

# Chaotic Whale Optimization Algorithm in Hyperparameter Selection in Convolutional Neural Network Algorithm

Akhmad Ridho <sup>1,\*</sup>, Alamsyah <sup>1</sup>

<sup>1</sup>Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Semarang, Indonesia  
\*Corresponding author: sase240@students.unnes.ac.id

## ARTICLE INFO

## ABSTRACT

**Article history**  
Received 22 October 2022  
Revised 31 October 2022  
Accepted 5 November 2022

**Keywords**  
Metaheuristic  
CNN Hyperparameters  
MNIST  
FashionMNIST  
CWOA

In several previous studies, Metaheuristic methods were used to search for CNN Hyperparameters. However, this research only focuses on searching for CNN Hyperparameters in the type of network architecture, network structure, and initializing network weights. Therefore, in this article, we only focus on searching for CNN Hyperparameters with network architecture type and network structure with additional regularization. In this article, the CNN hyperparameter search with regularization uses CWOA on the MNIST and FashionMNIST datasets. Each dataset consists of 60,000 training data and 10,000 testing data. Then, during the research, the training data was only taken 50% of the total data, then, the data was divided again by 10% for data validation, and the rest for training data. The results of the research on the MNIST CWOA dataset have an error value of 0.023 and an accuracy of 99.63. Then, the FashionMNIST CWOA dataset has an error value of 0.23 and an accuracy of 91.36.

This is an open access article under the [CC-BY-SA](#) license.



## 1 Introduction

Convolutional Neural Network (CNN) is a group of Deep Neural Networks to complete computer vision tasks (Buda et al., 2018; Y. Wang et al., 2019). The method can process large-scale data and build models of hierarchical data with complex distribution (Xie & Tian, 2019). With this ability, in the last few years research (2014; 2016; 2017) on topics other than computer vision, CNN is applied as a method of solving problems on that topic. Accordingly, the new CNN architecture also completed the task in several research topics and was more accurate when compared to the old architecture (Chen & Liu, 2019; Conneau et al., 2016; Qian et al., 2016). Plus, the support for Graphics Processing Unit (GPU) technology makes it easy to quickly implement the new CNN architecture on big data (Zhu et al., 2017).

The development of CNN architecture leads to deep network design, for example, AlexNet (Hinton et al., 2012) has a deeper architecture than LeNet (Lecun et al., 1998), as well as VGG (Simonyan & Zisserman, 2014), and GoogleNet (Szegedy et al., 2015). The advantage of designing a deep network architecture is to get a better feature representation, but it makes the network complex, difficult to optimize, and easy to overfit (Zhang et al., 2019). Moreover, the selection of hyperparameters from CNN as well as in Neural Network (NN) also affects the results obtained. In connection with that, in designing and using CNN, knowledge, experience, and expertise in the data field are required for optimal results (Bibaeva, 2018; B. Wang et al., 2018).

In 2018 to solve this problem, researchers Lee et al. (2018) et al applied a parameter-settings-free harmony search algorithm to find the optimal hyperparameters of the CNN method. In this study, the architecture is fixed and the hyperparameter search is limited to setting the convolutional layer and the pooling layer. Then, Strumberger et al (2019) in their research proposed the Tree Growth Algorithm (TGA) method to search for hyperparameters from CNN. In this study, hyperparameters focused only on managing the structure and architecture of the CNN

method. As a result, the research method has the smallest error value compared to several state-of-the-art methods. Furthermore, Sun et al (2020) in their research developed an automatic approach to developing the architecture and initialization of the weights of the CNN method for image classification problems. This approach uses an optimized Genetic Algorithm (GA) method. This research focuses on searching for architectural hyperparameters and initializing the CNN method weights. As a result of the research, the optimized Genetic Algorithm (GA) method obtained superior results in several datasets when tested using several state-of-the-art methods. Next, researchers Bacanin et al (2020) proposed two methods for selecting hyperparameters from CNN, namely Tree Growth with increased exploitation capabilities and Firefly algorithms with increased exploitation and exploration capabilities. In this research, hyperparameter search focuses on the number of convolutional layers, the number of filter convolutional layers, the size of the filter convolutional layer, the number of classification layers, and the number of hidden units of the classification layer. The results of the study, the method used in the study got the smallest error value. After that, researchers Ayla Gülcü & Zeki Kuş (2020) used a modified Microcanonical Optimization Algorithm (MOA) to select optimal hyperparameters on CNN. The results of this study were obtained using two approaches using constraints and not using limitations on hyperparameter searches. Then, the modified MOA method obtained the highest accuracy results in most of the datasets used in this study.

Some of the research results only focus on searching for hyperparameter network architecture, network structure, and initializing network weights on the CNN method. It makes previous research has limitations by not adding regularization and performing hyperparameter search of regularization on CNN. Therefore, this study focuses on searching for hyperparameter network architecture, network structure, regularization used in the CNN method network, and searching using different methods. The purpose of this research is to build a framework along with knowing the level of accuracy and error of the CNN method hyperparameter search automation using the Chaotic Whale Optimization Algorithm (CWOA) method which is included in the metaheuristic intelligence swarm class.

