ISSN No:-2456-2165

Convolutional Neural Network Architecture and Data Augmentation for Pneumonia Classification from Chest X-Rays Images

Ricki Chandra Hidayatullah
Department of Informatics,
Widyatama University Bandung, Indonesia

Sriyani Violina
Department of Informatics,
Widyatama University Bandung, Indonesia

Abstract:- In diagnosing pneumonia, a physician needs to perform a series of tests, one of which is by manually examining a patient's chest radiograph. In the case of a large amount of data, errors in the diagnostic process could occur due to human error and this, of course, can endanger the patient's life. Moreover, the conventional method above is also quite time-consuming. In this study, research was conducted to train classifiers using Convolutional Neural Network (CNN) to automatically recognize normal chest radiographs and chest radiographs with pneumonia. Several architectures are used to train the classifier from previous papers references that already proven to have high accuracy, namely VGG16, InceptionV3, VGG19, DenseNet121, Xception, and ResNet50. Besides that, we added data augmentation to this training. As the results, VGG16 architecture has the highest accuracy with training accuracy reaching 0.9824% and validation accuracy 0.9215% therefore, VGG16 could be the best option among the other architectures in automatically recognizing pneumonia from chest radiograph images.

***Keywords*:-** *Convolutional Neural Network, Deep Learning, Image Classification, Pneumonia Detection.*

I. INTRODUCTION

Pneumonia is a deadly disease that can affect anyone. In 2017 pneumonia accounts for 15% of the causes of death of children under the age of five years [1]. In diagnosing pneumonia, the doctor performs a series of tests using a Thorax X-ray and then diagnoses the image by looking at the characteristics that appear on the image. However, when diagnosing many cases, medical inaccuracies are possible to happen due to human error that certainly can endanger the patient's life. Therefore, the need for automatic detection methods using Deep Learning is expected to assist doctors in diagnosing pneumonia more accurately and quickly.

Deep Learning has an excellent feature for detecting certain medical issues, proven with some classification cases that already achieve accuracy of more than 90%. One of the Deep Learning algorithms used for classification is Convolutional Neural Network (CNN) Research conducted focuses on testing the CNN algorithm by using some pre-existing architecture, to find the best architecture for this case.

In this study we use selected CNN architectures which have proven accuracy above 90% (this section will be explained in the next point), this architecture will be tested with a total of 5842 image data divided into training data and test data by classifying two categories, namely normal and pneumonia [2].

II. PREVIOUS WORK

There have been various studies using Deep Learning technology with CNN algorithm which are widely used in various fields such as agriculture [3][4], industry [5][6], and the medical field that we are going to do at the moment. Several studies on the use of convolutional neural networks in the medical field have been carried out, such as research conducted by Kele Xu, Dawei Feng, and Haibo Mi in the use of Deep Convolutional Neural Networks to detect diabetic retinopathy using fundus images as the objects [7], the utilization of Convolutional Neural Network algorithms to detect brain tumor disease carried out by Alpana Jijja and Dinesh Rai [8], a research conducted by Terrance Deveries and Dhanesh Ramachandram that utilized the Convolutional Neural Network algorithm for the classification of skin lesions [9], and the use of CNN to resolve the Diabetic Foot Ulcer Classification case.[10]

In the aforementioned research, various additional techniques were carried out such as pre-processing, data augmentation, and combining with other algorithms. It aimed to get maximum results. In this study, we utilize data augmentation techniques. The input values will be generalized to each architecture test.

III. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network (CNN) is one of the Deep Learning methods which is a development of Multi-Layer Perceptron that is designed to process two-dimensional data. In the process, CNN is divided into two processes, the Feature Learning and Classification. The Feature Learning is the process of converting information from an image into numbers that present an image. This section consists of two layers, a convolutional layer and a max pooling layer (optional), plus an activation function to change values to a certain range, and numbers in the feature map will be converted into vector shapes, and then the data will be classified according to its type. We use several architectures in this test, the difference between each architecture is the number of layers and their placement.

