Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 1, February 2021, Pp. 19-26

CNN hyperparameter optimization using random grid coarse-to-fine search for face classification

Ade Nurhopipah*[1], Nurriza Amalia Larasati[2]
Department of Informatics, Universitas Amikom Purwokerto, Indonesia[1,2]

Article Info
Keywords:
CNN Optimization, Coarse-to-fine, Grid Search, Hyperparameter, Random Search

Article history:
Received: December 20, 2020
Accepted: January 5, 2021
Published: February 28, 2021

Cite:
Nurhopipah, A., & Larasati, N. A. (2021). CNN Hyperparameter Optimization using Random Grid Coarse-to-fine Search for Face Classification. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(1). https://doi.org/10.22219/kinetik.v6i1.1185

*Corresponding author.
Ade Nurhopipah
E-mail address:
ade_nurhopipah@amikompurwokerto.ac.id

Abstract
Convolutional Neural Network (CNN) is a recently used popular machine learning technique to classify images. However, choosing an optimum and efficient architecture is an inevitable challenge. The research goal was to implement CNN on face classification from low quality CCTV footage. The best model was gained from the hyperparameter optimization process used on CNN structure. The optimized hyperparameters were those connected to the structure network including activation function, the number of kernel, the size of kernel, and the number of nodes on the fully connected layers. Hyperparameter optimization strategy used was random grid coarse-to-fine search optimization approach. This approach combined random search, grid search, and coarse-to-fine technique that was easily and efficiently applied, yet worked well. Exhaustive-random search process was done by evaluating all selected activation functions and choosing another hyperparameters randomly. This was based on the assumption that activation functions were the most related hyperparameter to the model. The SELU activation function used in the next step was the one with the best average performance. Grid coarse-to-fine was conducted to optimize the number of kernel and the number of node on fully connected layer, while grid search was conducted to optimize the kernel size. This process aimed to locate optimal value gradually in hyperparameter which had high-dimensional space. Evaluation of the model resulted from the optimum hyperparameter was 97,56%.

1. Introduction

Applications using big data have been widely used in most industries and research. Some of them are the use of CCTV video footage, social media data, censor data, farming data, medical data, and even space research data. CCTV video data are abundant, since there are surveillance cameras applied in housing areas, industrial areas, education institutions, commercial companies, public places like city center, public transportations, etc. [1]. Surveillance video automation on CCTV is an important and interesting subject to learn. The challenge of this subject is how the data processing can recognize the object pattern, action, or situation correctly and precisely.

Image processing on CCTV video footage is the task of computer vision. Image classification is a part that covers a lot of applications such as object detection, localization, and segmentation. In machine learning algorithm for computer vision, Convolutional Neural Network (CNN) is the most commonly used technique for image classification. CNN algorithm is mostly used in recognizing patterns, such as traffic jam detection [2], human detection on night vision cameras [3], facial recognition for attendance system [4], and facial recognition on CCTV used to detect and track pedestrians [5]. Now, CNN has become a breakthrough in computer vision especially image classification, object detection and segmentation. The development of CNN is accelerating with new structures, new optimization techniques, and a large scale of dataset availability. CNN structure is getting deeper and more complex as well [6].

Selecting optimum hyperparameter is one of the main challenges in applying CNN. Optimum hyperparameter is network structure which has the best performance measured from the highest accuracy value.There is not any reliable specific approach to identify network structure which leads to an increasing performance significantly [7]. CNN performance depends on the hyperparameters in which its value can not be estimated from data. Hyperparameters on CNN can be classified into two types; they are hyperparameter defining network structure and the one defining network learning. The first one includes kernel size, kernel type, stride, padding, hidden layer, and activation function. Meanwhile, the other one includes learning rate, momentum, the number of epoch and batch-size [8]. Determining optimum combination of CNN hyperparameters is a challenging task for its computational complexity. It is because we deal with search space and high-dimensional dataset. Practically, optimal hyperparameter search can be conducted through different strategies, such as exhaustive search approach (testing all possible combinations), random search (generating random hyperparameter), grid search (systematic exploration), and heuristic search (based on a given heuristic function or a cost measure) [9].

