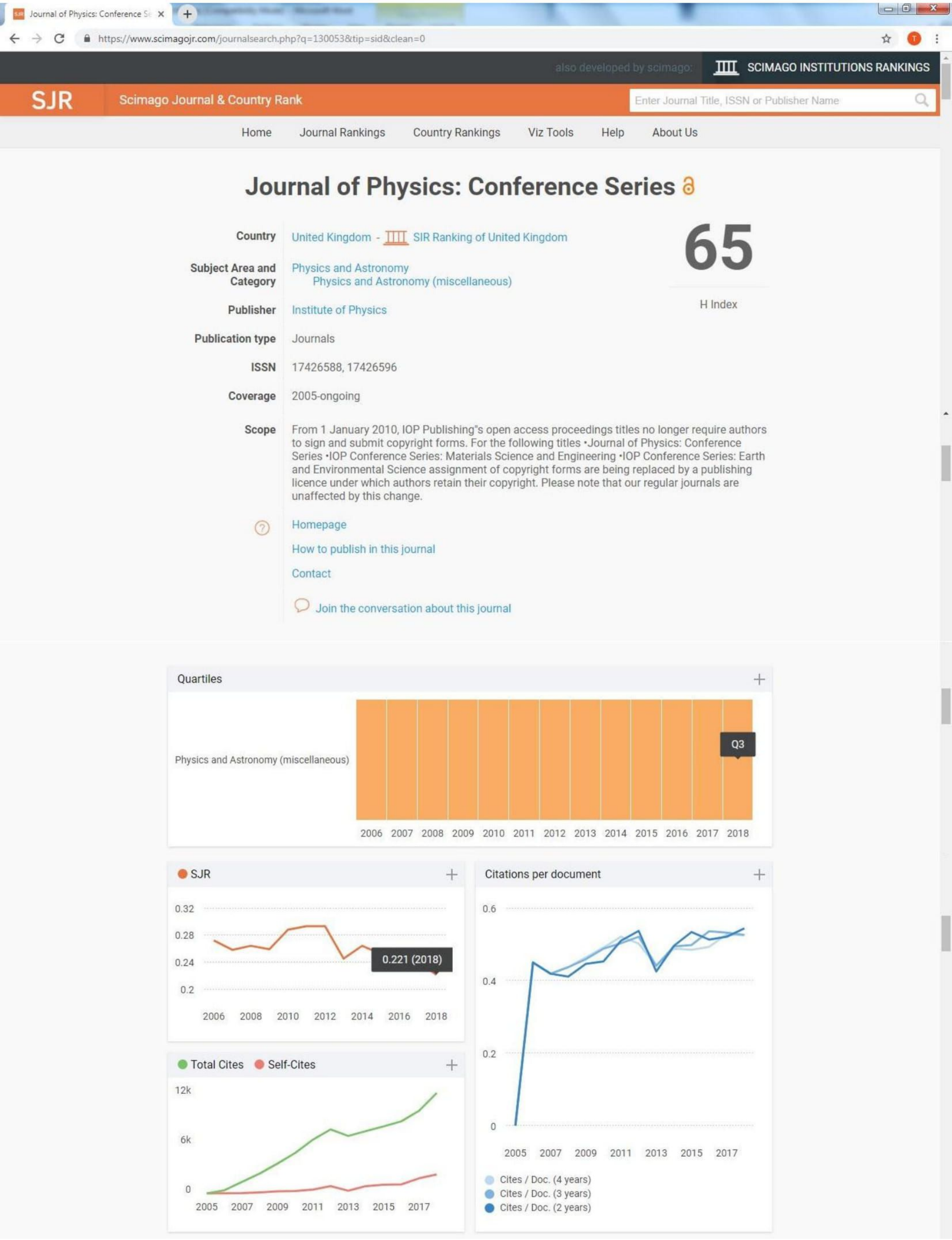

Journal of Physics: Conference Se
https://www.scimagojr.com/journalsearch.php?q=130053&tip=sid&clean=0
also developed by scimago:
SCIMAGO INSTITUTIONS RANKINGS
SJR
Scimago Journal & Country Rank
Enter Journal Title, ISSN or Publisher Name
Home
Journal Rankings
Country Rankings
Viz Tools
Help
About Us