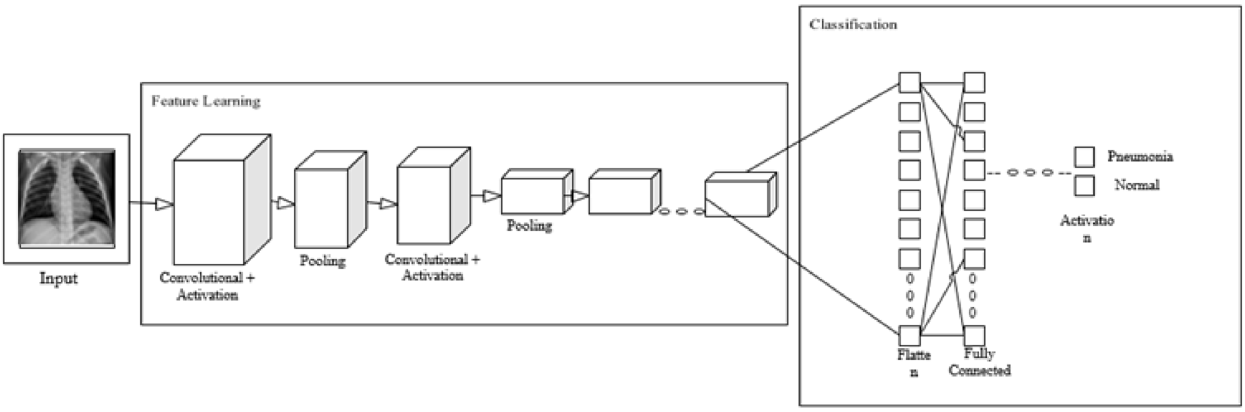


Fig 1:- Convolutional Neural Network

IV. CNN ARCHITECTURE

In this paper, we use several different architectures including VGG16 [11], InceptionV3 [12], VGG19 [13], Xception [14], DenseNet121 [15], and ResNet50 [16]. In the previous tests, these architectures have very high accuracy in research with different cases.

V. DATA AUGMENTATION

Data Augmentation is a data manipulation technique without removing the essence or core of the data. The following Table 1 is a data augmentation technique used in this study.

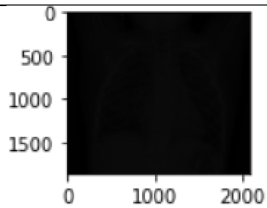
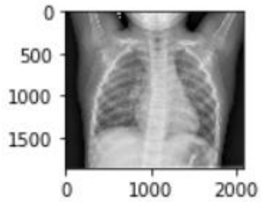
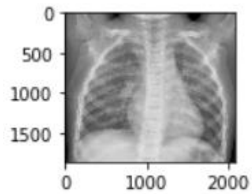
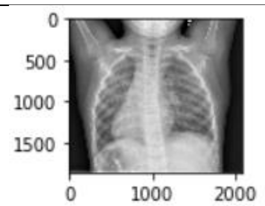
Before	Data Augmentation	Value	After
	Rescale	1./255	
	Shear Range	0.2	
	Zoom Range	0.2	
	Horizontal Flip	True	

