multiclass vehicle make and model in a grid-wise way. The reported average accuracy is 99.07%, but this approach suffers from different viewpoints and lighting variations. Siddiqui *et al.* [13] presented bag of Speeded-Up Robust Features (BoSURF) and a multiclass classifier support vector machine (SVM) system for VMMR, reporting the highest average accuracy of 94.84%. However, their method is still limited in wide range of illumination variations and non-frontal viewpoints. Clady *et al.* [14] designed a framework based on oriented contour points for recognizing vehicle type. Their framework achieved 93.1% accuracy over 50 vehicle models by employing three voting algorithms and a distance error. Zhang [15] studied two feature extractors, namely Gabor wavelet transform and Pyramid Histogram of Oriented Gradients (PHOG) for vehicle type classification.

Manuscript received May 30, 2017; revised December 6, 2017 and February 28, 2018; accepted April 18, 2018. This work was supported by the Ministry of Higher Education of Malaysia through the Fundamental Research Grant Scheme (FRGS) under Grant FRGS/2/2014/TK03/UM/02/6. The Associate Editor for this paper was Z. Duric. (Corresponding author: Joon Huang Chuah.)

The authors are with the Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia (e-mail: josephsoonfc@gmail.com; huiyingkhaw@gmail.com; jhchuah@um.edu.my; jjevan@um.edu.my).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2018.2833620

The classification reliability is enhanced by a second classifier ensemble, thus achieving an accuracy of 98.65%. Some of the aforementioned methods in VMMR are still hampered by the strong dependency of hand-crafted features, i.e. SIFT, which is inefficient and lacks of robustness against various distortions such as viewpoint variations, poor illumination, and noise.

Recently, some researchers have utilized deep learning techniques to extract features from low-level to high-level in the ITS domain. The high-level features are extracted based on stacked trainable stages and then classified by multiclass classifier. Dong *et al.* [16] adopted a semisupervised Convolutional Neural Network (CNN) for vehicle type classification. They introduced sparse Laplacian filters learning for the network and utilized softmax classifier, achieving accuracy of 88.11 %. Huang *et al.* [17] implemented PCA as pretraining strategy for CNN in order to improve the training procedure for vehicle logo recognition. They reported recognition accuracy of 99.07% while using 60 times lesser of initial training time. Yang *et al.* [18] implemented CNN on various viewpoints of each vehicle for vehicle model recognition. Liu *et al.* [19] proposed a Deep Relative Distance Learning (DRDL) approach which employs a two-branch deep CNNs to project raw vehicle images into an Euclidean space for vehicle re-identification purpose. Fang *et al.* [20] presented a coarse-to-fine CNN framework for fine-grained vehicle model recognition. Their framework extracted discriminative and hierarchical features from both global and local regions of vehicle images. They achieved an accuracy of 98.29% for fine-grained vehicle model recognition based on a few vehicle regions, such as whole frontal, center, left, and right parts.

Despite their outperforming result, most of these approaches depend on many vehicle parts for accurate recognition, hence, it can be extremely time-consuming. The local regional features, i.e. both front headlamps and license plates are emphasized for car model recognition due to its robustness against environmental variance [21]. Based on the result of our work, the recognition task may benefit from focusing only one side of vehicle headlamp since there is a symmetry axis at the middle part of vehicle frontal view. It is also noteworthy that vehicle headlamp inherits discriminative feature for each vehicle model. In addition, vehicle headlamp has less parameters to be computed in recognition system since its physical size is smaller than other parts, i.e. grille, bumper, hood, etc. More importantly, different from vehicle logo, headlamp is hardly and rarely replaceable thus making the vehicle recognition more relevant. Apart from that, it is often brought up to discussion that the training process of CNN takes time exponentially when a high performance Graphic Processing Unit (GPU) is not utilized for parallel computation.

To address the aforementioned issues, we leverage the merits of Principal Component Analysis (PCA) in generating convolutional filters while retaining the hierarchical structure and discriminative property of traditional CNN. In contrast to utilizing numerous global and local features of vehicle, PCANet based Convolutional Neural Network (PCNN) is able to learn significant features from only vehicle headlamp for VMMR. The feature maps learned by PCA convolutional filters are then imported into a flattened fully-connected layer

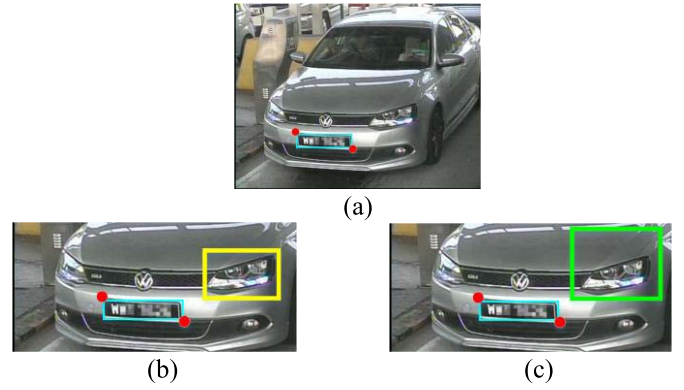


Fig. 1. The detection of car headlamp: the license plate is located, as indicated by blue rectangular box, for each original car images that are captured by the traffic monitoring camera in (a). The coarse segmentation of car headlamp as indicated by green square box in (c) detects a larger area of Region-of Interest (ROI) instead of precise segmentation of car headlamp as indicated by yellow square box in (b).

for classification. Thus, we have eliminated and skipped the exhaustive backpropagation (BP) training on the convolutional filters which are generated by PCA. Moreover, by considering only vehicle headlamp for VMMR, the computational cost is tremendously reduced without sacrificing its performance. The contributions of our work are summarized as follows:

- Proposed an end-to-end PCNN framework for vehicle model recognition, which can dramatically reduce the computational cost of the training procedure and concurrently preserving the discriminative property of CNN.
- Suggested a method which can automatically extract the hierarchical features and is robust against various distortions from vehicle headlamp image for VMMR.
- Proved that a simple yet efficient coarse segmentation technique is sufficient to accurately locate the position of vehicle headlamp. Therefore, the precise detection of global or local feature is no longer necessary for VMMR.

The rest of this paper is organized as follows. VMMR systems based on PCA and CNN are explained in Section II. In Section III, the description of dataset, experimental results and analysis are demonstrated. We conclude this work in Section IV.

