International Journal of Engineering & Technology, 7 (3.15) (2018) 86-90

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Performance Evaluation for Vision-Based Vehicle Classification Using Convolutional Neural Network

Raja Durratun Safiyah[1]*, Zaid Abdul Rahim[2], Syamsul Syafiq[3] , Zaidah Ibrahim[4#] and Nurbaity Sabri[5]

[1,2,3,4]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Shah Alam, Selangor, Malaysia
[5] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM Melaka) Kampus Jasin, Melaka, Malaysia
**Corresponding author E-mail: durratunsafiyah@gmail.com,*

Abstract

Vision-based vehicle classification is a very challenging task due to vehicle pose and angle variations, weather conditions, lighting quality, and limited number of available datasets for training. It can be applied for driver assistance system and autonomous vehicles. This paper conducted a performance evaluation for this task based on three Convolutional Neural Network (CNN) models, which are simple CNN, and pre-trained CNN models that are AlexNet and GoogleNet. A dataset of more than 7000 images from the Image Processing Group (IPG) has been used for training and testing and the results indicate that AlexNet achieves the best classification result that is 65.09%. This result is obtained because of the variations of the quality of the images.

Keywords: *Vision Based Vehicle Classification, Convolutional Neural Network (CNN), Deep Learning Training from Scratch, AlexNet, GoogleNet*

1. Introduction

Convolutional Neural Network (CNN) is an efficient and effective recognition, identification and classification algorithm which is globally used in image processing and pattern recognition. CNN consist of various features including simple structure, fewer training parameters and its ability to adapt in different areas and requirements. It has become a popular topic of discussion especially in voice analysis and image recognition area. CNN's capabilities to share its weight make it similar to biological neural networks by reducing the complex processes of network model and the number of weights [1].

In general, CNN structure consist of two layers, layer one is feature extraction layer, where the input of each neuron is connected to local or native accessible field of previous layer and extracts the local features. Once the features of local field are extracted, the relationship between both local field and other features will be established. Local receptive field will receive an input from a set of features in a small neighbourhood produced by the previous layer. With local receptive fields, vital visual features can be extracted such as oriented edges, end-points and corners. These features are then combined with the higher layers. Second layer in CNN is feature map layer which contains each computing layer in the network. It is a collection of several elements of feature map. Structure of feature map uses a complex function to act as the activation function in the convolution network. Each convolution layer in CNN is supported by a computing layer that is used to compute and calculate the local average [2].

The usage of CNN focuses in identification displacement, zooming and other forms of misrepresenting invariance of two dimensional images of graphics. Since the component of the recognition layer of CNN learns via preparing information, it keeps away from expressing the element extraction and certainly gains from the preparation information when CNN is being used. In addition, the neurons in a similar component in its feature map have the indistinguishable weight, so the system can continue its learning process simultaneously. This is the main advantage and major standpoint of the CNN concerning the neuronal system associated with each other. Due to the extraordinary structure of the CNN's local shared weights influences it to have an exceptional preferred usage in speech recognition and image processing areas. Specifically in multi-dimensional input such as vector image can directly enter the system, which maintains the simplicity and avoids complex data reconstruction in feature extraction and classification.

Visual analysis for vehicle classification has attracted the attention of major players and organizations that are involved with computer vision. Vehicle detection and classification is considered as a major and challenging issue due to its elements such as variations in objects, camera angle, quality of image, weather condition and lighting quality [3]. Furthermore, the limited resources of large scale vehicle dataset that is available makes this area more challenging for the researcher until the Image Processing Group (IPG) released their large scale vehicle dataset which consist of more than 7,000 images of vehicle images that are taken from different angles and images of roads that includes various objects and backgrounds. 3,425 images of vehicle rears were taken from different points of view, and 3,900 images were extracted from road sequences that do not contain vehicles [4]. Fig. 1 illustrates some sample images from this dataset. With the availability of this publicly available dataset, this paper presents a performance evaluation between simple CNN and two pre-trained CNN models which are AlexNet and GoogleNet.