## 2 The Proposed Method/Algorithm

The selection of optimal hyper-parameters on the CNN) has an impact on the results obtained. This is evidenced by the deeper development of architecture. With the deeper architectural design, you will get the advantages of better feature representation, but it will have an impact on network complexity and easy overfitting (Zhang et al., 2019). Plus using and designing the CNN architecture to solve problems that require knowledge and experience in the data field (Bibaeva, 2018; B. Wang et al., 2018). Researchers in overcoming these problems propose the CWOA method for automating the selection of hyperparameters from the CNN. The stages of the proposed method are in the form of a flowchart in Figure 1.



**Figure 1.** Flowchart proposed method

### 3 Method

#### 3.1 Convolutional Neural Network (CNN)

In the CNN method, there are three components used as the main layer in the learning process (Alom et al., 2018), including the Convolutional Layer, Sub-Sampling Layer, and Classification Layer. Convolutional Layer is a feature retrieval process that is useful for detecting certain patterns in the image. The way the process works is by changing the value of the image into the form of a filter matrix that is used during learning. The filter is used to detect the presence of certain features or patterns in the image. Furthermore, the Sub-sampling Layer is a layer that has a function as down sampling on the input feature maps from the previous layer. Next, the Classification Layer is

the last layer of CNN for the classification process of features that have been processed in the previous layer.

### 3.2 Regularization CNN with Dropout and Early Stopping

In the CNN training method, one way to speed up the training process, avoiding overfitting, is to use regularization. Therefore, in this study, to speed up the training process, two regularizations were used, it called Dropout and Early Stopping. According to Prechelt (2012) in Early Stopping the learning process of the CNN method was stopped when the error from the data validation increased. Then, the network weights use the value before being stopped during the learning process is stopped. In addition to using early stopping regularization, this study also uses Dropout regularization. The Dropout function avoids overfitting. This is evidenced in the study of Wu & Gu (2015) on the use of dropout in the CNN method and resulted in improved performance of CNN on several datasets compared to some state-of-the-art methods at that time.

### 3.3 Whale Optimization Algorithm (WOA)

In their research, Mirjalili & Lewis (2016) proposed the WOA as one of the methods belonging to the Metaheuristic group. The idea for this method came from observing the behavior of humpback whales in hunting prey. Described in the study, the working process of the WOA method consists of several stages, it is Encircling Prey, Bubble-net Attacking Method, and Search for Prey. Each of these stages represents the process of humpback whales in searching for prey. Then, from this study, it was found that the WOA method is quite competitive in balancing exploration and exploitation abilities when compared to several algorithms, such as PSO, GSA, DE, FEP. Then, for more details, the pseudocode of WOA can be seen in Table 1.

**Table 1.** Pseudocode WOA

Whale Optimization Algorithm
Initialize the search agent population $X_i (i = 1, 2, \dots, n)$
Calculate the <b>fitness</b> of each search agent
$X^*$ = best search agency
<b>while</b> ( $t <$ maximum number of iterations)
<b>for</b> any search agent
Update $a, \vec{A}, \vec{C}, l$ , and $p$
<b>if1</b> ( $p < 0.5$ )
<b>if2</b> ( $ A  < 1$ )
Update the current search agent position with the equation below
$\vec{D} = \left  \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right $
<b>else if2</b> ( $ A  \geq 1$ )
Choose a search agent randomly ( $X_{rand}$ )
Update current search agent position with equation below
$\vec{X}(t + 1) = X_{rand} - \vec{A} \cdot \vec{D}$
<b>end if2</b>
<b>else if1</b> ( $p \geq 0.5$ )
Update current search agent position with equation

$$\vec{X}(t + 1) = \vec{D} \cdot e^{bl} \cdot \cos \cos(2\pi l) + \vec{X}^*(t)$$

**end if1**

**end for**

Check if any search agent is passing through the search space and change it according to the value of  $a$

Calculate the **fitness** of each search agent

Update  $X^*$  if there is a better rating

$$t = t + 1$$

**end while**

**return  $X^*$**

### 3.4 Chaotic Whale Optimization Algorithm (CWOA)

The CWOA is a development of WOA using Chaos theory. The combination resulted in a significant performance increase in the exploration and exploitation capabilities of WOA (Kaur & Arora, 2018). By increasing this capability, avoids premature convergence in the local optimal solution and does not waste time in finding the global optimal solution in the metaheuristic method (Gharehchopogh & Gholizadeh, 2019). The increase in this ability is influenced by Chaos theory. Chaos theory is a deterministic equation that produces random motion (W. Z. Sun & Wang, 2017). Chaos theory in WOA in the form of chaotic maps. The list of chaotic maps used in this study can be seen in the Table 2.

**Table 2.** List Chaotic Map

Name	Chaotic map
Logistic	$x_{i+1} = ax_i(1 - x_i)$
Cubic	$x_{i+1} = ax_i(1 - x_i^2)$
Sine	$x_{i+1} = \frac{a}{4} \sin \sin(\pi x_i)$
Sinusoidal	$x_{i+1} = ax_i^2 \sin \sin(\pi x_i)$
Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 -$
Circle	$x_{i+1} = \text{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right) \sin \sin(2\pi x_k)\right)$
Iterative	$x_{i+1} = \sin \sin(a\pi/x_i)$
Tent	$x_{i+1} = \left\{ \frac{x_i}{0.7}, x_i < 0.7 \frac{10}{3}(1 - x_i), x_i \geq 0.7 \right\}$
Piecewise	$x_{i+1} = \left\{ \frac{x_i}{p}, 0 \leq x_i < p \frac{x_i - p}{0.5 - p}, p \leq x_i < 0.5 \right\}$