Exhaustive Search approach can not be used if the number of hyperparameters is big. The combination with another hyperparameter will cause a very inefficient search. Grid search approach is an approach method which tries all possibilities of specific ranges. This approach needs experience, intuition, or knowledge to determine the hyperparameter's interval limit that is going to be observed. Grid search is good to apply only when the number of hyperparameter has small dimension. Random search approach is a search using random combination of some hyperparameter value. This approach is more efficient to be applied in high-dimensional space. This strategy is easy to apply and works properly. Random search approach can be seen in research [8][10]. Another approach which is commonly used today is heuristic approach, such as bayesian [11][12][13], genetic algorithm [9][14], and evolutionary algortihm [7][15]. Heuristic approach is efficient eventhough it does not guarantee optimal solution. Combining two approaches can also be carried out such as in research [16] in which random search optimization is applied to be the basic use of grid search approach. Research [17] also combines random search approach and probabilistic greedy heuristic which is then called Weighted Random Search (WRS).

This research aims to implement CNN algorithm with the best network structure to classify images from CCTV video in office area. The data came from the previous research applied Counter-Propagation Neural Network (CPN) which produced accuracy image identification by 60% [18]. This low accuracy needs to be enhanced by applying more powerful algorithm such as CNN. The contribution on this research is the application of strategy to search CNN optimum hyperparameter by combining grid search and random search with coarse-to-fine technique. Random search approach is applied to limit the number of search spaces that will be done by taking random value in the determined range. The hyperparameter value will be applied as an input of searching process by using grid coarse-to-fine technique. This process aims to locate optimal value gradually in hyperparameter which has high-dimensional space. With this strategy, it is expected that optimal and efficient solution will be gained with higher accuracy value compared to the previous research.

2. Research Method

2.1 Image Pre-Processing

The data used in this research are 3000 face images of 21 people which are manually labeled. These images come from face detection process through resizing proces; hence images of 100x100 in size are produced. Original images are varied from 70x70 up to 140x140 in size; so resizing is commited so that the inputs have the same size. The pre-processing image process is done by improving the image quality (image enhancement) and data normalization. Image enhancement process is applied because the quality of the images are low. Those images are blurred, has lots of noise with minimum light and even incomplete. In this research, the image enhancement applied is denoising with Wavelet and Laplacian of Gaussian (LoG) for image sharpening. Facial image samples used are shown in Figure 1.

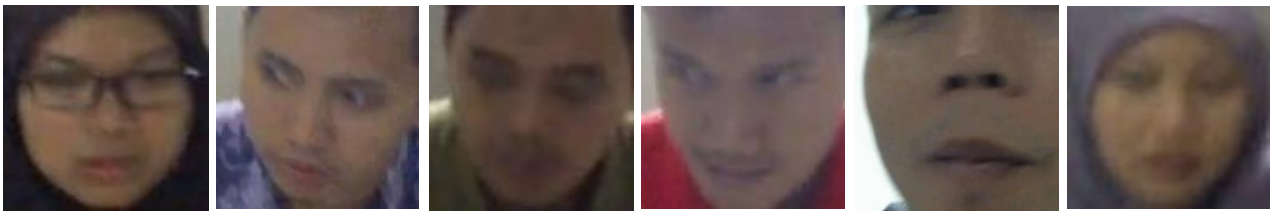

Figure 1. Sample Images from Dataset as Input

A CNN model will be made by learning dataset which is partitioned with *k*-fold validation technique. This method is used since it has good performance in learning small number of dataset. It also has stable performance, gives high accuracy in training set and good generalization to validation set and test set [19]. The model gained will give facial label prediction on each sample on the test.

2.2 Implementation of CNN

CNN is a Deep Learning architecture inspired by human's ability to recognize objects and environment rapidly. If computers can recognize or classify objects and environment by looking at low-level features, it can build a more abstract concept about what it knows through a series of convolutional layers. Therefore, this network structure is called Convolutional Neural Network [8]. Deep learning technique has engineering feature skills that can manipulate features automatically, so it's not necessary to build extraction feature models which are often complicated [20]. Based on the learning technique, deep learning can be classified into four parts, they are Deep Unsupervised Learning (DUL), Deep Supervised Learning (DSL), Deep Semi Supervised learning (DSSL) and Deep Reinforcement Learning (DRL) [21]. In this case, CNN belongs to DSL architecture.

