

## Determining Optimal Architecture of CNN using Genetic Algorithm for Vehicle Classification System

Wahyono\*<sup>1</sup> and Joko Hariyono<sup>2</sup>

<sup>1</sup>Departemen of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia

<sup>2</sup>Corporation & Investment Board, Yogyakarta, Indonesia

e-mail: \*<sup>1</sup>[wahyo@ugm.ac.id](mailto:wahyo@ugm.ac.id), <sup>2</sup>[joko\\_hariyono@jogaprovo.go.id](mailto:joko_hariyono@jogaprovo.go.id)

### Abstrak

*Convolutional Neural Network (CNN) adalah pembelajaran mesin yang menghasilkan akurasi yang baik untuk banyak masalah di bidang visi komputer, seperti segmentasi, pendeteksian, pengenalan, serta klasifikasi. Namun, hasil dan kinerja sistem dipengaruhi oleh arsitektur CNN. Pada penelitian ini, kami mengusulkan pemanfaatan komputasi evolusioner menggunakan algoritma genetika untuk menentukan arsitektur CNN optimal dengan strategi transfer-learning dari jaringan induk. Selanjutnya, CNN optimal yang dihasilkan digunakan sebagai model untuk kasus sistem klasifikasi jenis kendaraan. Untuk mengevaluasi efektivitas pemanfaatan komputasi evolusioner untuk CNN, percobaan akan dilakukan dengan menggunakan kumpulan data klasifikasi kendaraan.*

**Kata kunci**—CNN, sistem klasifikasi kendaraan, transfer learning, algoritma genetika

### Abstract

*Convolutional neural network is a machine learning that provides a good accuracy for many problems in the field of computer vision, such as segmentation, detection, recognition, as well as classification systems. However, the results and performance of the system are affected by the CNN architecture. In this paper, we propose the utilization of evolutionary computation using genetic algorithm to determine the optimal architecture for CNN with transfer learning strategy from parent network. Furthermore, the optimal CNN produced is used as a model for the case of the vehicle type classification system. To evaluate the effectiveness of the utilization of evolutionary computing to CNN, the experiment will be conducted using vehicle classification datasets.*

**Keywords**—convolutional neural network (CNN), CNN architecture, evolutionary computing, genetic algorithm, classification system, vehicle type classification

## 1. INTRODUCTION

Among several methods of machine learning, convolutional neural network (CNN) is a very popular method because of its ability to solve problems in computer vision domains, namely among others in segmentation, detection systems, classification systems, and other computer vision and video analysis applications [1]. It is improved version of multilayer perceptron or other networks [2], [3]. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. A CNN consists of an input and an output layer, as well as multiple hidden

layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

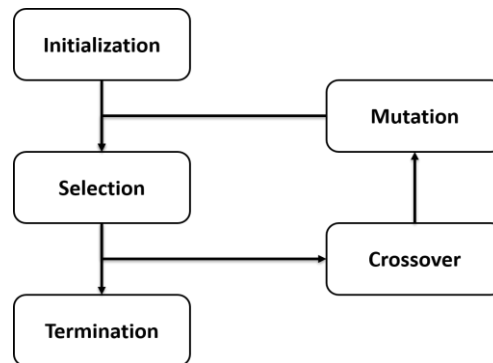


Figure 1. General framework of genetic algorithm

In many cases, CNN provides a very high level of accuracy compared to other machine learning methods, such as AlexNet for object classification, YOLO for object classification and segmentation, as well as GoogleNet for object recognition. The advantage of CNN compared to other machine learning methods is its ability to extract features automatically without human intervention. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

However, CNN's ability is very dependent on the architecture that is built [4]. The architecture of CNN is composed by the number of layers, the number and the size of convolution filters. In some cases, the deeper the CNN architecture (the greater the number of layers) will usually result in a better level of accuracy. However, there are some cases, the deeper the architecture, it will produce a worse level of accuracy. Therefore, determining the CNN architecture is a matter that needs attention before building a classification system that will be used later.

On the other hand, nowadays, evolutionary computing is increasingly being used to solve a problem with an optimization approach [5], [6]. One case is the use of evolutionary computing to determine the optimal parameters of a function [7]. When associated with CNN, the determination of CNN architecture is closely related to the selection of several parameters, including the number and size of the kernel, the number and type of layers and the type of activation function [3]. Therefore, the application of an evolutionary computational strategy can be applied in determining the optimal parameters in the formation of CNN architecture. This research proposes the utilization of evolutionary computation in determining the optimal CNN architecture for the case of vehicle type classification system.