II. FRAMEWORK OF PROPOSED PCANET-BASED CNN MODEL

A. Coarse Segmentation of ROI

The proposed vehicle headlamp detection system which utilizes coarse segmentation technique, is presented in Fig. 1. The original vehicle front-left view images are acquired from the traffic monitoring system as shown in Fig. 1 (a). These original images are first converted to grayscale with 8-bit resolution and then a License Plate Location (LPL) module is implemented. The LPL deploys a Sliding Concentric Window (SCW) segmentation approach, binarization, and followed by connected component labelling and also binary measurements [22]. After LPL, the upper left and bottom-right coordinates of vehicle license plate are identified as indicated by the red dots in Fig. 1. The output of the LPL module is the

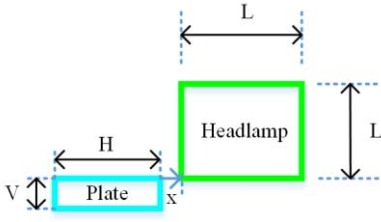


Fig. 2. Segmented area of car headlamp with size a size of $L \times L$.

license plate region with size of $H \times V$ as illustrated by the light blue rectangular box in Fig. 2.

According to Fig. 2, the region of headlamp is segmented coarsely based on the pixel coordinate at upper right corner of vehicle license plate. After LPL, the bottom left corner coordinate of headlamp is defined as the horizontal shifting of $x = 40$ pixels from the license plate's upper right corner coordinate. The larger segmented area of vehicle headlamp with size of $L \times L$ is applied to accommodate different headlamps of vehicle models in this work, as illustrated by the green box in Fig. 1(c) and Fig. 2. Hence the vehicle headlamps have a high probability of being located within the coarse segmented region. We only considered 38 vehicle models without including vehicles with larger size, i.e. suburban utility vehicle (SUV) and 4-wheel drive (4WD), in this work.

B. PCANet Based CNN (PCNN)

The structure of the PCANet based CNN is presented in Fig. 3. PCNN is an end-to-end system which its input is a coarsely segmented vehicle headlamp image with size 100×100 pixels, while its output is the prediction of vehicle model class. PCANet is able to learn the nonlinear relations from images, and hence it is robust against noises. In addition, PCANet has a low computational cost and is superior in its adaption to various conditions [23]–[25]. PCANet is an image classification network that is based on: 1) principal component analysis (PCA) for learning of multistage filter banks; 2) binary hashing; and 3) blockwise histogram. Aiming to reduce the computational cost, the proposed PCNN utilizes the PCA for learning of convolutional filters. In contrast to the PCANet, the PCNN utilizes non-linear activation, pooling layers and fully-connected layer which is able to not only extract discriminative features but also provide invariances, i.e. translation, rotation, scaling, etc. In PCNN, the inputs of each convolutional layers Conv1, Conv2, and Conv3 are headlamp images, output feature maps of Conv1, and output feature maps of Conv2, respectively.

1) *Convolutional Layer, Conv*: Assume that we have N input headlamp training images, $\{T_i\}_{i=1}^N$ of size $m \times m$, and suppose that the convolutional filter size (or patch size) is $k_s \times k_s$ at s -th stages. At first stage (or $s = 1$) or Conv1, we use a square patch size $k_1 \times k_1$ to collect every overlapping patches from i -th training image and normalize each patches by subtracting their respective patch mean. All patches are vectorized and combined into matrix form

$$\overline{X}_i = [\overline{x}_i^1, \overline{x}_i^2, \dots, \overline{x}_i^{\hat{m}\hat{m}}] \in \mathbb{R}^{k_s \times k_s} \quad (1)$$

where each \overline{x}_i^j denotes j -th vectorized and normalized patch in T_i , while $\hat{m} = m - (k_s/2)$. The mean normalization and vectorization are implemented across all input headlamp images and we obtain

$$X = [\overline{X}_1, \overline{X}_2, \dots, \overline{X}_N] \in \mathbb{R}^{k_s \times k_s} \times N\hat{m}\hat{m} \quad (2)$$

where \overline{X}_i is a mean-removed vector of image T_i .

We then compute the eigenvectors of VV^T . Suppose that the number of convolutional filters in layer s is F_s . PCA is deployed to minimize the reconstruction error while performing matrix transformation, as formulated by

$$\min_{V \in \mathbb{R}^{k_1 \times k_2 \times F_s}} \|X - VV^T X\|_F^2, \quad s.t. \quad V^T V = I_{F_s} \quad (3)$$

where I_{F_s} is the identity matrix with square size of F_s and $V = [V_1, V_2, \dots, V_m]$ is the $m \times m$ orthonormal filters. Thus F_s principal eigenvectors of XX^T are selected as PCA filters for convolutional stage as

$$W_f^s \doteq \text{mat}_{k_s, k_s}(q_f(XX^T)) \in \mathbb{R}^{k_s \times k_s}, \quad f = 1, 2, \dots, F_s \quad (4)$$

where $\text{mat}_{k_s, k_s}(v)$ is an operation to transform the vector form of $v \in \mathbb{R}^{k_s \times k_s}$ into a square size weight matrix $W \in \mathbb{R}^{k_s \times k_s}$ and $q_f(XX^T)$ represents the f -th principal eigenvector of XX^T . Hence, we retrieve the leading F_s principal eigenvectors of XX^T as our convolutional filters for the s -th stage. These extracted principal eigenvectors capture the main variance that garners maximum interpretation for the entire mean-removed training images.

In the convolutional stage, the weight filters generated by PCA are utilized to produce several output convolved feature maps at s -th stage, which are defined as $C_y^s, y = 1, 2, \dots, N \times F_s$ where y is the total number of output feature maps. The convolutional process is formulated as

$$C_y^s = R(T_i \otimes W_f^s) \quad (5)$$

where \otimes denotes 2D convolutional operation, T_i is the i -th input image or feature map, W_f^s corresponds to the f -th convolutional filter generated by PCA at s -th stage, and R is a Rectified Linear Unit (ReLU) as computed by $R(x) = \max(0, x)$. As compared to the traditional and common activation functions such as sigmoid tanh or sigmoid, the ReLU activation function provides the nonlinearity in the feature extraction process which is more effective for training purpose [26]. The convolution stages in layer Conv2 and Conv3 are similar to the Conv1, except with different size and number of convolutional filters.