Gauss/mouse

$$x_{i+1} = \{1, x_i = 0 \frac{1}{\text{mod}(x_i, 1)}, \text{ otherwise}$$

### 3.5 Overview Modification of Parameter Settings Free Harmony Search (PSFHS) Algorithm

The Harmony Search is a metaheuristic algorithm inspired by imitating artificial phenomena from the analogy of musical improvisation performance by musicians so that harmony becomes good (Geem et al., 2001). Having a simple concept, fast convergence, parameters that are not too difficult to set, simple models and concepts are the advantages of the HS algorithm (W. Sun & Chang, 2015). Then, research by Geem & Sim (2010) modified the method by adding steps to minimize the process of searching for the best parameters manually. The added stages are three, namely random tuning, rehearsal, and performance. Then, continued in the research by Lee et al. (2018) modified the performance stage to solve the CNN hyperparameter problem on the MNIST and CIFAR-10 datasets.

### 3.6 Research Dataset

This study using the MNIST dataset with a setting like this study (Gulcu & Kus, 2020). The distribution of data is done randomly with the amount of data taken 50% of the total training data in the dataset. Then, the data is divided into two more parts with a size of 10% taken for validation data and the rest for training data.

### 3.7 Hyperparameter Limitations and Architecture Setup

To reduce large computational resources, this research requires a search space limitation to find the optimal hyperparameter value from CNN. The limitations of this study are modified from research (Gulcu & Kus, 2020) and can be seen in Table 3. In addition, some hyperparameters are fixed and do not change. These hyperparameters can be seen in Table 4.

**Table 3.** Hyperparameter Limitations CNN

Name	Hyperparameter	Value
Convolutional layer	Filter size	3, 5, 7
	Feature map	32, 64, 96, 128, 160, 192, 224, 256
Subsampling layer	Filter size	2, 3
	Pooling Type	Max, Average
Classification layer	Number of Neurons	64, 128, 256, 512
Learning Process	Activation Function	ReLU, Leaky-ReLU
Regularization	Dropout rate	0.3, 0.4, 0.5

**Table 4.** Parameter Training CNN

Hyperparameter	Value
Batch size	64
Learning rate	1e-7
Optimizer	Adam
Loss function	Categorical Cross Entropy

In addition to hyperparameter restrictions, there are also limitations in choosing the CNN architecture. The architecture used in this research is the basic architecture of CNN. The pattern of the architecture itself is as follows:  $[CONVL - ACT - (POOL - DROP)] * N - [FULLC - ACT - DROP] * M$ . Where N is the number of Convolutional Blocks while M is the number of Fully connected Blocks, CONVL is Convolutional Layer, ACT is Activation Function, DROP is Dropout, and FULLC is Fully connected Layer or Classification Layer. Each block has a maximum

of two blocks and a minimum of one block, while the initial initialization architecture is randomly generated. In addition, the last layer of the model does not use a Fully connected Block but a predefined static layer.

### 3.8 Fitness and Parameter Settings

The fitness evaluation aims to determine the best agent owned by the population from the CWOA, WOA, and modification of PSFHS methods. The criteria for the agent must have the smallest error value of all agents in the population. The error value comes from the loss function of the CNN method during training, namely Categorical Cross Entropy. Furthermore, the chaotic map initialization value has a value of 0.7 in the CWOA method, while the HMCR value is 0.5 and the PAR value is 0.5 for modification of PSFHS.

## 4 Results and Discussion

### 4.1 Research Methodology

The selection of optimal hyperparameters on CNN has an impact on the results obtained. Therefore, this study uses the CWOA method to automate the selection of CNN hyperparameters and the stages in the research can be seen in Figure 2.

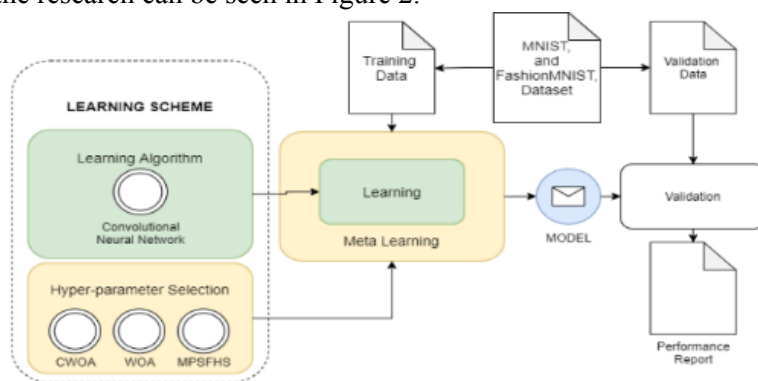


Figure 2. Research Methodology

### 4.2 Comparison CWOA, WOA, and Modification of PSFHS

In this study, the results of the CNN hyperparameter search from the method used by the researcher will be compared with the modified PSFHS method and the WOA method with the same test parameters. Then, the results of the error and the accuracy of these methods can be seen in Table 5.