Journal of Physics: Conference Series

Country	United Kingdom - SIR Ranking of United Kingdom
Subject Area and Category	Physics and Astronomy Physics and Astronomy (miscellaneous)
Publisher	Institute of Physics
Publication type	Journals
ISSN	17426588, 17426596
Coverage	2005-ongoing
Scope	From 1 January 2010, IOP Publishing"s open access proceedings titles no longer require authors to sign and submit copyright forms. For the following titles •Journal of Physics: Conference Series •IOP Conference Series: Materials Science and Engineering •IOP Conference Series: Earth and Environmental Science assignment of copyright forms are being replaced by a publishing licence under which authors retain their copyright. Please note that our regular journals are unaffected by this change.

65

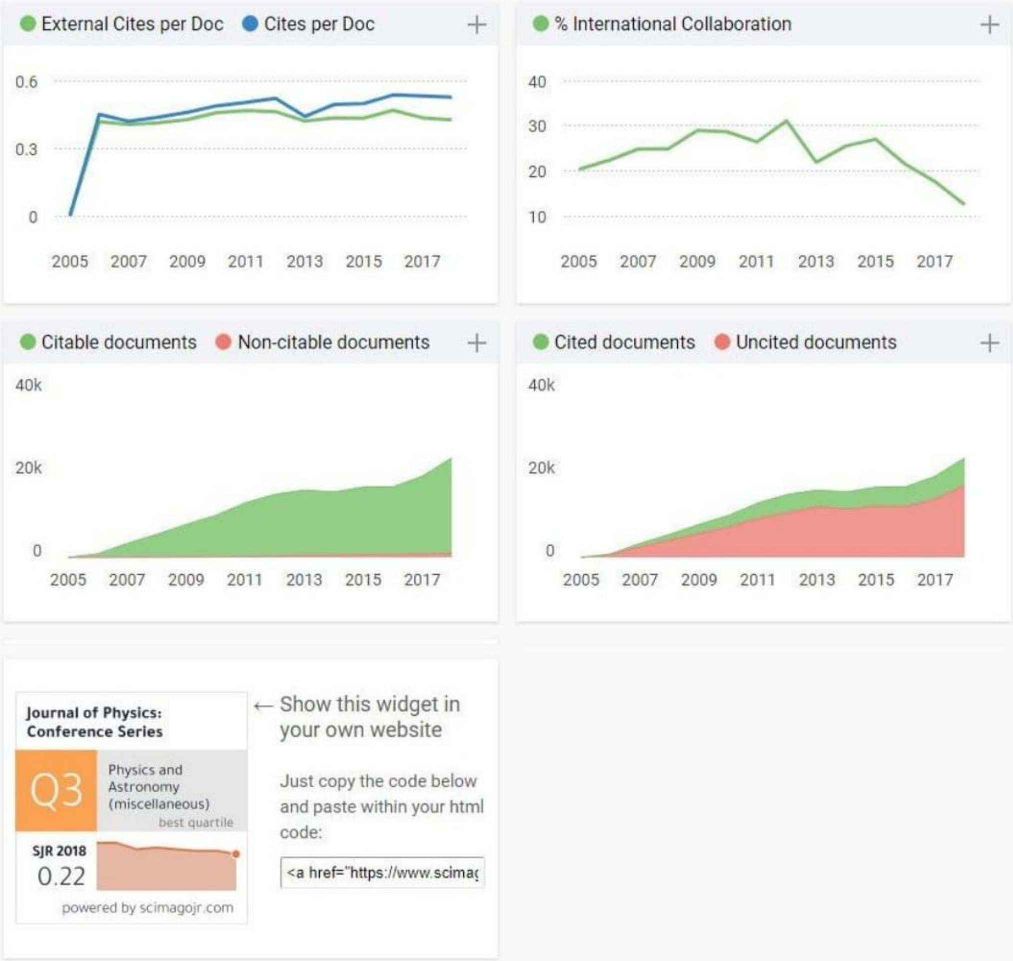
H Index

Homepage

How to publish in this journal

Contact

Join the conversation about this journal


● External Cites per Doc ● Cites per Doc
+
0.6
0.3
0
2005 2007 2009 2011 2013 2015 2017
● % International Collaboration
+
40
30
20
10
2005 2007 2009 2011 2013 2015 2017
● Citable documents ● Non-citable documents
+
40k
20k
0
2005 2007 2009 2011 2013 2015 2017
● Cited documents ● Uncited documents
+
40k
20k
0
2005 2007 2009 2011 2013 2015 2017
Journal of Physics: Conference Series
Q3
Physics and Astronomy (miscellaneous)
best quartile
SJR 2018
0.22
powered by scimagojr.com

← Show this widget in your own website

Just copy the code below and paste within your html code:

```
<a href="https://www.scimag
```

Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Appropriate CNN Architecture and Optimizer for Vehicle Type Classification System on the Toll Road

Appropriate CNN Architecture and Optimizer for Vehicle Type Classification System on the Toll Road

Windra Swastika, Martien Febriant Ariyanto, Hendry Setiawan, Paulus Lucky Tirma Irawan

Fakultas Sains dan Teknologi, Universitas Ma Chung, Villa Puncak Tidar N-01, Malang, Indonesia

windra.swastika@machung.ac.id