Copyright © 2018 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.



Fig.1: Sample images from [4].

Previously, research in object recognition uses handcrafted features such as texture features for fall activity recognition [5] leaf recognition [6] and color features have also being applied for fruit recognition [7]. The handcrafted feature is important to represent significant feature and classifier to obtain good recognition results. By applying this approach, various experiments need to be conducted in order to achieve the significant feature. However, currently, the object recognition research has progressed to Deep Learning (DL) where no handcrafted feature is required and yet the results produced are excellent.

2. Related Work

Since 1970s, the problem of vehicle detection or classification has been studied by several researchers and now, many studies show that it is possible to detect if a given object is a "vehicle" or a "non-vehicle". Research by Krizhevsky winner of Image Net LSVRC (ILSVRC)-2010 has proposed a deep convolution neural networks (DCNNs) algorithm where it managed to solve many classification issues [8]. A research on truck classification by Avery used vehicle length as the features. In this research 92% accuracy has been recorded using data from un-calibrated video [9]. Higher accuracy is achieved for vehicle classification using images gathered from range sensors where this research used laser intensity as the features and achieved 94% accuracy [11].

A combination of modified Scale Invariant Feature Transform (SIFT) and edge-detection algorithm in appearance-based method often implemented on vehicle classification. 98% classification rate is recorded for a comparison between car and minivan. Meanwhile, 96% classification rate is obtained for a comparison between car and taxi [12]. The appearance-based method used in this research as the learning based method is also applied to differentiate between moving object (trucks, people, bikes, cars and vans) by implementing multi-block local binary patterns [13]. Even though these researches obtain high accuracy results, the datasets used do not contain that many varieties in terms of angle and illumination changes.

3. Material and Methods

This research investigates the performance of three different types of CNN models in classifying whether the objects are vehicle or not. MATLAB version 2018a has been used to perform these experiments. The image dataset was processed by using the following computer specification: Intel® Intel Xeon Platinum 8160 2.1G Processor, 256GB (8x32GB) 2666MHz DDR4 ECC RDIMM memory, and 2 PCI Express® x16 Gen 3 graphics cards.

3.1 Image Dataset

This research uses a dataset of 3,245 images of vehicle rears from multiple angles and 3,900 images of road without vehicle [4]. To increase a high variability of these vehicle images, the selection of images is crucial to maximize the characteristic of the vehicle category. The dataset is divided into four categories which is far range, center or close range in front of camera, center or close range in the left, and close or middle range in the right. The image selected contain vehicle with background and non-background images. This include situation where the same vehicle will have different bounding position. In this research, images of 360x256 pixels are collected in highways of Turin, Madrid and Brussels are cropped into 64x64 pixels [4]. Fig. 2 illustrates some sample images [4]. Two hundred (2000) images from this dataset were captured under different weather conditions. Table 1 shows the distribution of the images acquired under different weather conditions.

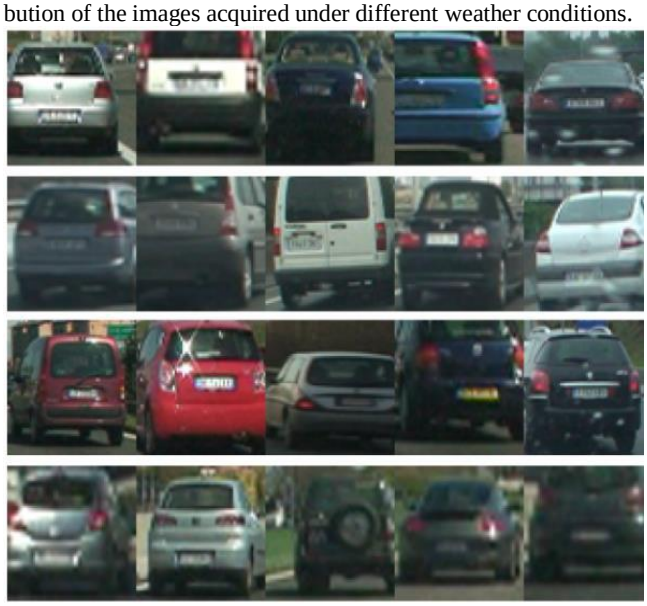


Fig. 2: Example of images dataset that collected by The Image Processing Group [4].

