Modeling of Convolutional Neural Networks for Detection and Classification of Three Vehicle Classes

Faqih Rofii¹, Gigih Priyandoko², Istiadi³, Mohammad Ifan Fanani⁴
¹,²,³,⁴ Departement of Electrical Engineering, Universitas Widyagama Malang

E-mail: faqih@widyagama.ac.id

Abstract. The increasing number of vehicles causes traffic density on each road segment, especially in urban areas. Planning and regulating vehicle traffic, requires traffic flow parameter data, which currently is still largely done manually by assigning several people to count vehicles passing divided by a certain time period. The development of artificial intelligence in the field of computer vision enables the observation of traffic data using smart cameras. This study aims to produce a Convolutional Neural Network (CNN) model for the detection and classification of three vehicle classes, namely bus, car and motorcycle. The architecture used to build CNN uses smallVGGNet with 3x3 convolution layers and one fully connected layer. The first three convolution layers are considered as feature extraction layers while the last one is for classification where there are three output types of vehicles. The model is trained using 3,000 different images according to the class. In the training phase, it was determined 75 epochs where the consumption time of each epoch was ± 30 minutes so that the total time needed was 37.5 hours. The results of tests that have been done show that CNN models that have been trained can classify vehicle types with an accuracy of between 63-100%.

Keyword: CNN; deep learning; vehicle traffic; detection; classification

1. Introduction
The number of vehicles is increasing every year. Based on data released by Badan Pusat Statistik (BPS) Indonesia in 2015, the number of motorcycle vehicles reached 98.9 million, passenger cars 13.5 million, freight cars 6.6 million and 2.4 million bus cars [1]. On the other hand, the width of the road does not increase, causing traffic density and congestion. To simplify planning and regulating vehicle traffic, traffic flow parameter data is needed, including: the number, density and speed of the vehicle [2] and to reproduce experimental data sample for calibrating traffic flow fundamental diagram [3]. Currently to obtain traffic flow data is done manually by assigning several people to count passing vehicles divided by a certain time span. Another way to measure the number of vehicles and classify their type is to use a piezoelectric sensor [4].

In the field of computer vision the study of the identification of objects in images still continues, how to duplicate the ability of humans to understand image information, so that computers can recognize objects in images like humans. Moving object detection in video streaming is a step to extract information in many applications including: surveillance, people tracking, traffic monitoring and video semantic annotations [5] and [6]. Intelligent monitoring includes the method of acquisition, processing, analysis and understanding of digital images and extraction of high-dimensional data from the real world.
to produce symbolic and numeric information [7]. To detect and track certain moving objects, the most important thing is to eliminate interference such as vehicle detection on intelligent transportation systems [8]. To reduce interference several methods have been applied, namely the Bayesian method, such as particle filter [9], Kalman filter [10], or using difference image methods such as background modeling [11] and Gaussian mixture model (GMM).

Video is a sequence of images called frames that are displayed with sufficient frequency, so that the human eye can understand the continuity of its contents. Detecting moving objects is the first step for further video analysis [12]. To analyze video sequences, object detection methods include: frame differencing, optical flow and background substraction; Object classification includes: shape-based, motion-based, color-based and texture-based, while object tracking methods include: point-based, kernel-based and silhouette-based [13]. Kim proposed a fast algorithm for the detection and tracking of movable multi-objects for intelligent video surveillance [14]. LeeCun and Schmiduber, stated that deep learning allows computational models consisting of several layers of processing to study data representations with different levels of abstraction [15]. This method has improved state-of-the-art speech recognition, visual object recognition, object detection and image, video, speech and audio processing.

Convolutional Neural Networks (CNN) is one of the deep learning methods that is able to identify feature extraction. This method has been used in image recognition, computer vision, and Natural Language Processing (NLP). CNN tried to imitate the image recognition system on human visual cortex, so that it has the ability to process image information [16]. In previous studies the implementation of Convolutional Neural Networks (CNN) for object detection has been able to identify and classify vehicles [15], can improve CNN structure and reduce loss of better feature information. Detection of Vehicle Make and Model Recognition [17] as well as detection and classification of vehicles based on vehicle rear view images.