Abstract. The growth in the number of vehicles in Indonesia causes traffic jam problems, including on the toll roads. Traffic jam problem on the toll roads occurs because the users must stop and make payments in the toll gate. Government built an Automatic Toll Gate Shelter (or *Gardu Tol Otomatis*/GTO) as an effort to reduce this problem. However, GTO can only be used by certain type of vehicles only. In this study, we developed a system that can classify type of vehicles so that GTO can be used for various types of vehicles that cross the toll road. The developed system will receive vehicle input to be classified. The learning process to do the classification is using the Convolutional Neural Network (CNN). The CNN algorithm is trained first with 2,930 vehicle images divided into 1,794 vehicles type 1 (van, jeep, and pick-up) image, 507 vehicle type 2 images (truck with 2 axle), and 631 vehicle type 3 images (truck with 3 axle). From the experimental results of CNN architecture and various parameters of the architecture, the best accuracy is found on MiniVGGNet architecture which applies Adadelta optimization function and input image parameter 64x64 and epoch 40. The result obtained from the network has accurate evaluation or out-sample accuracy of 73%.

1. Introduction

Traffic jam is a major problem faced by various residents of large cities in various countries, including Indonesia. In an effort to overcome this traffic jam, toll roads have been built by the government through Jasa Marga. It is expected that the traffic jam that occurs can be reduced by the user of toll roads.

Recently, Indonesian Government issuing E-Toll Cards using Automatic Toll Gate (GTO). The GTO does not require a gate keeper, so there is no payment process that takes time and causes traffic jam. However, the GTO can only be passed by family vehicles (type 1 vehicle). This still causes traffic jam in toll gate.

Based on this problem, it is necessary to automatically classify the vehicle type so that the GTO can be used by all types of vehicles. There was a previous study of classification of vehicle type by Amaluddin et al (2015) using Gaussian Mixture Model and Fuzzy Cluster Mean, as well as research by Irfan et al (2017) and Wu et al (2001) who used Artificial Neural Networks (ANN). The similarity of the three studies is the preprocessing to take the features of the vehicle. The preprocessing causes the results of the ANN to be very dependent on preprocessing which is not always successful. Based on this reason, a system that can classify vehicle type without preprocessing is needed.