Research on the type of vehicle classification system itself has been carried out by many previous researchers, [8], [9], and [10]. Bagus Pribadi [9] utilizes background modelling and image edge characteristics for vehicle type classification systems that are applied to CCTV video. Candradewi [8] utilizes a combined support vector machine classifier based on histogram of oriented gradient and Haar like-features. Whereas Muhammad Ifran [10] uses a multilayer perceptron to classify sample data by first extracting Haar-like feature-based features. Both researchers provide good accuracy. However, these results are strongly influenced by feature extraction methods used. To overcome this problem, convolutional neural networks can be one solution, where we do not need to determine the appropriate characteristics of training data manually. With CNN, the system will automatically search for the optimal features possessed by an object.

## 2. METHODS

### 2.1 General Framework of Genetic Algorithm

The genetic algorithm is one of optimization techniques inspired by natural genetics [6], [7]. Its search process is carried out between several alternative optimal individual based on a probabilistic function to obtain an optimal solution [5]. Compared to other optimization techniques, GA in many cases provides high accuracy as in the case of scheduling [11], financing [12], and even in robot control systems [13]. Therefore, the use of genetic algorithm in the case of CNN architecture selection is expected to provide optimal results with high accuracy. The framework of the GA used in this work for selecting optimal CNN is shown in Figure 1 and Algorithm 1.

---

#### Algorithm 1. Optimal CNN Architecture using genetic algorithm

---

- 1: **Generate**  $n$  initial child networks (ICN) as chromosomes using transfer learning strategy
  - 2: **Compute** accuracy of each ICN as fitness value
  - 3: **Sort** chromosome in descending order based on fitness value
  - 4: **repeat**
  - 5:   **for**  $i=1$  to  $k$  **do**
  - 6:     **select** randomly two chromosomes ( $ICN_1, ICN_2$ )
  - 7:      $child_i \leftarrow crossover(ICN_1, ICN_2)$
  - 8:      $child_i \leftarrow mutation(child_i)$
  - 9:     **compute** fitness value of  $child_i$
  - 10:   **end for**
  - 11: **merge**  $k$  new children to the population
  - 12: **sort**  $n+k$  ICN chromosomes in descending order based on fitness value
  - 13: **remove**  $k$  worst ICN chromosomes from the population
  - 14: **until** a fixed number of iteration
- 

At the initialization stage, there are five parameters which should be considered such as population size ( $n$ ), generation size (iteration process), probabilities of crossover and mutation ( $pc$  and  $pm$ ), and the length of chromosomes ( $l$ ).

Fixed number of iteration is used as the termination criteria in our algorithm in order to speed up the training process. In each iteration step,  $k$ -offspring (ICN chromosomes) are generated based on the crossover and mutation processes. These process will be explained further in section 2.4 and 2.5. These new offspring are added to the population and then sorted out together with the previously  $n$  chromosomes in descending order. To maintain the quality of the population,  $k$  worst ICN chromosomes are removed from the population. Hence, the number of population member is not changing in each iteration step (i.e. generation).

Reproduction is performed by considering crossover and mutation operators according to their probability values. Furthermore, individuals from the initial and new offspring populations of crossover and mutation results are combined for the selection process. Selection is done by calculating the fitness of each individual. The best individuals are individuals who have the best fitness after a fixed number of iteration. The detail of each process will be explained in the next subsection for the case of determination of optimal architecture of CNN.

### 2.2 Chromosome Initialization Based on Transfer Learning

In this stage, we conducted chromosome initialization based on transfer learning from pre-trained CNN. The pre-trained CNN is regarded as parent network, while the chromosome networks are regarded as child network in initial population. Technically speaking, it is possible for us to create completely new initial networks in the population with random weight initialization. However, since we have limited dataset and resources to train the network which demands huge dataset to obtain a good accuracy, partial transfer learning is utilized to adopt the

partial knowledge from parent networks. Intuitively, this is in accordance with human behavior that can solve a problem only with some limited knowledge.

