Vehicle Detection and Type Classification Based on CNN-SVM

Stephen Karungaru, Lyu Dongyang, and Kenji Terada

***Abstract*—In this paper, we propose vehicle detection and classification in a real road environment using a modified and improved AlexNet. Among the various challenges faced, the problem of poor robustness in extracting vehicle candidate regions through a single feature is solved using the YOLO deep learning series algorithm used to propose potential regions and to further improve the speed of detection. For this, the lightweight network Yolov2-tiny is chosen as the location network. In the training process, anchor box clustering is performed based on the ground truth of the training set, which improves its performance on the specific dataset. The low classification accuracy problem after template-based feature extraction is solved using the optimal feature description extracted through convolution neural network learning. Moreover, based on AlexNet, through adjusting parameters, an improved algorithm was proposed whose model size is smaller and classification speed is faster than the original AlexNet. Spatial Pyramid Pooling (SPP) is added to the vehicle classification network which solves the problem of low accuracy due to image distortion caused by image resizing. By combining CNN with SVM and normalizing features in SVM, the generalization ability of the model was improved. Experiments show that our method has a better performance in vehicle detection and type classification.**

***Index Terms*—Vehicle detection, vehicle classification, Yolov2-tiny, AlexNet, spatial pyramid pooling, CNN, and SVM.**

I. Introduction

Vehicle detection and classification are the most important topics in the field of Advanced Driver Assistant Systems (ADAS) [1], [2] and Intelligent Transportation Systems (ITS) [3], [4]. With the development of image processing and pattern recognition technology, vision-based vehicle detection and classification have become a research hotspot in recent days, because machine vision is non-contact, convenient, and cheap.

Vehicle detection is a process of extracting vehicle targets from a region of interest in a video sequence or image using various image processing algorithms. Depending on whether the image is still or not, vehicle detection methods can be divided into 2 categories: vehicle detection based on motion information and vehicle detection based on features. Vehicle detection based on motion information is mainly aimed at the target vehicle during the movement. The methods include background difference, frame difference, optical flow, etc. However, these methods can only detect moving vehicles and the detection effect is greatly affected by illumination. Vehicle detection based on features mainly adopts vehicle appearance features such as vehicle symmetry, lights, edges, and colors [5], [6]. However, these features can only work well in certain circumstances, which is not universal. With the rapid development of artificial intelligence, a series of excellent object recognition algorithms have been proposed. Compared with traditional methods, using deep learning to locate objects is more accurate and generalizable. Since the real-time ability of the algorithm is our primary consideration, we choose Yolov2-tiny for vehicle detection.

Vehicle classification is the process of identifying and classifying the area extracted in the previous vehicle detection. Firstly, the feature description of the region of interest is established and then sent to the classifier to get the vehicle classification information. According to different feature extraction methods, vehicle classification can be divided into traditional methods and deep learning methods.

The traditional machine learning required the extraction of features manually and inputting them into various classifiers, such as HOG plus SVM [7] or DPM plus SVM [8], to implement classification. However, the feature selection is vital for accuracy, but different environments and features could cause the performance to drift. Choosing the most suitable features to describe the objects is, therefore, a great challenge. Deep learning based on CNN, uniquely solves this problem because of automatic optimal feature extraction through learning.

Therefore, in this work, the following two-step approach is proposed. For vehicle detection, a method of target extraction using the Yolov2-tiny is proposed. In the process of network training, we perform network parameter adjustment and K-means clustering. Experiments show that our method has strong real-time performance, high recall rate, and relatively satisfactory accuracy, which can easily extract the vehicle position in the video sequence. For type classification, the improved CNN network is used for feature extraction to make up for the disadvantage of poor generalization of manual feature extraction. We modify the network based on AlexNet and add SPP to solve the problem of low classification accuracy caused by image resizing and rescaling. After the CNN training is completed, we perform secondary training on the SVM. This step reduces the overfitting of the network, enhances the generalization ability of the model, and further improves the accuracy of the network. Experiments show that we are successful in the transformation of the two parts, but there is still some room for improvement.

The details of the proposed method are provided in detail in the sections below.

Manuscript received March 11, 2020; revised September 7, 2020.