Convolution Neural Network (CNN) is one of the developments of ANN that specializes in image recognition. There was a study by Lu et al (2015) on the introduction of food types, research by Li et al

(2015) on face detection, and research by Ciresan et al (2016) on handwriting reading. The three studies were carried out on the basis of CNN deep learning without feature extraction.

This research will focus on creating a system prototype that can perform vehicle recognition based on digital images automatically. Moreover, the appropriate CNN architecture and optimizer in the classification system prototype will also be investigated.

2. Materials and Method

The process of developing a vehicle classification system starts with collecting data for training using Google Image. Data obtained from Google image will go through manual checks before it can be used as training data. The criteria of data training image is that the vehicle must be clear and not smaller than the background. The image that has been selected will be resized according to the needs of the CNN architecture.

To get appropriate network in this study, we use 4 parameters in the CNN. First parameter is the type of CNN architecture. There are three CNN architectures that will be tested in this study, namely ShallowNet, LeNet, and MiniVGGNet. ShallowNet is a shallow architecture that only has one convolution layer. The process in ShallowNet starts by obtaining an image as the Input Layer, then proceed with the convolution. After going through the convolution, the matrix arrangement of the convolution layer results is changed from 3-dimensional form to two dimensions in the Fully Connected Layer.

The second architecture used in this study is LeNet. LeNet is an architecture with 2 layers of convolution and 1 layer fully-connected. The process starts by obtaining an image as Input, then convolution is performed to produce a convolution layer. There is also a pooling layer to reduce the size of the matrix before the second convolution process. The last pooling is performed to reduce the size of the matrix in the previous process before finally converted from the 3-dimensional matrix to 2 dimensions in a hidden layer. The last process is to create a fully-connected layer to become the output layer.

The last CNN architecture used in this study is MiniVGGNet. MiniVGGNet has 6 convolution layers and 2 layers fully-connected. The process in MiniVGGNet starts by obtaining image as the Input Layer. The input image is then forwarded to 2 convolution layers, then followed by subsampling process to obtain pooling layer. This process is repeated twice as shown in the Figure 1. The last process is to change the pooling layer matrix from 3 dimensions to 2 dimensions, then forwarded the fully connected layer 2 times.

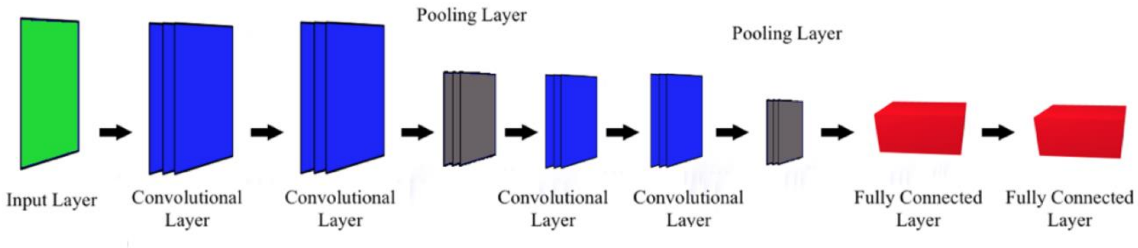


Figure 1. Architecture of MiniVGGNet.

The second parameter is the use of optimizer. There are 3 optimizers tested for each CNN architecture. They are Adadelta, Adam, and SGD. The third parameter is image sizes as input for the CNN. The image size that will be used are 32x32, 48x48, and 64x64. Finally, the network will also be tested with different numbers of epochs during the training process. Each combination of the four parameters will be a model of the results of the training.

Two types of data will be used in this study. The first data is the image obtained from Google Image and used as both training and testing data (in-sample data). The second data is obtained from photos taken directly using a smartphone for the evaluation process (out-sample data). All data grouped into 3 types: sedans, jeeps, pick-ups (group 1), trucks with 2 axles (group 2) and trucks with 3 axles (group 3).

