Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 8, No. 2, April, 2024

Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification

Khang Wen Goh [1], Sugiyarto Surono [2*], M. Y. Firza Afiatin [2], K. Robiatul Mahmudah [2], Nursyiva Irsalinda [2], Mesith Chaimanee [3], Choo Wou Onn [1]

[1] *Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia.*

[2] *Mathematics Study Program, Ahmad Dahlan University, Yogyakarta, Indonesia.*

[3] *Faculty of Engineering and Technology, Shinawatra University, Pathum Thani 12160, Thailand.*

Abstract

Deep learning, specifically the Convolutional Neural Network (CNN), has been a significant technology tool for image processing and human health. CNNs, which mimic the working principles of the human brain, can learn robust representations of images. However, CNNs are susceptible to noise interference, which can impact classification performance. Choosing the right activation function can improve CNNs performance and accuracy. This research aims to test the accuracy of CNN with ResNet50, VGG16, and GoogleNet architectures combined with several activation functions such as ReLU, Leaky ReLU, Sigmoid, and Tanh in the classification of images that experience Poisson noise. Poisson noise is applied to each test data to evaluate CNN accuracy. The data used in this study consists of three scenarios of different numbers of classes, namely 3 classes, 5 classes, and 10 classes. The results showed that combining ResNet50 with the ReLU activation function produced the best performance in class recognition in each scenario of the number of classes experiencing Poisson noise interference. The model achieved 97% accuracy for 3-class data, 95% for 5-class data, and 90% for 10-class data. These results show that using ResNet50 with the ReLU activation function can provide excellent resistance to Poisson noise in image processing. It was found that as the number of classes increases, the accuracy of image recognition tends to decrease. This shows that the more complex the image classification task is with a larger number of classes, the more difficult it is for CNNs to distinguish between different classes.

Keywords:

Activation Function;
Classification;
Convolutional Neural Network;
Poisson Noise.

Article History:

Received:	23	November	2023
Revised:	26	February	2024
Accepted:	07	March	2024
Published:	01	April	2024

1- Introduction

Deep Learning (DL) is one of the widely used technological tools because to its ability to represent images [1]. DL algorithms were designed with the aim of mimicking the function of the human brain, which has many hidden layers [2]. Deep learning (DL) algorithms have demonstrated notable success across diverse fields [3]. In the context of image classification, DL methods are often the first choice [4]. Another one of the most popular DL methods used for many image processing and human health applications is the convolutional neural network [5]. Convolutional neural network (CNN) is a machine learning algorithm that has shown to make good results for image classifications [6]. In recent years, CNN has attracted much attention due to its feature extraction capabilities and application to image classification [7]. CNN is an excellent classification method and is used widely today. One of the reasons is that CNNs provide high accuracy [8].

* **CONTACT**: sugiyarto@math.uad.ac.id

DOI: http://dx.doi.org/10.28991/ESJ-2024-08-02-014

Based on research by Ali Narin on coronavirus detection using the CNN method with the ResNet50 architecture, the accuracy value is 96.1% for dataset-1, 99.5% for dataset-2 and 99.7% for dataset-3 [9]. Then other research conducted by Sarah Mohd Ashhar on lung cancer classification using the CNN method with GoogleNet architecture resulted in an accuracy value of 94.53% [10]. Furthermore, another study conducted by Ebru Erdem on the detection of pneumonia using the CNN method with VGG16 architecture resulted in an accuracy value of 88.78% [11].

Despite the fact that CNNs have proven to be effective in classifying images, they are highly susceptible to noise. The impact of such noise interference on CNN performance in image classification is one of the most important and interesting issues in the field of image classification [12–14]. So it is important to choose the necessary activation function to increase the flexibility and efficiency of the network [15]. The activation function is a key component of convolutional networks, which can map non-linear characteristics [16]. The performance of CNNs can be significantly affected by the selection of an appropriate activation function. The activation function plays an important role in the output calculation process. The type of activation function used is very important in determining the output value of the CNN [17].

Momeny et al. used the advantages of the CNN method for the classification of images subject to uniform noise based on their previous research on the classification of images subject to noise. They recommended in his research that the type of noise be replaced with Poisson noise [18].

Based on this information, we intend to test the accuracy of the CNN method by using three different architectures, specifically VGGNet Medium, GoogleNet and ResNet50, to classify images that have been contaminated by Poisson noise. In addition, in this series of experiments we will also look at using different activation mechanisms such as ReLU, Leaky ReLU, Sigmoid and Tanh.

2- Method

2-1- Resize

Resize is the process where the size of an image, either a digital image or a photograph, is changed to make it smaller or larger. The nearest neighbor interpolation technique is one of the techniques used for resizing. This method uses the information of the value pixel to generate a new image of the image to be resized [18]. Mathematically, resize can be represented as follows [19]:

$$b_{ij} = a_{\lfloor i \times x \rfloor \lfloor j \times y \rfloor} \tag{1}$$

where b_{ij} is the pixel value of the i-th row of the j-th column of the resized image, a is the image element to be resized, i is the row index, j is the column index, x is the ratio between the number of rows in the input image and the output image, and y is the ratio between the number of columns in the input image and the output image.