In accordance with research that continues to develop in the field of image and video processing, as well as deep learning with the CNN method, this research proposes CNN modeling for vehicle detection and classification through mapping the features / special features of different types of vehicles. The purpose of the research is to produce CNN models that have been trained to be able to identify and classify various types of vehicles. The performance of the CNN models that have been trained is measured from the level of accuracy and losses to the number of epochs.

2. Convolutional Neural Networks

Convolutional Neural Networks (CNN, CNNs or ConvNets) is an algorithm for deep learning. CNN is a multi-layer perceptron (MLP) specifically designed to recognize two-dimensional images. CNN mimics how the human brain works to recognize objects that are seen, this technique is called image recognition. CNN can be expressed as an artificial neural network that is specifically made to process array data. Data in the form of time series can be considered as one-dimensional arrays, while image data, are considered as two-dimensional arrays of image pixels. CNN uses a mathematical operation in the form of convolution, at least at one network layer.

CNN uses three basic ideas, namely: local receptive fields, shared weights, and pooling [18]. A local receptive field is a connection that occurs in a small local area in the input image data. Unlike in a feed forward network, the input is fully connected to the next hidden node for each neuron. Conversely, CNN input only makes connections in a small area. Each neuron in a hidden layer will be connected to a small field from the previous layer, called the local receptive field. For example, if the plane has a 3 × 3 area, neurons from the first convolutional layer will correspond to 9 pixels from the input layer.

Shared weights on CNN are sharing the same weighting parameters for all neurons. In the convolutional layer, neurons are arranged into several parallel layers, called feature maps. Each neuron in the feature map is connected to a local receptive field. For each feature map, all neurons share the same weighting parameters known as filters or kernels. CNN consists of a convolution layer, a pooling / subsampling layer and a fully connected layer. The convolution layer consists of neurons arranged in such a way as to form a filter.
Figure 1. Basic Structure of CNN

The convolution layer produces an initial feature map in the input data. Convolution is formed by shifting the filter in the input data [15]. The pooling layer consists of a filter with a certain stride and size that shifts throughout the feature map area. Pooling that is commonly used is max pooling and average pooling. The purpose of the pooling layer is to reduce the dimensions of the feature map, thereby speeding up computation because fewer parameters need to be updated and to overcome overfitting. The output of the pooling layer is a multidimensional array, so it needs to be flattened, i.e., re-forming the feature map into a vector as input for the fully-connected layer. FC-Layer is an MLP that has several hidden layers, activation functions, output layers and loss functions.

3. Method

CNN modeling has two important stages, namely the training and testing phase. The training phase aims to build a CNN model to get the CNN architecture consisting of the number of layers, neurons and the weight of each neuron and its activation function. At the training stage, training data is needed, to train the CNN model that has been built. The second stage, namely testing aims to test the ability of the model that has been trained. These stages are explained in figure 2.

Figure 2. CNN Modeling

The model was built using python with tensorflow and Keras to build deep neural networks, open_cv for image processing and vgg_CNN as CNN architecture. Figure 2 is a CNN modeling process, starting from creating a dataset, dividing the dataset into training and test data, building the CNN architecture, training and evaluating the model. To model the CNN network, a number of training datasets are needed to identify certain relationships and common features associated with objects. The dataset is built using Microsoft's search engine Bing Image Search API. Filtering is done to adjust the downloaded image parameters with the model. The dataset is a matrix representation of the image filtered with the image
label and stored in the vehicles dataset. Table 1 describes the image parameters as a dataset of vehicles that are classified as buses, cars, and motorcycles.

The next stage divides the dataset into training data and validation data. The choice of type and amount of training data is very important to produce a model with good accuracy. To get trained model performance, a model test is performed using validation data. In this study, the dataset used was 1,000 images in each class of vehicles, with a total of 3,000 images. Each image of each class is divided into 75% as training data and 25% as test / validation data.

