DOI: 10.15849/IJASCA.221128.10

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 3, November 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Improvement on I-Devices Using L-GCNN Classifier for Smart Mosque Simulation

Juniardi Nur Fadila*, M. Amin Hariyadi*, Ajib Hanani, Johan Ericka W.P., Okta Qomaruddin Aziz

UIN Maulana Malik Ibrahim Malang, Indonesia
e-mail: juniardi.nur@uin-malang.ac.id
e-mail: adyt2002@uin-malang.ac.id
e-mail: ajib@uin-malang.ac.id
e-mail: johan@uin-malang.ac.id
e-mail: okta.qomaruddin@uin-malang.ac.id
**Corresponding Author*

Abstract

I-Device (Intelligent Devices) is one of the fastest growing devices since the beginning of this decade. Some of its major problems are accuracy and performance. This study aims to present an improvement in the performance of those devices. We used a simulation application for I-Devices to conduct the experiment. The simulation was built based on classifying results using Logarithmic learning for Generalized Classifier Neural Networks (L-GCNN). The output was a simulation that will be implemented on a smart mosque system. L-GCNN itself was a modification method of GCNN to improve the processing speed and have high accuracy as a classifier method. This method will take a role when the given parameters meet the conditions of the devices to take an action. To simplify the understanding of the simulation models, we used a game application to make an interactive simulation for our project in an environment that represents the real-world condition of the mosque. The result of this study shows that the devices could make a decision by themselves accurately. Additionally, using LGCNN models, we could reduce the processing iteration compared to other models. The experiment results show that LGCNN has an average value of 90% in accuracy, precision, recall, and f1.

Keywords: *Automation, Classifier, L-GCNN, Neural Network, Decision.*

1 Introduction

The growth of intelligent devices in this decade made a big change in how we build something. Especially with industrial 4.0, future technology become more and more complex and advanced [1]. A dream of a smart building that could act independently by the condition became true with the emergence of the latest Internet of Things (IoT) technology that can perform various types of functions [2]. Many experiments that used IoT in designing a building, produce a good results in improving its function like a smart greenhouse or smart home [3]. As a big country with the majority of citizens having

Received 18 September 2022; Accepted 2 November 2022

Islamic religion, Indonesia has a lot of mosques, but just a small portion of it has used today's technology. This research wants to apply IoT technology to a mosque. However, that system must have advance intelligent to know how it must take an action based on input given by the user or the condition. Today machine learning studies give us many options for the algorithm to improve the forecasting and classification process[4]. A study about disaster support systems is one of many studies that used machine learning to make some early warning systems using forecasting models. For example, the tsunami prediction system by Novianty et al,[5] used a neural network as a basic algorithm for prediction intelligence. On the other hand, machine learning also can be used to train some devices like robots to know about planning their path like research conducted by Arenas et al [6].

This study wants to present the implementation of a machine learning system in IoT devices to make it an I-Devices or smart device. One of the most popular algorithms for giving an artificial consciousness to a device is Artificial Neural Network (ANN) [7]. ANN adopts a human neural network system to generate some action based on the responses given to the system [8]. In 1988, the Radial Based Function (RBF) neural networks model was introduced by Broomhead and Lowe [9]. This model contains three base layers namely the Input layer, Hidden Layer, and Output layer. The difference between the RBF model and conventional neural networks is that conventional neural networks process multiple connection layers but not the RBF model [10]. There is only one single hidden layer in RBF that makes it more simple than a conventional neural network.

Several studies develop this neural network algorithm into a classifier algorithm so it can be used for classifying objects like fruits, animals, diseases, or even sign language. Ozyildirim et al did research on the advance of neural network models. They release some research that improved the neural network performance based on several different approaches [7], [11]–[13]. Apart from ANN backpropagation, a generalized classifier neural network (GCNN) is an adaptation of a feed-forward ANN model that does not require iterative training [13]. Each layer on the model consists of a different number of neurons and each layer is fully connected to the next layer.

Implementation of IoT technology that uses an intelligent system, needs more effort to be directly implemented. In addition, we need more effort in making a precise experimental setup. Because of those reasons, this paper will conduct the experiment in a simulation-based model. In a simulation-based approach, as a first step, we will make a model of our system in the virtual world according to its environment and parameter. Then we use many scenarios to get the best initial input and conduct the test for our proposed method according to those scenarios.