Table 1:- Data Augmentation

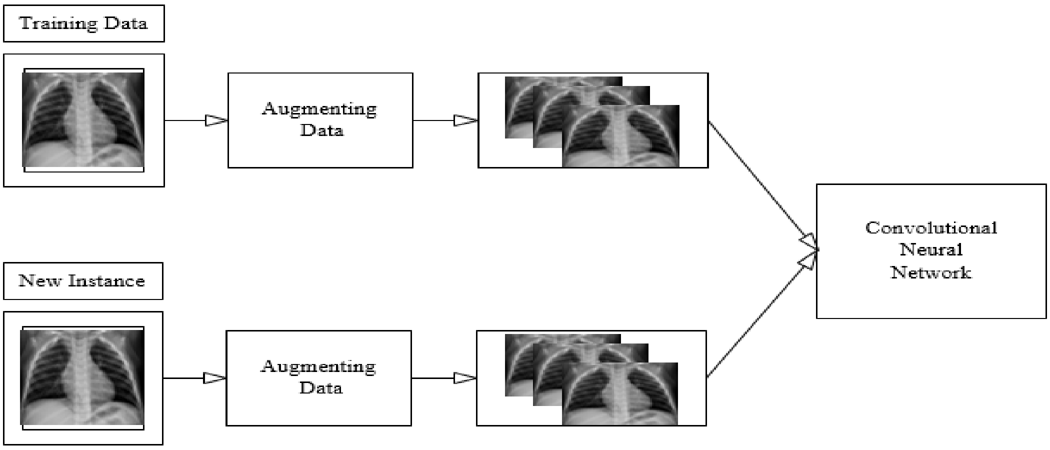


Fig 2:- Data Augmentation in Convolutional Neural Network

VI. HYPER-PARAMETERS OPTIMIZATION

Hyper-parameters are parameters in the learning process whose values are set before the learning process begins. Different training algorithms require different hyper-parameters. Hyper-parameters greatly affect the speed and quality of the learning process. In this research, we utilize Adam as an optimizer with a learning rate of 0.001. We use it since it has the advantage of being able to handle a sparse gradient on noisy problems [17]. The Adam optimizer can be represented in the mathematical form below:

$$v_t = \beta_1 * v_{t-1} - (1 - \beta_1) * g_t \quad (1)$$

$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2 \quad (2)$$

$$\Delta\omega_t = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} * g_t \quad (3)$$

$$\omega_{t+1} = \omega_{t+} \Delta\omega_t \quad (4)$$

ω^j = for each parameter
η = initial learning rate
g_t = gradient at time t along ω^j
v_t = exponential average of gradients ω^j
s_t = exponential average of squares of gradients along ω^j
g_t = gradient at time t along
β_1, β_2 = gradient at time t along

In addition, the loss function that we use is categorical cross-entropy, as shown in Fig. 2 and can be defined Eq. (5) and Eq. (6).

$$CE = -\sum_{i}^{c} t_i \log(f(s)_i) \quad (5)$$

$$f(s)_i = \frac{e^{si}}{\Sigma_j^C e^{s_j}} \quad (6)$$

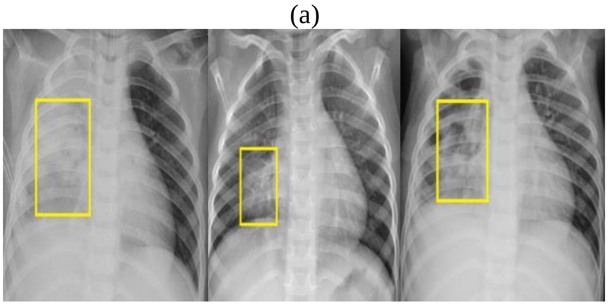
Fig 3:- The Cross-Entropy Loss Function

There are 11 epochs assigned to each architecture.

VII. DATASET

In this paper, our CNN was trained to use X-Ray Images. We modified the dataset into 2 folders namely train and validation which contained subfolders for each image category, normal and pneumonia. There were 1342 training data for normal images and 3876 for pneumonia images. Also for validation data, there were 234 normal images and 390 pneumonia images. Based on the information obtained, the data were obtained from the chest x-rays images (anterior-posterior) selected from a retrospective cohort of pediatric patients aged one to five years old from Guangzhou Women's and Children's Medical Center [2].

(a)



ISSN No:-2456-2165

(b)

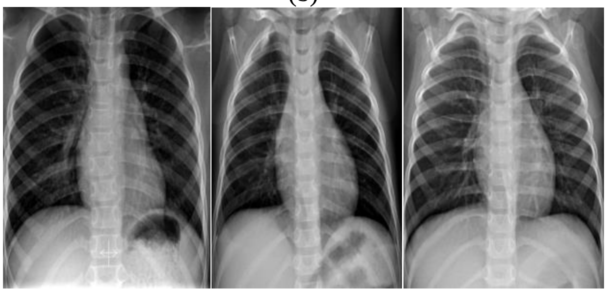


Fig 4:- a. Pneumonia, b. Normal

VIII. COMPARISON RESULT

Based on our experiment, the values of the parameters that affected the data were generalized, but the architectural forms were different. The size of the image input forwarded to the network was 224 x 224. The original image was in various sizes, with an average of more than 1000 x 1000 pixels. It was done to eliminate unnecessary parts. The following is a graph of each test with 11 epochs, where x-axis information is epoch and y-axis is the level of accuracy and loss:

➤ *VGG16 Architecture*

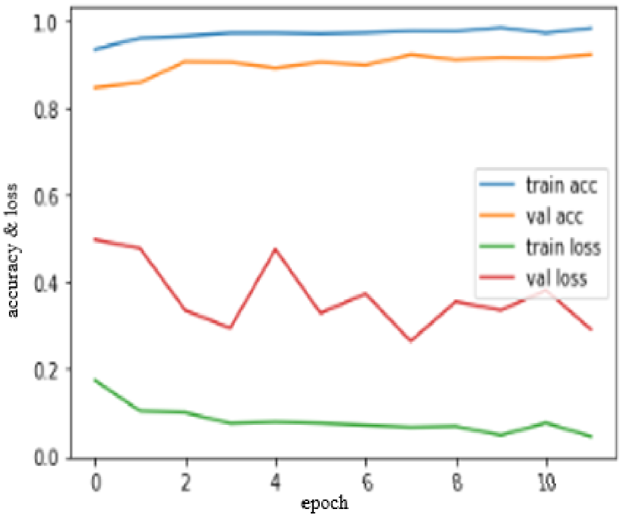


Fig 5:- Accuracy and loss of VGG16 architecture

The accuracy produced in the VGG16 test, training and validation accuracy tends to rise until the final epoch, there is a high increase in validation loss at the 4th epoch, and back down at the next epoch, although there is still an increase in the next epoch but not as big as the 4th epoch and the results this shows a positive pattern until the last epoch.

➤ *InceptionV3 Architecture*

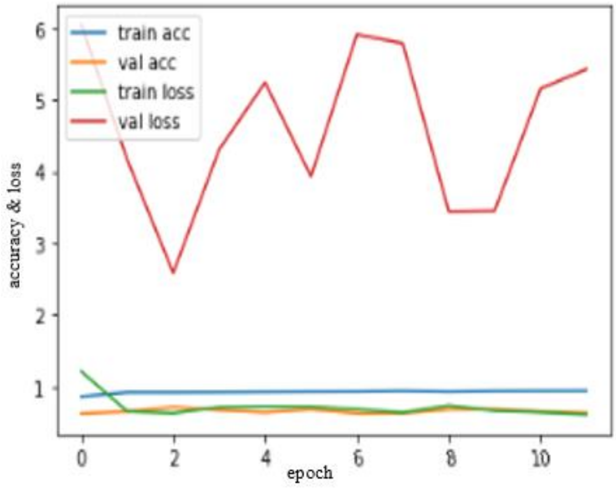


Fig 6:- Accuracy and loss of InceptionV3 architecture

In InceptionV3, training accuracy shows that the graph tends to rise while validation accuracy is not stable. The highest increase in validation accuracy occurs in the 2nd epochreaching 0.7163% up 0.0641% from the previous epoch, validation loss shows a graph that tends to rise, the highest in the 7th epochamounted to 5.9134%, and until the last epoch validation loss showed unstable results.