Initial child networks (ICN) are formed based on grandparent network (pre-trained CNN) by acquiring partial information (partial transfer learning). In this case, we transferred the partial value of filter from parent network as initial value of filter in the child network. Figure 2 depicts the illustration how the initial child networks are formed from parent network. Supposed that we have parent network consists of input layer, three convolution layers, one full connected layer, and output layer. Each convolution layer has five filters with a certain size. In forming ICN, the number of layers could be the same or fewer with the parent network. However, the number of filters adopted from grandparent network to the ICN would be fewer than parent network. In the example, each child network only contains 3 filters on the convolution layer with same size and initial value. In this research, we randomly select the number of filters to be transferred when creating the child network.

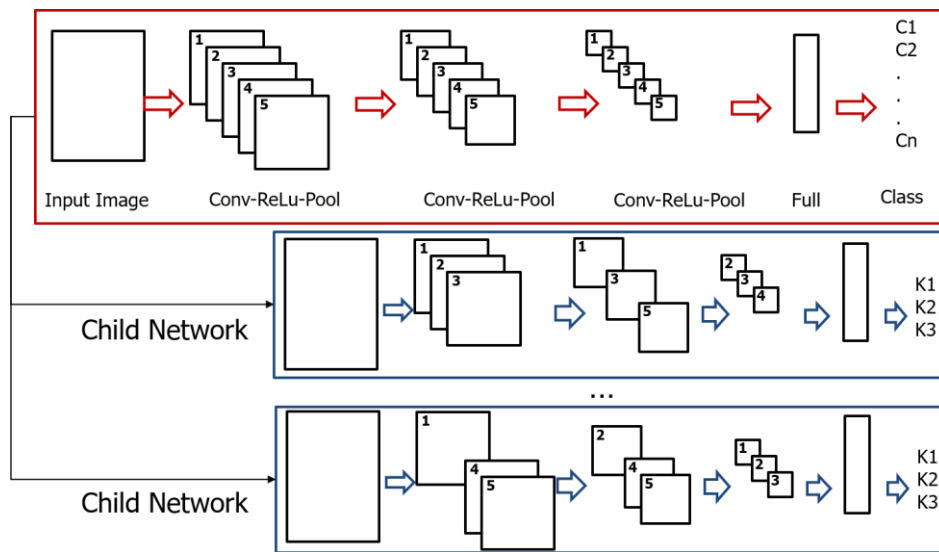


Figure 2. Child network process to initialize population of GA

In addition, the number of child network in initial population will be 10 chromosomes. Each chromosome is represented by three pairing-genes. Each gene represents the structure of convolution layer, while the first and the second number of each pairing-gen are representing the size and the number of filters. For example, chromosome {7 10, 5 12, 3 15} is representation of the child network with three convolution layers. The first layer contains 10 filters with size of 7x7, the second layer 12 filter with size of 5x5, while the third layer composed of 15 filters with size of 3x3. To be noted that we defined one layer of network as composition of convolution-ReLu and max pooling layer.

### 2.3 Fitness Value Calculation and Selection

The fitness value of each chromosome is calculated by conducting training strategy on child network with 100 epochs. In this case, we apply them on the problem of vehicle types classification system. The chromosomes are then sorted out based on their fitness value. It is utilized to determine whether the object candidates are target objects in the system. There are four classes used in our classification module, namely human, vehicle, baggage, and other objects. Noted that although we have four classes in object classifiers, each object is assigned as target for a certain monitoring task. For instance, in abandoned object detection, the target object is baggage, while in illegally parked vehicle detection, the target object is vehicle. Here, we use object vehicle as illustration.