2 Related Work

Several studies for optimizing neural networks to get a more accurate result have been done frequently in this decade. Many applications of the neural network method had become popular for classifying or forecasting. Studies on biomedical territory dabbled in using this method for identifying a disease by its pattern like research conducted by Masoud [14]. That study said that using 4 types of modification of Convolutional Neural Networks (CNN), they obtained 79% average sensitivity, accuracy, and specificity in the classification of Arterial Wedge Pressure. CNN itself is one kind of neural network model that is very popular for classifying using pictures as inputs [15]. Outside biomedical case, other studies conducted using neural network was for developing artificial intelligence for

a specific purpose. Research by Panchal et al [16] shows that neural networks are used for predicting move patterns in an automatic chess game. It identified chess moved to compare with traditional chess algorithm, minimax. That two algorithms can learn and perceive chess rules and patterns. On the IoT topic itself, many types of machine learning methods already obtained good results[17], [18]. Furthermore, the modification of neural network algorithms is a focus of several studies. Shraddha et al, identified that instead of using the gaussian RBF kernel on a generalized classifier neural network (GCNN), it is better to use the Laplace kernel and optimal smoothing parameter calculated using a population-based Sine Cosine Algorithm [19].

Although it needs more time to process, a study in identification for sign language using LGCNN, a modification method of GCNN by Darlis et al[20], indicates that the tested algorithm provides better accuracy compared to conventional GCNN, and several studies support that statement. For example, the research of Mabruroh, et al, experimented with using LGCNN for giving bots some knowledge to take an action in a game[21]. In another case, LGCNNs are able to classify human emotion based on Electro Encephalography (EEG)[22].

3 Problem Formulations or Methodology

3.1 Simulation Approach

One of the important things about this study is the approach of not using a set of physical equipment for experimenting with the program. Instead of trying the proposed algorithm directly on the device, this study uses a simulated-based approach that creates a virtual environment using game-based simulation to simulate every possible condition of the I-Device on the smart mosque. This study uses a game as our object for simulation mainly because we can minimize the cost of the developing system. In the next development, we can connect the system with metaverse technology, so each user of the system can join a meta-environment to create interaction between the device and other users [23]–[25]. Today metaverse technology became advanced technology because of the impact of the COVID-19 pandemic era. Every aspect of thing on the world became on meta-world from economics to academics [26].

3.2 Generalized Classifier Neural Network

Several studies have developed some adaptations of neural network models to improve their performance such as the Generalized classifier neural network (GCNN). This model used the RBF kernel to generate the results. As a classifier algorithm based on the gaussian RBF kernel, GCNN has a good reputation for generating high accuracy in classifying objects[13]. GCNN itself is one of the feed-forward neural networks in comparison to Probabilistic Neural Networks (PNN) and Generalized Regression Neural Networks (GRNN)[7]

But these benefit sacrifice GCNN computation speed. To manage those problems, the latest study offers a solution using a Logarithmic learning generalized classifier neural network (LGCNN)[20], [21], [27]. The difference of this method with other RBF neural network is in using a logarithmic function to calculate the error for each epoch[12]. With this function, it will decrease the iteration to get the minimum error. Like GCNN this

method contains 5 primary layers: input, pattern, summation, normalization, and output layer.

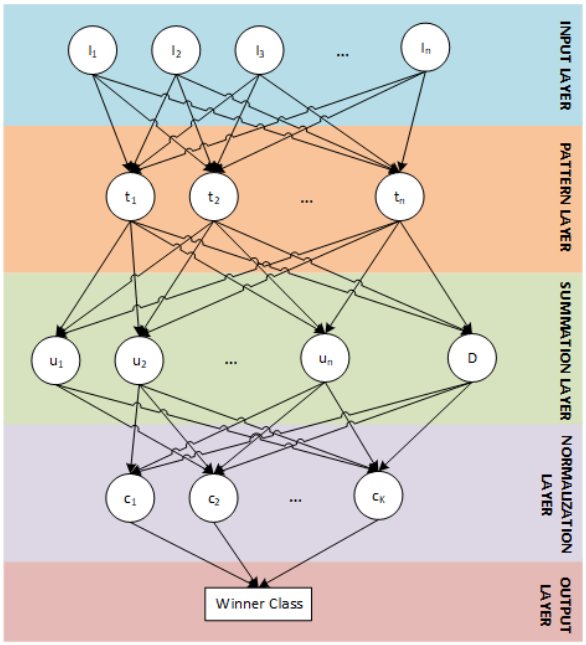

Figure 1. GCNN Architecture by Juniardi, 2022

Fig. 1 shows the general architecture of GCNN. The first layer is an Input vector x_t from our data. The number of neurons in this layer is equal to the number of properties in our data. It will transfer to the pattern layer that consists of neurons in an equal number of the training set. As a first step, the data need to be normalized before continuing to the next step. For each neuron t_i calculate Gaussian RBF kernel based on Euclidian value between training data with the test data, and the output R_i will be calculated as definition 3.1.