Table 5. Training results dataset MNIST

Method	MNIST	
	Accuracy	Error
<b>CWOA Chaotic Map 6</b>	<b>99.63</b>	<b>0.023</b>
CWOA Chaotic Map 8	99.43	0.024
CWOA Chaotic Map 1	99.43	0.025
CWOA Chaotic Map 10	99.63	0.025
CWOA Chaotic Map 2	99.43	0.025
CWOA Chaotic Map 7	99.23	0.026
CWOA Chaotic Map 4	99.46	0.026
CWOA Chaotic Map 9	99.33	0.026
CWOA Chaotic Map 5	99.33	0.026
WOA	99.36	0.027
CWOA Chaotic Map 3	99.43	0.027
Modification of PSFHS	99.19	0.035

In Table 5 the MNIST dataset model CWOA method with Chaotic Map 6 has an error value of 0.023 and an accuracy of 99.63, then, the WOA method has an accuracy value of 99.36, and an error value of 0.027, while the modified method from PSFHS has an accuracy value of 99.19, and

the error of 0.035. Seeing these results, the error value of the CWOA method with Chaotic Map 6 obtained the smallest error value and the largest accuracy value.

**Table 6.** Training results dataset FashionMNIST

Method	FashionMNIST	
	Accuracy	Error
<b>CWOA Chaotic Map 8</b>	<b>91.36</b>	<b>0.23</b>
CWOA Chaotic Map 9	91.73	0.23
CWOA Chaotic Map 7	91.53	0.24
CWOA Chaotic Map 2	90.60	0.24
CWOA Chaotic Map 4	91.60	0.24
CWOA Chaotic Map 6	91.46	0.24
CWOA Chaotic Map 10	91.13	0.24
CWOA Chaotic Map 3	91.23	0.24
CWOA Chaotic Map 1	91.69	0.24
WOA	91.00	0.24
CWOA Chaotic Map 5	91.60	0.25
Modification of PSFHS	90.96	0.25

Furthermore, in the FashionMNIST dataset model in Table 6, the CWOA method with Chaotic Map 8 obtains an error value of 0.23 and an accuracy of 91.36, then, the WOA method has an error value of 0.24 and an accuracy of 91.00, while the modification of PSFHS has an error value of 0.25 and an accuracy of 90.96. Seeing these results, the error value of the CWOA method with Chaotic Map 8 obtained the smallest error value and the largest accuracy value.

### 4.3 Comparison Method State-of-the-art

In addition to comparing the test results between the three methods used in this study to obtain good quality test results, it is necessary to compare the results obtained with several previous studies. Some of these studies can be seen in Table 7, Table 8, and Table 9. Although in some previous studies (Bacanin et al., 2020; Gulcu & Kus, 2020; Lee et al., 2018; Strumberger et al., 2019; Y. Sun et al., 2020) has different test conditions and in common with this study only the dataset used. Moreover, the differences in the computational machines used by researchers in this study with different previous studies make the comparison of this approach not entirely realistic.

**Table 7.** Training results dataset FashionMNIST

Method	Error
TGA-CNN (Strumberger et al., 2019)	0.23
<b>EvoCNN (Y. Sun et al., 2020)</b>	<b>0.11</b>
$E^3FA$ -CNN (Bacanin et al., 2020)	0.21
EE-TGA-CNN (Bacanin et al., 2020)	0.19
CWOA Chaotic Map 6	0.023
WOA	0.027
Modification of PSFHS	0.025

In Table 7, the test uses the MNIST dataset by comparing the error values between methods. From the table, it shows that the CWOA method with Chaotic Map 6 has the smallest error value. Then, the WOA method and the modified method from PSFHS have the second and third smallest values seen from the order of Table 7. Therefore, Table 7 can show that the method used by the researcher has the smallest error value when compared to several previous research methods which in his research did not implement regularization, namely EvoCNN, TGA-CNN,  $E^3FA$ -CNN, and EE-TGA-CNN.

**Table 8.** Training results dataset FashionMNIST

Method	Accuracy
modifikasi PSF-HS-CNN (Lee et al., 2018)	99.30
$\mu O$ (Gulcu & Kus, 2020)	<b>99.64</b>



CWOA Chaotic Map 6	99.63
WOA	99.36
Modification of PSFHS	99.19

Furthermore, in Table 8 the test is carried out by comparing the accuracy values between methods. The highest accuracy value is obtained by the  $\mu$ O method while the CWOA method with chaotic Map 6 has the largest accuracy value after the  $\mu$ O method. The WOA method has the greatest accuracy value after the CWOA method with Chaotic Map 6. Then, the modified PSFHS method that the researchers did has a smaller accuracy result compared to previous studies that used the same method and dataset, namely the PSF-HS-CNN modification.

**Table 9.** Training results dataset FashionMNIST

Method	Accuracy
$\mu$ O (Gulcu & Kus, 2020)	93.78
<b>EvoCNN</b> (Y. Sun et al., 2020)	99.00
CWOA Chaotic Map 8	91.36
WOA	91.00
Modification of PSFHS	90.96