2) *Pooling Layer 1, Pool1*: Pooling process is to significantly downsize the input feature maps, hence maximizing the invariance of the output to translation and rotation. More importantly, this stage will also mitigate the number of the parameters, thereby prevent overfitting and also enhance computational efficiency in the network. Each output feature map is obtained in the $(s+1)$ -th stage by max pooling process that is executed on the corresponding feature map from previous layer in the s -th stage as

$$C_y^{s+1} = \text{down}(C_y^s) \quad (6)$$

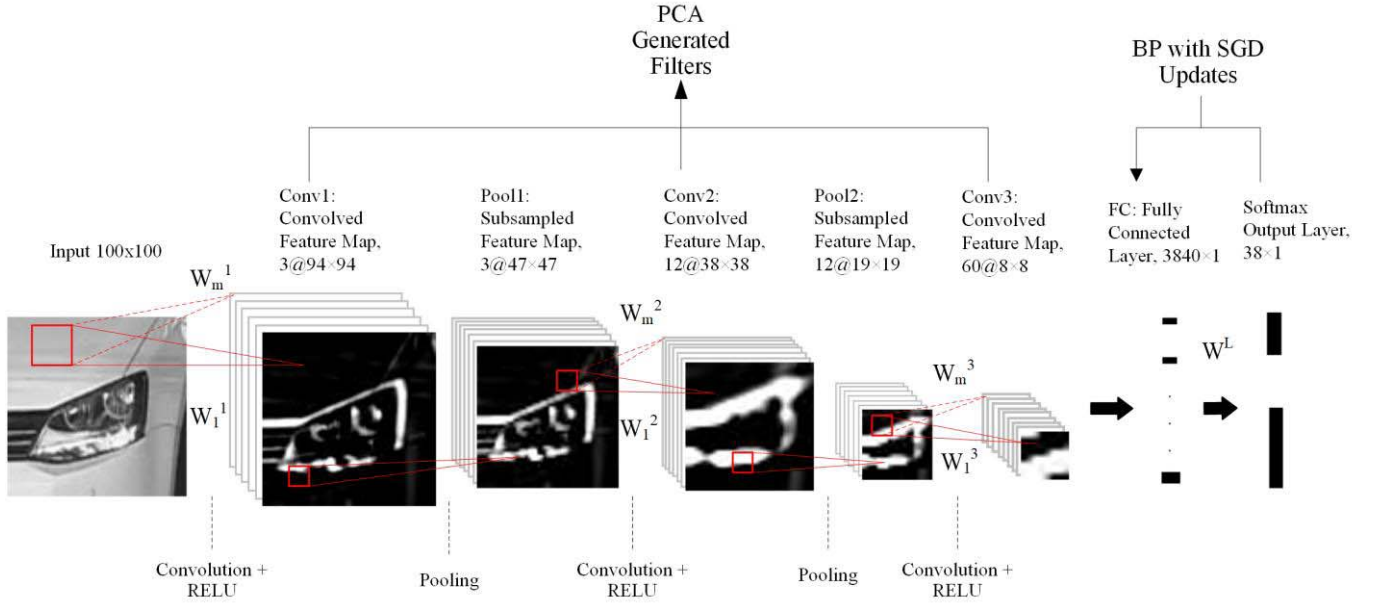


Fig. 3. Architecture of PCNN for VMMR. It primarily consists of generating convolutional filters using PCA and training fully-connected layer using BP with SGD updates. The proposed network comprises three convolutional layers, two pooling layers, and one fully-connected layer. The number in front of represents the number of feature maps, whereas the number behind of represents the size of the feature maps.

where $down(\cdot)$ is a pooling process which downsamples the input feature maps. We deploy a non-overlapping rectangular regions of dimension $(p_x \times p_y)$ for detecting and choosing maximum response over input feature maps of previous layer, where the downsampling factor, $p_x = p_y = 2$ and stride=1. Thus the size of the output feature map in this layer will become half of the size of input feature map, $m/2$ and $m/2$. Eventually multi-scale features of the vehicle headlamp images will be aggregated at the end of this process. The pooling stages in layer Pool2 will repeat the same procedure as in Pool1.

3) *Fully-Connected Layer, FC*: Linear classification is implemented at the end of fully-connected layer. We utilized softmax classifier (or multinomial logistic classifier) in the final layer of the network to classify multiple classes of vehicle headlamp or model. The feature vector with fixed dimensions, which is the output of the previous Conv3 layer, will be fed as input to the softmax classifier. The output of softmax classifier produces the probabilities for each headlamp images correspond to its vehicle model classes, in which the one with the highest value is equivalent to the predicted vehicle model class. To generate output probabilities for multi-class vehicle model classification problem, the softmax classifier which its response variable h can take on different R values, can be generalized as

$$P(h = r|x; W^L) = \frac{\exp(w_r^T x)}{\sum_{i=1}^R \exp(w_i^T x)} \quad (7)$$

where h is the vehicle model class label, $x \in \mathbb{R}^{(K+1) \times 1}$ is the K dimensional feature vector, $w \in \mathbb{R}^{(K+1) \times 1}$ denotes the weight vector, and $W^L = [w_1, w_2, \dots, w_N] \in \mathbb{R}^{(K+1) \times N}$ is the weight parameters at the fully-connected layer with each w_i corresponds to different category of weight parameters.

4) *Cross Entropy Loss Function*: To measure the error between the actual class and the predicted class that is enumerated at the softmax layer, we utilize the loss function, $E(W)$ as

$$E(W) = - \left[\sum_{i=1}^N \sum_{r=1}^R 1\{h^{(i)} = r\} \log \frac{\exp(w_r^T x^{(i)})}{\sum_{j=1}^R \exp(w_j^T x^{(i)})} \right] \quad (8)$$

where $1\{\bullet\}$ denotes the indicator function, so that $1\{a \text{ true class}\} = 1$ or $1\{a \text{ false class}\} = 0$. We have added a regularization term (or weight decay term), D into Eq.(8) in order to prevent over-fitting problem, which is given by,

$$D = \frac{\lambda}{2} \sum (W^L)^2 \quad (9)$$

where W^L is the weight parameter at the fully-connected layer and λ is the weight decay parameter and we use $\lambda = 10^{-5}$ in our experiment.