Definition 3.1. Gaussian RBF kernel

$$r_j = \exp\left\{\frac{-\|x_t - t_j\|^2}{2\sigma^2}\right\}, 1 \le j \le N$$

where N is the total number of training data and σ represents the smoothing parameter of the model. when training pattern t_i classifying x_i into i_{th} class, then it will set the value of x_i with 0.9, and for the rest class will be 0.1 as shown as definition 3.2.

Definition 3.2. Labelling vector

$$y(j,i) = \begin{cases} 0.9, If\ x_j belongs\ to\ i_{th} class, 1 \le i \le K \\ 0.1, \quad The\ other\ classs, \ 1 \le j \le N \end{cases}$$

Where $y(j,i)$ is the label vector for j^{th} training data and K is the number of classes. Pattern layers will transfer the result to the summation layer that has $K+1$ neurons. That additional neuron is the denominator neuron that will be used for further calculation. The calculation of diverges effect term value $d(j,i)$ for i^{th} class on j^{th} training data will be processed using definition 3.3.

Definition 3.3 diverge effect term

$$d(j,i) = \exp\{y(j,i) - y_{max}\} \times y(j,i)$$

This y_{max} represent the maximum value of $y(j,i)$ where it will be assigned as 0.9 for the first iteration. The diverge effect terms value will classify data based on the class. The exponential function offers restrictions on over-fitting problems and convergence to minimum error. Furthermore, on this summation layer, denominator D and output of the summation layer u_i generated by using definition 3.4 and 3.5.

Definition 3.4 Summation Layer

$$u_i = \sum_{j=1}^{N} d(j,i) \times r(j) \ , \ 1 \leq i \leq K$$

Definition 3.5 Denominator

$$D = \sum_{j=1}^{N} r(j)$$

where $r(j)$ is pattern layer output and $d(j,i)$ is diverge effect term calculated by definition 3.3. The summation layer result will be normalized on the normalization layer to produce the winner class. By dividing the output layer with denominator D we obtained c value for each class given as definition 3.6.

Definition 3.6 Class Value

$$c_i = \frac{u_i}{D} \ , 1 \leq i \leq K$$

This $c_i$value determined the rank of the winner class. As shown in definition 3.7, the first rank of the class will be the winner neurons or the final action of the input pattern.

Definition 3.7 Winner class

$$[o, id] = \max(c)$$

The neural network algorithm relies on error calculation for generating new weights for the next iteration. Because of that, the gradient descent-based formula takes a role in minimizing the square error of the system. The formula for calculating square error is shown as Definition 3.8

Definition 3.8 Mean Squared Error

$$e = (y - f)^2$$

This y belongs to the desired value and f is a value from the output of the neural network. For converging e to its minimum value, a derivative of definition 3.8 is calculated by w. The function can be seen as theorem 3.1

Theorem 3.1 Error derivative by w

$$\frac{\partial e}{\partial w} = 2 \times \frac{\partial(-f)}{\partial w}$$

For GCNN Winner neuron value will be stored for updating diverge effect term then new smoothing parameter will be calculated using Gradient decent approach shown as definition 3.9.

Definition 3.9 New smoothing parameter

$$\sigma_{new} = \sigma_{old} + lr \times \Delta e$$

Where lr is the representation of learning rate and Δe was a gradient of square error e. Refers to definition 3.8, GCNN calculates the square error as formula as shown in definition 3.10.

Definition 3.10. GCNN squared error

$$e = (y(z, id) - c_{id})^2$$

This cost function e was the least square cost function based on regression used on ordinary GCNN and it produce better classification results.

4 The Proposed Method

4.1 Logarithmic Learning Neural Network

While it generated good result, GCNN facing a convergence problem as it need a long-time process to converge. To try to tackle this problem, LGCNN come up an idea with replacing the least square cost function with a logarithmic cost function.

Logarithmic cost function takes a place to reduce iterations of process reaching minima. The formula for e calculation on LGCNN is given as definition 4.1.

Definition 4.1. Logarithmic cost function

$$e = (y(z, id) \times \log(c_{id})) + \big(1 - y(z, id)\big) \times \log(1 - c_{id})$$

This introduced method replaced the conventional one that is still based on regression analytics. Though it has more complexity, it greatly reduces iteration to convergence. After calculated cost function e, σ_{new} generated by calculating the gradient of the cost function using derivative of definition 4.1 as shown in theorem and lemma 4.1.