The authors are with the Faculty of Technology and Science, Department of Computer science, Tokushima University, Japan (e-mail: karungaru@tokushima-u.ac.jp).

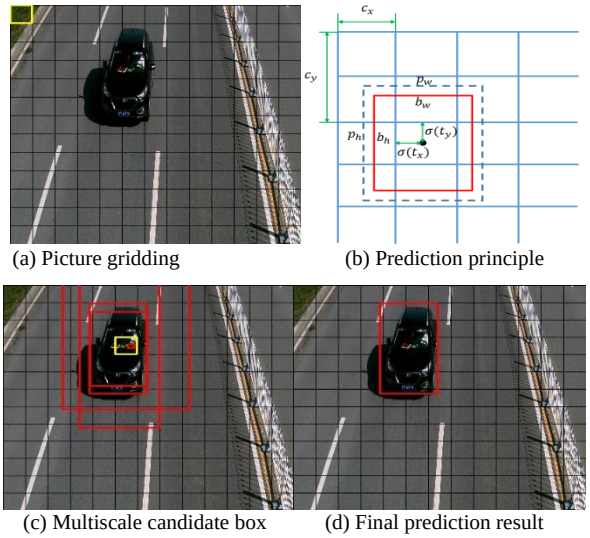
doi: 10.18178/ijmlc.2021.11.4.1052

II. Vehicle Detection

As an important part of ITS, vehicle detection is a prerequisite step for the subsequent vehicle type recognition. Therefore, an accurate and efficient vehicle detection method is vital. The vehicle detection method based on motion information can detect moving vehicles in video frames. However, the algorithm fails for stationary objects and cannot detect objects in a single image. Feature-based vehicle detection is not universal because it's difficult to find suitable features to describe the object. Therefore, we use the Yolov2-tiny [9], [10] deep learning algorithm to detect vehicles, which has stronger applicability and generality.

A. Object Extraction Principle of Yolov2-Tiny

The recognition process of Yolov2-tiny is shown in Fig. 1. First, the image is divided into grids. As shown in Fig. 1(a), the input image is divided into a 13×13 black grids, and a sliding window (yellow rectangle) traverses all the grids to predict and extract the candidate bounding boxes. Fig. 1(b) is the principle of bounding box prediction. Yolov2-tiny uses the idea of an anchor box, as in Faster-RCNN [11] for reference, to predict the position of candidate bounding box by regression of the position. The dotted box in the figure is the anchor box, which is a series of prior boxes, and the red solid box is the bounding box.



(a) Picture gridding (b) Prediction principle

(c) Multiscale candidate box (d) Final prediction result

Fig. 1. Yolov2-tiny detection procedure.

The prediction box is obtained by predicting the relative offset of the center point from the upper left corner of the grid where it is located and its size relative to the anchor box. Each prediction box predicts 5 parameters using the following equations:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

$$Pr(\text{object}) * IOU = \sigma(t_o) \tag{5}$$

In eq. (1) and (2), b_x and b_y are the coordinates of the center point of the prediction box. c_x and c_y are the coordinates of the upper left corner of the current grid. t_x and t_y are the offset value of the center point of the prediction box. The offset value is limited in the current grid by the sigmoid function, which makes the model easier to converge during training. In eq. (3) and (4), b_w and b_h are the width and height of the prediction box. p_w and p_h are the width and height of the anchor box. t_w and t_h are the width and height ratio of the prediction box to the anchor box.

Since the single-scale candidate box is prone to misdetections, Yolov2-tiny uses five different scales of anchor boxes. According to the principle in Fig. 1(b), a grid can predict five scale candidate bounding boxes. The prediction results are shown in Fig. 1(c). The yellow box is the current position of the sliding window, and the red boxes are the five prediction candidate bounding boxes of the current grid.