Four operations that become the basic structure of CNN are convolution, activation function, pooling and fully connected layers. CNN architecture is a series of convolutional layers which is equipped with activation function, and the continuation of the other structure. The size of the image is smaller but deeper. After convolutional and pooling layers, feed forward neural network with fully connected layer is added with an activation function, and the last layer, the output layer is a predictive layer with softmax function which shows the estimated probability [22]. CNN architecture is as shown in Figure 2.

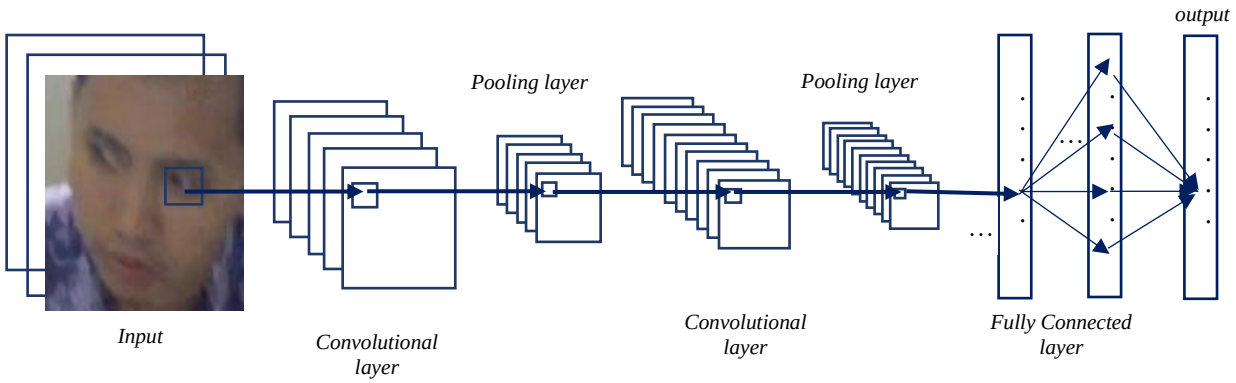

Figure 2. Convolutional Neural Network Architecture

The purpose of convolutional layers is to extract the image's features. Each of them uses various kernels to detect and extract features such as edge detection, sharpening, blurring, and so on. Therefore, by applying CNN, a specific feature extraction algorithm is no longer needed since CNN can extract information through the convolution process. Convolutional operation can be shown in Equation 1. Notation / is the operated image, *K* is the kernel and *S* is convolutional result [23].

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n) \quad (1)$$

After completing the process of convolution, an activation function is applied to extract non-linear features. Pooling subsampling layers aims to lessen dimension on image features. Therefore, the most valuable information stays. The purpose of fully connected layers is to classify features into some classes based on the dataset labels [9].

2.3 Network Structure Hyperparameters

The optimized hyperparameters in this research are the one related to the network structures, including activation function, the number and size of kernel, and the number of nodes in fully connected layers. Activation function is the point of CNN model's complexity problem. Non-linear activation enables neural network to have an authentic artificial intelligent. ReLu, one of the most used activation function, is one of popular functions since it learns data farther in a network with many layers [24]. However, ReLu function also has flaws because it allows the existence of necrosis phenomena. The effect of activation function in CNN model is learnt in research [25]. This research compares sigmoid, Tanh, ReLu, leaky ReLus softplus-ReLu and new activation function suggestion. Research [26] compares three activation functions: ReLu, sigmoid, and ELU. In general, based on the experiment conducted in that research, it is found that ELU is better than ReLU and Sigmoid.