Theorem 4.1. LGCNN cost function

$$\frac{\partial e}{\partial \sigma} = y_{(z,id)} \times \left(\frac{\frac{\partial c_{id}}{\partial \sigma}}{c_{id}}\right) + \big(1 - y_{(z,id)}\big) \times \left(\frac{\frac{\partial c_{id}}{\partial \sigma}}{c_{id}}\right)$$

$$\frac{\partial c_{id}}{\partial \sigma} = \frac{b(id) - l(id) \times c_{id}}{D}$$

$$b(id) = 2 \times \sum_{j=1}^{p} d(j, i) \times r(j) \times \frac{dist(j)}{\sigma^3}$$

$$l(id) = 2 \times \sum_{j=1}^{p} r(j) \times \frac{dist(j)}{\sigma^3}$$

For updating the value of diverge effect term, y_{max} updated with the value of the winner class c_{id} each iteration.

4.2 System Architecture

Our study used a flux sensor for detecting the brightness of the environment. They are placed spread over the environment in which depending on the pattern it will decide which lamp will take an action. The diagram of our system can be seen in fig 2.

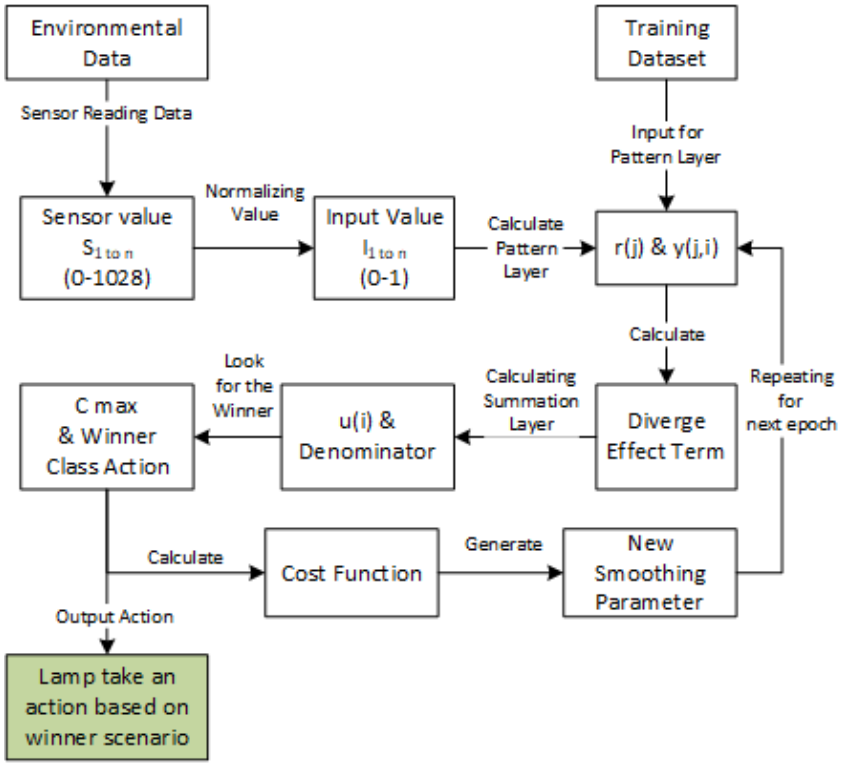

Figure 2. Proposed System Architecture

Training data used for the proposed architecture was for determining the best solution by calculating Euclidian value from input to the given training data then those values will help to obtain the RBF activation function. The training set combines several values of the sensor and generates action. That action controls several lamps' activation on simulation. Each action will be trained with 3 different given data and each data contain 25,50, and 100 data to be trained. Several scenarios will be conducted to test the model's response.