It then calculates the confidence $\sigma(t_o)$ of each prediction box according to eq. (5). When there is an object in the prediction box, $Pr(object)$ is 1. Otherwise, $Pr(object)$ is 0. IOU represents the coincidence rate between the predicted box and the real box of the object, as follows:

$$IOU = \frac{S(pred) \cap S(truth)}{S(pred) \cup S(truth)} \tag{6}$$

Each prediction box predicts the object probability of C categories at the same time and multiplies the object probability $Pr(C_i \mid object)$ of each category and the confidence $\sigma(t_o)$ of the prediction box to get the scores $score_i$ of each category of the box, as is shown in Eq. (7):

$$score_i = Pr(C_i \mid object) * s(t_o) \tag{7}$$

The algorithm sorts $score_i$ and compares its maximum value with the set threshold to filter out potential candidate bounding box. Finally, it uses non-maximum suppression (NMS) to filter these candidate boxes to obtain the final recognition result, as shown in Fig. 1(d).

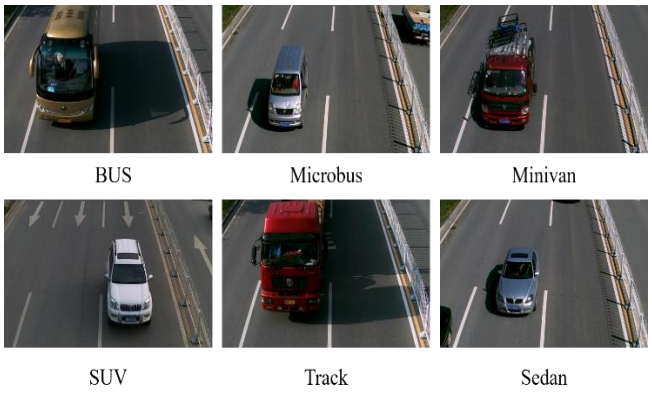


Fig. 2. BIT-Vehicles dataset.

B. BIT Datasets

The dataset used in this work is based on BIT-Vehicle [12], which is a dataset produced by the Beijing Institute of Technology. The dataset consists of 9850 front images of

vehicles taken at traffic checkpoints on highways. The dataset provides the location information of the vehicle in every image. These images have a variety of lighting conditions and background interference (such as guardrails, lane lines, etc.), as well as vehicles in a variety of colors. The vehicle dataset is divided into six categories: Bus, Microbus, Minivan, Sedan, SUV, and Truck. The total per vehicle types are 558, 883, 476, 5,922, 1,392, and 822, respectively. Fig. 2 shows some examples in the dataset.

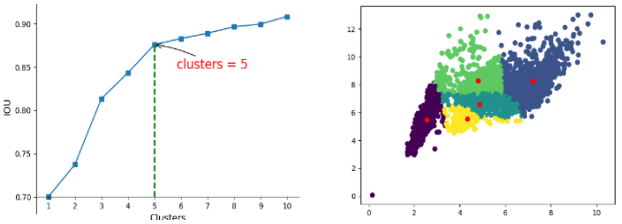
The function of vehicle detection in this section is only to locate the object, that is, to distinguish whether the extracted object is a vehicle or not. Therefore, the number of vehicle categories in our dataset is 1, and all are positive samples. We change the category of all samples in the XML file provided by the original dataset to "object". After that, we randomly allocate 80% as the training dataset and 20% as the testing dataset to form our dataset.

C. K-Means Clustering

The recognition performance of the Yolov2-tiny network is closely related to the selection of anchor box's width and height. The anchor box parameters in the Yolov2-tiny network configuration file are determined according to VOC2007 and VOC2012 datasets, which are not universal. Therefore, we use the K-means algorithm to cluster the ground truth of the vehicles' bounding box in the BIT dataset.

The clustering number k has a great influence on the clustering effect. Unreasonable k value will lead to the final output of the K-means algorithm being the local optimal solution rather than the global optimal solution. Fig. 3(a) reflects the relationship between k and IOU. We can find that when k equal to 5, the comprehensive performance of the network is the best. Fig. 3(b) shows the clustering distribution of the dataset when k equal to 5. The red points in the figure are the clustering center points, and each category set is distinguished by different colors.

The width and height of the anchor box correspond to the coordinates of the five clustering center points are (2.55, 5.46), (4.32, 5.53), (4.80, 8.26), (4.87, 6.58) and (7.21, 8.25) respectively.



(a) IOU-Cluster relationship curve (b) k = 5 Cluster distribution

Fig. 3. Result of clustering.