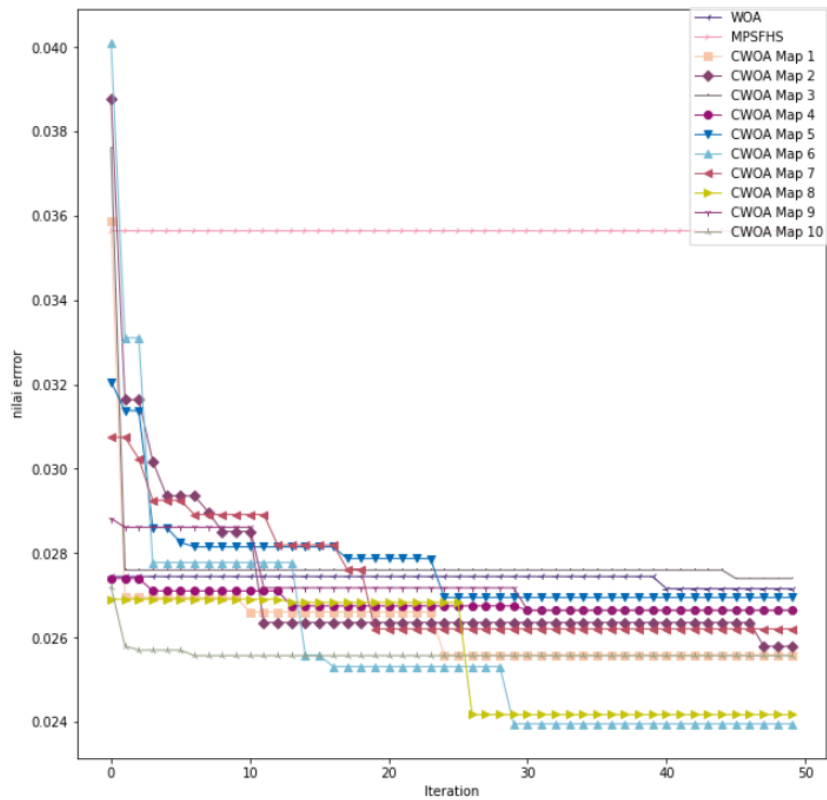
Then, in Table 9, the test uses the FashionMNIST dataset by comparing the accuracy values between methods. The greatest accuracy value is obtained by the EvoCNN method and the  $\mu$ O method. Both methods have a greater accuracy value when compared to the methods used in this study, it is the CWOA method with Chaotic Map 8, the WOA method, and the modified PSFHS method. More results from other methods can be seen in Appendix A.

## 5 Conclusion

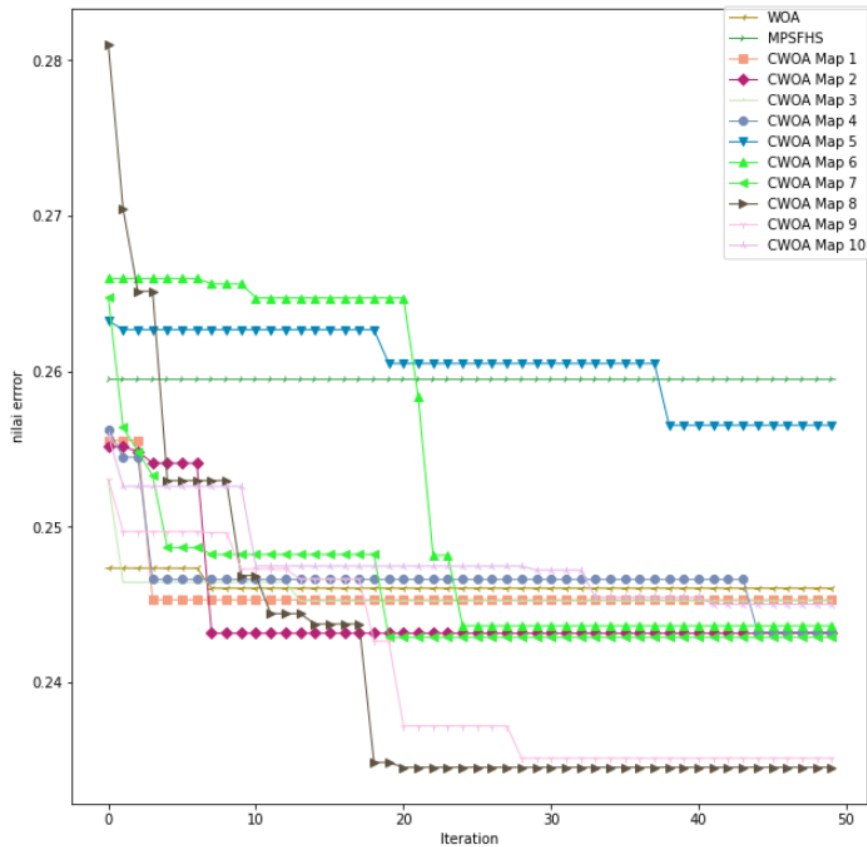
Judging from the results of this study, the use of regularization in searching for hyperparameters of the CNN method also affects the accuracy and error results obtained during the training process. This is evidenced by the method used in the study which has a smaller error value compared to several previous studies. However, it is different for the accuracy results on the MNIST dataset where the method in this study is not superior to any of the previous studies. Meanwhile, in the FashionMNIST dataset, the accuracy value obtained by the method in this study also does not have the greatest accuracy value compared to previous studies.