Table 1: Division of images captured according to weather conditions.

1000 (vehicle) & 1000 (non-vehicle)	20%	Sunny
	20%	Cloudy
	20%	Medium Condition
	20%	Poor Illumination
	10%	Light Rain
	5%	Bad Resolution Camera
	2.5%	Tunnels

3.2. Convolutional Neural Network Structures

For this research, we use three well-known CNN models, which are CNN from scratch, and pre-trained CNN models that are AlexNet and GoogleNet. All three models were tested and trained by using one single computer system and the training phase were allocated with the same processor and memory allocation to ensure all models received the same capabilities in completing this experiment. The main parameter that was evaluated was the average validation accuracy percentage, which is the percentage of the number of images that were recognized by the various models of CNN.

4. Basic Convolutional Neural Network

Convolutional Neural Network (CNN) training is the predefined set of programs that is available in Matlab program which is specifically built for image recognition program. For this experiment, this structure required an input image of 28-by-28-by-1 pixels and we change the default codes to enable this structure to accept and run a training phase for our images dataset. Fig. 3 showed the sample coding to change the image size to 64 by 64 pixels and in colour format.

```
layers = [
    imageInputLayer([64 64 3])
```

Fig. 3: Sample coding to set the image size.

Next step is to define the CNN architecture whereby the default set of layers has been predefined in this structure by Matlab program as showed in Fig. 4.

```
layers = [
    imageInputLayer([64 64 3])

    convolution2dLayer(3,8,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,32,'Padding',1)
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(4)
    softmaxLayer
    classificationLayer];
```

Fig. 4: Default CNN layers in Matlab version 2018a.

```
options = trainingOptions('sgdm', ...
    'MaxEpochs',4, ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain,layers,options);
```

Fig.5: Default training option set in Matlab version 2018a.

These layers consist of Image Input Layer, Convolutional Layer, Batch Normalization Layer, ReLU Layer, Max Pooling Layer, Fully Connected Layer, Softmax Layer, and Classification Layer. Next phase is the training process whereby in this phase the software trains the network on the training data and calculates the accuracy on the validation data at regular intervals during training as shown in Fig. 5. An epoch is a full training cycle on the entire training data set. The maximum number of epochs is 4 for this structure.

The final phase of this structure is to train the network. The training progress plot shows the mini-batch loss and accuracy and the validation loss and accuracy as illustrated in Fig. 6 and the details regarding the training phase is listed in Table 2.

Table 2: Training plot details for Deep Learning Training from Scratch

Parameters	Value
Validation Accuracy	56.30%
Training Status	Completed
Elapsed Time	1 min 23 sec
Number of Epoch	4
Number of Iteration	92
Validation Frequency	3



Fig. 6: Training plot graph shows the mini-batch loss and accuracy and the validation loss and accuracy.

Fig. 7 showed the result that has been achieved from this CNN structure. It is able to calculate the number of images in each category. labelCount is a table that contains the labels and the number of images for each label.

```
labelCount = countEachLabel(imds)
```

labelCount = 4×2 table

	Label	Count
1	Far	1950
2	Left	1950
3	MiddleClose	1475
4	Right	1950

Fig.7: The system is able to detect, categorize and calculate the number of images for four different categories that has been set from the dataset.