TABLE I: YOLOV2-TINY CONFIGURATION PARAMETERS

momentum	decay	max batches	learning rate	steps	scales
0.9	0.0005	2000	0.001	200, 1000	0.1,0.1

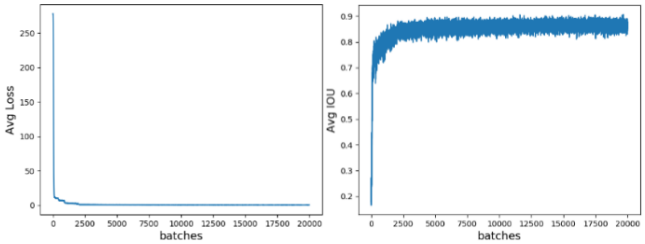
D. Vehicle Detection Results

The selection of training parameters determines the convergence speed and performance of the network. Through repeated experiments, the final network configuration parameters are shown in Table I.

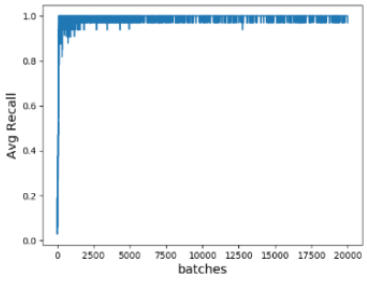
In the experiment, the hardware environment is CPU Intel Core i5-9400F, memory is 16GB, GPU 8GB GTX-1070ti, software environment is VS2015 + OPENCV, and the operating system is windows 10 professional edition.

The training results are shown in Fig. 4. Fig. 4(a) shows the average loss during the training process. It can be found that the average loss approaches 0 as the training progresses. Fig. 4(b) shows the relationship between IOU and batches in the training process. When the training is completed, the final IOU exceeds 80%. Fig. 4(c) is the average recall rate curve during the training process. Although the recall rate fluctuated, when the training batch reaches 2500 steps, the recall rate stabilized to more than 95%.

After training, we use the deep neural network (DNN) module in the OPENCV extension module to migrate the model trained in the Darknet to VS2015 and test the testing dataset in this environment.



(a) Result of the average loss (b) Result of average IOU



(c) Result of average Recall

Fig. 4. Training results of Yolov2-tiny.

The testing threshold of the detection network is set as 0.8. The testing data on the testing dataset is as follows:

1) The average recall rate is 94.45%, and the detection rate of vehicles is very high. All the targets in the testing dataset are extracted. However, some objects are not detected because their confidence is lower than the threshold value. This problem can be solved by reducing the threshold value properly on the premise of ensuring the detection accuracy. In addition, we also carried out migration experiments on the network under different datasets and found that when the environment changes drastically, the recall rate is significantly reduced. This part of the problem can only be solved by increasing training samples or using data enhancement.
2) The average IOU is 82.25%. In the initial selection of the neural network, the algorithm operation speed was taken as the primary consideration. So we adopted the simplified Yolov2 neural network, and abandon a certain network depth, which leads to the decline of detection rate. But this IOU is still able to meet most experimental situations.
3) The average recognition speed is 38.78ms, and the detection rate exceeds 25FPS. This detection rate can fully meet the speed requirements in general video

processing.

Fig. 5 is the detection result of one sample in the testing dataset.

Fig. 5. Yolov2-tiny detection results.

III. Vehicle Classification

As noted before, manually extracted features cannot accurately describe objects. Therefore, we use a combination of CNN and SVM to classify objects. After constructing the network, we verify its effectiveness on the UA-DETRAC [13] dataset. Our CNN is based on AlexNet [14].

A. Structure of Our Algorithm

As shown in Fig. 6, the network in this paper consists of a vehicle feature extractor and a vehicle classifier. In step 1, we train an AlexNet model consisting of a convolutional neural network (CNN) + fully connected layer (FC), so that CNN can learn to obtain accurate feature descriptions of objects. In step 2, we use the trained CNN as a feature extractor and then send the extracted features to the SVM for training. Finally, we build a recognition network composed of CNN + SVM.

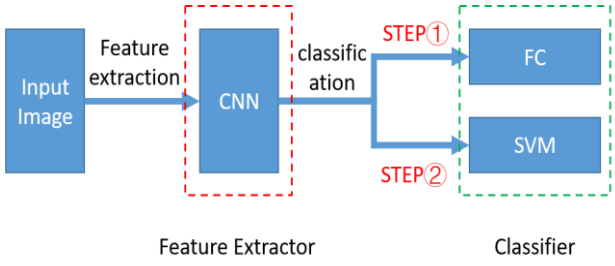


Fig. 6. Network composition.