## 6 Appendices

### 6.1 Appendix A



**Figure B1.** The fitness results of each tested algorithm in dataset MNIST



**Figure B2.** The fitness results of each tested algorithm in dataset FashionMNIST

## 6.2 Appendix B

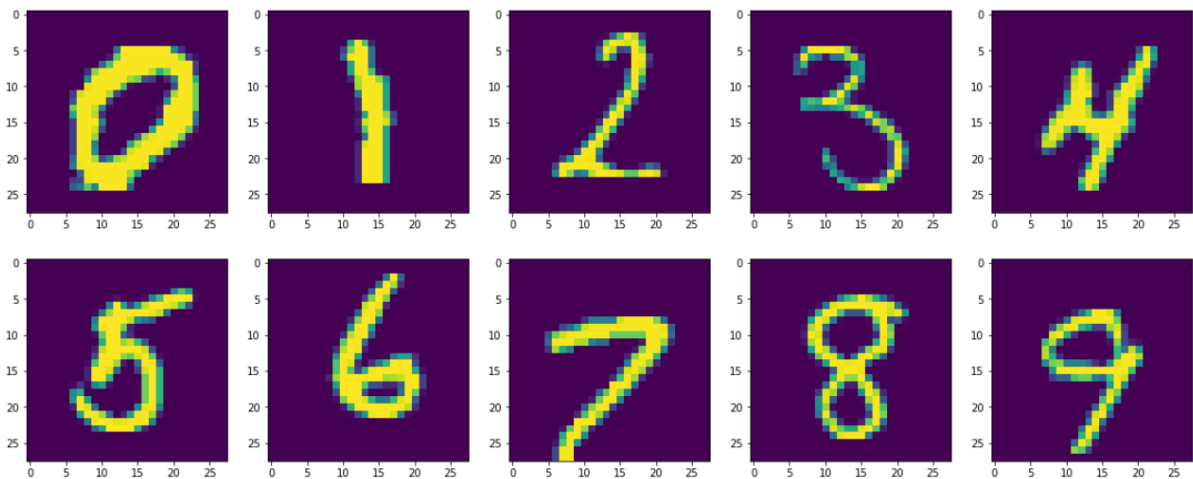