Kernel in CNN network is a spatial representation in the model. The most used kernel is the odd-sized kernel, while the even-sized kernel is seldom used and explored. The even-sized kernel has asymmetric receptive field which causes pixel movement in feature maps. Location offset is accumulated when layering some convolution, so spatial information is eroded [27]. There is no direct relation between the kernel size and the accuracy. If the feature is small, the application of large kernel might cause a loss of information from the feature. Either way, for large-sized feature, the small kernel can not cover the whole information of that feature. However, if convolutional layer is added, information coming from small feature will be accumulated to form big-sized information feature. Both applying large-sized kernel and adding convolutional layer will enlarge the parameter and the model's complexity.

The number of hidden layers and nodes can also affect model optimization. The model with a lot of hidden layers makes it expressive in learning complicated connection between the input and the output. Yet, with limited training, the connection is just a noise. That information only shows up in training data, not in test data which causes the model over fitting. Shallow architecture needs more nodes in fully connected layer to produce better performance. Meanwhile, deep architecture needs fewer nodes [28].

2.4 *CNN* Optimization

CNN Algorithm has various hyperparameters that need to be noticed. The more complicated the CNN structure is, the more difficult it is to determine hyperparameter combination used. This research will use random search approach, grid search and coarse-to-fine technique.

2.4.1 Grid Search

Grid search approach is easy to apply and reliable in low dimensional spaces. In addition, it's parallelization is trivial [10]. Grid search is a method of hyperparameter searching by testing model to all determined value. For example, if the optimization problem is to find maximum objective function, evaluate Equation 2, then Equation 3 will get maximum value. Value $\Delta x = x_{max} - x_{min}$, and *n* is the number of sample points. Taking more samples will make the function approach the real maximum value. Grid search is efficient when there is a little value to be evaluated.

$$f = \left(f(x_{min}), f\left(x_{min} + \frac{\Delta x}{n}\right), \dots, f\left(x_{min} + (n-1)\frac{\Delta x}{n}\right)\right) \quad (2)$$

$$\tilde{f} = \max_{0 \le i \le n-1} f_i \quad (3)$$

2.4.2 Random Search

Random search is an easy-to-use optimization strategy, but often works better than other optimization approaches. Random search has all the practical advantages of grid search and provides efficiency in the high dimensional search space [10]. In random search approach, the sample point is not taken regularly yet randomly in interval between x_{min} dan x_{max}. Since the samples are randomly taken, the optimization result is also different. The risk of this method is that the samples taken can not get even close to the real optimal value. However, that probability is quite small. If a constant probability distribution is conducted, it will also be probable that a sample producing optimal value is selected randomly.

2.4.3 Coarse-to-Fine Technique

When we want to find optimal value $f(x)$ between $x_{min}$and x_{max},coarse-to-fine optimization is a good technique to get optimal value based on these steps [29].

1. Search in area $R_1 = (x_{min}, x_{max})$. Symbolize optimal value in this area with (x_1, f_1).
2. Search in smaller area $R_2 = (x_1 - \delta x_1, x_1 + \delta x_1)$. Assume that the real optimal value is in that interval. Symbolize optimal value in this area with (x_2, f_2).
3. Repeat step 2 with δx_2 is smaller than δx_1. Stop when the optimal value (x_i, f_i) in area R_{i+1}does not change.

By carrying out this technique, there will be a chance to get faster optimal value with fewer searches.

2.5 Optimization Design

The value which limits hyperparameter space search in this research is shown in Table 1. Each of *convolutional layer*, *pooling layer*, and *fully connected layer* uses two layers. If we do systematic search in all hyperparameter combination, a lot of model combination will need evaluation. In this research, all options of activation function will be evaluated, assuming that activation functions are the most related parameter to the model. In each evaluation of activation function, each hyperparameter's random value is taken for 50 times per activation function type. Therefore, the experiment is done 350 times (50x7 activation functions), The best result will be used as the next input process.