B. SPP

AlexNet requires an input of 227×227 pixels. However, in vehicle classification, the detected vehicle scales are often inconsistent. If we resize the input image uniformly, the aspect ratio of the image will change, causing distortions to the original image, which in turn, negatively affects the recognition accuracy. In this work, spatial pyramid pooling layer (SPP) [15] is introduced between the last convolutional layer and the fully connected layer or Support Vector Machine (SVM) to standardize the dimensions of the features, avoiding the need to fix input size at the beginning. This process improves the over-fitting phenomenon in the network training process to some extent.

The principle of SPP is shown in Fig. 7. First, we divide the feature map into 2^n grid cells. Then, we perform max pooling among these cells. Finally, we expand and combine these features to get $\sum_n 2^n * N$ features, N being the number of feature maps. In the paper, we choose 1×1 and 2×2 scales, so we can get (1+4)×N features after SPP.

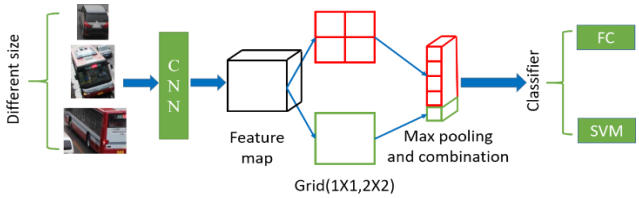


Fig. 7. Principle of SPP.

C. Proposed AlexNet Improvements

We add SPP into the original AlexNet, which enables the use of any input image size. Since the size of the images in the datasets is almost always medium and small, we reduce the size and number of the kernel on the premise of retaining the overall framework of AlexNet. Moreover, we delete some convolution layers which have little effect on the results. The improved AlexNet structure is shown in Table II. Since the input image size is not restricted, the width and height of the output feature map are unknown and we mark them with a question mark.

TABLE II: The CNN Structure of Improved AlexNet

Name	Kernel	Stride	Activation	Output
Conv1	5×5×96	1	Relu	?×?×96
Pool1	3×3×96	2		?×?×96
LRN				
Conv2	3×3×128	1	Relu	?×?×128
Pool2	3×3×128	2		?×?×128
LRN				
Conv3	3×3×128	1	Relu	?×?×128
Conv4	3×3×100	1	Relu	?×?×100
SPP(1×1,2×2)				500

Compared with the original AlexNet, we performed the following changes:

1) Using SPP to replace the Pool3 layer: We normalized features through the SPP layer instead of resizing images at the beginning, which avoids the loss of accuracy caused by image distortion.
2) Fewer layers: The original AlexNet convolute three times between Conv3 and Conv5. Since it has little effect on the results, we delete the Conv layer which makes the model size smaller.
3) Smaller kernel size: We replaced the 11×11 kernel of the original Conv1 layer with 5×5 kernel, and replaced the kernel of the other layers with 3×3 kernels.
4) Fewer feature maps, fewer model parameters, and faster recognition: The maximum number of feature maps of the proposed AlexNet is 128, while the maximum number of feature maps of the original AlexNet is 384.

D. Optimal Parameters of SVM

Because the UA-DETRAC dataset is from the real road video, there are different lighting conditions and some occlusion in the dataset, as is shown in Fig. 8 (http://detrac-db.rit.albany.edu/).

We crop the vehicles from the UA-DETRAC dataset with bounding box data to build a classification dataset. Because of the different lighting conditions, etc. we propose CNN-SVM to achieve high classification accuracy. The SVM performs better than the FC network.