Figure C1. Rxample classes in the MNIST dataset

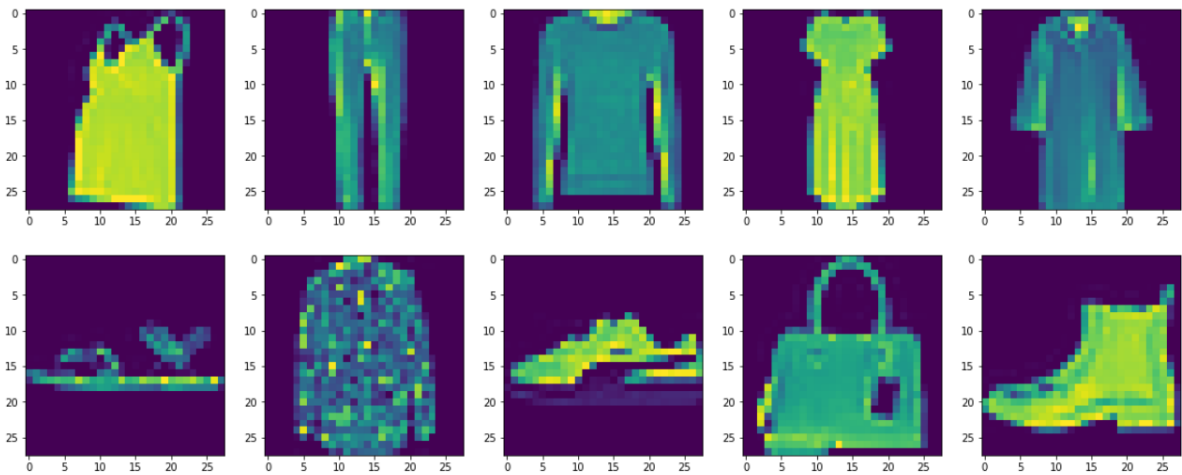


Figure C2. example classes in the FashionMNIST dataset

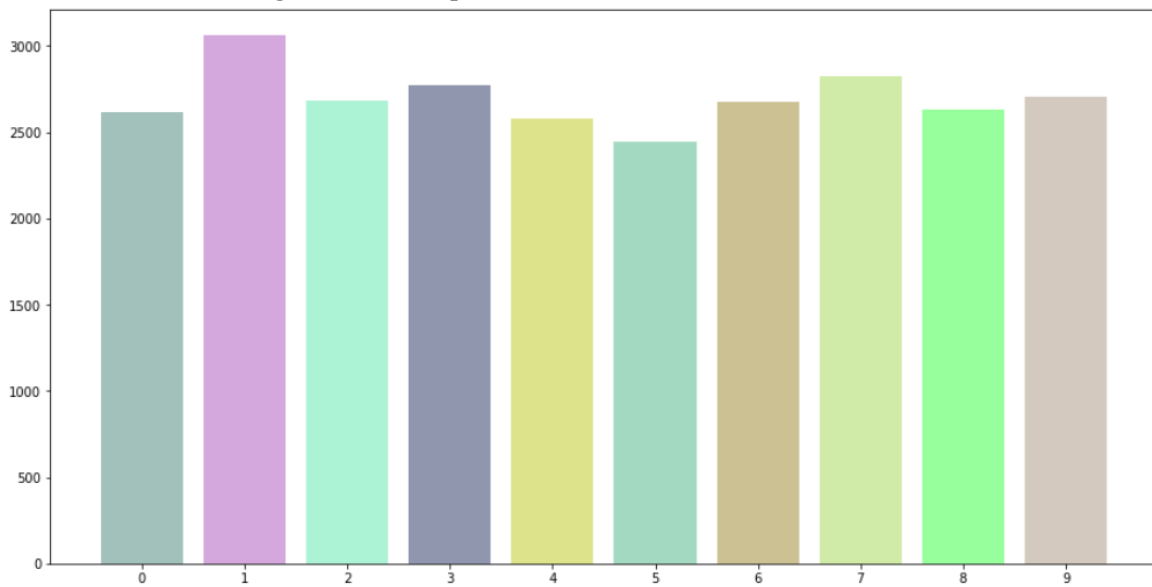
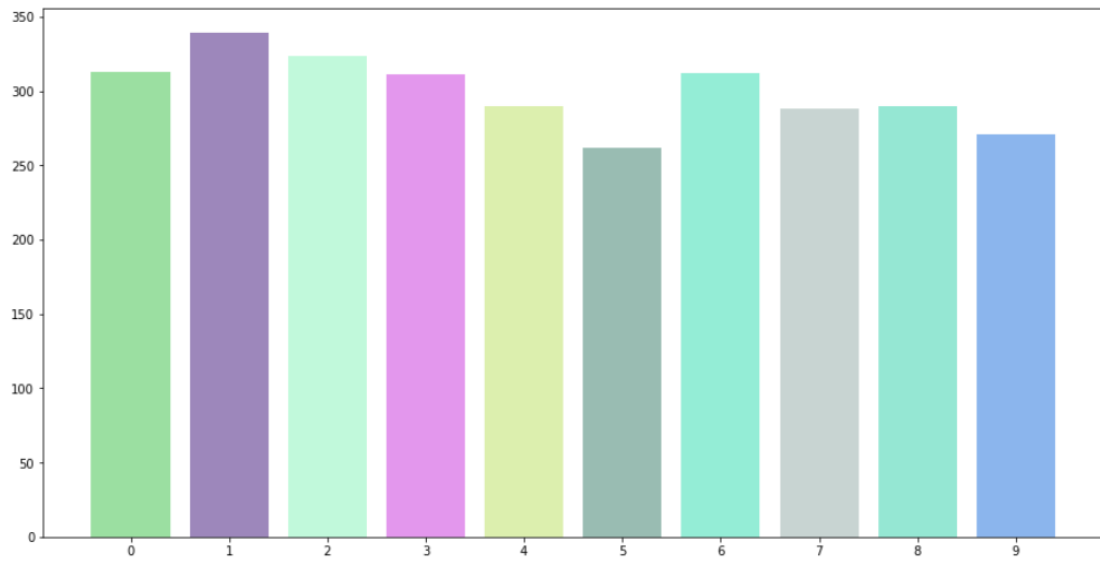
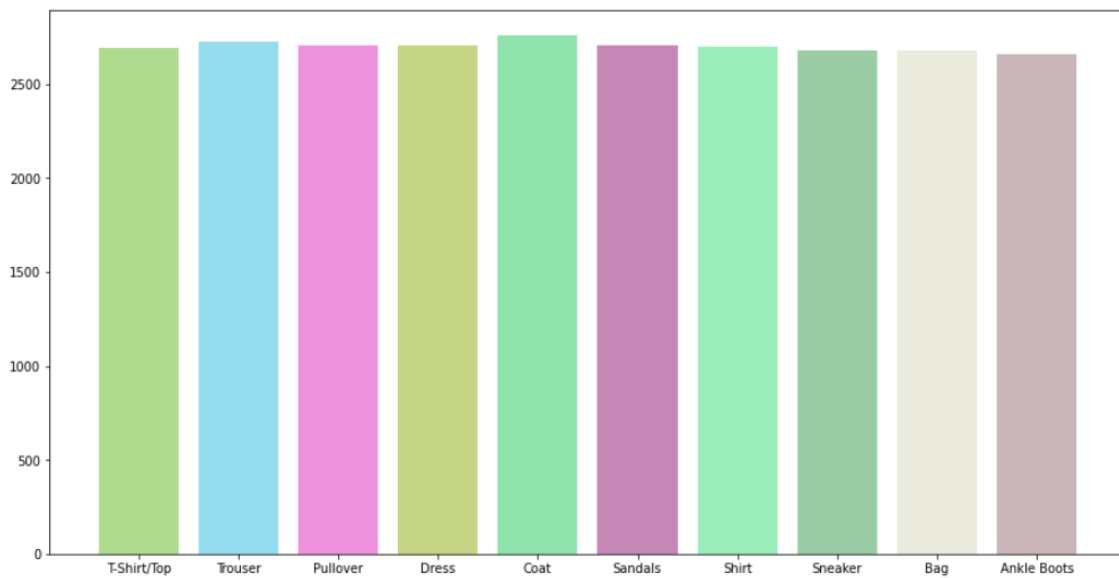


