Layers Modification of Convolutional Neural Network for Pneumonia Detection

Wahyudi Setiawan* **and Fitri Damayanti**

Department of Informatics, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan, Jawa Timur, Indonesia

*wsetiawan@trunojoyo.ac.id

Abstract. Pneumonia is a bacterial, virus and fungi infection that attacks respiratory function. The disease causes air sacs in the lungs inflamed and swollen. It conditions produce lungs filled with fluid and mucus. Generally, the detection of pneumonia was done by chest x-ray images. This study discusses the detection of pneumonia through x-ray images using Convolutional Neural Network. The CNN model was Visual Group Geometry VGG16 and VGG19. As a comparison, we used the modified CNN 35 layer. The experiment using public data from Chest X-Ray Images - Kaggle. Data consist of 2 classes: normal and pneumonia with a total of 624 images. The results using VGG16 show a performance measure of sensitivity 92.75%, specificity 96.8%, and accuracy 94.1%. The result of VGG19 has sensitivity 96.6%, specificity 94.3%, and accuracy 95.7%. For CNN 35 layer has sensitivity 95.1%, specificity 98.5%, and accuracy 96.3%.

1. Introduction

Pneumonia is a disease that attacks the respiratory organs in humans. The disease infects the lungs, more precisely the alveoli which are small sacs filled with air when breathing. Pneumonia is caused by viruses, bacteria, and fungi. In patients with pneumonia, alveoli are filled with pus and fluid which makes breathing painful and limits oxygen intake [1], [2].

Pneumonia is a disease that causes death by 15% in children under five years, 808,694 children died in 2017 [3]. In Indonesia, the data on children under five years with pneumonia was 2% [4]. This is certainly a special concern of the authorities, especially in the health sector to take preventive and curative measures against patients with pneumonia.

In general, the examination procedure in pneumonia patients consists of pulse oximetry, which is a measurement of oxygen levels in the blood, chest x-ray photo, blood test to confirm infection and identification of the type of organism, urine test, and sputum sample checking. A long-time examination and big financial become a problem. An alternative to early detection of pneumonia is by taking chest x-ray images, then through pattern recognition, it can be classified as having pneumonia or not [5], [6].

This article presents the detection of pneumonia through chest x-ray images using the Convolutional Neural Network (CNN). This method has effectiveness in recognition, It can extract features and classifications automatically. There are previous studies that have detected pneumonia including Stephen et al using CNN with 4 Convolutional layers, 4 max-pooling layers, 1 flatten, 1 drop out, 7 dense classification layer. Total of 19 layers with 300 data. Augmented in 7 ways: rescale,

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

rotation, width-shift, heigh-shift, shear-range, zoom-range, and horizontal-flip. The classification results have an average accuracy of 93,012% [7].

Donthi et al. classify 2 classes of normal and abnormal chest x-ray images. The CNN model used was not detail explained, there was only information: 32 batch size, 668 steps per epoch and requires 3 epochs. The accuracy was 68.8% to 78.9% [8].

Furthermore, Urey et al. used CNN and residual network to classify 2 classes of chest x-ray images. CNN was modified with 6 scenarios. First, use the CheXNet model with 121 layers [9]. Second, CNN without input modification. Third, CNN with different color schemes. Fourth, CNN with an increase in the value of image contrast. Fifth, modification data with a lightened image. Sixth, the image from scenario 5 is applied to the Residual Network. The results show an accuracy of 63.74% to 78.73% [10].

Saraive et al., 2018 used data of 5,863 from 2 classes. The CNN was 27 layers consisting of an input layer, 7 convolutional layers, 9 ReLU, 2 Batch Normalization, Flatten, 2 Dense, Dropout 0.6 and 0.4, 2 softmax and output layer. It has an accuracy of 95.3% [11].

In the research above, performance measures still have opportunities for improvement. CNN architecture can use existing and modified models. This study has a contribution. First, a new CNN architecture with simpler layers than the existing CNN architecture. Second, the classification accuracy was better than the existing CNN. Third, there is a visualization of features at certain layers of CNN.

2. Material and Method

The research methodology starts with image input on the CNN model, then after it is processed using the gradient descent of the momentum optimization algorithm it produces a classification output. The classification results are validated using 10 k-cross validation. In the final stage, the performance measures sensitivity, specificity, and accuracy are calculated. The test scenario uses VGG16, VGG19, and modified CNN architectures. The research method shows at Figure 1.