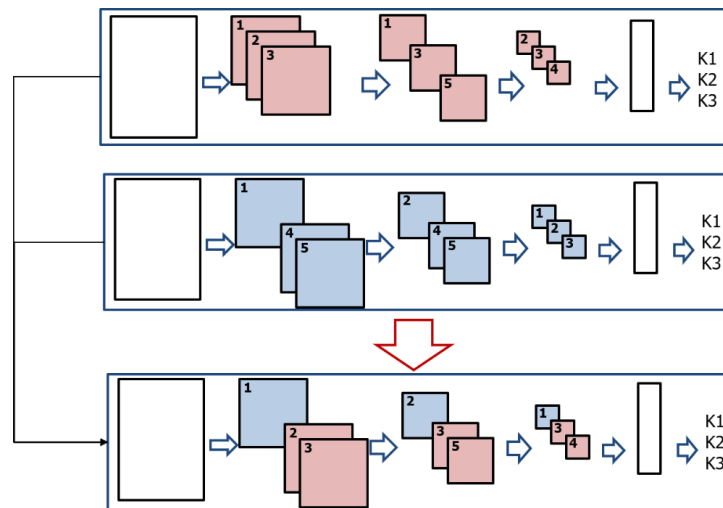


Figure 3. Crossover process between two individual networks

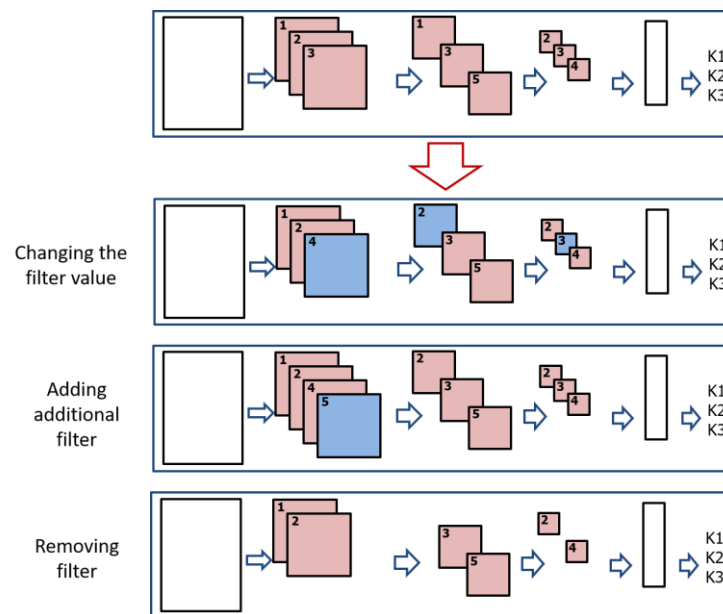


Figure 4. Mutation on individual networks with three possible processes such as changing the filter value, adding additional filter, and removing filter

The fitness value calculation is based on accuracy computation. In the calculation, true classification (TC), false classification (FC) and accuracy were used as measurement protocols. TC is defined as the number of data which are corrected classified, for instance sedan image is classified as sedan. FC is defined as the number of data which are wrongly classified, for instance sedan image is classified as bus. Accuracy is defined as ratio between TC and the total of data in the testing set.

The selection process is done to obtain the best chromosome in the population for next generation. The selection process is performed by elitism, which sorts out the chromosomes based on fitness value from the highest to the lowest ones and then keeps  $n$  best chromosome (child networks) as a new population in the next generation, while the remaining chromosomes will be removed. After the selection process, the best chromosome with the highest fitness value is selected as optimal CNN on each generation.