2-2- Min Max Scale

Min-max scaler is a normalization method used to transform all values in a dataset into a range between 0 and 1. This method utilizes the smallest and largest values in the dataset to perform the transformations [19]. Using B data, the min-max scaler can be mathematically represented as follows:

$$r_{ij} = \frac{b_{ij} - \min(B)}{\max(B) - \min(B)} \tag{2}$$

where r_{ij} is the pixel value of the i-th row of the j-th column of the normalized image, b_{ij} is the pixel value of the i-th row of the j-th column of the image to be normalized, $\min(B)$ is the minimum value in data B, and $\max(B)$ is the maximum value in data B.

2-3- Noise

Noise is a common problem in image processing that often arises from the process of taking images by utilizing cameras and transmitting information on communication systems. As a result, the transmitted image often experiences disturbances that result in a mismatch with the original image. Noise causes the pixel intensity value in the image to not reflect the actual image value [20].

Noise in digital images can generally be explained as random variations of brightness or color information in the captured image. This variation arises due to the influence of external factors. A visual example of an image with noise can be seen in Figure 1.

Figure 1. Image Noise

Mathematically, the image function affected by noise can be described as follows:

$$A(i,j) = H(i,j) + B(i,j) \tag{3}$$

where $A(i,j)$ is the noisy image function, $H(i,j)$ is the original image function, and $B(i,j)$ is the noise function.

There are various types of noise that are often found in digital images, one of which is Poisson noise [21]. Poisson noise is generated based on the Poisson distribution and is caused by the random appearance of photons in the image [22]. The Poisson distribution can be shown as follows:

$$X{\sim}POI(x,\lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!} \tag{4}$$

where x is the value of the random variable X and λ is the average parameter that controls the intensity level of the Poisson distribution.

2-4- Convolutional Neural Network (CNN)

Convolutional Neural Networks is the type of neural network that is routinely used for simplifying images to improve analysis and classification. It reduces the need for intensive human intervention and preprocessing, which is a major advantage of this network [23]. CNNs are used in a variety of image processing tasks, including image recognition, image segmentation, and object detection [19]. Convolutional and pooling layers are responsible for the extraction of important features from the data, while a fully connected layer is used to map these features to the final output, such as in the classification task [24]. The basic construction of this CNN can be seen in Figure 2.

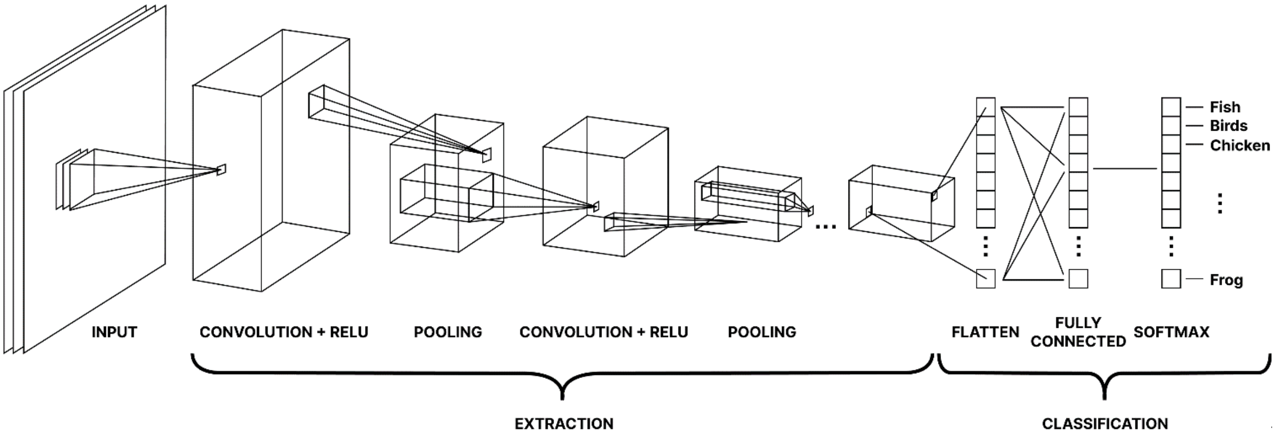

Figure 2. Convolutional Neural Network (CNN)

An important element in CNN that plays an important role in extracting features from the input image isvolutional. A convergence operation is a mathematical process through which two matrices are used: the input image and the kernel matrix. This multiplication is a process in which both the elements of these two matrices are multiplied and then summarised. Usually, the convolution operation is indicated by the * operator. In mathematics, the convolution operation can be represented as the follows [25].

$$z_{i,j}^{h} = \sum_{m=1}^{M}\sum_{n=1}^{N} a_{(i-1)s+m,(j-1)s+n} * k_{m,n} + b^{h} \tag{5}$$

where $z_{i,j}^{h}$ is the output of the convolutional operation with i-th row, j-th column, and h-th channel. b^{h} is the bias of the h-th channel. $k_{m,n}$ is an element of the K matrix, while $a_{(i-1)s+m,(j-1)s+n}$ is an element of the input image matrix. The parameters M and N are the kernel sizes, while s is the stride size that determines the shift of the kernel during convolutional operation.

2-4-1- CNN Architecture

In this research, the CNN architectures used are VGG-Net-Medium, ResNet50 and GoogleNet architectures. The training process was conducted using 100 epochs with a batch size of 64. These architectures are presented in Figures 3 to 5.