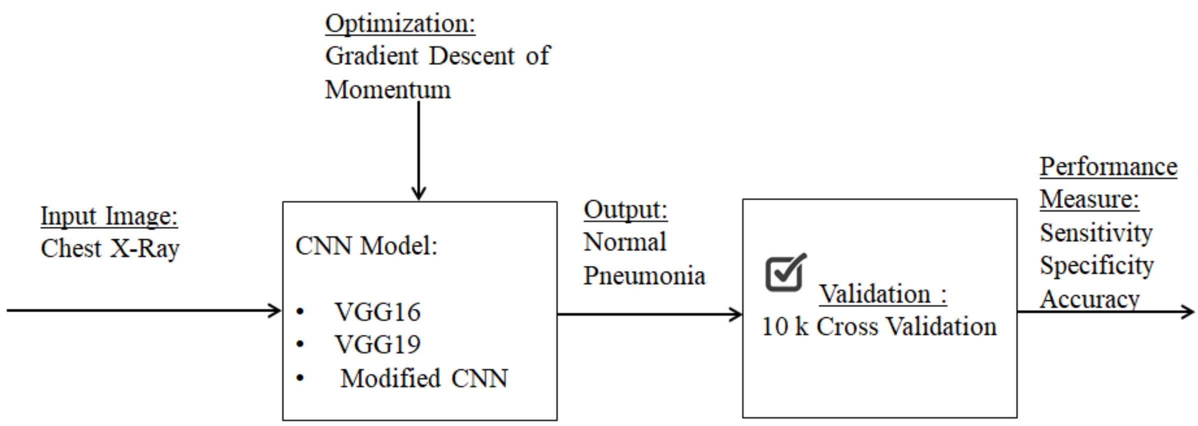


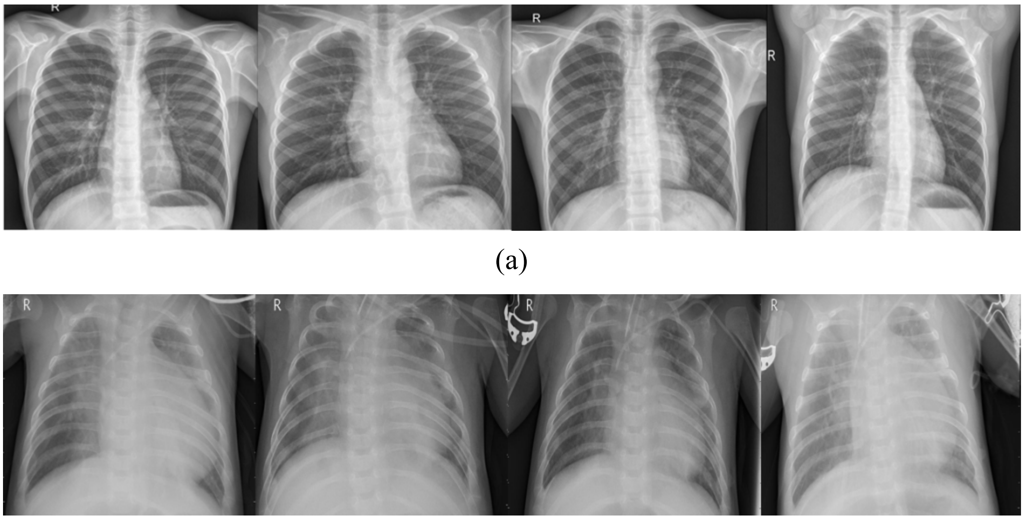
Figure 1. Research Method of Deep CNN for Pneumonia Detection

2.1 Data

Experimental using public data from Chest X-Ray Images - Kaggle. Data consist of 2 classes: normal and pneumonia. The total data is 5860 images. Testing data 624, training data 5218, validation data 18 images. In this study, the experiment was carried out on data testing only. The complete data are shown in Table 1. An example of data representation is shown in Figure 2 [12].