The result from the model will be applied to the simulation with some object in the virtual mosque. That mosque building has lamps that need to be active by the condition of the given sensor. The map of simulation can be seen in fig. 3

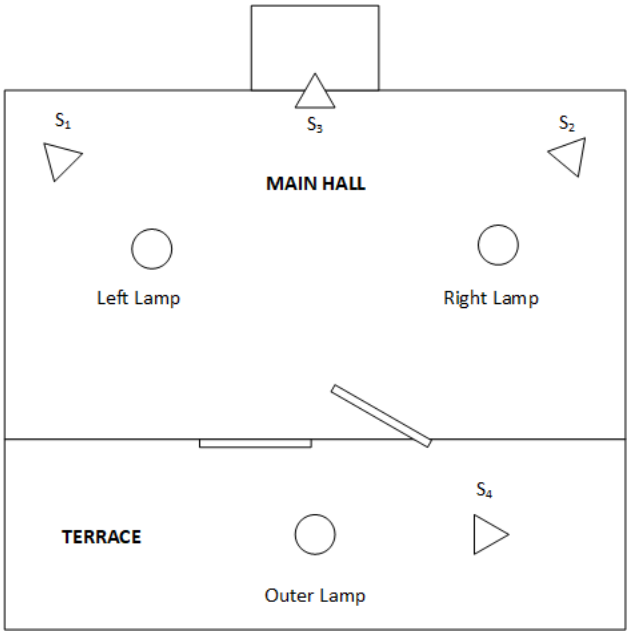

Figure 3. Room design on mosque simulation

There are two places for lamp placement, inside and outside. Sensors with a total of 4 sensors were spread inside building and outside the building. When the flux sensor detected a fluctuation, one of the actions in table 1 will be selected based on the value of the sensors.

Table 1: Action Detail for Lamp

ACTION ID	LEFT LAMP	RIGHT LAMP	OUTER LAMP
0	0	0	0
1	0	0	100
2	0	100	0
3	0	100	100
4	100	0	0
5	100	0	100
6	100	100	0
7	100	100	100

4.2. Measuring performance

Classification models need to measure their performance to be perfectly running[28]. This performance evaluation normally can be measures by accuracy, precision, and recall or sensitivity of the algorithm based on the confusion matrix. A confusion matrix is a very popular measurement used while solving classification problems. It can be applied to binary classification as well as to multiclass classification problems[29]. Calculating accuracy means that we analyze how accurate our method predicting or classifying some pattern. Accuracy value will form by calculating true positive, true negative, false positive, and false negative with formula definition 4.2.

Definition 4.2. Accuracy Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision shows the accuracy of desired data with prediction results given by the algorithm[30]. As shown in definition 4.3.

Definition 4.3. Precision Confusion Matrix

$$Precision = \frac{TP}{TP + FP}$$

Recall or sensitivity expresses the correctness of the model to find back the information. The process of recall can be seen in definition 4.4.

Definition 4.4 Recall Confusion Matrix

$$Recall\,/Sensitivity = \frac{TP}{TP + FN}$$

Technically, a good classifier will generate a value closer to 1 for $precision$ and $recall$. $F1 - score$ metric come to take $precision$ and $recall$ into account as given definition 4.5.

Definition 4.5. F1 Score

$$F1_{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

5 Results, Analysis and Discussions

Several experiments have been conducted to test how accurate the LGCNN in making decisions for simulated Intelligent Lamp (I-Lamp). We compare the use of conventional GCNN with L-GCNN to accommodate lamp action. A test for analyzing the converging performance produces a result shown as σ-graph in figure 4.

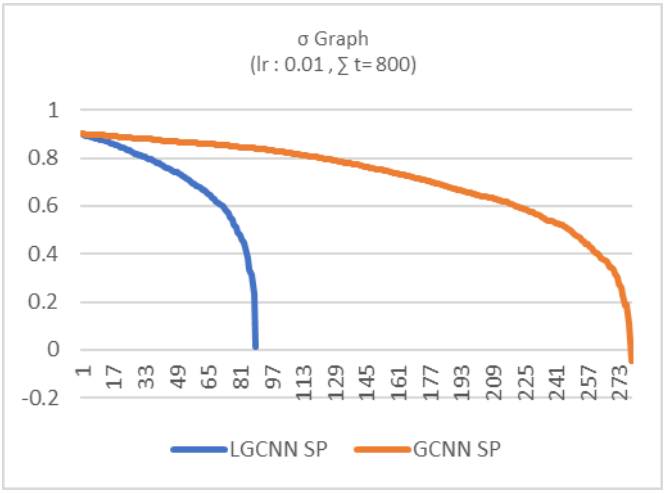

Figure 4. Smoothing Parameter Graph for 800 data training

As we can see in the figure, with 800 training data and a learning rate initial value is 0.01, the LGCNN graph became convergence faster than GCNN. LGCNN took around 90 times of iteration to be convergence while GCNN need over 273 times of iteration. Furthermore, a different graph pattern shows as the result of comparing these two methods.