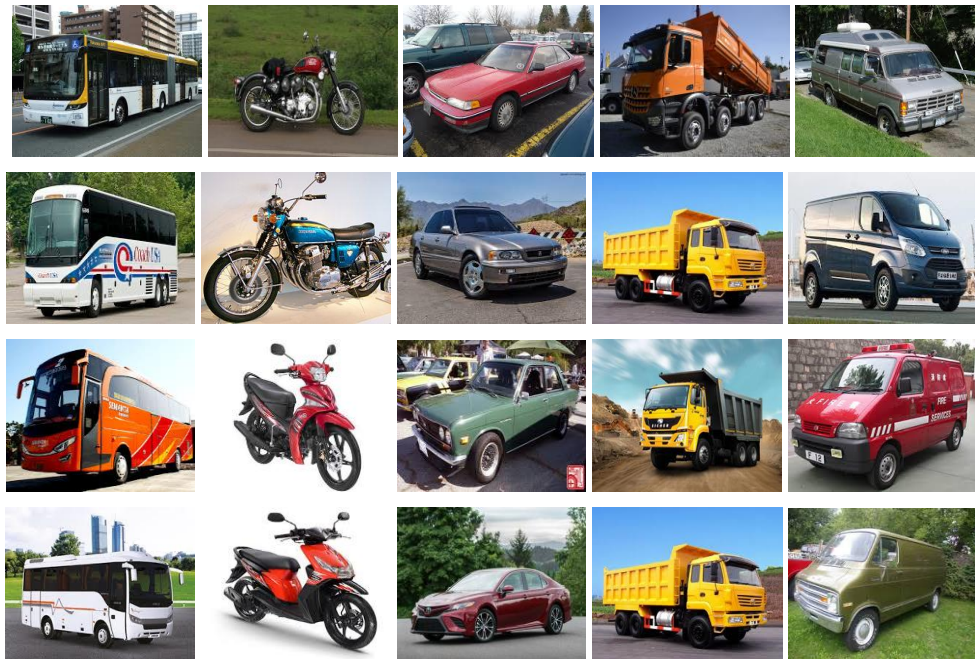


Figure 5. Selected samples of vehicle classes (bus, motorcycle, sedan, truck, and vans)

#### 2.4 Crossover Operation

Crossovers are used to generate new network with different genes from the previous individual i.e. parents. In this work, the crossover is done by randomly swapping the weight values of certain filter between two individual networks in certain layer location. The process is possible to be done since the structure of individual network are the same e.g. the size of filter. Figure 3 illustrate how crossover process between two individual networks carried out.

#### 2.5 Mutation Operation

The mutation operator on the genetic algorithm modifies an offspring (a child's chromosome) after crossover operation. The mutation is performed based on probability. In this work, the mutation process is done by either randomly changing the value of filter; adding additional filter or removing the filter in certain layer. Figure 4 shows the illustration of mutation process on individual network.

Table 1. Initial Chromosomes (Child Networks)

| No. | Layer 1  | Layer 2   | Layer 3   | Layer 4   | Layer 5   |
|-----|----------|-----------|-----------|-----------|-----------|
| 1   | [3x3,10] | [3x3, 15] | N/A       | N/A       | N/A       |
| 2   | [3x3,05] | [3x3, 10] | [3x3, 15] | N/A       | N/A       |
| 3   | [3x3,10] | [3x3, 15] | [3x3, 20] | N/A       | N/A       |
| 4   | [3x3,10] | [3x3, 10] | [3x3, 12] | [3x3, 12] | N/A       |
| 5   | [5x5,10] | [5x5, 15] | [3x3, 10] | [3x3, 15] | N/A       |
| 6   | [5x5,10] | [5x5, 12] | [5x5, 15] | N/A       | N/A       |
| 7   | [5x5,10] | [5x5, 12] | [5x5, 15] | [5x5, 10] | N/A       |
| 8   | [3x3,15] | [3x3, 10] | [3x3, 10] | [3x3, 05] | N/A       |
| 9   | [3x3,20] | [3x3, 15] | [3x3, 10] | [3x3, 05] | [3x3, 05] |
| 10  | [5x5,20] | [5x5, 15] | [5x5, 10] | [5x5, 05] | [5x5, 05] |

### 3. RESULTS AND DISCUSSION

#### 3.1 Experiment Setup and Parameter Setting

The proposed method has been implemented using C++ programming language using PC under Windows operating system. The utilization of GA to obtain the optimal architecture of CNN has been implemented in vehicle type classification problem. The problem contains five classes of vehicles such as bus, motorcycle, sedan, truck, and vans, as shown in Figure 5. We collected the dataset from internet in which each class contains 200 samples with size 64x64 pixels. The dataset which is then divided into training and testing data with the proportion of 75% and 25%.

Furthermore, we generate 10 chromosomes (initial child networks) as initial population. The child network is formed by using transfer learning strategy from AlexNet which have small number of layer. On each initial child network (ICN) the minimum number of layers is 2 while the maximum number of layers is 5. Table 1 shows 10 initial child networks which are generated in our experiments.

Table 2. Determining optimal epoch

| Epoch | Accuracy (%) |       |         |
|-------|--------------|-------|---------|
|       | Min          | Max   | Average |
| 10    | 60.00        | 80.00 | 77.00   |
| 15    | 64.00        | 86.00 | 78.50   |
| 20    | 68.00        | 90.00 | 81.50   |

#### 3.2 Optimal Epoch for Fitness Value Calculation

As mentioned before, fitness value is calculated by performing CNN network using certain architectures and computing the accuracy for those network. Considering this process, we need to determine first the optimal number of epoch in training data. We conduct CNN process on initial child networks (population) by setting three epoch values, such as 10, 15 and 20. These values are chosen due to resource constraint. Accuracy is calculated using the average of 3-fold cross-validation. Table 2 shows the results of the determining epoch number from 10 initial child networks which are generated in initial population stage. It can be seen that the increasing the number of epoch obtained higher accuracy. Based in this experiment, for performing fitness value calculation we set the number of epoch become 20.