Table 1. Total Data of Chest X-Ray Images - Kaggle

Data	Normal	Pneumonia
Testing	234	390
Training	1342	3876
Validation	9	9
Total	1585	4275



(a)

(b)

Figure 2. Chest X-Ray (a) Normal (b) Pneumonia from Kaggle

2.2 Convolutional Neural Network

CNN is a deep learning algorithm, which means a reliable algorithm for recognition of large amounts of data and processed with high-speed machines. CNN layers are composed of the input layer, hidden layer, and output layer. If on a normal NN hidden layer consists of 1 or 2 layers, CNN has a hidden layer up to hundreds. It has 2 main parts: feature extraction and classification. The feature extraction section is convolutional layers, max pooling, batch normalization, rectified linear units and dropouts. while the classification part is fully connected layer and softmax. The training process is carried out using the concept of backpropagation, which is to improve the weight and bias iteratively until the desired conditions are reached, i.e the maximum epoch has been reached or overfitting has occurred. Overfitting is a condition where the results of the accuracy of the testing phase are slowed down compared to the accuracy of the training. Overfitting stopped the iterative process to prevent lower accuracy values [13].

The CNN model used was an architecture that joins in the ImageNet project competition which classifies millions of data in tens of thousands of classes [14] - [16]. This study uses the CNN model from Visual Geometry Group-Oxford University. It is a research group that has a focus in Artificial Intelligence. Two VGG models used are VGG16 and VGG19.

2.2.1 VGG16 dan VGG19

Visual Geometry Group (VGG) created the VGG16 network architecture with 41 layers and VGG19 with 47 layers. VGG simplifies the process by creating 3 × 3 filters for each layer. The use of uniform and smaller filter sizes on VGG can produce more complex features and lower computing when compared to AlexNet. The difference between VGG16 and VGG19 is shown in Table 2 [17].

Table 2. Comparison of VGG16 and VGG19 Layers

Layer	VGG16	VGG19
Size of Layer	41	47
Image Input Size	224x224 pixel	224x224 pixel
Convolutional Layer	13	16
Filter Size	64 & 128	64,128,256, & 512
ReLU	5	18
Max Pooling	5	5
FCL	3	3
Drop Out	0.5	0.5
Softmax	1	1

2.2.2 CNN architecture model with a simple layer

A novelty was done by modifying the VGG architecture. The type of layer that can be added to architecture is batch normalization (BN) [18]. The target output is to make architecture with a simpler number of layers and increase the accuracy of the classification results when compared to the VGG architecture. The new CNN architecture, shown in Figure 3.

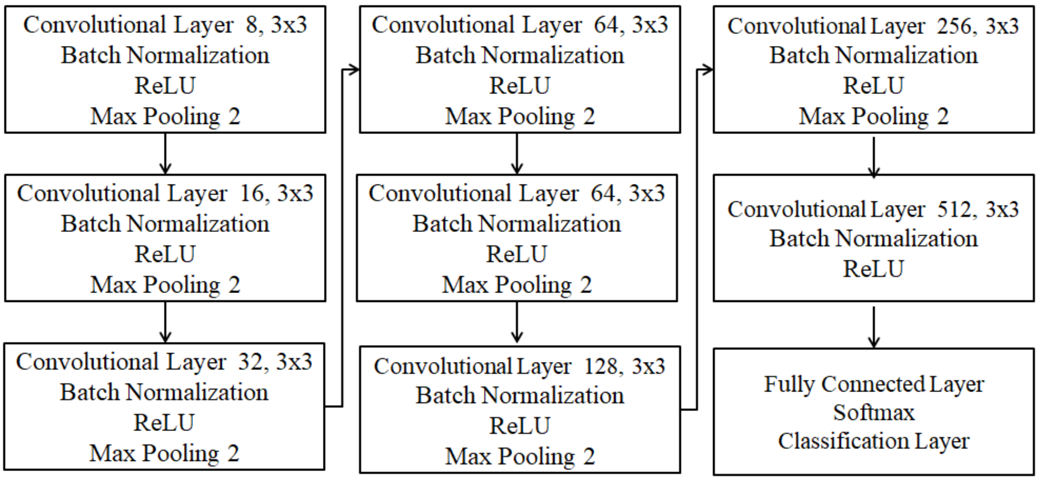


Figure 3. CNN architecture 35 layers

The architecture has 35 layers with an input layer, 8 convolutional layers that have a 3x3 filter with dimensions 8, 16, 32, 64, 64, 128, 256, and 512. Furthermore, there are 8 batch normalization layers, 8 Rectified Linear Units, 7 max-pooling sizes 2 with stride 2, fully connected layer (FCL), Softmax and classification layer or output layer.