5. AlexNet

Pre-trained CNN model, AlexNet, has been trained for over a million images and can classify images into 1,000 different categories such as flower, animals, computer and many more. This architecture used rich features as input and produces a label for each image category as an output. It consist of a total of eight layers which is five convolution layers, three pooling layers and the remaining layers are fully connected layers that contain 60 million trainable parameter [14]. The first element of the layers property of the network is the image input layer. This layer requires input images from three colour channel (red green blue) with the size of 227-by-227-by-3 as shown in Fig. 8.

```
inputSize = net.Layers(1).InputSize
```

```
inputSize = 1×3
   227   227     3
```

Fig. 8: Input size required by AlexNet.

The next phase of this experiment is to train the network. Automatic image resize are set on the datastore code to resize multi

size images into the required size (227-by-227-by-3). Configuration operation is performed to the last three layers of the pre-trained. It must be fine-tuned for the new classification problem. The new fully connected layer is specified according to the dataset categories (left, right, far and c Close) as shown in Fig. 9.

```
numClasses = numel(categories(imdsTrain.Labels))

numClasses = 4
```

Fig. 9: Number of classes is set into a new data set classes.

Training option is the next step in this experiment whereby in this phase the software trains the network on the training data and calculates the accuracy of the validation data at regular intervals during training as shown in Fig. 10 and six epochs has been set which is the default number of epoch for this structure.

```
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',1e-4, ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'ValidationPatience',Inf, ...
    'Verbose',false ,...
    'Plots','training-progress');
```

Fig.10: Training option and number of epoch used for AlexNet platform.

Fig. 11 shows the training progress plot consist of mini-batch loss and the validation loss and accuracy as shown in Fig.11. The details regarding the training phase is listed in Table 3.

Table 3: Training plot details for AlexNet.

Parameters	Value
Validation Accuracy	65.09%
Training Status	Completed
Elapsed Time	906 min 38 sec
Number of Epoch	6
Number of Iteration	3072
Validation Frequency	3

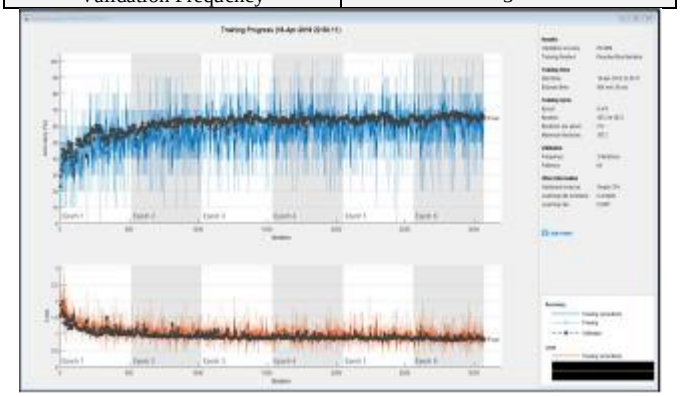


Fig. 11: Training plot graph shows the mini-batch loss and accuracy and the validation loss and accuracy of AlexNet.

5. Google Net

GoogleNet is one of the compelling deep CNN pre-trained models. GoogleNet has also been trained on over a million images and can classify images into 1000 object categories as AlexNet. GoogleNet consists of two outer convolutional layers, two outer pooling layers, three sets of top-1 and top-5 loss functions for three classifiers with a regularization dropout of 0.7, 0.7 and 0.4 respectively for the three classifiers and nine inception layers. In each inception layer, there exist six convolution layers and one pooling layer [15]. The first element of the layers property of the network is the image input layer. This layer requires input images of size 224-by-224-by-3, where 3 is the number of colour channels as shown in Fig. 12.

```
net.Layers(1)

ans =
  ImageInputLayer with properties:

                 Name: 'data'
            InputSize: [224 224 3]

   Hyperparameters
    DataAugmentation: 'none'
       Normalization: 'zerocenter'

inputSize = net.Layers(1).InputSize;
```

Fig. 12: Input size required by GoogleNet.