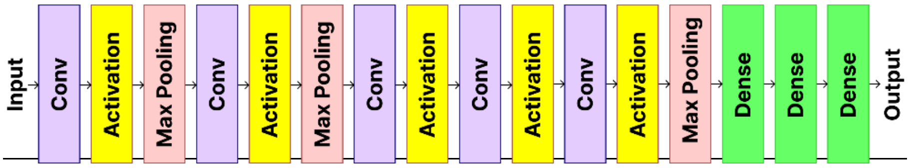

Figure 3. VGG-Net-Medium Architecture

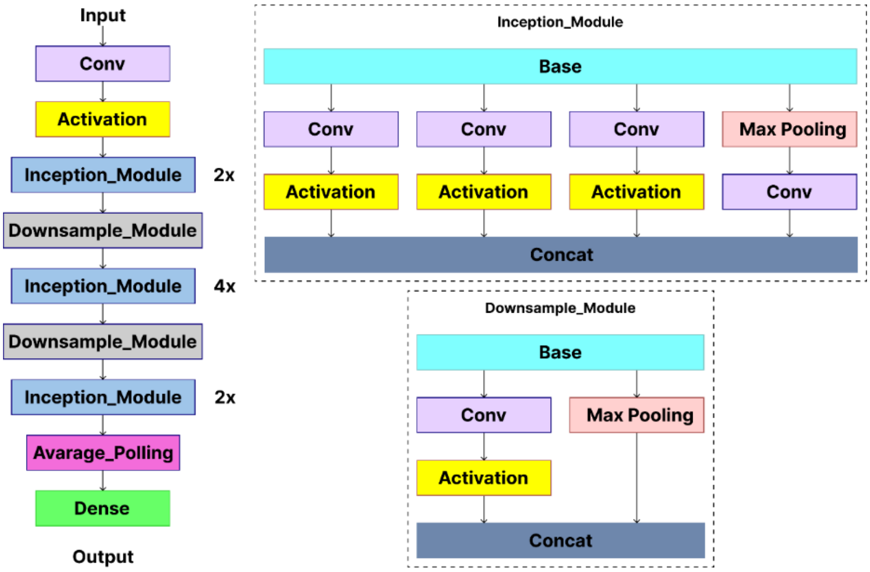

Figure 4. ResNet50 Architecture

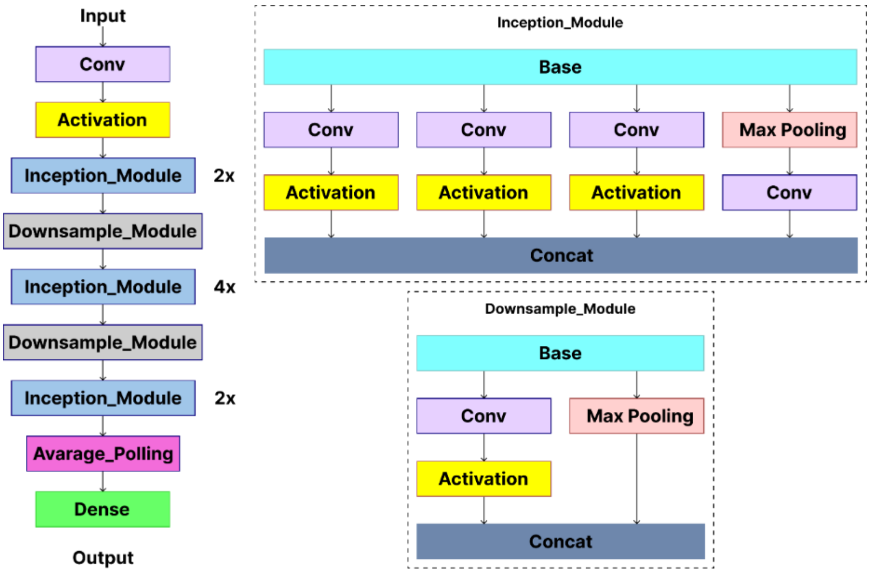

Figure 5. GoogleNet Architecture

2-5- Activation Function

The activation function is used to output a hidden layer and an output layer in a CNN. The input of the activation function is the result of a linear combination of the input and its weights. In CNN, some activation functions that are often used are as follows:

- Rectified Linear Unit (ReLU) function

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}, \ range = [0, \infty) \tag{6}$$

- Leaky ReLU function

$$f(x) = \begin{cases} \alpha x, & x \leq 0 \\ x, & x > 0 \end{cases}, \ range = (-\infty, \infty) \tag{7}$$

- Sigmoid function

$$f(x) = \frac{1}{1+e^{-x}}, \ range = (0,1) \tag{8}$$

- Hyperbolic Tangent (Tanh) function

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \ range = (-1,1) \tag{9}$$

- Softmax function

$$f(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{N} e^{x_k}}, \ \ j = 1,2,\ldots,N, \ \ range = (-1,1) \tag{10}$$

2-5-1- Pooling

In the Convolutional Neural Network (CNN) architecture, the pooling layer plays an important role. Its main function is to perform subtraction or sub-sampling on the feature map generated from the previous convolution operation [26]. Thus, this layer transforms a large feature map into a smaller one while retaining important information or features. The pooling layer is analogous to the convolution method, where the step and kernel are fixed before pool manipulation [27]. Various types of pooling methods can be applied, such as tree pooling, gate pooling, median pooling, maximal pooling, Global Average Pooling (GAP), and worldwide maximal pooling. Maximal, minimal, and GAP pooling are the most common and widely used methods [28].