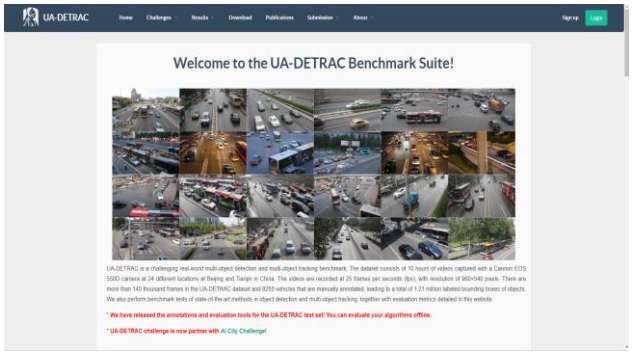
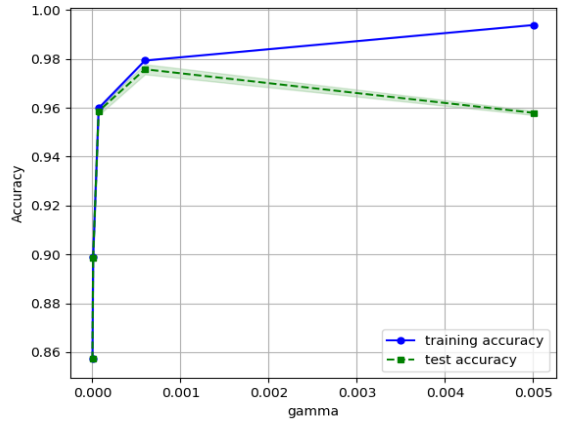
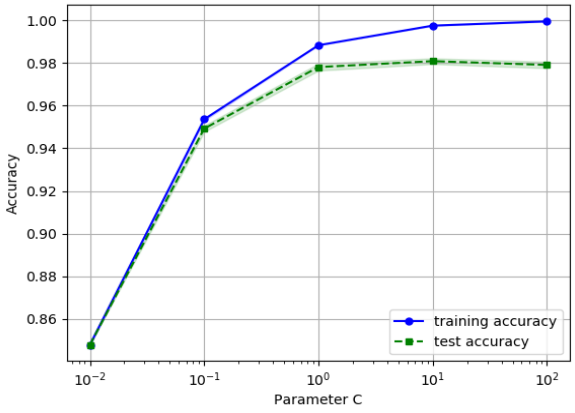


Fig. 8. Example images.

However, the performance of SVM mainly depends on the penalty factor C and the kernel parameter gamma. Therefore, the choice of the parameter is very necessary for the SVM. We select the optimal parameters of SVM using the K-fold cross-validation (using SKLearn). In this paper, we choose CV as 5. The value range of C is $lg^d, d = -2, -1, 0, 1, 2$. The value range of gamma is $lg^{0.925d-6}, d = 0, 1, 2, 3, 4$, Fig. 9 shows the result of cross-validation. From the image, the optimal gamma = 0.0006 and the optimal C = 10.



(a) Parameter gamma



(b) Parameter C

Fig. 9. Results of cross-validation.

E. UA-DETRAC Datasets

UA-DETRAC datasets consist of 10 hours of videos captured using a Canon EOS 550D camera at 24 different locations in Beijing and Tianjin in China. The videos are recorded at 25 frames per second (fps), with a resolution of 960×540 pixels [13]. Fig. 10 shows some examples in the datasets. Green boxes are vehicle regions and red boxes are misclassified candidate locations.

The UA-DETRAC dataset provide a training set containing 83,791 frames and 577,899 annotated bounding boxes. There are 4 types of object including 5936 vehicles (i.e., "car": 5177, "bus": 106, "van": 610, "others": 43).

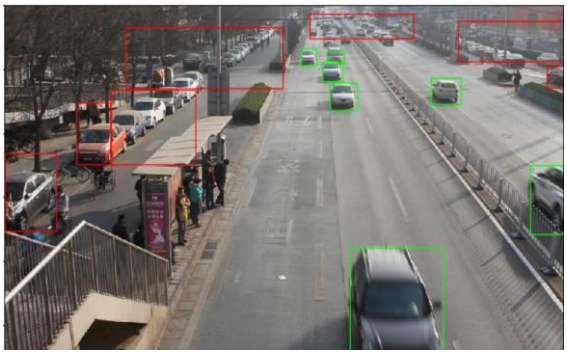


Fig.10 Scenes in UA-DETRAC.