### 3.3 Determining Crossover and Mutation Probabilities

In the next experiment, the crossover probability (pc) and the mutation probability (pm) were examined by combining the probability value of both operators. Both probabilities were set to the combination of 0.3, 0.5, and 0.8, respectively. So there will be 9 experiments. Next, due to resource limitation, we only utilize 5 population size for determining the optimal probabilities of crossover and mutation processes. Each combination pc and pm sizes is tested 3 times and we select the average of the highest fitness value. It can be seen in Table 3 that the highest fitness value is 0.8167 with a crossover probability value of 0.5 and a mutation probability of 0.8. Therefore, we use these parameters for the rest of our experiments.

Table 3. Results in combination of crossover and mutation Probabilities

| $i^{\text{th}}$ trial | PC  | PM  | Fitness Value |
|-----------------------|-----|-----|---------------|
| 1                     | 0.3 | 0.3 | 0.7633        |
| 2                     | 0.3 | 0.5 | 0.7933        |
| 3                     | 0.3 | 0.8 | 0.7967        |
| 4                     | 0.5 | 0.3 | 0.7633        |
| 5                     | 0.5 | 0.5 | 0.7789        |
| 6                     | 0.5 | 0.8 | 0.8167        |
| 7                     | 0.8 | 0.3 | 0.7911        |
| 8                     | 0.8 | 0.5 | 0.7511        |
| 9                     | 0.8 | 0.8 | 0.6767        |

### 3.4 The effect of mutation process

In the mutation stages, there are type types processes conducted in our experiment such as changing the filter value, adding new filter, and removing filter in the convolutional network. Based on our experiment, changing the filter value gains better accuracy of 75% that two other process types of 65% and 67%, respectively. Therefore, changing the filter value will have the

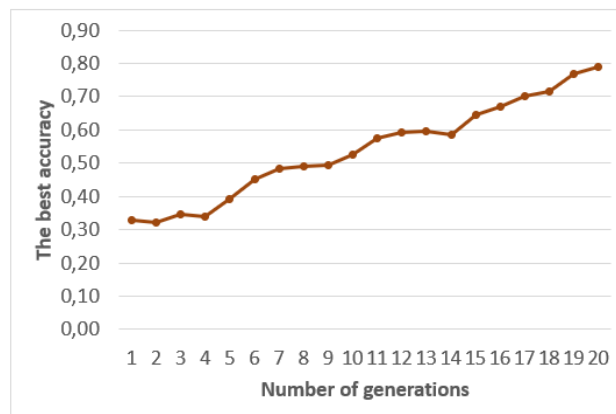


Figure 6. Training results on Genetic Algorithm

highest probability to be chosen comparing to other types in mutation stage.

### 3.5 Results on Training with Genetic Algorithm

Since the genetic algorithm is totally random, we conducted three trials on genetic algorithm training in order to get optimal CNN. The experiment is run until 20 generations in which we evaluate the best accuracy on each generation. The experiments of GA were conducted to evaluate the best fitness value on each generation by setting crossover and mutation probabilities as 0.5 and 0.8, obtained from previous experiment. Figure 6 depicts the



average accuracy on training versus the number of generations. As we can see, the average accuracy is getting increase while the number of generations is increasing. However, in certain generation, the accuracy is also decreased (e.g. 13th generation to 14th generation). Overall after conducting 20 generation, the training obtains the average accuracy as much as 79% from three trials. From this experiment, we obtain the optimal network is {(3,10), (3,15), (3,20)}.

### 3.6 Results on Vehicle Classification Problem

After obtaining optimal architecture of CNN in the previous experiment, it is then used to evaluate the testing data of vehicle classification problem. Table 4 shows the confusion matrix of vehicle classification problem. As shown in Table 4, the experiment found that the average accuracy of ICN obtained was 70.40%. Among five classes, the motorcycle class has the greatest accuracy of 88% compared to other classes. This is because the motor class is more visually different than other classes. Furthermore, because motorcycle has different appearance that others, other type of vehicles is not classified as motorcycle. On contrary, Bus and Vans classes have the lowest accuracy, since they are visually similar.