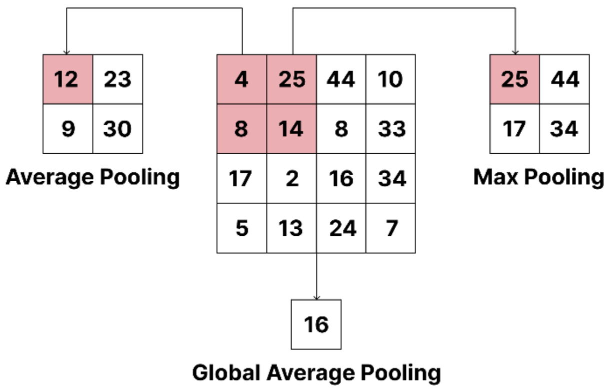

Figure 6. Pooling

Max pooling selects the largest element in each pooling region and is shown in the following equation [25]:

$$p_{i,j}^{h} = \max\left(z_{(i-1)s+m,(j-1)s+n}\right) \tag{11}$$

The average pooling method is a step to take the average of the data in the pooling region. In the average pooling method, this method takes the average of the elements in each region. Mathematically it can be shown as follows [25]:

$$p_{i,j}^{h} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} z_{(i-1)s+m,(j-1)s+n} \tag{12}$$

where $p_{i,j}^{h}$ is the output of the pooling operation of the i-th row, j-th column, and h-th channel. While $z_{(i-1)s+m,(j-1)s+n}$ are elements of the input image matrix. The parameters M and N are the pooling sizes, while s is the stride size that determines the shifting of the pooling window during the operation.

2-5-2 Fully Connected

The Fully Connected (FC) layer, is usually located at the end of the CNN architecture. In this layer, each neuron is connected to all neurons in the respective previous layer, this is referred to as the Fully Connected (FC) approach. It is used as a CNN classifier [29]. This layer follows the basic method of a conventional multi-layer perceptron neural network, as it is a type of Artificial Neural Network (ANN) so it is feed-forward in nature. The input of the FC layer is from the last pooling or convolution layer. This input is a vector, which is created from the feature map after the process of smoothing. The output of the FC layer is the final output of the CNN, as illustrated in Figure 7 [28].

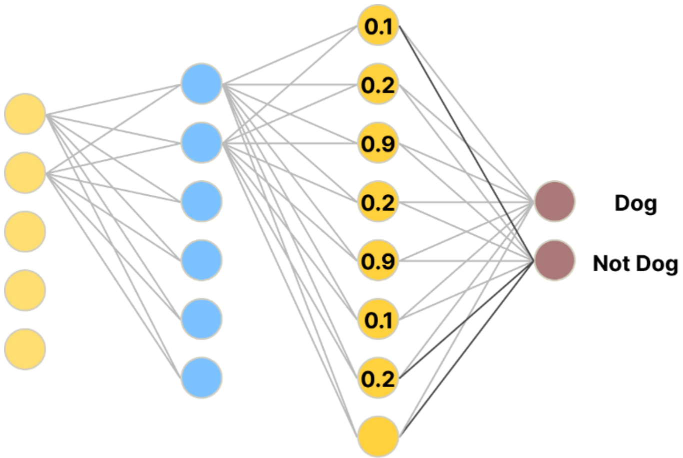

Figure 7. Fully Connected Layer

2-5-3- Loss Function

In terms of optimization algorithms, there is a function used to measure the goodness or badness of the solution found, and this function is often called a loss function. The goal of the optimization algorithm is to get a result that minimizes or maximizes the value of the loss function [30]. The value generated by the loss function is often called the loss. Loss functions play an important role in simplifying complex information to a single scalar number, allowing the comparison and ranking of different solutions. The selection of an appropriate loss function is very important, because poor loss function selection can produce poor results in the optimization process [31]. One type of loss function that is commonly used, especially in the case of multiclass classification, is cross entropy. Cross entropy is commonly used to measure the degree of difference between the probability distribution estimated by a model and the actual probability distribution. This can help to measure the extent to which our model really understands and is able to estimate the data correctly [32].

Mathematically, the cross entropy formula can be represented as follows:

$$L(t,\hat{y}) = -\sum_{i=1}^{n} t_i \ln(\hat{y}_i) \tag{13}$$

where $L(t,\hat{y})$ is the loss value, n is the number of classes, i is the data index, t_i is the i-th true value, and $\hat{y}_i$ is the i-th predicted value.

2-5-4- Backpropagation

Backpropagation is a process in the Neural Network (NN) algorithm that serves to improve the weights in each layer in an architecture [15]. In contrast to the feedforward algorithm, this algorithm moves in reverse in the process so that the weight update process is done from the last layer to the first layer. The calculation for the backpropagation algorithm uses an optimization method. In this research, the optimization method applied is adaptive moment estimation.