C. Backpropagation and Optimization Technique

We make use of backpropagation (BP) to minimize the error or loss function by only updating weight parameters at the fully-connected layer, W^L and its bias, B^L . The error or delta which we propagate backwards via our network can be related as sensitivities of each unit or node with respect to perturbations of the parameter at the fully-connected layer. It is given by

$$\nabla_{W^L} E(W) = - \sum_{i=1}^N 1\{h^{(i)} = r\} - P(h^{(i)} = r|x^{(i)}; W^L) \quad (10)$$

where $\nabla_{W^L} E(W)$ denotes the gradient in response to the weight parameter, W^L at the fully-connected layer. Stochastic Gradient Descent (SGD) optimization technique is implemented to update the weight parameter of fully-connected layer in every iteration. All weights and biases are initialized randomly based on the normalized initialization of $U\left[-\left(\sqrt{6}/\sqrt{n_i+n_{i+1}}\right), \left(\sqrt{6}/\sqrt{n_i+n_{i+1}}\right)\right]$ by LeCun *et al.* [27], where n_i and n_{i+1} are the number of neurons in layer i and $i+1$ respectively. BP computes the gradient while SGD updates the weight parameter using a minibatch of training images. The new update in every iteration is given by,

$$W_{i+1}^L = W_i^L - \alpha \nabla_{W^L} E(W) \quad (11)$$

where α is the learning rate. The weight parameter update is calculated by using the cost and gradient with respect to a training set with minibatch size. In our approach, we use an initial $\alpha = 0.01$ and minibatch size of 256 images for each SGD iteration. The utilization of minibatch in SGD minimizes the variance in the parameter update and thereby achieving more stable convergence during training process. In order to further speed up the rate of convergence and avoid the problem of being trapped into local minima while training by SGD, a momentum approach is introduced into our model. A momentum fraction γ of the update vector is incorporated into the current SGD update which leads Eq.(11) to

$$v_{i+1} = \gamma v_i + \alpha_i \nabla_{W^L} E(W)_i \quad (12)$$

where α denotes the learning rate at the i -th iteration and γ is set to 0.9. The new update of the weight and bias parameter is then defined as

$$W_{i+1}^L = W_i^L - v_{i+1} \quad (13)$$

and

$$b_{i+1}^L = b_i^L - v_{i+1} \quad (14)$$

respectively, where v_i denotes the i -th iteration of velocity vector which is of the same dimension as the weight parameter, W^L , and $\nabla_{W^L} E(W)_i$ is the derivative or gradient of the loss function computed at i -th iteration with respect to W^L . The learning rate α_{i+1} is annealed in each iteration as

$$\alpha_{i+1} = \alpha_i \cdot (1 + 0.0005 \cdot i)^{-1} \quad (15)$$

where α_i is the learning rate at the i -th iteration. After combining all the aforementioned steps, we have our optimized parameters which are updated and obtained by the algorithm for PCNN as stipulated in Algorithm 1.

The time needed in training procedure of traditional CNN grows exponentially due to the parameters updating process by backpropagation training. Numerous works have been

conducted by [23], [24], and [28] to research the effectiveness of using PCANet for image classification. Hence in our experiment, we have implemented the strategy of PCANet in generating filters of convolutional stage only. Aiming to leverage the discriminative topology of CNN and speed up the training procedure simultaneously, we initialize the weight parameters of fully-connected layer and then update its parameters iteratively. We validate the testing dataset for

Algorithm 1 PCANet Based CNN (PCNN)

```

function PCNN (T, Wconv, WL, bL, α, γ)
    Generate Wconv using PCA
    Initialize WL and bL randomly
    repeat
        Calculate E(W) in function (8)
        Calculate ∇WL E(W) in function (10)
        repeat
            Update WL using SGD in function (13)
            Update bL using SGD in function (14)
        until function (8) is minimized
    until reaching total number of iterations
    Return (WL, bL)
end function

```

each epoch in order to seek for the after-updated network with optimal classification performance. As shown in Fig. 4, the training procedure of the network begins to converge at the 101th iterations. In fact, our model only requires 36 minutes to train the PCNN until it achieves its highest recognition accuracy at the 3351th iteration, as illustrated in Fig. 4. All experiments were performed in MATLAB 2015 on a desktop with an Intel Core i5 central processing unit and 8-GB random access memory, without using Graphic Processing Units (GPU).

III. EXPERIMENTS

A. Datasets

We evaluated the effectiveness and robustness of our proposed model based on dataset generated from PLUS Malaysia North-South Expressway (NSE). The dataset incorporates vehicle headlamps from the latest top 38 most common vehicle models in Malaysia. A total of 15960 outdoor vehicle images were captured from surveillance cameras located at the electronic road pricing toll system. As illustrated in Fig. 3, the image comprises some useful vehicle information, i.e. vehicle license plate, vehicle make logo, and headlamp details. We ascertained the unique feature of the headlamp from the left front-side for classification of vehicle model in our proposed system. The cars in the acquired images are classified into 38 car models based on the left front-side headlamp.

The proposed system has implemented the coarse segmentation technique to locate and detect the region-of-interest (ROI) which is the headlamp region. For this purpose, the car license plate has been initially detected using Sliding Concentric Window (SCW) approach. Using license plate as a reference point, we then located and segmented the region of left frontal-side headlamp. The vehicle headlamp images were converted into 100×100 pixels in the grayscale format and then normalized. The original images captured by the camera exhibit various outdoor imaging conditions such as illumination. In order to validate the robustness of the proposed system, we have further augmented the dataset by introducing various distortions, i.e. translation, rotation, and noise, into the images.

A training dataset with 13300 vehicle headlamp images and a testing dataset with 2660 vehicle headlamp images belonging