Training option is the next step in this experiment whereby in this phase the software trains the network on the training data and calculates the accuracy of the validation data at regular intervals during training as shown in Fig. 13 and six epochs has been set which is the default number of epoch for this structure.

```
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',1e-4, ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',3, ...
    'ValidationPatience',Inf, ...
    'Verbose',false ,...
    'Plots','training-progress');
```

Fig. 13: Training option and number of epoch used for GoogleNet.

The training progress plot shows the mini-batch loss and accuracy and the validation loss and accuracy as illustrated in Fig. 14 and the details regarding the training phase is listed in Table 4.

Table 4: Training plot details for GoogleNet.

Parameters	Value
Validation Accuracy	64.22%
Training Status	Completed
Elapsed Time	3265 min 33 sec
Number of Epoch	6
Number of Iteration	3072
Validation Frequency	3

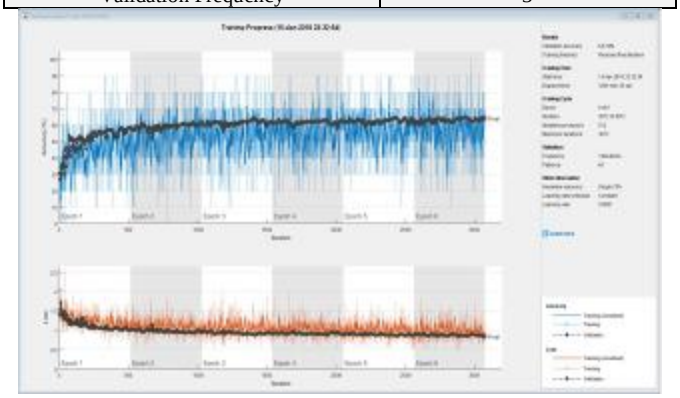


Fig. 14: Training plot graph shows the mini-batch loss and accuracy and the validation loss and accuracy of GoogleNet platform.

Fig. 15 showed the results that have been achieved by GoogleNet with the predicted labels and probabilities. GoogleNet is able to determine each vehicle angle, which consist of Left, Right, Middle Close and Far positions. The overall accuracy of GoogleNet is 64.22% and the loss percentage of 35.78% which is due to low quality of images and variations of vehicle positions, weather conditions and lighting qualities.



Fig.15: Sample results produced by GoogleNet.

6. Conclusion and Findings

All findings in this section were based on the experiments from three selected CNN models as illustrated in Table 5.

Table 5: Summary of performance produced by basic CNN, AlexNet and GooglNet.

Parameters	Basic CNN	AlexNet	GoogleNet
Average Validation Accuracy	56.30%	**65.09%**	64.22%
Training Status	Completed	Completed	Completed
Elapsed Time	1 min 23 sec	906 min 38 sec	3265 min 33 sec
Number of Epoch	4	6	6
Number of Iteration	92	3072	3072
Validation Frequency	30	3	3

Based on the results from this experiment, AlexNet has produced the highest average validation accuracy percentage followed by GoogleNet and CNN. From the dataset that has been used for this experiment, the top two pre-trained network models have recognized only 60% of the average images. This is due to the poor quality of images in the dataset which contributes of more than 35% average loss of accuracy and recognition percentage. Nevertheless, the system was able to recognize and separated two main categories in this dataset which are vehicle and non-vehicle.

For vision-based vehicle classification dataset, it is understandable that the quality of the images was poor due to its main resources, which are video sequences (acquired with a forward-looking camera mounted on a vehicle). This resource effected the overall images quality used in this experiment as images in the dataset was captured during various weather conditions such as sunny, cloudy, medium conditions (neither very sunny nor cloudy), poor illumination (down/dusk), light rain, with bad resolution cameras, and in tunnels (with artificial light).

These results have proven that even with a good pre-trained network model such as AlexNet and GoogleNet, the quality of images in a dataset play some important roles in ensuring the best accuracy result produced by these large-scale pre-trained network models. AlexNet has showed a small margin of higher percentage over GoogleNet because AlexNet extracts features that are slightly more general and more effective for a dataset and used a relatively simple layout, compared to GoogleNet. GoogleNet requires the longest training time because of the large number of layers in its architecture. Future work includes experimenting with other pre-trained CNN models with other publicly available datasets.