Adaptive Moment Estimation (ADAM) is a parameter optimization method commonly used in NNs with large amounts of data. As an optimization method, ADAM is able to adapt to the system to optimize parameters [33]. In the NN backpropagation algorithm, ADAM is used to update the weight and bias values so that they become optimal. Mathematically, ADAM iteration can be expressed as follows:

$$w_{new} = w_{old} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}} \tag{14}$$

Using the values $m_0 = 0$, $v_0 = 0$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, then:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{15}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(g_t)^2 \tag{16}$$

and

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{17}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{18}$$

$$g_t = \frac{dL(w_{old})}{dw_{old}} \tag{19}$$

3- Results and Discussion

Before starting the research, the data set will undergo a series of data preparation steps. In the first step, the images are converted into the same size of 150×150 pixels to ensure consistency of analysis and ease the computational burden. The data is then separated into training and test data for the model training and evaluation process. Furthermore, as an experiment to test the performance of the model under more complex conditions, Poisson noise will be inserted in the test data to simulate the variability of pixel intensity in the image. Finally, a normalization process is run to change the color intensity distance of each color channel into a distance of 0 to 1, which aims to maintain stability and efficiency in the model training process. The required dataset is miniImageNet, which has 60000 images sorted into one hundred classes, but due to hardware limitations, only the first 3 classes, 5 classes, and 10 classes of the total 100 classes are implemented [34].

The research was conducted by utilizing VGG-Net-Medium, ResNet50, and GoogleNet architectures, with various activation functions such as ReLU, Leaky ReLU, Sigmoid, and Tanh. The device required has specifications of 32 GB RAM, Intel Core i7-9800F CPU, and NVIDIA GeForce GTX 1050 GPU. Evaluation of classification results that include confusion matrix is very useful for generating various metrics to get the performance of the classification model [35]. The metric formulations used are shown in Equations 20 to 23. The results obtained by using these formulas are presented in Table 1 and Figure 8.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FN} \tag{20}$$

$$Precision = \frac{TP}{TP+FP} \tag{21}$$

$$Recall = \frac{TP}{TP+FN} \tag{22}$$

$$F1-Score = \frac{2 \cdot Recall \cdot Precision}{Recall+Precision} \tag{23}$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

Table 1. Calculated metrics of the CNN architecture

Number of Classes	Architecture	Activation Functions	Accuracy	Precision	Recall	F1-Score
3	VGG16	ReLU	85%	85%	85%	85%
		Leaky ReLU	88%	87%	88%	88%
		Sigmoid	86%	86%	86%	86%
		Tanh	82%	83%	82%	82%
	ResNet50	ReLU	97%	97%	98%	97%
		Leaky ReLU	93%	93%	93%	93%
		Sigmoid	39%	13%	33%	19%
		Tanh	29%	10%	33%	15%
	GoogleNet	ReLU	94%	94%	94%	94%
		Leaky ReLU	96%	96%	96%	96%
		Sigmoid	88%	89%	89%	88%
		Tanh	86%	87%	87%	86%
5	VGG16	ReLU	81%	82%	81%	81%
		Leaky ReLU	84%	84%	84%	84%
		Sigmoid	76%	77%	76%	76%
		Tanh	78%	79%	78%	78%
	ResNet50	ReLU	95%	95%	95%	95%
		Leaky ReLU	91%	90%	90%	90%
		Sigmoid	22%	4%	20%	7%
		Tanh	21%	4%	20%	7%
	GoogleNet	ReLU	94%	95%	95%	95%
		Leaky ReLU	93%	94%	94%	94%
		Sigmoid	92%	92%	92%	92%
		Tanh	87%	88%	87%	87%

10	VGG16	ReLU	71%	70%	70%	70%
		Leaky ReLU	74%	73%	73%	72%
		Sigmoid	64%	64%	63%	63%
		Tanh	68%	68%	67%	66%
	ResNet50	ReLU	90%	90%	90%	89%
		Leaky ReLU	81%	81%	81%	80%
		Sigmoid	10%	1%	10%	2%
		Tanh	8%	1%	10%	1%
	GoogleNet	ReLU	88%	88%	88%	88%
		Leaky ReLU	87%	87%	87%	86%
		Sigmoid	75%	76%	74%	73%
		Tanh	80%	79%	80%	79%