The total number of first data is 2,932 images with the distribution of 1,794 images representing group 1, 507 images representing group 2 and 631 images representing group 3. The total numbers of second data is 150 images, with the distribution of 50 group 1 vehicle images, 50 group 2 vehicle images, and 50 group 3 vehicle images.

The classification accuracy is based on true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values. Accuracy percentage is obtained using the formula:

$$\text{Accuracy} = ((TP + TN) / (TP + TN + FP + FN)) * 100\% \qquad [1]$$

3. Results and Discussion

The choice of type of architecture for training data greatly affects the speed of network training. This is because of differences in the number and type of layers in each architecture. The training process for architectures with a small number of layers will be faster than many layers of architecture. In addition to the type of architecture, differences in training speed can also occur due to the use of Graphic Processing Unit (GPU) during the training process. Table 1 shows the speed comparison of training process using CPU and GPU from different input image size.

Table 1. Speed comparison of training process for 1 epoch (in second)

Image size	**ShallowNet**		**LeNet**		**MiniVGG**	
	GPU	CPU	GPU	CPU	GPU	CPU
32x32	0,79s	2,35s	0,91s	16,77s	3s	39s
48x48	1,39s	5,88s	1,55s	39,82s	7s	75s
64x64	2,24s	9,06s	2,46s	63,75s	12s	135s

As shown in table 1, each architecture shows different speed of training. One factor that affects the speed is the numbers of convolution layer and pooling layer in each respective architecture. ShallowNet has only 1 convolution layer and the training speed is the fastest compared to the other 2 architectures. On the other hand, MiniVGG has the largest numbers of convolution and pooling layers. It takes 3,7 times longer compared to ShallowNet for 1 epochs training using GPU and 16 times longer for 1 epochs training using CPU.

The use of GPU also greatly affects the training speed. For example, in training using the GPU for MiniVGG architecture, each epoch takes 3s, 7s and 12s for 32x32, 48x48 and 64x64 image size respectively. This training time is much faster compared to training using CPU, where each epoch takes 39s, 75s and 135s for 32 x 32, 48x48 and 64x64 image size respectively. This speed difference is large, but not as large as the speed difference using LeNet architecture.

For the classification accuracy, we use 2 types of evaluation. The first one is evaluation using in-sample. In this evaluation, we used 2,932 images (794 images of group 1 vehicle, 507 images of group 2 vehicle and 631 images of group 3 vehicle) for both training and testing. The accuracy percentage is calculated using [1]. Table 2 is the accuracy of in-sample data using ShallowNet, LeNet and MiniVGGNet with different optimizer. All networks were trained using 40 epochs with 64x64 image size.

Table 2. In-sample Accuracy

Architecture	**Optimizer**		
	Adadelta	**Adam**	**SGD**
ShallowNet	99,72%	99,69%	99,62%
LeNet	61,19%	99,62%	99,76%
MiniVGGNet	99,08%	98,19%	98,43%

The second evaluation is using out-sample data. This evaluation was conducted to verify the accuracy of the CNN network in the first evaluation. Evaluation was carried out using 150 vehicle images with the distribution of 50 images for each group. All evaluation data that will be used are new images that are excluded in the training process. The out-sample data is obtained by taken photo directly on the road. Examples of out-sample data are shown in the Figure 2.

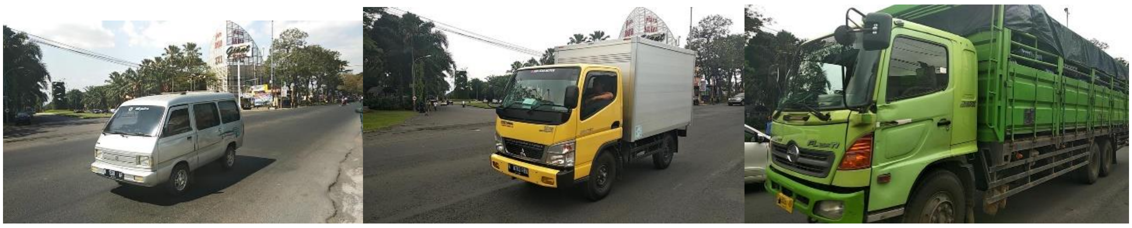


Figure 2. Examples of out-sample data. Left-right: group 1, group 2 dan group 3 vehicle

The accuracy of out-sample data is shown in the Table 3.