Table 1. Range Value of Hyperparameter

Hyperparameter	Range
Activation function	[ReLU, ELU, Sigmoid, SELU, tanh, softsign, softplus]
Number of kernel	[10,11,…, 100]
Kernel size	[2x2,3x3,4x4,5x5,6x6,7x7]
Fully connected layer *node*	[10,11,…, 100]

Coarse-to-fine technique with grid search optimization is applied to hyperparameter of kernel number and node number to fully connected layer. This technique is applied to reduce the number of search and to locate optimum value in a quite big dimension. This experiment is conducted for 60 times (3 iteration x 5 samples x 2 layers of kernel number x 2 layers of node number). In this point, there is optimization of kernel size hyperparameter which can be done with grid search technique for 12 times (6 kernel size x 2 layers). In each experiment, 50 iterations are used to limit the training procces. Computation time is not considered since in this research, optimization methods comparison is not

commited. Time efficiency considered by limiting the numbers of experiments. Figure 3 shows optimization design in this research.

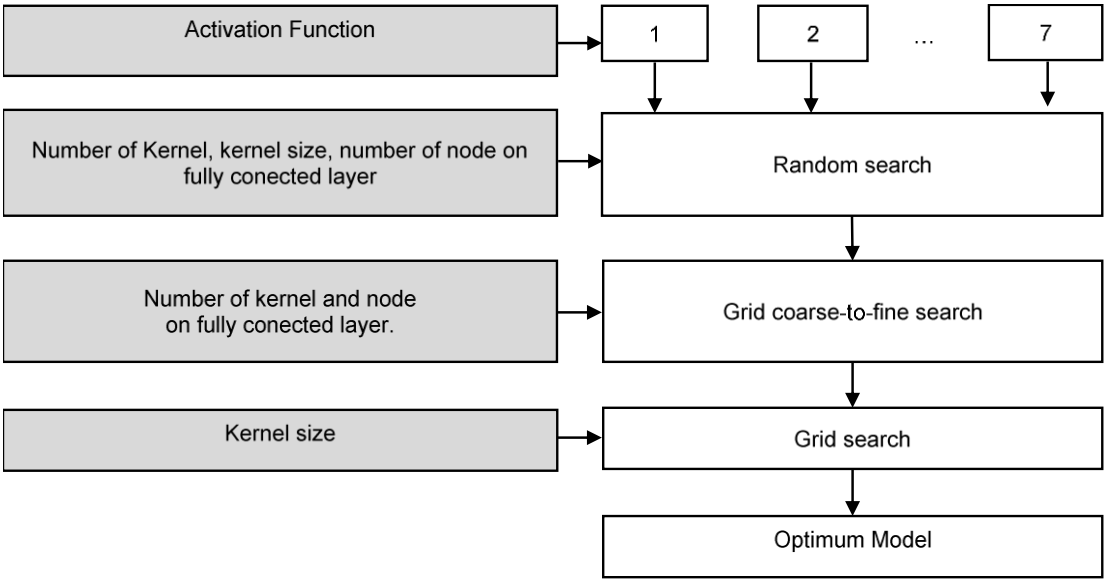

Figure 3. Design of CNN Hyperparameter Optimization

3. Results and Discussion

A CNN model that has not been optimized was used to evaluate image enhancement result. Table 2 shows that the result of denoising wavelet evaluation gave the best result by 92.89% accuracy value, while the loss function value with cross entropy was 0.2990. The result of model evaluation with CNN before optimization had given better accuracy compared to the previous research which used CPN algorithm. Next, dataset coming from denoising wavelet would be the input of optimum CNN hyperparameter search process.

Tabel 2. Accuracy Value and Loss Function Based on Image Enhancement

Image Enhancement	Original	LoG	Wavelet
Accuracy	0.8956	0.9200	0.9289
Loss function value	0.3910	0.3202	0.2990

The next process was exhaustive-random search process by evaluating all activation functions and choosing another hyperparameters randomly. The model evaluation on testing process produced accuracy value as shown in Table 3. It shows that the best accuracy average was given by SELU function by 0.9572 and the least accuracy was produced by sigmoid function by 0.7999. Meanwhile, the smallest deviation standard was gained from ELU function.