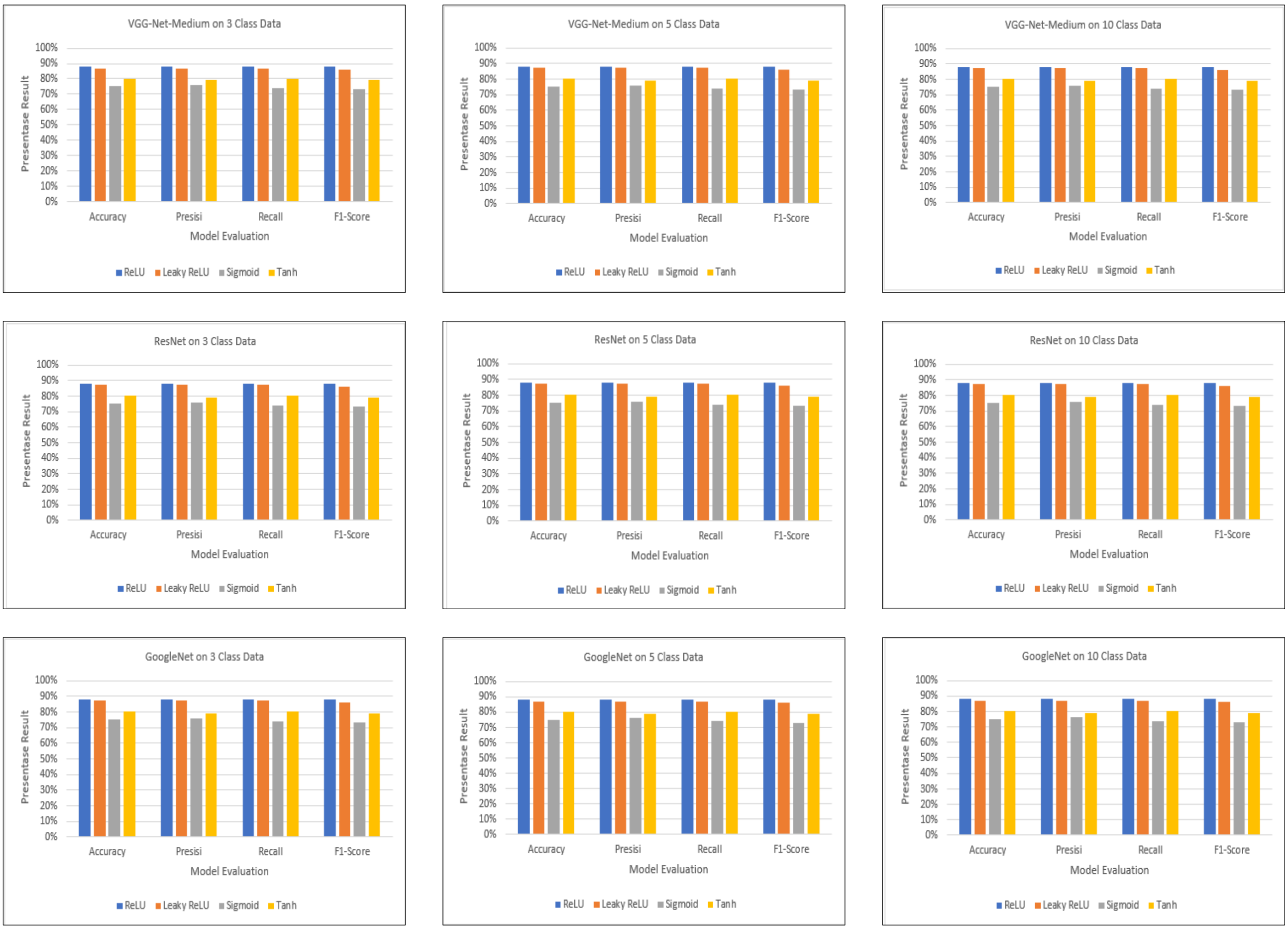

Figure 8. Performance Evaluation

Table 1 and Figure 8 show that in 3 test scenarios using different numbers of classes, namely 3 classes, 5 classes, and 10 classes, it was found that the combination of the ResNet50 architecture utilizing the ReLU activation function showed accuracy results reaching 97% for 3-class data, 95% for 5-class data, and 90% for 10-class data. The accuracy results of each architecture are shown in Figure 9.

Figure 9 shows that there is a tendency for image detection accuracy to decrease as the number of classes increases. This means that the more complex an image classification task is with an increasing number of classes, the more complex the Convolutional Neural Network (CNN) task is to separate the different classes.