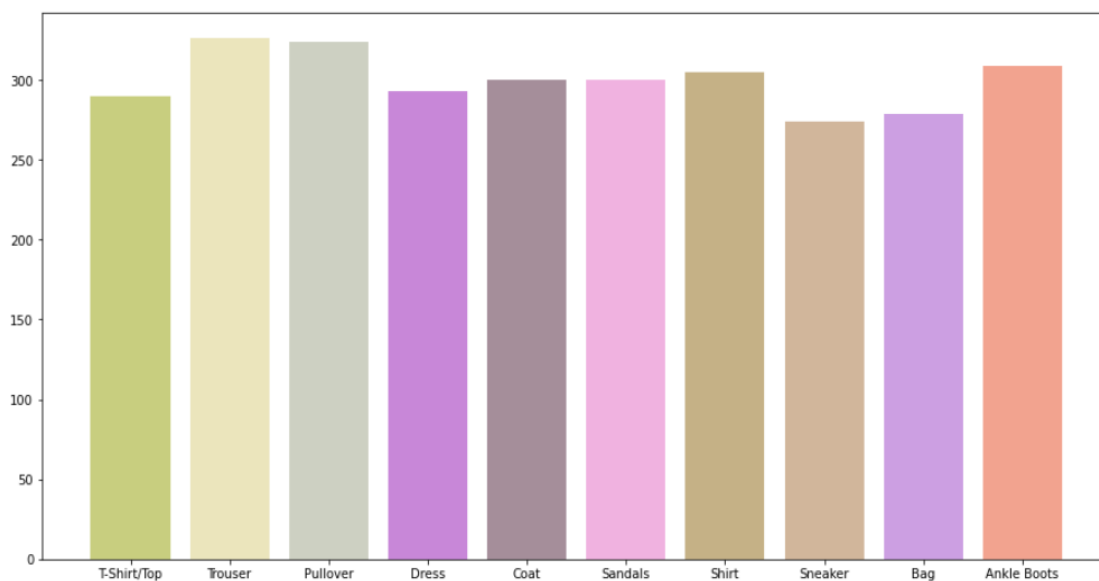
Figure C3. Training data from the MNIST dataset after a random selection process



**Figure C4.** Validation data from the MNIST dataset after a random selection process



**Figure C5.** Data training FashionMNIST



**Figure C6.** Data validation FashionMNIST

## References

- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., van Esesn, B. C., Awwal, A. A. S., & Asari, V. K. (2018). *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches*.
- Bacanin, N., Bezdán, T., Tuba, E., Strumberger, I., & Tuba, M. (2020). Optimizing Convolutional Neural Network Hyperparameters by Enhanced Swarm Intelligence Metaheuristics. *Algorithms*, 13(3), 67.
- Bibaeva, V. (2018). Using Metaheuristics for Hyper-Parameter Optimization of Convolutional Neural Networks. *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1, 1–6.
- Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259.
- Chen, Z., & Liu, G. (2019). DenseNet+Inception and Its Application for Electronic Transaction Fraud Detection. *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 1(c), 2551–2558.
- Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very Deep Convolutional Networks for Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1107–1116.
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), 60–68.
- Geem, Z. W., & Sim, K. B. (2010). Parameter-setting-free harmony search algorithm. *Applied Mathematics and Computation*, 217(8), 3881–3889.
- Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, 48(March), 1–24.
- Gulcu, A., & Kus, Z. (2020). Hyper-Parameter Selection in Convolutional Neural Networks Using Microcanonical Optimization Algorithm. *IEEE Access*, 8, 52528–52540.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. 1–18.
- Kaur, G., & Arora, S. (2018). Chaotic whale optimization algorithm. *Journal of Computational Design and Engineering*, 5(3), 275–284.
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

- Lee, W.-Y., Park, S.-M., & Sim, K.-B. (2018). Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, 172(July), 359–367.
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67.
- Prechelt, L. (2012). *Early Stopping — But When?* (Vol. 0, Issue 1998, pp. 53–67). [https://doi.org/10.1007/978-3-642-35289-8\\_5](https://doi.org/10.1007/978-3-642-35289-8_5)
- Qian, Y., Bi, M., Tan, T., & Yu, K. (2016). Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12), 2263–2276. <https://doi.org/10.1109/TASLP.2016.2602884>
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Strumberger, I., Tuba, E., Bacanin, N., Jovanovic, R., & Tuba, M. (2019). Convolutional Neural Network Architecture Design by the Tree Growth Algorithm Framework. *Proceedings of the International Joint Conference on Neural Networks, 2019-July(July)*, 1–8.
- Sun, W., & Chang, X. (2015). An Improved Harmony Search Algorithm for Power Distribution Network Planning. *Journal of Electrical and Computer Engineering*, 2015, 1–6.
- Sun, W. Z., & Wang, J. S. (2017). Elman Neural Network Soft-Sensor Model of Conversion Velocity in Polymerization Process Optimized by Chaos Whale Optimization Algorithm. *IEEE Access*, 5(c), 13062–13076.
- Sun, Y., Xue, B., Zhang, M., & Yen, G. G. (2020). Evolving Deep Convolutional Neural Networks for Image Classification. *IEEE Transactions on Evolutionary Computation*, 24(2), 394–407.
- Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.
- Wang, B., Sun, Y., Xue, B., & Zhang, M. (2018). Evolving Deep Convolutional Neural Networks by Variable-Length Particle Swarm Optimization for Image Classification. *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8.
- Wang, Y., Luo, X., Ding, L., Fu, S., & Wei, X. (2019). Detection based visual tracking with convolutional neural network. *Knowledge-Based Systems*, 175, 62–71. <https://doi.org/10.1016/j.knosys.2019.03.012>
- Wu, H., & Gu, X. (2015). Towards dropout training for convolutional neural networks. *Neural Networks*, 71, 1–10.
- Xie, L., & Tian, Q. (2019). Towards multi-modality invariance: A study in visual representation. In *Multimodal Behavior Analysis in the Wild* (pp. 323–348). Elsevier.
- Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., & Yu, B. (2019). Recent advances in convolutional neural network acceleration. *Neurocomputing*, 323, 37–51.
- Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8–36.