However, the original UA-DETRAC dataset is too large for effective processing. Therefore, we sample every 10 frames in the training set to create our datasets. 80% is assigned as the training set and the remainder as the testing set. Thereafter, the objects and labels on the frame image are saved based on the annotated bounding box data. Finally, training data with 49076 images (i.e., "car": 41358, "bus": 2729, "van": 4686, "others": 303) and a testing data with 12053 images (i.e., "car": 10148, "bus": 662, "van": 1166, "others": 77) is obtained.

F. Vehicle Classification Results

In the experiment, the hardware environment was CPU Intel Core i7-9700k, memory 16GB, GPU NVIDIA RTX2080ti, and software environment was Python + Pytorch.

Table III shows the comparative testing results of three methods (original Alexnet, improved Alexnet, and improved Alexnet+SVM) on epoch 10. The results of each category are shown in the recall.

TABLE III: TESTING RESULT ON EPOCH 10

Methods	Car	Bus	Van	Others
AlexNet	98.17%	91.24%	68.44%	49.35%
Improved AlexNet	98.37%	93.81%	76.07%	71.43%
Improved AlexNet+SVM	98.87%	95.02%	76.50%	74.03%

Comparing the improved AlexNet with AlexNet, we can note that the recall of each category of the improved network has increased, especially for the "van" and "others" categories. Therefore, the improved CNN part is much better than Alexnet and it proves that adding the SPP method can improve the performance of the network.

Comparing the improved AlexNet + SVM with the improved AlexNet, the addition of SVM makes the recall of each category further increase, which proves that SVM has better generalization than FC and can fine-tune the accuracy to some extent.

Fig. 11 shows the overall accuracy curve of the three networks. The horizontal axis represents the number of epochs, and the vertical axis represents total accuracy.

From Fig. 11, we can see that the accuracy of the improved AlexNet + SVM is higher than the other two. Compared with the original AlexNet, the total accuracy is improved by

1.73%. In current situations where the accuracy is already high, this improvement is quite considerable.

Fig. 12 shows the final testing result of the improved AlexNet + SVM in the form of the confusion matrix. We can see that the accuracy of "car" and "bus" is high enough, while the accuracy of "Van" and "others" still has room for improvement. This might have been caused by data imbalance. Therefore, our next step is to solve this problem by over-sampling and under-sampling the datasets.

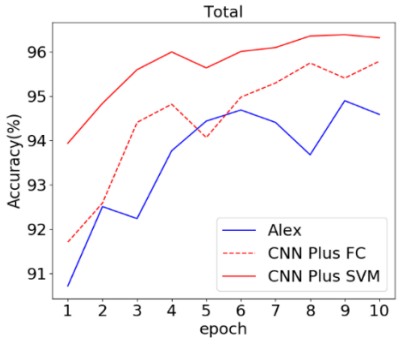


Fig. 11. Result of total accuracy.

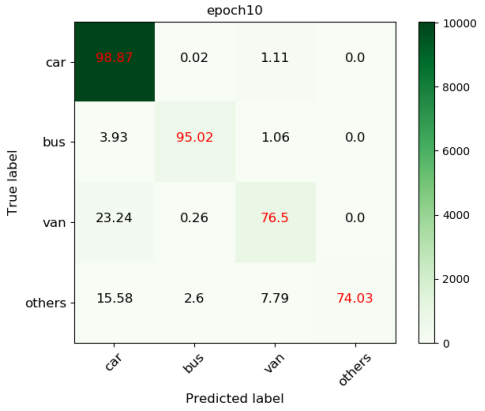


Fig. 12. Result of the confusion matrix.

In this work, we concentrated on improving the classification accuracy. However, we can confidently conclude that our method is faster than the original AlexNet. Our improved method has fewer layers, fewer feature maps, and smaller kernel size.

IV. Conclusion and Feature Work

In the paper, we discussed vehicle detection and type classification. For vehicle detection, a method of target extraction using the Yolov2-tiny is proposed. In the process of network training, we performed network parameter adjustment and K-means clustering. We used Darknet to train the detection part. Experiments show that our method has strong real-time performance, high recall rate, and relatively satisfactory accuracy, which can easily extract the vehicle position in the video sequence.