2.3 Optimization with Gradient Descent

Gradient Descent (GD) works by providing initial parameters. It moves iteratively with parameters on a network according to the direction of the gradient. The goal is to direct the parameters to the optimal point. In GD there is a variable learning rate. Learning rate α is a variable that contributes to the speed of learning. The greater the value of learning rate, the faster the learning, but the risk of divergence or oscillation will be obtained. Conversely, if the learning rate is too small, convergence takes a long time and can potentially be trapped in local optima, which is smaller than the nearest point but can be greater than the point that is far away located [19]. Figure 4 is the movement of the gradient descent function with big and small learning rates.

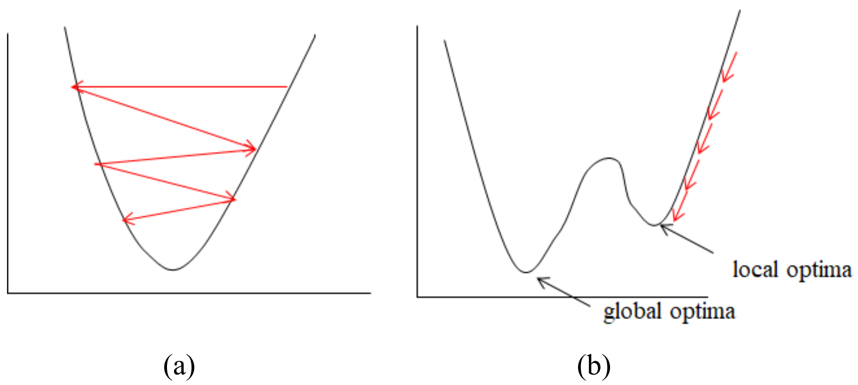


Figure 4. Movement of the gradient descent function A. learning rate is large, B. learning rate is small [19].

An efficient optimization path is the main problem in the gradient descent algorithm. The learning rate is set to have a big value at the beginning of an iteration, then it is set to gradually decrease at the middle or end of the iteration. This can be solved with momentum.

2.3.1 Gradient Descent with Momentum

Momentum is a method for GD acceleration by utilizing gradient information in the previous steps. The accumulation of the gradient is useful for controlling the oscillation effect, and it is hoped that the optimization path can be more stable. The following is the GDM equation [19]:

$$m_0 = 0 \quad (1)$$

$$g_t := \nabla_{\theta_{t-1}} L\,(\theta_{t-1}) \quad (2)$$

$$m_t := g_t + \beta m_{t-1} \quad (3)$$

$$\theta_t := \theta_{t-1} - \alpha\, m_t \quad (4)$$

with m_0 = initial momentum, m_{t-1} = momentum , m_t = the next momentum

2.4 Performance Measures

Performance measures using standards of accuracy, sensitivity, and specificity. These measurements require True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. TP is the amount of positive data or pneumonia that is classified correctly by the system. TN is the amount of negative or normal data that is classified correctly by the system. FN is the amount of negative data but is classified incorrectly by the system. FP is the amount of positive data but is classified incorrectly by the system. Measurement of TP, FP, TN, and FN uses a confusion matrix as shown in Figure 5 [20].

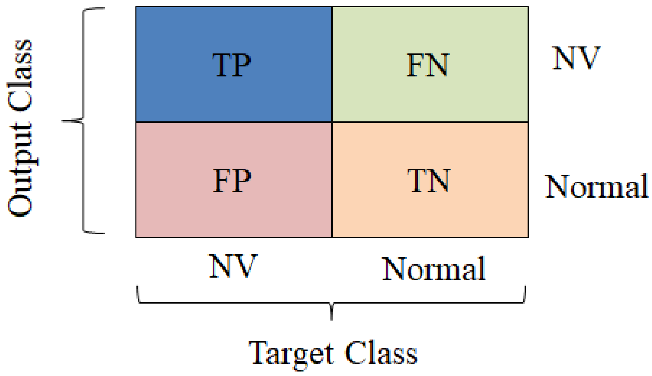


Figure 5. Confusion Matrix. *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), and *False Negative* (FN). NV = Neovascularization.