Acknowledgement

The authors would like to thank Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, for sponsoring this research.

References

[1] H. Salman, J. Grover, and T. Shankar, "Hierarchical Reinforcement Learning for Sequencing Behaviors," vol. 2449, pp. 2352–2449, 2018.

[2] C. H. Samer, K. Rishi, and Rowen, "Image Recognition Using Convolutional Neural Networks," *Cadence Whitepaper*, pp. 1–12, 2015.

[3] J. T. Lee and Y. Chung, "Deep Learning-Based Vehicle Classification Using an Ensemble of Local Expert and Global Networks," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017–July, pp. 920–925, 2017.

[4] E. T. S. I. T. Grupo de Tratamiento de Imágenes (GTI), "Vehicle Images Database," 2011. [Online]. Available: https://www.gti.ssr.upm.es/data/Vehicle_database.html. [Accessed: 17-Apr-2018].

[5] W. Ong Vui Jiunn, N. Sabri and Z. Ibrahim, "Image-based Human Fall Recognition using Gaussian Mixture Model and Support Vector Machine", International Journal of Control Theory and Applications, vol. 9,number 44, 2016

[6] Z. Ibrahim, N. Sabri and N. N. Mohd Manghor, "Leaf Recognition Using Texture Features for Herbal Plant Identification', International Journal of Electrical Engineering and Computer Science (IJEECS),Vol.9, No. 1 2018, pp.152-156.

[7] N. Sabri and Z. Ibrahim, "Palm Oil Fresh Fruit Bunch Ripeness Grading Identification using Color Features", Journal of Fundamental and Applied Science, 2017, 9(4S), pp. 563-579.

[8] A. Krizhevsky and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," pp. 1–9.

[9] R.P. Avery, Y. Wang, G.S. Rutherford, "Length-Based Vehicle-Classification Using Images from Uncalibrated Video Cameras," in:Proceedings of the 7th International IEEE Conference on IntelligentTransportation System, pp.737-742, 2004.

[10] G. Zhang, R.P. Avery, Y. Wang, "A Video-Based Vehicle Detection andClassification System for Real-Time Traffic Data Collection UsingUncalibrated Video Cameras," Transportation Research Record: Journalof the Transportation Research Board, 1993: 138-147, 2007.

[11] G. Moussa and K. Hussain, "Laser Intensity Automatic Vehicle-Classification System," North American Travel Monitoring Expositionand Conference (NATMEC)", Washington, DC, USA, August 6-8,2008. G. Zhang, R.P. Avery, Y. Wang, "A Video-Based Vehicle Detection andClassification System for Real-Time Traffic Data Collection UsingUncalibrated Video Cameras," Transportation Research Record: Journalof the Transportation Research Board, 1993: 138-147, 2007.

[12] X. Ma, W. Eric, and L. Grimson, "Edge-Based Rich Representation forVehicle Classification", Proc. Int. Conf. Computer Vision, vol. 2, pp.1185- 1192, 2005

[13] L. Zhang, S.Z. Li, X. Yuan, S. Xiang, "Real-Time Object Classificationin Video Surveillance Based On Appearance Learning," in: Proceedingsof IEEE Conference on Computer Vision and Pattern Recognition, 1-8,2007.

[14] E. Okafor, M. A. Wiering, E. Okafor, P. Pawara, F. Karaaba, and O. Surinta, "Comparative Study Between Deep Learning and Bag of Visual Words for Wild-Animal Recognition Comparative Study Between Deep Learning and Bag of Visual Words for Wild-Animal Recognition," no. December, 2016.

[15] L. F. Rodrigues, M. C. Naldi, and J. F. Mari, "HEp-2 Cell Image Classification Based on Convolutional Neural Networks," *2017 Work. Comput. Vis.*, pp. 13–18, 2017.