Table 1. Vehicle dataset parameters

| Images     | Class | Parameters                                                                 |
|------------|-------|---------------------------------------------------------------------------|
|            | bus   | Bus images are classified as a large vehicle with a size, length 7.5 - 12.5 m, width 2.2 - 2.5 m, and height 3 - 3.2 m |
|            | car   | Image cars are classified as medium vehicle with size, length 3.5 - 5.4 m, width 1.4 - 1.9 m, and height 1.5 - 2.3 m. |
|            | motorcycle | Motorcycle images are classified as small vehicles with a size, maximum height of 1.3 m, width 1 m, and length 2 m. |

Preprocessing dataset aims to resize the vehicle image to fit the dimensions required by the CNN architecture. After the size, color, height and width of the image obtained the separation of the values of each component red, green, and blue from the color image is done. Furthermore, a new image is created to accommodate the changed image. This image measuring 64 x 64 pixels is processed and converted into a matrix as input for CNN.

Table 2. Composition of training and validation data

| Class      | Training data (images) | Validation data (images) | Number of images |
|------------|------------------------|--------------------------|------------------|
| Bus        | 750                    | 250                      | 1.000            |
| Car        | 750                    | 250                      | 1.000            |
| Motorcycle | 750                    | 250                      | 1.000            |
| Total (images) | 2250                  | 750                      | 3.000            |

4. Results And Discussion

This study uses smallVGGNet with 3x3 convolution layers to build the CNN architecture. The build command is declared for the implementation of this architecture which contains four input image parameters, namely: width, height, depth and number of classes. The input image is in an RGB color space with a depth value of 3. A total of three convolution layers are used in this study plus one fully connected layer. The three initial convolution layers are considered as feature extraction layers while the last one is the classification layer, with three output types of vehicles namely bus, car, and motorcycle. Figure 3 shows the CNN architecture that was built.

The first convolution layer consists of 32 filters with a size of 3x3, activation functions using the Rectified Linear Unit (ReLU) and Maxpooling with 32 filters. The second convolution layer consists of 64 filters of 3x3 size, with the activation function of ReLU and Maxpooling with 64 filters. The third convolution layer consists of 128 filters with a size of 3x3, with the activation function of ReLU and
Maxpooling with 128 filters. At this convolution layer Batch Normalization, MaxPooling and Dropout are applied. Batch Normalization is used to normalize the activation of the input volume before being passed to the next layer, thereby effectively reducing the number of epochs in the training phase. While dropout serves to reduce overfitting, improve accuracy, and allow network generalization. The last layer is fully connected, or the layer that connects all the outputs of the third convolution layer, with the ReLU activation function and three outputs as the final prediction of CNN.

Training the model using a computer with Windows 7 Ultimate 64-bit OS, processor: i3-2350M, 2.30GHz, 4096MB RAM and GPU: Intel (R) HD Graphics 3000. The training takes ± 30 minutes for each epoch of 75 epochs, so that the total time needed is 37.5 hours where the time required for the GPU to complete the training process every epoch for ± 8 minutes. The results of the training carried out are shown by the graph in Figure 4.

Figure 3. CNN Architecture

Figure 4 can be seen that increasing the epoch from 1 to 40 can significantly increase training accuracy, which is 0.43 to 0.85. However, above epoch 45 the increase in training accuracy is slow although it continues to increase until the accuracy reaches 0.88. The accuracy of the validation data is very volatile at the beginning of the epoch between 1 to 27 epochs, but the overall trend is increasing to reach 0.85 at
75 epochs. While the loss in the training data shows a sharp decrease in 1 to 35 epoch, which is 0.78 to 0.5, and continues to decrease gently until it reaches a loss of 0.48 at 75 epochs. While the loss in the validation data is very volatile along the number of epochs, although the trend decreases to 0.5.

| Class      | Accuracy | Recall | F1-Score |
|------------|----------|--------|----------|
| Bus        | 0.84     | 0.91   | 0.87     |
| Car        | 0.90     | 0.77   | 0.79     |
| Motorcycles| 0.83     | 0.88   | 0.88     |

We tested the trained model to predict images that were not in the dataset. The images tested represent three types of vehicle classes namely: bus, car and motorcycle. The test results are shown in Figure 5, which shows that CNN models that have been trained can classify vehicle types with an accuracy of between 63-100%.