➤ *VGG19 Architecture*

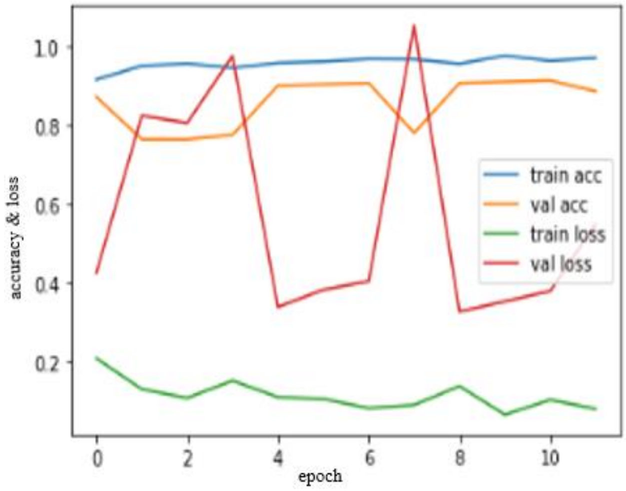


Fig 7:- Accuracy and loss of VGG19 architecture

Tests on VGG19 architecture, training accuracy tends to rise, while the validation accuracy produced up and down but stable every 2 epochs, validation loss shows a very high increase on the 7th epoch reaching 1.0539% and on the next epoch an increase but not greater than the 7th epoch, and until the last epoch, validation loss still shows an increasing graph.

➤ *Xception Architecture*

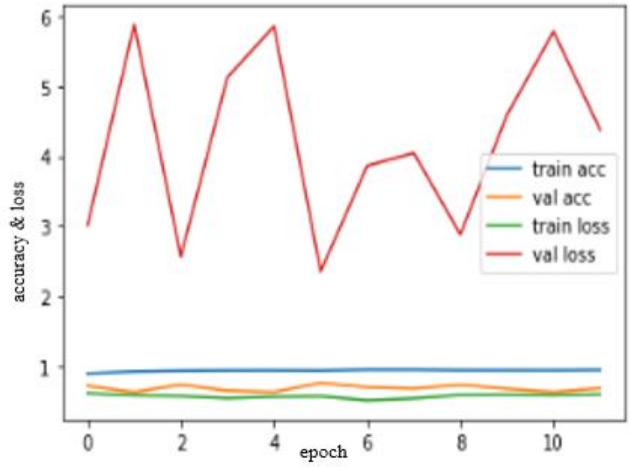


Fig 8:- Accuracy and loss of Xception architecture

The accuracy training generated in this test tends to increase, and validation accuracy shows an unstable graph until the last epoch, the highest validation loss in this test can be seen in 1st, 4th, and 10th epochs, and we estimate the occurrence of overfitting conditions in this test in terms of the loss validation movement until the last epoch.

➤ *DenseNet121 Architecture*

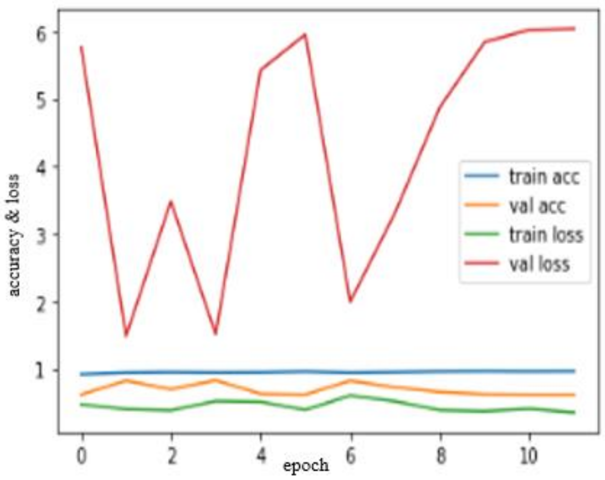


Fig 9:- Accuracy and loss of DenseNet121 architecture

In DenseNet121 testing, training accuracy tends to increase from the initial epoch to the last epoch by reaching an accuracy above 0.92%, while validation accuracy shows a graph that falls on the last 4 epochs, the highest validation accuracy is found in 3rd epoch with an accuracy reaching 0.8349%, and further up to the last epoch only in the range of 0.6% - 0.71%.