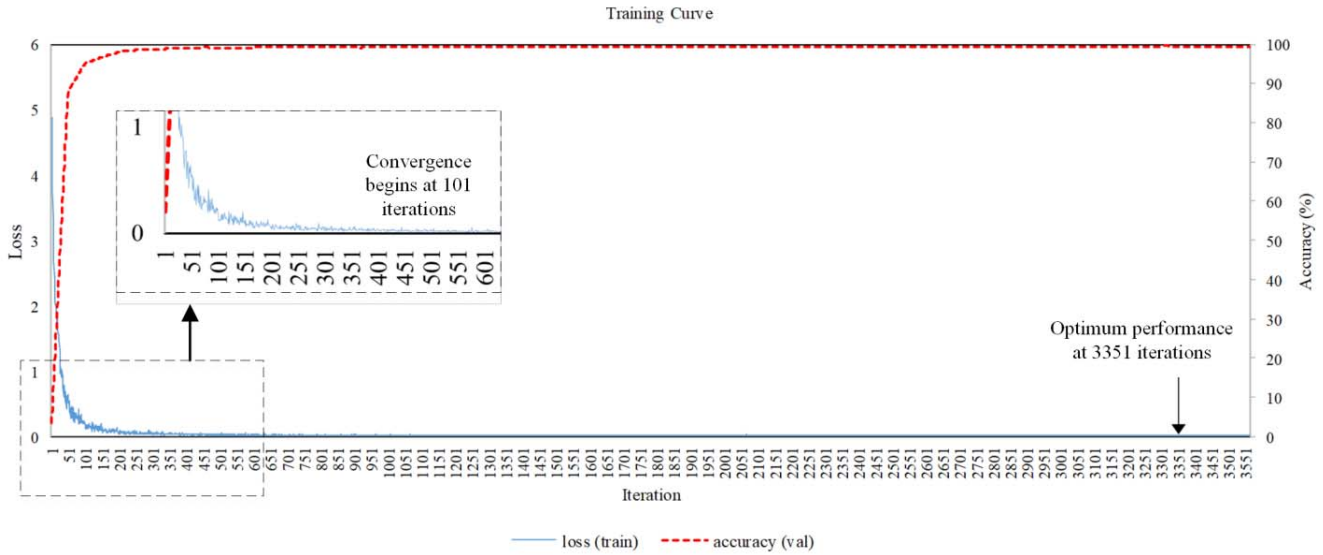


Fig. 4. Training curve of vehicle headlamp images using PCNN model. The network begins to converge after 101 iterations.



Fig. 5. Sample of original image.



Fig. 6. Headlamp images of similar model name, Almera but in different year of release, which (a) year of 2013 and (b) year of 2015 (facelift).

to 38 vehicle model classes were generated. These 38 classes consist of 11 different vehicle makes (or manufactures) that are commonly available on the road. Some similar vehicle models are considered to be different classes as they were released in different years, i.e. Nissan Almera released in year 2013 as shown in Fig. 6 (a), and its facelift model released in year 2015 as illustrated in Fig. 6 (b).










A total of 38 classes of vehicle model were collected in this work for performance evaluation, namely, Ford Fiesta, Nissan Almera N17, Nissan Almera N17 Facelift, Nissan Livina L10, Nissan Sylphy G11, Peugeot 207, Volkswagen Jetta, Volkswagen Polo, Suzuki Swift Second Generation, Suzuki Swift Third Generation, Honda Accord Eight Generation, Honda City Fourth Generation, Honda City Fifth Generation, Honda City Sixth Generation, Honda Civic FD, Honda Civic FB4, Honda Jazz Third Generation, Toyota Altis E140, Toyota Avanza F600, Toyota Camry XV40, Toyota Vios NCP42, Toyota Vios NCP93, Toyota Vios NCP150, Proton Exora, Proton Iriz, Proton Persona CM6, Proton Preve P3-21A,

Proton Saga BLM, Proton Saga FLX, Perodua Alza, Perodua Axia, Perodua Bezza, Perodua Myvi First Generation, Perodua Myvi Second Generation, Perodua Myvi Third Generation, Perodua Viva, Hyundai Elantra MD, and Kia Cerato K3, as summarized in Table I.

B. Result and Analysis

A PCANet based CNN (PCNN) was proposed to evaluate the performance of vehicle model recognition system. In the proposed system, we have selected the number and size of convolutional kernel from the optimized performance via multiple experiments. The vehicle model classification accuracy obtained is based on the 100% correctly segmented vehicle headlamp region from the vehicle image. According to Table I, the average accuracy of vehicle model recognition is 99.51%. Our proposed method works well for the left front-side view which utilizes the vehicle headlamp as ROI. There are only a total of 13 incorrect recognized images which belong to these classes such as Honda Jazz, Toyota Avanza, Toyota Vios NCP42, Toyota Vios NCP150, Proton Exora, Proton Preve, Proton Saga FLX, Perodua Myvi First Generation, and Perodua Myvi Second Generation. A comparison with the most recent and relevant works are tabulated in Table II. Our proposed model has outperformed other reported works for recognizing vehicle model even only one feature or ROI (headlamp) is utilized. The training procedure and testing procedure of our system are 36 minutes and 123 milliseconds per image, respectively. Hence, the computational cost of our proposed method is low and very efficient for real-time applications. The low computational cost is contributed by the utilization of coarse-segmented vehicle headlamp since vehicle headlamp is discriminative enough to represent distinct vehicle model. In addition, the advantage of utilizing vehicle headlamp as local feature is that only a few parameters need to be computed in the system in lieu of other extracted most discriminative features and also further reduced the computational cost of the

TABLE I
PERFORMANCE ACCURACY OF VMRR BY PCNN

Make	Ford	Nissan				Peugeot	Volkswagen		Suzuki		Hyundai
Model											
Year/Generation	B299	N17	N17 Facelift	L10	G11	2009	MK6 GP	MK5 CKD	Second Gen.	Third Gen.	MD
True	70	70	70	70	70	70	70	70	70	70	70
False	0	0	0	0	0	0	0	0	0	0	0
Accuracy (%)	100	100	100	100	100	100	100	100	100	100	100
Make	Honda							Toyota			
Model											
Year/Generation	Eighth Gen.	Fourth Gen.	Fifth Gen.	Sixth Gen.	FD	FB4	Third Gen.	E140	F600	XV40	NCP42
True	70	70	70	70	70	70	68	70	69	70	69
False	0	0	0	0	0	0	2	0	1	0	1
Accuracy (%)	100	100	100	100	100	100	97.14	100	98.57	100	98.57
Make	Toyota		Proton					Perodua			
Model											
Year/Generation	NCP93	NCP150	2015	2015	CM6	P3-21A	BLM	FLX	2014	2014	2016
True	70	68	69	70	70	68	70	69	70	70	70
False	0	2	1	0	0	2	0	1	0	0	0
Accuracy (%)	100	97.14	98.57	100	100	97.14	100	98.57	100	100	100
Make	Perodua				Kia		Average Accuracy				
Model											
Year/Generation	First Gen.	Second Gen.	Third Gen.	2014	K3	K3					
True	69	68	70	70	70	70					
False	1	2	0	0	0	0					
Accuracy (%)	98.57	97.14	100	100	100	100					

system. It should also be emphasized that the proposed vehicle model recognition is implemented without using GPU for both training and testing procedure. The experimental results demonstrate that the proposed model is suitable for real-time applications.

C. Discussion

In order to validate the robustness of our proposed model, we have further introduced some distortions into 1330 testing images, i.e. shifting, rotation, noise, and illumination. These distorted images are of poorer quality than those from the real-time public traffic. The distortion level of every experiments are from low to high including the extreme situations.