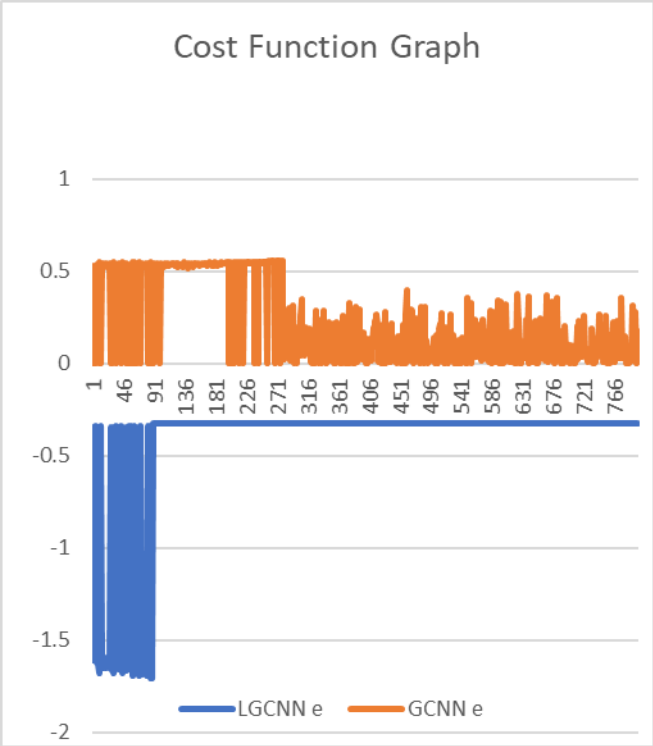

Figure 5. Chart between GCNN and LGCNN (= 0.01)

Comparation between GCNN square error and LGCNN cost function can be seen in figure 5. In figure 5, it is safe to say that the LGCNN cost function generates a negative value while GCNN positive. According to figure 4, when the smoothing parameter converges, there is a change on the cost function graph as the value of cost function in the models became closer to zero.

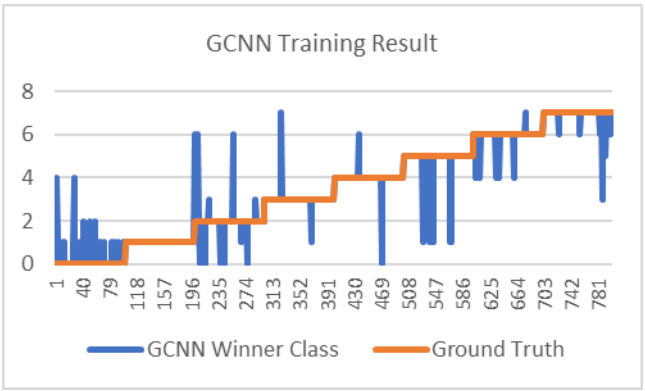

Figure 6. GCNN training result compared with the training data ground truth

The training result of the GCNN model shown in figure 6, denotes that the GCNN model still has several errors. The spike on the graph shows the class that was misinterpreted by

the model as it makes 7% wrong action. Meanwhile, the LGCNN result is better than GCNN. As the graph shows in fig.7, the LGCNN model provides fewer mistakes than GCNN with only 2.875% wrong action.

Another performance test using a confusion matrix has been calculated to evaluate the two models. The result is shown in table 2 as accuracy value and F-1 score between the two models.

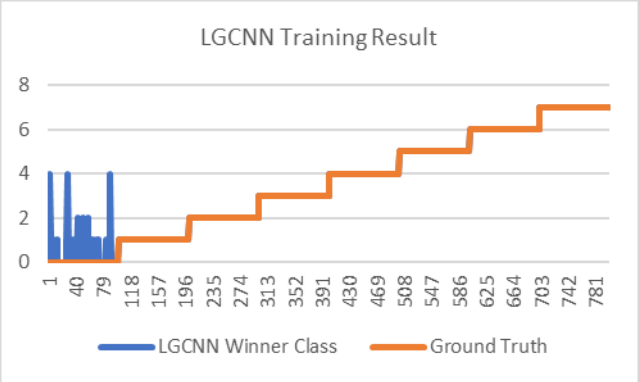

Figure 7. LGCNN training result compared with the training data ground truth

Table 2. Accuracy and F-1 Score of two model

Dataset	GCNN		L-GCNN	
	Accuracy	F1-Score	Accuracy	F1-Score
125	88%	93.02%	87.20%	92.38%
250	72.80%	84.49%	89.20%	93.85%
375	74.66%	80.30%	88.27%	92.31%
500	90.00%	90.74%	88.40%	89.95%
Average	81%	87%	88%	92%

According to comparison of performance between GCNN and LGCNN in Table 2, it can be seen that in Accuracy, LGCNN models produce an average score 7% better than GCNN models. And along with the accuracy F-1 score of those two methods, indicates that LGCNN obtained 5% greater than GCNN. by average, LGCNN models outperform GCNN by accuracy and F-1 score.