Table 3. Accuracy Value and Loss Function Based on Activation Function

Activation Function		ReLU	ELU	Sigmoid	SELU	tanh	softsign	softplus
Accuracy	Max	0.9644	**0.9733**	0,9200	0.9711	**0.9733**	0.9711	0.9578
	Min	0.8000	0.8644	**0,5178**	0.9267	0.8600	0.8511	0.6800
	Average	0.9033	0.9544	0,7999	**0.9572**	0.9508	0.9383	0.9055
	St.dev.	0.0437	**0.0180**	0,0921	0.0474	0.0229	0.0272	0.0474

The loss function value of this experiment showed that there was a small value on the training and validation process, yet showed big value during the testing. For instance, by applying the best hyperparameter on SELU function, the loss function value produced on the training was 1.5239e-04, and 1.9222e-04 on the validation. Yet, on the testing, the loss function value was 0.1292. It showed that the model could be overfitting. However, the accuracy value gained was quite satisfying. Further research can be conducted to find the cause and solution to that overfitting. Figure 4 showed loss function graph from the research, arranged from the smallest to the biggest. The loss function average of sigmoid produced the highest value by 0.9443 and had high diversity. SELU had the smallest deviation standard by 0.0628 and showed more stable result, while tanh had the smallest average loss function value by 0.1327.

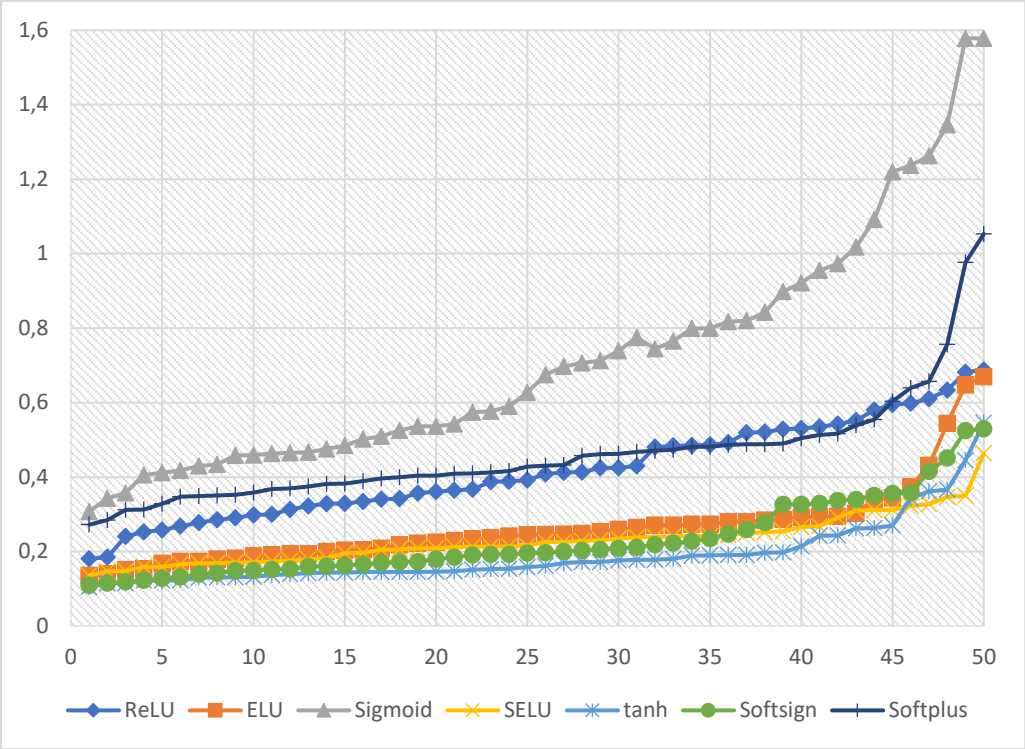

Figure 4. Graph of Loss Function Value Based On Activation Function

The activation function used in the next step was the one with the best average performance, SELU function (Scaled Exponential Linear Unit). This function was first introduced in research [30] as the one that prevented vanishing phenomenon and exploding gradients. SELU function is defined with the formula in Equation 4.