Table 3. Out-sample Accuracy

Architecture	**Optimizer**		
	Adadelta	**Adam**	**SGD**
ShallowNet	59%	58%	59%
LeNet	33%	62%	62%
MiniVGGNet	73%	71%	71%

Accuracy obtained from out-sample data as shown in Table 3 is lower compared to in-sample accuracy. This shows that the network can recognize images in the training data up to 99%. However, the network was not good at generalizing new images. The best accuracy for out-sample data is 77% which is obtained from MiniVGGNet with Adadelta optimizer.

There are several things that affect the accuracy for out-sample data. First is image resolution and orientation in the data training. This study uses training images obtained from Google Image. Those images have different resolution and orientation. This is a major problem because all training images will be resized to square resolution (32x32, 48x48, and 64x64). All training images with a square resolution will be resized proportionally. However, training images with rectangular size will distort and cause changes to the characteristics of the object (vehicle) itself. The second factor that affect the accuracy is the similarity of group 2 and 3 vehicles. CNN classifies using the characteristics obtained in the convolution process. Consequently, all images that are similar but have different group will confuse the network. This led to a lack of weight improvement during the training process and eventually failed to correctly classify the image.

4. Conclusion

Based on the results obtained in this study, it can be concluded several things as follows:

1. We have developed a system prototype that can automatically classify vehicle using the Convolutional Neural Network method.
2. Testing various architectures and parameters used for CNN network training shows that networks with 64x64 input image get higher accuracy than images with 32x32 and 48x48 inputs. In epoch parameters, networks with epoch 40 numbers produce better accuracy than networks with epochs 20 and 30. CNN with MiniVGGNet architecture gets better accuracy than CNN with the ShallowNet architecture.

3. Based on evaluation, the best CNN network is MiniVGGNet with Adadelta optimization function, 64x64 input image size, and 40 epochs. Up to 73% accuracy is obtained from the network evaluation.

References

[1] Amaluddin, F., Muslim, M.A. and Naba, A., 2015. Klasifikasi Kendaraan Menggunakan Gaussian Mixture Model (GMM) dan Fuzzy Cluster K Means (FCM). *Jurnal EECCIS*, *9*(1), pp.19-24.

[2] Ciresan, D.C., Meier, U., Gambardella, L.M. and Schmidhuber, J., 2011, September. Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (pp. 1135-1139). IEEE.

[3] Haykin, S., 1994. *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

[4] Irfan, M., Sumbodo, B.A.A. and Candradewi, I., Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, *7*(2), pp.139-148.

[5] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[6] Li, H., Lin, Z., Shen, X., Brandt, J. and Hua, G., 2015. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5325-5334).

[7] Lu, Y., 2016. Food Image Recognition by Using Convolutional Neural Networks (CNNs). *arXiv preprint arXiv:1612.00983*.

[8] Rosebrock, A. 2017, Deep Learning for Computer Vision with Python, 1st edition, PyImageSearch.

[9] Ruder, S., 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[10] Saverese, S. & Ermon, S., 2016. CS231n: Convolutional Neural Networks for Visual Recognition. [Online] Available at: http://cs231n.stanford.edu/ [Accessed 27 June 2018].

[11] Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.

[12] Wu, W., QiSen, Z. and Mingjun, W., 2001. A method of vehicle classification using models and neural networks. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd* (Vol. 4, pp. 3022-3026). IEEE.