➤ *ResNet50 Architecture*

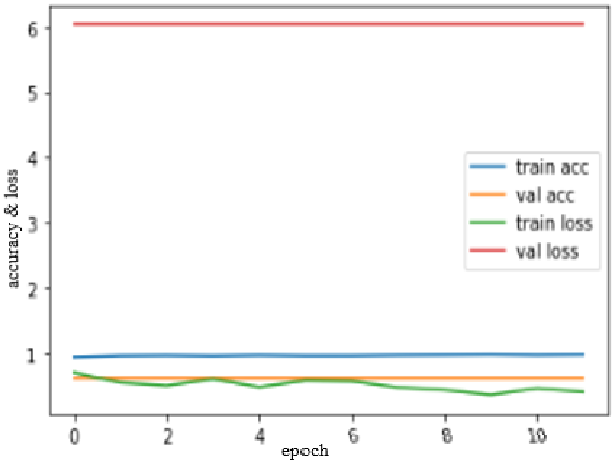


Fig 10:- Accuracy and Loss of ResNet50 Architecture

In the ResNet50 test, the resulting training accuracy is very high and stable from the first to the last epoch, reaching accuracy above 0.93%, while the validation accuracy is very low from the beginning to the end unchanged at 0.6250%, this architecture shows poor conditions in validation and possible conditions that occur where architecture cannot predict new data properly.

The implementation of architectures above produces four parameter values that describe the performance of the model in classifying input images. These values are training accuracy, validation accuracy, training loss, and validation loss. Accuracy is defined as the percentage accuracy of predictions while loss illustrates the inaccuracy of predictions in classification problems. Table 2 is the final test result on each architecture.

ISSN No:-2456-2165

Architecture	Training Accuracy (%)	Validation Accuracy (%)	Training Loss (%)	Validation Loss (%)
VGG16	**0.9824**	**0.9215**	**0.0458**	**0.2911**
InceptionV3	0.9404	0.6364	0.6120	5.4271
VGG19	0.9707	0.8862	0.0811	0.5496
Xception	0.9440	0.6827	0.5984	4.3780
DenseNet121	0.9689	0.6250	0.3587	6.0443
Resnet50	0.9732	0.6250	0.4040	6.0443

Table 2:- Result of testing on every architecture

Architecture is said to be good if the training accuracy and validation accuracy increase with each epoch. If validation accuracy tends to decrease while training accuracy increases, the architecture is estimated to have overfitting. Overfitting occurs because the model is made to focus on specific training data so that it cannot make precise predictions if given a new dataset. This is because the model studies the details and noise in the data training. By looking at the results of each architecture we can see that all architectures have training accuracy above 0.9000% with the highest validation accuracy reaching 0.9215%. From several architectures, we conclude that VGG16 architecture is the best architecture in this test.

IX. ABOUT VGG16

VGG16 is a convolutional neural network architecture initiated by Karen Simonyan and Andrew Zisserman from the Department of Engineering Science, Oxford University [18]. The following is the arrangement of VGG16 architecture:

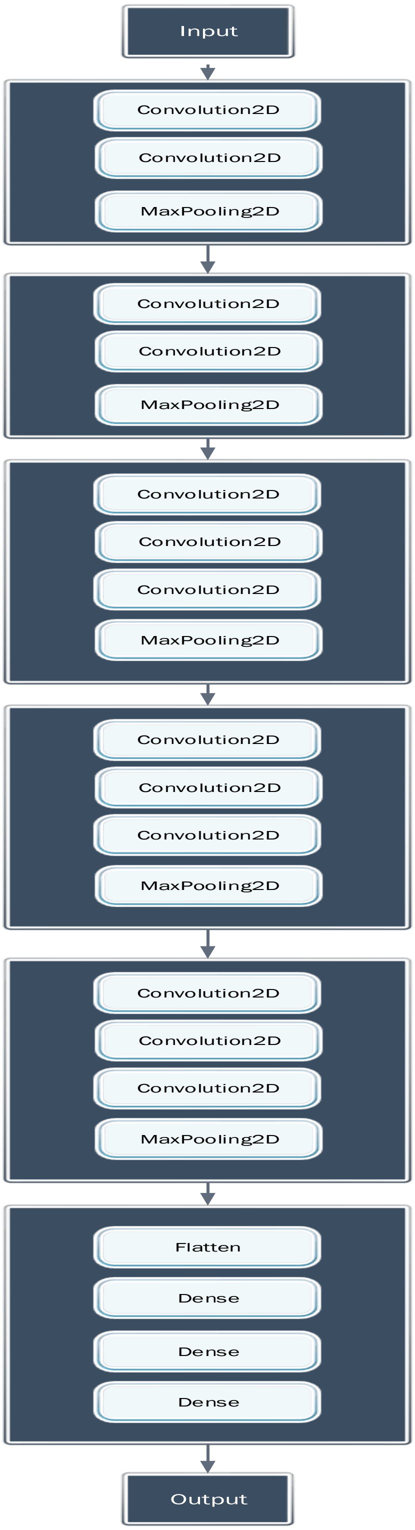


Fig 11:- Architecture of VGG16

This architecture has 13 convolutional layers with ReLU activation and 5 max pooling layers in the feature extraction section and has 3 fully connected classification processes with softmax activation. In the ImageNet Challenge 2014 architecture, this architecture won the top ranking for classification and localization [19].

X. CONCLUSION

In this paper, we introduce CNN which automatically classifies normal X-ray images and pneumonia X-rays. The test is carried out using several selected architectures namely VGG16, InceptionV3, VGG19, Xception, DenseNet121, and ResNet50 to find the best architecture. We added data augmentation techniques and generalized hyper-parameter values in each test. VGG16 architecture is the only architecture that shows a pattern of increasing training and validation accuracy which tends to increase with increasing number of epochs, and the final result of this architectural accuracy is highest of all architectures, with Training and validation accuracy reaching 0.9824% and 0.9215%. For future work we consider including a brightness balance step to improve accuracy in this case.

REFERENCES

[1]. "Pneumonia," *Word Health Organization*, 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/pneumonia. [Accessed: 29-Nov-2019].
[2]. P. Mooney, "Chest X-Ray Images (Pneumonia)," *Kaggle*, 2017. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. [Accessed: 29-Nov-2019].
[3]. A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. February, pp. 70–90, 2018.
[4]. C. McCool, T. Perez, and B. Upcroft, "Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1344–1351, 2017.
[5]. D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Ann. - Manuf. Technol.*, vol. 65, no. 1, pp. 417–420, 2016.
[6]. R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated Time Series Convolutional Neural Architecture: An Intelligent Fault Diagnosis Approach for Electric Machine," *IEEE Trans. Ind. Informatics*, vol. 13, no. 3, pp. 1310–1320, 2017.
[7]. K. Xu, D. Feng, and H. Mi, "Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image," *Molecules*, vol. 22, no. 12, 2017.
[8]. A. Jijja and D. Rai, "Efficient MRI segmentation and detection of brain tumor using convolutional neural network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 4, pp. 536–541, 2019.
[9]. T. DeVries and D. Ramachandram, "Skin Lesion Classification Using Deep Multi-scale Convolutional Neural Networks," 2017.
[10]. M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, "DFUNet: Convolutional Neural Networks for Diabetic Foot Ulcer Classification," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. PP, pp. 1–12, 2018.
[11]. A. Romero Lopez, X. Giro-I-Nieto, J. Burdick, and O. Marques, "Skin lesion classification from dermoscopic images using deep learning techniques," *Proc. 13th IASTED Int. Conf. Biomed. Eng. BioMed 2017*, pp. 49–54, 2017.
[12]. X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," *2017 2nd Int. Conf. Image, Vis. Comput. ICIVC 2017*, pp. 783–787, 2017.
[13]. M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," *J. Comput. Sci.*, vol. 30, pp. 174–182, 2019.
[14]. Y. Wang, C. Wang, and H. Zhang, "Ship classification in high-resolution SAR images using deep learning of small datasets," *Sensors (Switzerland)*, vol. 18, no. 9, 2018.
[15]. Y. M. Park, J. H. Chae, and J. J. Lee, "DenseNet 을 활용한 식물 잎 분류 방안 연구," vol. 21, no. 5, pp. 571–582, 2018.
[16]. H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, "Label Refinery: Improving ImageNet Classification through Label Progression," no. 1, 2018.
[17]. D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
[18]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
[19]. "Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)," ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014). [Online]. Available: http://www.image-net.org/challenges/LSVRC/2014/results. [Accessed: 04-Jan-2020].