Table 4. Confusion matrix of vehicle classification problem

|            | Bus | Vans | Truck | Motorcycle | Sedan |
|------------|-----|------|-------|------------|-------|
| Bus        | 60% | 20%  | 10%   | -          | 10%   |
| Vans       | 28% | 54%  | 14%   | -          | 4%    |
| Truck      | 6%  | 14%  | 72%   | -          | 8%    |
| Motorcycle | -   | 2%   | 6%    | 88%        | 4%    |
| Sedan      | 4%  | 8%   | 10%   | -          | 78%   |

## 4. CONCLUSIONS

The utilization of genetic algorithm to obtain optimal architecture of CNN has been conducted. The process has been done to solve the problem of vehicle type classification. The framework consists of four main stages: chromosome initialization, selection, crossover, and mutation processes. Based on the experiment, the optimal architecture of CNN after implementing GA with 10 populations and 20 generation achieves 70.40% accuracy for classifying the vehicle type. The motor class has the greatest accuracy compared to other classes. This is because the motor class is more visually different than other classes.

## ACKNOWLEDGEMENTS

This research was supported by Research Grant of Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia (BPPTN Funding Grant No. 0092/J01.1.28/PL.06.02/2018).

## REFERENCES

- [1] E. P. I. W. Suartika, A. Y. Wijaya, and Soelaiman, R., "Image Classification Using Convolutional Neural Network (CNN) on Caltech 101", *Jurnal Teknik ITS* Vol. 5, No. 1, 2016. ISSN: 2337-3539 (2301-9271 Print)

- [2] E. S. Marquez, J. S. Hare, and M. Niranjana, "Deep Cascade Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol.29, no. 11, pp. 5475-5485, November 2018. Available: <https://ieeexplore.ieee.org/document/8307262>.
- [3] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, vol.29, no.9, pp. 2352-2449, September 2017. Available: <https://ieeexplore.ieee.org/document/8016501>.
- [4] M. Zufar and B. Setiyono, "Convolutional Neural Networks for Real-Time Face Recognition", *Jurnal Sains dan Seni ITS* Vol. 5 No. 2, pp.2337-3520 (2301-928X Print), 2016.
- [5] J. He and G. Lin, "Average Convergence Rate of Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, vol.20, no.2, pp.316-321, 2016. Available: <https://ieeexplore.ieee.org/document/7122298>.
- [6] D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre, "Level-Based Analysis of Genetic Algorithms and Other Search Processes". *IEEE Transactions on Evolutionary Computation*, vol.22, no.5, pp.707-719, 2018. Available: <https://ieeexplore.ieee.org/document/8039236>.
- [7] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic Learning Particle Swarm Optimization", *IEEE Transactions on Cybernetics*, vol.46, no.10, pp. 2277-2290, 2016. Available: <https://ieeexplore.ieee.org/document/7271066>.
- [8] I. Candradewi, "Video processing for vehicle classification based on support vector machine", Master thesis, Graduate School of Computer Science, Universitas Gadjah Mada, Yogyakarta, 2015.
- [9] B. Pribadi and M. Naseer, "Vehicle Type Classification System Through Digital Image Technique", *SETRUM – Volume 3, No. 2, December 2014*
- [10] M. Irfan, "Digital Image-Based Vehicle Classification System with the Multilayer Perceptron Method", *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, Vol.7, No.2, October 2017, pp. 139-148
- [11] A. K. Wardana and S. Hartati, "Scheduling System of Pencak Silat Based on Genetic Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 11, no. 2, p. 177, July 2017 [Online]. Available: <https://jurnal.ugm.ac.id/ijccs/article/view/24214>. [Accessed: 24-September-2018]
- [12] Wahyono, C. Puspitasari, M. D. Fauzi, K. Kasliono, W. S. Mulyani, and L. Kurnianggoro, "An Optimal Stock Market Portfolio Proportion Model Using Genetic Algorithm" *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 12, no. 2, July 2018 [Online]. Available: <https://jurnal.ugm.ac.id/ijccs/article/view/36154>. [Accessed: 24-December-2018]
- [13] K. Hakiim, A. Darmawan, and Faizah, "Optimization of PID Control using Genetic Algorithms for Quadrotor Flights," *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 7, no. 2, October 2017 [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/2321>. [Accessed: 29-September-2018]