2.4.1 *Validation*

Validation of testing results using k-fold cross-validation with k = 10. Cross-validation divides data into 10 equal parts. Nine parts are used for training and one part is for validation. Then select one other part for validation and the remaining 9 parts for training and so on. The cross-validation scheme is shown in Figure 6 [21].

Figure 6. 10-*fold cross-validation*

2.5 *System Requirements*

The test was carried out on a computer with Core i7 7700 specifications, GPU 1060 6 Gb D5 amp, Solid State Drive, Double Data Rate 4 16 Gb, and MSI Z270 Motherboard.

3. Result and Discussion

At the experimental steps, the output results at each of the CNN layers ie features can be displayed visually. Figure 7 shows an example of the visualization of features in the initial, middle, and final layers of the chest x-ray dataset. It can be seen in the picture that in the last layer the feature pattern is more clearly visible when compared to the initial layer.

(a) (b) (c)

Figure 7. Feature Visualization at (a) the first (b) middle (c) the last of Convolutional Layer of VGG16

Furthermore, the test was carried out with 15 scenarios. It varying the type of architecture, maximum epoch, learning rate, and minibatch-size used. The architecture used is VGG16 and VGG19, minibatch 4 and 8, max-epoch 50,100 & 200, learning rate 1e-3, 1e-4 & 1e-5. Scenarios with variations in variable values are made to analyze the effect of changes in variable values. The test scenarios are in Table 3, while the results are in Table 4.

Table 3. The Testing Scenarios

No	Architect.	Minibatch-size	Max-epoch	Learning rate	No	Architect.	Minibatch-size	Max-epoch	Learning rate
1	VGG16	8	50	1e-4	9	VGG19	8	50	1e-4
2		4	50	1e-4	10		4	50	1e-4
3		8	100	1e-4	11		8	100	1e-4
4		4	100	1e-4	12		4	100	1e-4
5		8	200	1e-4	13		8	200	1e-4
6		4	200	1e-4	14		4	200	1e-4
7		8	50	1e-3	15		8	50	1e-5
8		8	50	1e-5					

Table 4. Result of the experiment

Scenario	Sensitivity (%)	Specificity (%)	Accuracy (%)	Time	Stop training at an epoch
1	91.94	95.24	93	3 min 9 sec	10
2	96.4	85.7	92	4 min 31 sec	10
3	99.1	87.3	94.1	4 min 21 sec	13
4	92.7	96.8	94.1	2 min 48 sec	9
5	94.2	95.9	94.7	3 min 31 sec	8
6	96.3	83.5	90.9	3 min 4 sec	7
7	62.6	-	62.6	1 min 49 sec	5
8	96.3	84.6	91.4	4 min 26 sec	13
9	96.6	94.3	95.7	3 min 37 sec	10
10	96.6	93	95.2	8 min 22 sec	16
11	97.4	93.1	95.7	5 min 1 sec	13

12	96.6	94.3	**95.7**	5 min 24 sec	14
13	94.8	90.1	93	2 min 49 sec	6
14	98.1	81.9	90.9	3 min 11 sec	6
15	94.2	94	94.1	7 min 29 sec	20

The results show the highest accuracy using the VGG19. The accuracy obtained up to 95.7% with a computation time of 5 min 24 sec. The variables used are minibatch-size 4, max-epoch 100, and learning rate 1e-4. From 15 scenarios, training only conducted up to epoch 5 to 16. This shows the ineffectiveness of the epoch value that set too high. In the learning rate variable with a value of 1e-3, it was unable to calculate specificity due to the limited specifications of the system used. By using the same variable in scenario 12, try out the modified architecture. A comparison of results is shown in Table 5.

Table 5. Performance Measures Comparison with the VGG

Architecture	Layer	Sensitivity	Specificity	Accuracy
VGG16	41	92.7	96.8	94.1
VGG19	47	96.6	94.3	95.7
Proposed Method	35	95.1	98.5	96.3

Test results show that the CNN 35 layer has better accuracy compared to VGG16 and VGG19. Accuracy of 96.3%, sensitivity of 95.1% and specificity of 98.5%. Furthermore, the validation using 10 k-cross validation on the results of the proposed method. The results of the validation were shown in Table 6.