In vehicle type classification, the improved CNN network is used for feature extraction to make up for the disadvantage of poor generalization of manual feature extraction. We modify the network based on AlexNet and add SPP to solve the problem of low classification accuracy caused by image resizing and rescaling. After the CNN training is completed, we perform secondary training on the SVM. This step reduces the overfitting of the network, enhances the generalization ability of the model, and further improves the accuracy of the network. Experiments show that we are successful in the transformation of the two parts, but there is still some room for improvement.

Future works are as follows. To solve the problem of poor network mobility in vehicle detection, we consider increasing the number of datasets and training the network on multiple training datasets. To solve the problem of low accuracy of some categories due to data imbalance in vehicle classification, a data augmentation method will be adopted in the future. Moreover, our work is still separate for each part. Integrating the two perfectly is a future task.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Karungaru proposed, supervised, and prepared the final manuscript. The other research was done by Mr. Lyu including running the experiment and writing the original draft. Dr. Terada revised the paper and confirmed the research results. All authors have approved the final version.

References

[1] T. Wang, "Ph.D. forum: Real-time lane-vehicle detection for advanced driver assistance on mobile devices," in *Proc. IEEE International Conference on Smart Computing*, 2017.

[2] J. Arróspide and L. Salgado, "Video based vehicle detection and tracking for driver assistance systems," *Securitas Vialis*, vol. 7, pp. 1-9, 2014.

[3] X. Chen, L. Liu, Y. Deng *et al.*, "Vehicle detection based on visual attention mechanism and AdaBoost cascade classifier in intelligent transportation systems," *Optical and Quantum Electronics*, 2019, vol. 51, no. 8, pp. 263.

[4] Y. Mao and P. Shi, "Noise reduction algorithm of vehicle detection in intelligent transportation system," *Intelligent Transportation Systems*, 2003.

[5] C. Lv and T. Shan, "Multi-dimensional image edge localization method based on edge symmetry algorithm," *Multimedia Tools and Applications*, pp. 1-15. , 2019

[6] S. S. Pillai and B. Radhakrishnan, "Night-Time vehicle detection using tail lights: A survey," *International Journal of Engineering Research and General Science*, vol. 4, no. 2, 2016.

[7] Y. Xu, G. Yu, Y. Wang *et al.*, "A hybrid vehicle detection method based on viola-jones and HOG+ SVM from UAV images," *Sensors*, vol. 16, no. 8, p. 1325, 2016.

[8] A. Kurniawan, R. Saputra, M. Marzuki *et al.*, "The implementation of object recognition using Deformable Part Model (DPM) with Latent SVM on lumen robot friend," in *Proc. International Conference on Engineering and Technology Development (ICETD)*, 2017.

[9] J. Redmon, S. Divvala, R. Girshick *et al.*, "You only look once: Unified, real-time object detection," in *Proc. the IEEE Conference on Computer Vision and pattern Recognition*, 2016, pp. 779-788.

[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," arXiv preprint, 2017.

[11] S. Ren, K. He, R. Girshick *et al.*, "Faster r-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91-99, 2015.

[12] Z. Dong, Y. Wu, M. Pei *et al.*, "Vehicle type classification using a semi-supervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247-2256, 2015.

[13] L. Wen, D. Du, Z. Cai *et al.*, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," ArXiv preprint arXiv:1511.04136, 2015.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 2, 2012.

[15] K. He, X. Zhang, S. Ren *et al.*, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. European Conference on Computer Vision*, Springer, Cham, 2014, pp. 346-361.

Copyright © 2021 by the authors. This is an open-access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited (CC BY 4.0).

Stephen Karungaru received his B.S degree from the Department of Electrical/ Electronics, MOI University. He received a master's and a doctoral degree from the Department of Information Science and Intelligent systems, Faculty of Engineering, Tokushima University, in 2001 and 2004. He became an associate professor in the Institute of Advanced Science and Technology, Tokushima University since 2004. His research interests are face detection, recognition, neural networks, and genetic algorithms.

Lyu Dongyang is a dual degree graduate student. He was studied at Tokushima and Nantong Universities. His research interests are in artificial intelligence and machine vision.

Kenji Terada received a doctoral degree from Keio University in 1995. In 2009, he became a professor in the Department of Information Science and Intelligent Systems, University of Tokushima. His research interests are in computer vision and image processing. He is a member of the IEICE, SICE, SCIE, and JSPE.