![Figure 5](image)

Figure 5. Accuracy of the CNN model for classifications of vehicle classes that are not datasets

### 6. Conclusion

The model proposed in this study is intended for the detection and classification of three vehicle classes, namely bus, car and motorcycle based on CNN architecture. The architecture used to build CNN using smallVGGNet with 3x3 layers of convolution, has been trained with 3,000 different images according to the class. The results of tests that have been done show that CNN models that have been trained can classify vehicle types with an accuracy of between 63-100%. To improve the accuracy of the model output predictions, it is possible to develop more varied training and validation data, and to consider CNN architecture in terms of the number of layers, neurons, training algorithms and network structures.

### 7. Acknowledgement

The authors gratefully wish to acknowledge the Lembaga Penelitian dan Pengabdian Masyarakat Universitas Widyagama Malang and the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia for the financial support throughout the course of this research.

### 8. References

[1] Statistik B P 2012 Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis tahun 1987-2013 Diakses dari http://www.bps.go.id/tab_sub/view.php
[2] Mathew T V and Rao K K 2006 Fundamental Parameters Of Traffic Flow NPTEL (may 2006)
[3] Zhang J, Qu X and Wang S 2018 Reproducible generation of experimental data sample for calibrating traffic flow fundamental diagram Transportation Research Part A: Policy and Practice 111 41–52
[4] Santoso D R and Nurriyah A L 2011 Development of a simple traffic sensor and system with vehicle classification based on PVDF film element Sensors & Transducers 126 74
[5] Hu W-C, Chen C-H, Chen T-Y, Huang D-Y and Wu Z-C 2015 Moving object detection and tracking from video captured by moving camera Journal of Visual Communication and Image Representation 30 164–180
[6] Prakash U M and Thamaraiselvi V G 2014 Detecting and tracking of multiple moving objects for intelligent video surveillance systems *Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on (IEEE)* pp 253–257

[7] Forsyth D and Ponce J 2011 *Computer vision: a modern approach* (Upper Saddle River, NJ; London: Prentice Hall)

[8] Yang Z and Pun-Cheng L S 2018 Vehicle detection in intelligent transportation systems and its applications under varying environments: A review *Image and Vision Computing* 69 143–154

[9] Breitenstein M D, Reichlin F, Leibe B, Koller-Meier E and Van Gool L 2009 Robust tracking-by-detection using a detector confidence particle filter *Computer Vision, 2009 IEEE 12th International Conference on (IEEE)* pp 1515–1522

[10] Stenger B, Thayananthan A, Torr P H and Cipolla R 2006 Model-based hand tracking using a hierarchical bayesian filter *IEEE transactions on pattern analysis and machine intelligence* 28 1372–1384

[11] Vargas M, Milla J M, Toral S L and Barrero F 2010 An enhanced background estimation algorithm for vehicle detection in urban traffic scenes *IEEE Transactions on Vehicular Technology* 59 3694–3709

[12] Muchsin M, Rofii F and Jaenuri I 2014 RANCANG BANGUN PROTOTYPE MONITORING KEMANAN RUMAH BERBASIS CLOSED CIRCUIT (CCTV) DENGAN DETEKTOR GERAK *Widya Teknika* 22

[13] Parekh H S, Thakore D G and Jaliya U K 2014 A survey on object detection and tracking methods *International Journal of Innovative Research in Computer and Communication Engineering* 2 2970–2979

[14] Kim J S, Yeom D H and Joo Y H 2011 Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems *IEEE Transactions on Consumer Electronics* 57

[15] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *nature* 521 436

[16] Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Yang S, Wang Z and Loy C-C 2015 Deepid-net: Deformable deep convolutional neural networks for object detection *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 2403–2412

[17] Tian B, Li Y, Li B and Wen D 2014 Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance *IEEE Transactions on Intelligent Transportation Systems* 15 597–606