$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4)$$

Maximum accuracy gained from SELU activation function was 0,9711 with loss function = 0.2090. The randomly gained hyperparameter value was: the number of kernel on the first convolutional layer was 54, and the one on the second layer was 63. The kernel size on the first convolution was 7, and the one on the second convolution was 5. The number of node on the first fully connected layer was 73 and the one on the second layer was 44. Next, with this hyperparameter, grid coarse-to-fine was conducted to optimize the number of kernel and the number of node on fully connected layer and grid search to optimize the kernel size. The evaluation result is shown in Table 4.

Table 4. Optimum Value with Grid Coarse-to-fine search Method

Hyperparameter	Layer	Optimum value	Accuracy
Number of kernel	1	59	0.9733
	2	30	0.9689
Number of fully connected layer node	1	52	0.9667
	2	91	0.9756
Kernel size	1	7	0.9756
	2	5	0.9756

Table 4 shows that grid-coarse-to-fine process did not fully produce localization of the best optimum value. In the number of kernel of the second layer and the number of node of the first fully connected layer, this method was jammed on the local optimum so it produced smaller accuracy value compared to the one produced on the previous random search result. Grid search process on the number of kernel did not give improvement on accuracy as well. It means that the previous random value had been the optimum value. Finally the best model of hyperparameter optimization result is shown in Table 5. Face classification using CNN with this hyperparameter produced accuracy up to 97.56%. Model accuracy of this research was better than the previous one using Counter-Propagation Neural Network (CPN) by 60%. The optimum hyperparameters resulted from this research couldn't be generalized for other cases. However, the method applied could be an alternative way to determine optimum hyperparameters on other cases.

Table 5. Hyperparameter Optimum Value

Hyperparameter	Layer	Optimum value
Activation function	-	SELU
Number of kernel	1	59
	2	63
Kernel size	1	7
	2	5
Number of fully connected layer node	1	73
	2	91

4. Conclusion

This research applies hyperparameter optimization of CNN network structure by combining random search, grid search, and coarse-to-fine technique. The produced model is implemented on face detection images that have low quality. The combination of these three approaches can lessen the number evaluation that needs to be done and produce optimum models as expected. The application of CNN algorithm before image enhancement produces accuracy of 89.56 %. Denoising wavelet improves image quality; therefore it improves accuracy value by 92.89%. Hyperparameter optimization of activation function using exhausted-random-search improves accuracy up to 97.11%. Next, by using grid coarse-to-fine method to other hyperparameters, the accuracy was slightly increased up to 97.56%.

Acknowledgement

The researcher would like to thank Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Of Amikom University of Purwokerto that has given a big support to conduct this research.