6 Conclusion

Neural networks as classifier algorithms can be used for many things including making decisions for choosing action on smart devices. LGCNN as the suggested model can optimize GCNN weakness very well. With ±90% of average accuracy to f1 score, LGCNN

could be used in a simulation to provide the right decision. Further experience it is suggested to add more variable and different case to test the consistency of the model. More training data with increasing iteration process are also suggested to measure the efficiency of the model.

ACKNOWLEDGEMENTS

The researcher expresses his deepest gratitude to the Ministry of Religion and UIN Maulana Malik Ibrahim Malang and the LP2M team of universities because with the help of these parties this research can run well. We also thank all fellow researchers and the mosque officer who became the center of research that researchers have carried out in completing their research

References

[1] R. Agus Zandra Kurniawan, S. Wahjuni, and S. Nidya Neyman, "Secure Communication Protocol for Arduino-based IoT Using Lightweight Cryptography," vol. 12, no. 2, 2022.

[2] S. Villamil, C. Hernández, and G. Tarazona, "An overview of internet of things," Telkomnika (Telecommunication Computing Electronics and Control), vol. 18, no. 5, pp. 2320–2327, Oct. 2020, doi: 10.12928/TELKOMNIKA.v18i5.15911.

[3] N. Bafdal and I. Ardiansah, "Application of Internet of Things in Smart Greenhouse Microclimate Management for Tomato Growth," vol. 11, no. 2, 2021.

[4] S. M. Basha and D. S. Rajput, "Survey on Evaluating the Performance of Machine Learning Algorithms: Past Contributions and Future Roadmap," in Deep Learning and Parallel Computing Environment for Bioengineering Systems, Elsevier, 2019, pp. 153–164. doi: 10.1016/b978-0-12-816718-2.00016-6.

[5] A. Novianty, C. Machbub, S. Widiyantoro, and I. Meilano, "Tsunami Potential Prediction using Seismic Features and Artificial Neural Network for Tsunami Early Warning System," vol. 12, no. 1, 2022.

[6] J. O. Pinzón-Arenas, R. Jiménez-Moreno, and A. Rubiano, "Path Planning for Robotic Training in Virtual Environments using Deep Learning," vol. 12, no. 1, 2022.

[7] M. Oral, S. Kartal, and B. M. Özyıldırım, "A cluster based approach to reduce pattern layer size for generalized regression neural network," Pamukkale University Journal of Engineering Sciences, vol. 24, no. 5, pp. 857–863, 2018, doi: 10.5505/pajes.2017.76401.

[8] G. di Franco and M. Santurro, "Machine learning, artificial neural networks and social research," Qual Quant, vol. 55, no. 3, 2021, doi: 10.1007/s11135-020-01037-y.

[9] D. S. B. Roomhead ' and D. Id Lowe, "Com plex Sy st em s 2 (1988) 321-355 M ult ivar ia ble Functional Interpolation and Adaptive N etworks."

[10] S. See et al., "Radial Basis Function (RBF) Neural Network: Effect of Hidden Neuron Number, Training Data Size, and Input Variables on Rainfall Intensity Forecasting," vol. 9, no. 6, 2019.

[11] B. M. Ozyildirim and M. Avci, “One pass learning for generalized classifier neural network,” Neural Networks, vol. 73, 2016, doi: 10.1016/j.neunet.2015.10.008.

[12] B. M. Ozyildirim and M. Avci, “Logarithmic learning for generalized classifier neural network,” Neural Networks, vol. 60, pp. 133–140, Dec. 2014, doi: 10.1016/j.neunet.2014.08.004.

[13] B. Melis and M. Avci, “Generalized classifier neural network,” Neural Networks, vol. 39, pp. 18–26, 2013, doi: 10.1016/j.neunet.2012.12.001.

[14] M. Fetanat, M. Stevens, P. Jain, C. Hayward, E. Meijering, and N. H. Lovell, “Fully Elman Neural Network: A Novel Deep Recurrent Neural Network Optimized by an Improved Harris Hawks Algorithm for Classification of Pulmonary Arterial Wedge Pressure,” IEEE Trans Biomed Eng, vol. 69, no. 5, 2022, doi: 10.1109/TBME.2021.3129459.