1) *Robustness Against Shifting*: We have further shifted the after coarse segmented headlamp images as illustrated

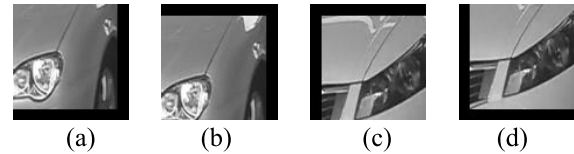


Fig. 7. Horizontal & vertical shifting of (a) $-10, -10$. (b) $-10, 10$. (c) $10, 10$. (d) $10, -10$ pixels.

in Fig. 7. Half of the testing dataset is shifted in either the horizontal direction or vertical direction, or both. As presented in Table III, the recognition accuracy within 5 pixels of shifting remains above 90%. The accuracy starts to decrease significantly when the shifting exceeds 10 pixels. The accuracy is above 70% for all cases except when the shifting is 10 pixels in

TABLE II
COMPARISON OF STATE-OF-THE-ART PERFORMANCE

Methods, Year	#Class	#Features/ ROIs Needed	Computational Cost	Accuracy (%)
Petrovis and Cootes [10], 2004	77	1 (whole front)	-	93.00
Lee [29], 2006	24	1 (whole front)	-	94.00
Clady [14], 2008	50	1 (whole front)	-	93.10
Psyllos et al. [22], 2011	11	1 (whole front)	1) Training: 1428 ms 2) Testing: 793 ms/ image	88.59
Zhang [15], 2013	21	1 (whole front)	-	98.65
Hsieh et al. [12], 2014	29	1 (whole front)	-	97.96
Llorca et al. [30], 2014	52	1 (whole rear)	-	92.21
He et al. [21], 2015	30	1 (whole front)	-	92.47
Fang et al. [20], 2016	281	4 (whole front)	1) Training: 40 minutes (with GPU)	98.29
This work	38	1 (left front-side headlamp)	1) Training: 36 minutes (without GPU) 2) Testing: 123 ms/ image (without GPU)	99.51

TABLE III
ROBUSTNESS TO SHIFTING IN HORIZONTAL AND VERTICAL DIRECTION

Horizontal Shift (Pixel) \ Vertical Shift (Pixel)	-10	-5	0	5	10	
-10	70.11	80.34	83.31	82.22	72.67	Accuracy (%)
-5	83.23	92.48	96.65	95.68	88.38	
0	88.76	97.67	99.51	98.20	89.21	
5	86.77	96.43	97.37	92.41	81.17	
10	73.16	82.33	84.74	78.65	68.46	

both horizontal and vertical direction. As shown in Fig. 7 (c), the shifting of 10 pixels can cause the headlamp to lose important edges information. Since the edges are of the unique features for each vehicle model class, losing a little part of it will sacrifice the classification performance. Therefore, it can be observed that images with complete shape or edges of headlamp images yield better performance.

2) *Robustness Against Scaling*: Similar to the previous experiment, 1330 headlamp images are randomly selected from the testing dataset. In this experiment, different scaling factors at the range from 0.6 to 1.4 are implemented on the randomly selected testing images.

Examples of some scaled headlamp images are presented in Fig. 9. The graph of average accuracy with different scaling factors is also plotted in Fig. 8. The classification accuracy drops significantly when the scaling factors are below 0.7 and above 1.4. It can be observed that the proposed method is robust against scaling which its accuracy stays above 80% when its scaling factors are between 0.8 and 1.3.

3) *Robustness Against Rotations*: After coarse segmentation of vehicle headlamp, 1330 testing images from 38 vehicle model classes are then rotated within -20° to 20° , as illustrated in Fig. 11. The degrees of rotation have simulated most of the extreme cases in traffic monitoring camera. The average recognition accuracy across every testing images is shown in Fig. 10. Our proposed model has achieved accuracy of above 81% within -15° to 15° of rotation degrees.

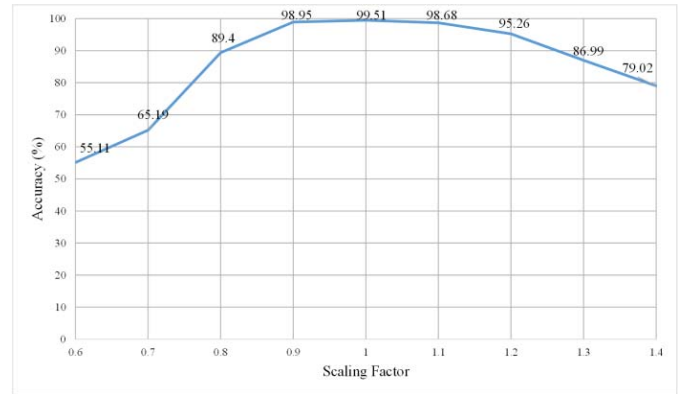


Fig. 8. Robustness against scaling.

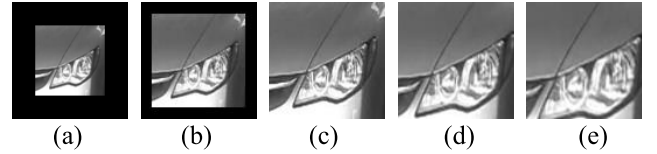


Fig. 9. Headlamp images with scaling factors of (a) 0.6, (b) 0.8, (c) 1, (d) 1.2 (e) 1.4.

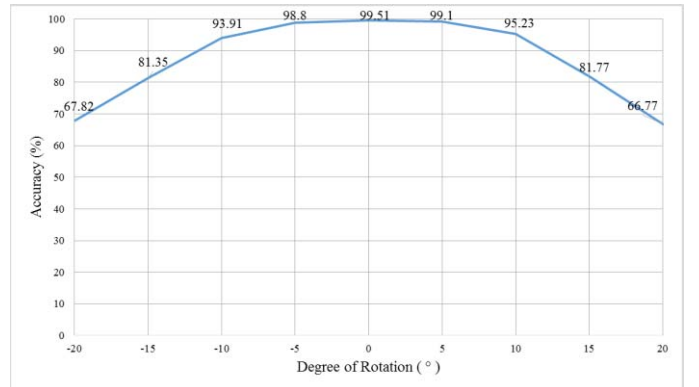


Fig. 10. Robustness against rotations.

The accuracy decreases significantly when the rotation degree exceeds $\pm 15^\circ$. It can be observed that the proposed model is robust against various rotation degrees within -15° to 15° .