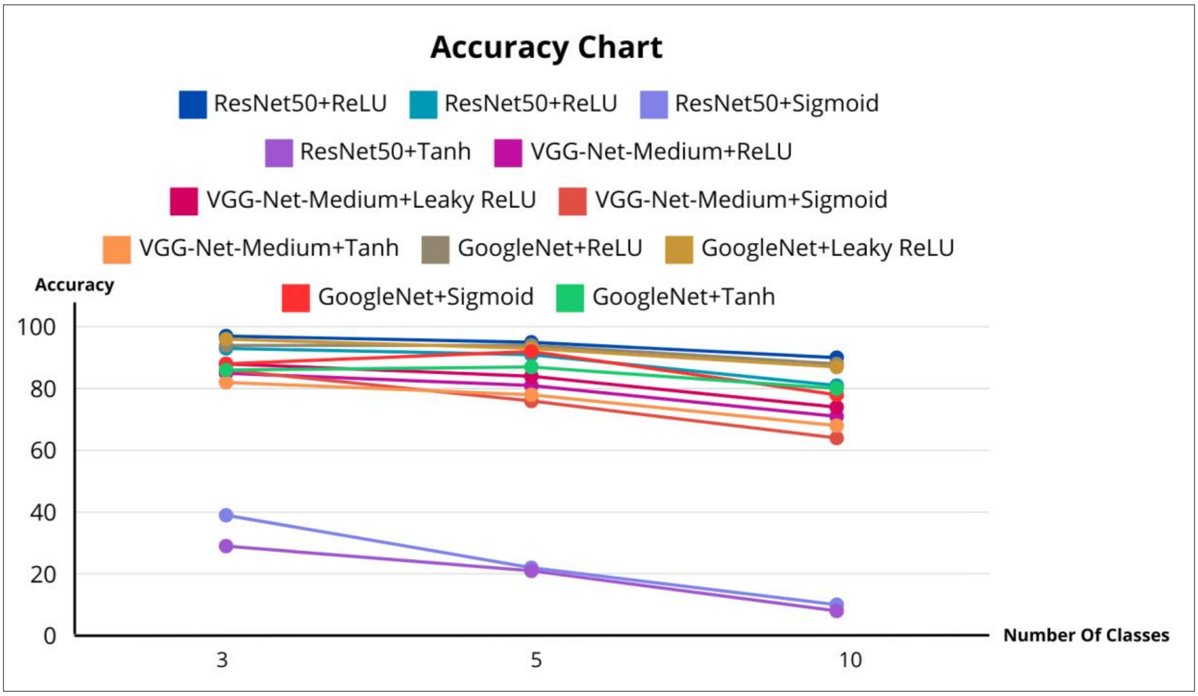

Figure 9. Accuracy Chart

4- Conclusion

In this study, we have run various tests to test how good a Convolutional Neural Network (CNN) with various architectures is at handling image classification problems caused by Poisson noise. We found that the CNNs on the three architectures we tested were able to handle this problem very well, as evidenced by the accuracy obtained. Meanwhile, by observing the differences in the use of some activation functions, we found that applying ReLU to ResNet50 resulted in higher accuracy than other architectures, such as VGG-Net-Medium and GoogleNet. In addition, we also found that ReLU was significantly better than the use of other activation functions such as Leaky ReLU, Sigmoid, and Tanh. These results emphasize the importance of activation function selection in CNN model development.

In addition, we also indicated that the more complex the image recognition task becomes with the increase in the number of classes, the more difficult it is for CNNs to distinguish between different classes, which may result in a decrease in image recognition accuracy. Overall, this study shows that CNNs with different architectures have great potential for addressing the problem of image classification perturbed by Poisson noise. However, the selection of activation functions and the understanding of their impact on CNN performance are important factors to consider in the development of better models in the future.

5- Declarations

5-1- Author Contributions

Conceptualization, G.K.W. and S.G.Y.; methodology, S.G.Y.M.C.; programming, G.K.W., M.Y.F.A., and N.S.A.; validation programming, G.K.W.M.C.; writing—original draft preparation, M.Y.F.A. and K.R.M.; writing—review and editing, C.W.O. and M.C. All authors have read and agreed to the published version of the manuscript.

5-2- Data Availability Statement

Data sharing is not applicable to this article.

5-3- Funding

This study was funded by from Universitas Ahmad Dahlan the fiscal year 2023 with the contract number: 03/RIA/LPPM-UAD/VI/2023 and the Ministry of Education and Culture for the fiscal year 2023 with the contract number: 075/E5/PG.02.00.PL/2023,0254.8/LL5-INT/AL.04./2023, 007/PDUPT/LPPM UAD/IV/2023.

5-4- Acknowledgements

The authors gratefully acknowledge FAST Ahmad Dahlan University Indonesia, especially Mathematic Laboratory, and Faculty of Data Science and Information Technology, INTI International University, Nilai Malaysia for supporting this work.

5-5- Institutional Review Board Statement

Not applicable.

5-6- Informed Consent Statement

Not applicable.

5-7- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

6- References

[1] Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. Applied Sciences (Switzerland), 10(10), 3443. doi:10.3390/app10103443.

[2] Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India. doi:10.1109/icsccc.2018.8703316.

[3] Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. International Journal of Pattern Recognition and Artificial Intelligence, 34(13), 2052013. doi:10.1142/S0218001420520138.

[4] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, 2018. doi:10.1155/2018/7068349.

[5] Mohammed, N. A., Abed, M. H., & Albu-Salih, A. T. (2022). Convolutional neural network for color images classification. Bulletin of Electrical Engineering and Informatics, 11(3), 1343–1349. doi:10.11591/eei.v11i3.3730.

[6] Badža, M. M., & Barjaktarović, M. C. (2020). Classification of brain tumors from MRI images using a convolutional neural network. Applied Sciences (Switzerland), 10(6), 1999. doi:10.3390/app10061999.

[7] Zhao, J., & Li, L. (2021). Research on image classification algorithm based on convolutional neural Network-TensorFlow. Journal of Physics: Conference Series, 2010(1), 12096. doi:10.1088/1742-6596/2010/1/012096.

[8] Ashraf, S., Kadery, I., Chowdhury, A. A., Mahbub, T. Z., & Rahman, R. M. (2019). Fruit Image Classification Using Convolutional Neural Networks. International Journal of Software Innovation, 7(4), 51–70. doi:10.4018/IJSI.2019100103.

[9] Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Analysis and Applications, 24(3), 1207–1220. doi:10.1007/s10044-021-00984-y.

[10] Ashhar, S. M., Mokri, S. S., Rahni, A. A. A., Huddin, A. B., Zulkarnain, N., Azmi, N. A., & Mahaletchumy, T. (2021). Comparison of deep learning convolutional neural network (CNN) architectures for CT lung cancer classification. International Journal of Advanced Technology and Engineering Exploration, 8(74), 126–134. doi:10.19101/IJATEE.2020.S1762126.

[11] ERDEM, E., & AYDİN, T. (2021). Detection of Pneumonia with a Novel CNN-based Approach. Sakarya University Journal of Computer and Information Sciences, 4(1), 26–34. doi:10.35377/saucis.04.01.787030.