[15] I. Putu, A. Dharmaadi, D. Witarsyah, A. Bayupati, G. Made, and A. Sasmita, “Face Recognition Application Based on Convolutional Neural Network for Searching Someone’s Photo on External Storage,” vol. 12, no. 3, 2022.

[16] H. Panchal, S. Mishra, and V. Shrivastava, “Chess Moves Prediction using Deep Learning Neural Networks,” 2021. doi: 10.1109/ICACC-202152719.2021.9708405.

[17] Z. Ahmad et al., “Anomaly detection using deep neural network for iot architecture,” Applied Sciences (Switzerland), vol. 11, no. 15, Aug. 2021, doi: 10.3390/app11157050.

[18] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for IoT intrusion detection system,” Simul Model Pract Theory, vol. 101, May 2020, doi: 10.1016/j.simpat.2019.102031.

[19] S. M. Naik, R. Prasad, K. Jagannath, and V. Kuppili, “OPTIMIZED LAPLACIAN GENERALIZED CLASSIFIER NEURAL NETWORK.”

[20] D. Heru Murti and dan Wijayanti Nurul Khotimah, “PENGENALAN SISTEM ISYARAT BAHASA INDONESIA MENGGUNAKAN KOMBINASI FITUR STATIS DAN FITUR DINAMIS LMC BERBASIS L-GCNN.”

[21] I. Mabruroh and D. Herumurti, “Adaptive Non Playable Character in RPG Game Using Logarithmic Learning For Generalized Classifier Neural Network (L-GCNN),” Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 2019, doi: 10.22219/kinetik.v4i2.755.

[22] S. B. Musa, “Klasifikasi Emosi Sinyal EEG berdasarkan Empirical Mode Decomposition dan Wavelet Packet Decomposition menggunakan Logarithmic Learning for Generalized Classifier Neural Network,” Tesis Its, 2017.

[23] S. Lee, G. Ha, H. Kim, and S. Kim, “A collaborative Serious Game for fire disaster evacuation drill in Metaverse,” Journal of Platform Technology, vol. 9, no. 3, 2021.

[24] T. Ai, “Metaverse Theory,” SSRN Electronic Journal, 2021, doi: 10.2139/ssrn.3840764.

[25] E. Shin and J. Hyun Kim, ″The Metaverse and Video Games: Merging Media to Improve Soft Skills Training ☆,″ Journal of Internet Computing and Services, no. 1, 2022.

[26] Y. Wang, L.-H. Lee, T. Braud, and P. Hui, "Re-shaping Post-COVID-19 Teaching and Learning: A Blueprint of Virtual-Physical Blended Classrooms in the Metaverse Era," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.09228

[27] A. Baso Kaswar, "Sistem Klasifikasi Jenis Jeruk Impor Menggunakan Metode Klasifikasi Logarithmic Generalized Classifier Neural Network (LGCNN) Orange Import Type Classification System Using Logarithmic Generalized Classifier Neural Network (LGCNN) Classification Method."

[28] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," Inf Sci (N Y), vol. 507, 2020, doi: 10.1016/j.ins.2019.06.064.

[29] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering, Elsevier, 2020, pp. 83–106. doi: 10.1016/B978-0-12-818366-3.00005-8.

[30] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021.

Notes on contributors

Juniardi Nur Fadila is a lecturer at the Department of Informatics, UIN Maulana Malik Ibrahim of Malang, East Java, Indonesia. His main teaching and research interests include Artificial Intelligent, Robotics and Multimedia programming. He has published several research articles in international and national journals of informatics concern to his interest.

M. Amin Hariyadi is a lecturer at the Department of Informatics Engineering and Informatic Master Degree Program, His main teaching and research interests include Artificial Intelligent, Robotics and Image Processing. He has published several research articles in international and national journals of informatics concern to his interest.

Ajib Hanani is a lecturer at the Department of Informatics Engineering and Informatic Master Degree Program, his main teaching and research interests include Robotics and Internet of Things, Embedded System. He graduated from UIN Maulana Malik Ibrahim for bachelor Degree and continue on Brawijaya University for his Master Degre. Although he is a lecture he handled journal system in UIN Maulana Malik Ibrahim Malang.

Johan Ericka Wahyu is a lecturer at the Department of Informatics Engineering and Informatic Master Degree Program, His main teaching and research interests include System & Network, Internet of Things, and Embedded System. He has published several research articles in national journals of informatics concern to his interest. Although he is a lecture, in UIN Maulana Malik Ibrahim, he have a task to manage Faculty Information Website.