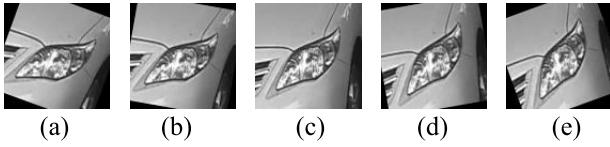


Fig. 11. Headlamp images with rotation degree of (a) -20° , (b) -10° , (c) 0° , (d) 10° (e) 20° .

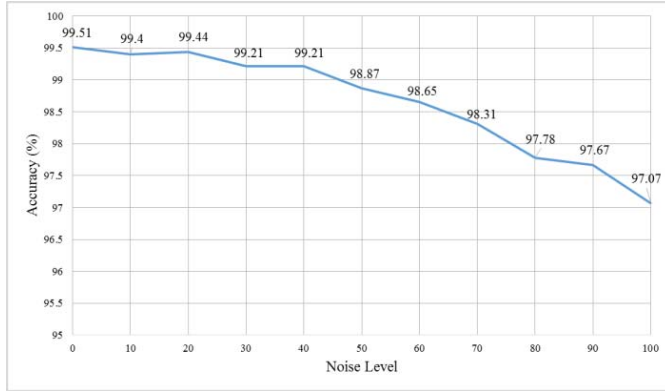


Fig. 12. Robustness against noise.

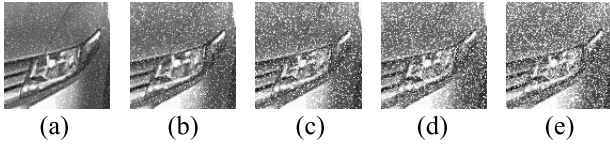


Fig. 13. Examples of headlamp images degraded with Gaussian noise level of (a) 20, (b) 40, (c) 60, (d) 80, (e) 100.

4) *Robustness Against Noises*: Due to various imaging conditions, 1330 testing images are randomly selected from the testing dataset and then introduced with Gaussian noise. Different levels of Gaussian noise are introduced into the headlamp images in order to validate the robustness of our proposed model. We have introduced Gaussian noise within noise level, σ of 10 to 100 into the randomly selected images using method by Zhang *et al.* [31]. The classification accuracy slightly decreases when the noise level increases from 10 to 100. The average accuracy stays above 97% across all the noise levels as presented in Fig. 12. Examples of headlamps images degraded by Gaussian noise are illustrated in Fig. 13. Some of the images are seriously degraded by noise which are hardly recognizable by observers, as shown in Fig. 13 (e).

5) *Exceptional Consideration*: To further validate the generalization of the system under various lighting conditions, some images with vehicle headlamp in operating mode are also investigated as illustrated in Fig. 14 (a) and (b). These images are taken in low light conditions by the traffic monitoring camera. We have tested these images using the proposed model without any further preprocessing process or illumination conversion. Although the headlamp is in operating mode, our proposed model can still recognize the vehicle model based on the headlamp images. The unusual cases are explicitly shown in the Incorrect Classification column

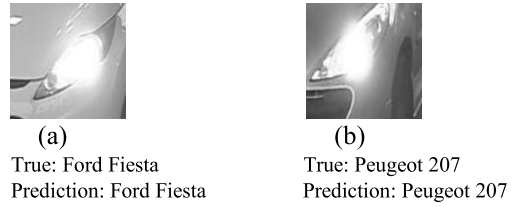


Fig. 14. Sample headlamp images of prediction results.

TABLE IV
ANALYSIS OF WRONGLY CLASSIFIED HEADLAMP IMAGES





Incorrect Classification	Correct Classification
 True: Proton Exora Prediction: Proton Preve	 True: Proton Exora Prediction: Proton Exora
 True: Perodua Viva Prediction: Perodua Myvi Third Gen.	 True: Perodua Viva Prediction: Perodua Viva

TABLE V
COMPARISONS OF PCNN WITH CNN

Methods	Training Procedure	GPU Computation	Accuracy
CNN	1 hour	With	98.68%
PCNN	36 minutes	Without	99.51%

of Table IV, in which the headlamps are slightly different from their original appearances. Thus, we presume that these headlamps might have been self-modified or customized by the vehicle owner. As the most significant features learned by our model are the headlamp structures and edges, the result shows that it will be somehow affected by the intractable physical modification on headlamp.

6) *Result Comparisons*: We have also applied the CNN method of MatConvNet [32] to our dataset. As shown in Table V, the proposed PCNN method outperforms the CNN method with 99.51% over 98.68%. It is also worth noting that the accuracy of the proposed PCNN is achieved without being implemented in parallel on GPU which is in contrast to the CNN run. It is obvious that the PCNN reduces computational time and also enhances the classification accuracy simultaneously.

To validate the effectiveness of our proposed method, we implement PCNN on the vehicle surveillance-nature CompCars dataset [18]. It is realized that a car model is grouped under a same class regardless of its year of release in CompCars dataset. In fact, headlamps of a same model but released at different years might have different physical structure and features. In other words, a same class of model might inherit more than one headlamp designs. Moreover, the functionality

TABLE VI

COMPARISON OF THIS WORK WITH OTHER STATE-OF-THE-ART METHODS USING SURVEILLANCE-NATURE COMPCARS DATASET

Methods	#classes	Accuracy
Zhang[15]	281	83.78%
Hsieh et al.[12]	281	51.70%
Fang et al.[20]	281	98.29%
This work	357	89.83%

of our proposed recognition method which is based on headlamp classification is rather sensitive to the headlamp features. Hence we have further grouped the initial vehicle model classes from 281 to 357 according to the vehicle make, model, and its year of release. Fang *et al.* [20] has implemented the methods of Zhang [15] and Hsieh *et al.* [12] on the CompCars dataset and the results are presented in Table VI. In comparison with other state-of-the-art methods, our proposed method achieved an accuracy of 89.83% from 357 vehicle models. The proposed method handles the highest number of classes in addition to achieving promising result. More importantly, this brings to another major achievement of this work which is its ability to classify not merely vehicle make and model (or inter-model) but also subtlety of its year of release (or intra-model variation) based on vehicle headlamp features. To the best of our knowledge, this in-depth analysis has not been addressed by any other work in the literature.