References

[1] G. Sreenu and M. A. Saleem Durai, "Intelligent Video Surveillance: a Review through Deep Learning Techniques for Crowd Analysis," *J. Big Data*, vol. 6, no. 1, pp. 1–27, 2019. https://doi.org/10.1186/s40537-019-0212-5
[2] J. Kurniawan, S. G. S. Syahra, C. K. Dewa, and Afiahayati, "Traffic Congestion Detection: Learning from CCTV Monitoring Images using Convolutional Neural Network," *Procedia Comput. Sci.*, vol. 144, pp. 291–297, 2018. https://doi.org/10.1016/j.procs.2018.10.530
[3] J. H. Kim, H. G. Hong, and K. R. Park, "Convolutional Neural Network-Based Human Detection in Nighttime Images Using Visible Light Camera Sensors," *Sensors (Switzerland)*, vol. 17, no. 5, 2017. https://dx.doi.org/10.3390%2Fs17051065
[4] M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "FaceTime - Deep Learning Based Face Recognition Attendance System," *SISY 2017 - IEEE 15th Int. Symp. Intell. Syst. Informatics, Proc.*, no. October, pp. 53–57, 2017. https://doi.org/10.1109/SISY.2017.8080587
[5] D. Acharya, K. Khoshelham, and S. Winter, "Real-time Detection and Tracking of Pedestrians in CCTV Images Using a Deep Convolutional Neural Network," *CEUR Workshop Proc.*, vol. 1913, no. April, pp. 31–36, 2017.
[6] H. Choi, "CNN Output Optimization for More Balanced Classification," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 17, no. 2, pp. 98–106, 2017. http://dx.doi.org/10.5391/IJFIS.2017.17.2.98
[7] E. Bochinski, T. Senst, and T. Sikora, "Hyper-Parameter Optimization for Convolutional Neural Network Committees Based on Evolutionary Algorithms," *Proc. - Int. Conf. Image Process. ICIP*, pp. 3924–3928, 2018. https://doi.org/10.1109/ICIP.2017.8297018
[8] N. M. Aszemi and P. D. D. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 269–278, 2019. https://dx.doi.org/10.14569/IJACSA.2019.0100638
[9] S. Loussaief and A. Abdelkrim, "Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 252–266, 2018. https://dx.doi.org/10.14569/IJACSA.2018.091031
[10] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
[11] J. Wu, X. C. Hao, Z. L. Xiong, and H. Lei, "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019. https://doi.org/10.11989/JEST.1674-862X.80904120
[12] D. P. Tran, G. N. Nguyen, and V. D. Hoang, "Hyperparameter Optimization for Improving Recognition Efficiency of an Adaptive Learning System," *IEEE Access*, vol. 8, no. 1, pp. 160569–160580, 2020. https://doi.org/10.1109/ACCESS.2020.3020930
[13] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 2951–2959, 2012.
[14] L. Xie and A. Yuille, "Genetic CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 1388–1397, 2017. https://doi.org/10.1109/ICCV.2017.154
[15] A. Baldominos, Y. Saez, and P. Isasi, "Hybridizing evolutionary computation and deep neural networks: An approach to handwriting recognition using committees and transfer learning," *Complexity*, vol. 2019, 2019. https://doi.org/10.1155/2019/2952304
[16] D. Motta *et al.*, "Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes," *PLoS One*, vol. 15, no. 7, pp. 1–30, 2020. https://doi.org/10.1371/journal.pone.0234959
[17] R. Andonie and A. C. Florea, "CCC Publications Weighted Random Search for CNN Hyperparameter Optimization," 2020. https://doi.org/10.15837/ijccc.2020.2.3868
[18] A. Nurhopipah and A. Harjoko, "Motion Detection and Face Recognition For CCTV Surveillance System," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 2, p. 107, 2018. https://doi.org/10.22146/ijccs.18198
[19] A. Nurhopipah and U. Hasanah, "Dataset Splitting Techniques Comparison For Face Classification on CCTV Images," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 4, pp. 341–352, 2020. https://doi.org/10.22146/ijccs.58092
[20] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika, 2018.
[21] Suyanto, K. N. Ramadhan, and S. Mandala, *Deep Learning ; Modernisasi Machine Learning untuk Big Data*. Bandung: Informatika, 2019.
[22] Aurélien Géron, "Deep Computer Vision Using Convolutional Neural Network," in *Hands-On Machine Learning with Scikit-Learn, Keras & tensorFlow*, 2nd ed., Sebastopol: O'Reilly Media, Inc., 2019, pp. 445–496.
[23] I. Goodfellow, Y. Bengio, and A. Courville, "Convolutional networks," in *Deep Learning*, Cambridge: MIT Press, 2016, pp. 321–359.
[24] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. https://doi.org/10.1038/nature14539

[25] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, 2020. https://doi.org/10.3390/app10051897

[26] R. Hazra, A. Kumar, and B. Baranidharan, "Effect of Various Activation Function on Steering Angle Prediction in CNN based Autonomous Vehicle System," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 3806–3811, 2019. https://doi.org/10.35940/ijeat.B4017.129219

[27] S. Wu, G. Wang, P. Tang, F. Chen, and L. Shi, "Convolution with even-sized kernels and symmetric padding," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–12, 2019.

[28] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, no. April, pp. 112–119, 2020. https://doi.org/10.1016/j.neucom.2019.10.008

[29] U. Michelucci, *Applied Deep Learning*. Dübendorf, Switzerland, 2018. https://doi.org/10.1007/978-1-4842-4976-5

[30] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," in *In Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 972–981.