Table 6. Validation with 10 k-cross validation

Scenario	Sensitivity (%)	Specificity (%)	Accuracy (%)	Time(sec.)	Stop training at an epoch
1	95.1	98.5	96.3	64	12
2	91.5	87	89.94	46	9
3	90.4	93.5	91.4	31	6
4	92	96.8	93.6	54	10
5	95.8	95.6	95.7	38	7
6	93.3	91.2	92.5	53	10
7	95.6	87.8	92.5	47	9
8	95.7	90.3	93.6	41	8
9	94.9	92.8	94.1	48	9
10	96.5	91.7	94.7	67	12
Average	94.08	92.52	93.43	48.9	9.2

The results of 10-k cross-validation validation showed an average sensitivity, specificity, and accuracy of 94.08%, 92.52%, and 93.43% respectively. For the average computational time is 48.9 sec with an average epoch of 9.2.

4. Conclusion

The experiment has been carried out to classify Normal Chest X-Ray and Pneumonia images. The results show the best accuracy using the CNN 35 layer architecture with an accuracy of 96.3%. After validation with 10 k cross-validation, the average accuracy was 93.43%, the average sensitivity was 94.08%, and the specificity was 92.52%. The CNN 35 layer consists of an input layer, 8 convolutional layers of 3x3 size with dimensions ranging from 8,16, 32, 64,128, 256 and 512. Furthermore, there are 8 batch normalization layers, 8, Rectified Linear Units (ReLU), 7 max-pooling with sizes 2, Fully Connected Layer, Softmax, and Classification Layer. To test the reliability of the architecture that has been made, the next study can be classified more than 2 classes of chest x-ray images.

Acknowledgement Thanks to Paul Mooney for pneumonia chest x-ray images and also Visual Geometry Group for the CNN Architectures

References

[1] Kementrian Kesehatan RI, "Buletin Jendela Epidemiologi Pneumonia Balita," 2010.
[2] National Institute for Health and Care, "Pneumonia in adults : diagnosis and management," 2014.
[3] World Health Organization, "Pneumonia," 2019.
[4] A. Anwar, I. Dharmayanti, P. Teknologi, I. Kesehatan, M. Badan, and P. Kesehatan, "Pneumonia pada Anak Balita di Indonesia Pneumonia among Children Under Five Years of Age in Indonesia," no. 29, pp. 359–365, 2013.
[5] R. R. Watkins and T. L. Lemonovich, "Diagnosis and Management of Community-Acquired Pneumonia in Adults," pp. 1299–1306, 2011.
[6] W. Osler, "Pneumonia," Kaggle, 2006, pp. 155–163.
[7] O. Stephen, M. Sain, U. J. Maduh, and D. Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare," vol. 2019, 2019.
[8] A. Donthi, A. Tammanagari, and A. Huang, "Pneumonia Detection using Convolutional Neural Networks," vol. 21, no. 1, p. 27599, 2018.
[9] P. Rajpurkar *et al.*, "CheXNet : Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," pp. 3–9, 2017.
[10] D. Y. Urey, C. J. Saul, C. D. Taktakoglu, and C. V Apr, "Early Diagnosis of Pneumonia with Deep Learning," 2019.
[11] S. Saraiva, A.A., Ferreira, N.M.F., de Sousa, L.L., Costa, N.J.C., Sousa, J.V.M., Santos, D.B.S., Valente., V., Soares, "Classification of Images of Childhood Pneumonia using Convolutional Neural Networks," 2018.
[12] P. Mooney, "Data of Chest X-Ray kaggle," 2019. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.
[13] W. Setiawan, M. . Utoyo, and R. Rulaningtyas, "Classification of neovascularization using convolutional neural network model," *TELKOMNIKA*, vol. 17, no. 1, pp. 463–473, 2019.
[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification," 2015.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012.
[16] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.
[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015, pp. 1–14.
[18] F. Schilling, *The Effect of Batch Normalization on Deep Convolutional Neural Networks*. Stockholm, Sweden: KTH Royal Institue of Technology, School of Computer Science and Communication, 2016.
[19] S. Ruder, "An overview of gradient descent optimization," 2017.
[20] J. M. Banda, R. A. Angryk, and P. C. Martens, "Steps Toward Large-scale Solar Image Data Analysis to Differentiate Solar Phenomena," *Sol. Phys.*, no. May 2013, pp. 1–28, 2013.
[21] Hastie and Tibshirani, "K -Fold Cross-Validation," 2009. [Online]. Available: http://statweb.stanford.edu/~tibs/sta306bfiles/cvwrong.pdf.