IV. CONCLUSION

A PCANet based CNN (PCNN) has been proposed for vehicle model recognition. Our proposed method eliminates the need of precise detection and segmentation of local feature from vehicle images. Discriminative features that are extracted automatically via the hierarchical PCNN network are robust against various shifting, rotations, and noises. The generation of convolutional filters by PCA and the utilization of only one significant local feature which is the vehicle headlamp, have greatly reduced the computational cost of the training procedure and at the same time achieved exceptional results. Our proposed system has been experimented based on the real-time traffic monitoring dataset which comprises the latest and most common vehicle models in Malaysia. To validate the robustness of the proposed system, the testing dataset is further introduced with various distortions, i.e. shifting, rotation, and noise. In addition, our system is also evaluated on the comprehensive CompCars dataset, and achieved exceptional results. Furthermore, the parallel implementation of GPU with the proposed system will greatly reduce the computational cost and make it more suitable for real-time application. In the future work, we will focus on a more challenging dataset with various viewpoints and distortions.

ACKNOWLEDGMENT

The authors would like to thank PLUS Malaysia, the builder of the North-South Expressway (NSE), for providing real-time traffic data in this work.

REFERENCES

- [1] M. Wafy and A. M. M. Madbouly, "Efficient method for vehicle license plate identification based on learning a morphological feature," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 389–395, Aug. 2016.
- [2] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, Nov. 2005.
- [3] Q. Lu, W. Zhou, L. Fang, and H. Li, "Robust blur kernel estimation for license plate images from fast moving vehicles," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2311–2323, May 2016.
- [4] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, Sep. 2006.
- [5] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a SIFT-based enhanced matching scheme," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 322–328, Jun. 2010.
- [6] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1185–1192.
- [7] M. J. Leotta and J. L. Mundy, "Vehicle surveillance with a generic, adaptive, 3D vehicle model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1457–1469, Jul. 2011.
- [8] Z. Zhang, T. Tan, K. Huang, and Y. Wang, "Three-dimensional deformable-model-based localization and recognition of road vehicles," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 1–13, Jan. 2012.
- [9] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," in *Proc. 8th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2011, pp. 373–378.
- [10] V. S. Petrovic and T. F. Coates, "Analysis of features for rigid structure vehicle type recognition," in *Proc. BMVC*, 2004, pp. 1–10.
- [11] A. Psyllos, C. Anagnostopoulos, and E. Kayafas, "SIFT-based measurements for vehicle model recognition," in *Proc. 19th IMEKO World Congr. Fundam. Appl. Metrol.*, 2009, pp. 2103–2108.
- [12] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 6–20, Feb. 2014.
- [13] A. J. Siddiqui, A. Mammeri, and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of SURF features," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3205–3219, Nov. 2016.
- [14] X. Clady, P. Negri, M. Milgram, and R. Poulenard, "Multi-class vehicle type recognition system," in *Artificial Neural Networks in Pattern Recognition*. Berlin, Germany: Springer, 2008, pp. 228–239.
- [15] B. Zhang, "Reliable classification of vehicle types based on cascade classifier ensembles," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 322–332, Mar. 2013.
- [16] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015.
- [17] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1951–1960, Aug. 2015.
- [18] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3973–3981.
- [19] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2167–2175.
- [20] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 78, pp. 1782–1792, Jul. 2017.
- [21] H. He, Z. Shao, and J. Tan, "Recognition of car makes and models from a single traffic-camera image," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3182–3192, Dec. 2015.
- [22] A. Psyllos, C.-N. Anagnostopoulos, and E. Kayafas, "Vehicle model recognition from frontal view image measurements," *Comput. Standards Interfaces*, vol. 33, no. 2, pp. 142–151, Feb. 2011.
- [23] F. Gao, J. Dong, B. Li, and Q. Xu, "Automatic change detection in synthetic aperture radar images based on PCANet," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1792–1796, Dec. 2016.

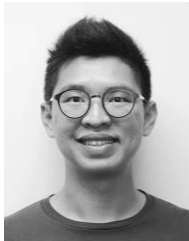
- [24] T.-H. Chan, K. Jia, S. Gao, J. Lu, and Z. Zeng, Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.
- [25] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C.-O. Chow, "Image noise types recognition using convolutional neural network with principal components analysis," *IET Image Process.*, vol. 11, no. 2, pp. 1238–1245, Dec. 2017. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2017.0374>
- [26] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognit.*, vol. 36, no. 7, pp. 1645–1659, Jul. 2003.
- [27] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Germany: Springer, 1998, pp. 9–50.
- [28] J. Wu, S. Qiu, R. Zeng, Y. Kong, L. Senhadji, and H. Shu, "Multilinear principal component analysis network for tensor object classification," *IEEE Access*, vol. 5, pp. 3322–3331, 2017.
- [29] H. J. Lee, "Neural network approach to identify model of vehicles," in *Advances in Neural Networks—ISNN*, Berlin, Germany: Springer-Verlag, 2006, pp. 66–72.
- [30] D. F. Llorca, D. Colás, I. G. Daza, I. Parra, and M. A. Sotelo, "Vehicle model recognition using geometry and appearance of car emblems from rear view images," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 3094–3099.
- [31] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531–1549, 2010.
- [32] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, QLD, Australia, 2015, pp. 689–692.



Hui Ying Khaw (S'15) received the B.Eng. degree (Hons.) from the Department of Electrical Engineering, University of Malaya, in 2014, where she is currently pursuing the Ph.D. degree. Her research interests include image processing, image denoising, and machine learning.



Joon Huang Chuah (M'07–SM'14) received the B.Eng. degree (Hons.) from Universiti Teknologi Malaysia, the M.Eng. degree from National University of Singapore, and the M.Phil. and Ph.D. degrees from University of Cambridge. He is currently a Senior Lecturer with the Department of Electrical Engineering, Faculty of Engineering, University of Malaya. His main research interests include image processing, computational intelligence, IC design, and scanning electron microscopy. He is also a Chartered Engineer registered under the Engineering Council, U.K., and also a Professional Engineer registered under the Board of Engineers, Malaysia. He is a Committee Member and a Communications Officer of the IEEE Computational Intelligence Society Malaysia Chapter.



Foo Chong Soon (S'15) received the B.Eng. degree (Hons.) from the Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of Malaya. His research interests include computer vision, machine learning, optimization, and intelligent transportation systems.



Jeevan Kanesan is currently an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Malaysia. He has published over 40 papers in peer-reviewed journals and conferences. His current research interests are optimal control, expert system, and nature-inspired metaheuristics.