[12] Xu, Z., Guo, X., Zhu, A., He, X., Zhao, X., Han, Y., & Subedi, R. (2020). Using deep convolutional neural networks for image-based diagnosis of nutrient deficiencies in rice. Computational Intelligence and Neuroscience, 7307252. doi:10.1155/2020/7307252.

[13] Li, Q., Shen, L., Guo, S., & Lai, Z. (2020). Wavelet Integrated CNNs for Noise-Robust Image Classification. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, United States. doi:10.1109/cvpr42600.2020.00727.

[14] Pei, Y., Huang, Y., Zou, Q., Zhang, X., & Wang, S. (2021). Effects of Image Degradation and Degradation Removal to CNN-Based Image Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(4), 1239–1253. doi:10.1109/TPAMI.2019.2950923.

[15] Busch, N. A., & Micheli-Tzanakou. (1989). Neural network design: methodology. International Joint Conference on Neural Networks, Washington, United States. doi:10.1109/ijcnn.1989.118439.

[16] Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. Applied Sciences (Switzerland), 10(5), 1897. doi:10.3390/app10051897.

[17] Fausett, L. V. (2006). Fundamentals of neural networks: architectures, algorithms, and applications. Pearson Education India, Noida, India.

[18] Momeny, M., Latif, A. M., Agha Sarram, M., Sheikhpour, R., & Zhang, Y. D. (2021). A noise robust convolutional neural network for image classification. Results in Engineering, 10, 100225. doi:10.1016/j.rineng.2021.100225.

[19] Phaladisailoed, T., & Numnonda, T. (2018). Machine Learning Models Comparison for Bitcoin Price Prediction. 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia. doi:10.1109/iciteed.2018.8534911.

[20] Hendarto, J. (2017). Denoising Grayscale Images Using the Fractal Method. PRODUKTIF: Jurnal Ilmiah Pendidikan Teknologi Informasi, 1, 1-12. (In Indonesian).

[21] Owotogbe, J. S., Ibiyemi, T. S., & Adu, B. A. (2019). A comprehensive review on various types of noise in image processing. International Journal of Scientific & Engineering Research, 10(11), 388-393.

[22] Thakur, R. S., Chatterjee, S., Yadav, R. N., & Gupta, L. (2021). Image De-Noising with Machine Learning: A Review. IEEE Access, 9, 93338–93363. doi:10.1109/ACCESS.2021.3092425.

[23] Desai, M., & Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). Clinical EHealth, 4, 1–11. doi:10.1016/j.ceh.2020.11.002.

[24] Patil, A., & Rane, M. (2021). Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. Smart Innovation, Systems and Technologies, 195, 21–30. doi:10.1007/978-981-15-7078-0_3.

[25] Goodfellow, I., Bengio, Y., & Courville, A. (2016). A Book on Deep Learning. MIT Press Book, Cambridge, United States.

[26] Zhou, Z., Yuan, H., & Cai, X. (2023). Rock Thin Section Image Identification Based on Convolutional Neural Networks of Adaptive and Second-Order Pooling Methods. Mathematics, 11(5), 1245. doi:10.3390/math11051245.

[27] Monnet, M., Gebran, H., Matic-Flierl, A., Kiwit, F., Schachtner, B., Bentellis, A., & Lorenz, J. M. (2023). Pooling Techniques in Hybrid Quantum-Classical Convolutional Neural Networks. 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, United States. doi:10.1109/qce57702.2023.00074.

[28] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), 53. doi:10.1186/s40537-021-00444-8.

[29] Rafidison, M. A., Ramafiarisona, H. M., Randriamitantsoa, P. A., Rafanantenana, S. H. J., Toky, F. M. R., Rakotondrazaka, L. P., & Rakotomihamina, A. H. (2023). Image Classification Based on Light Convolutional Neural Network Using Pulse Couple Neural Network. Computational Intelligence and Neuroscience, 2023, 1–17. doi:10.1155/2023/7371907.

[30] Abdulkadirov, R., Lyakhov, P., & Nagornov, N. (2023). Survey of Optimization Algorithms in Modern Neural Networks. Mathematics, 11(11), 2466. doi:10.3390/math11112466.

[31] Kinsley, H. & Kukieła, D. (2020). Neural Networks from Scratch in Python. Available online: https://nnfs.io/ (accessed on March 2024).

[32] Mao, A., Mohri, M., & Zhong, Y. (2023). Cross-entropy loss functions: Theoretical analysis and applications. International Conference on Machine Learning, 23-29 July, 2023, Honolulu, United States.

[33] Singarimbun, R. N., Nababan, E. B., & Sitompul, O. S. (2019). Adaptive Moment Estimation to Minimize Square Error in Backpropagation Algorithm. 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), Medan, Indonesia. doi:10.1109/icosnikom48755.2019.9111563.

[34] Kaggle. (2021). miniImageNet: re-processed miniImageNet dataset for few-shot learning research. Kaggle, San Francisco, United States. Available online: https://www.kaggle.com/datasets/arjunashok33/miniimagenet (accessed on March 2024).

[35] Deepak, S., & Ameer, P. M. (2019). Brain tumor classification using deep CNN features via transfer learning. Computers in Biology and Medicine, 111, 103345. doi:10.1016/j.compbiomed.2019.103345.