JOURNAL MEDICAL INFORMATICS TECHNOLOGY
medinftech.org ISSN 2988-7003 (online)

Optimizing Cataract Detection in Fundus Images through EfficientNet-Based Classification

Andi Ibrahim[1], Edi Sabara[2], Winarlin Dirsam[3] and Faruq Aziz[4]*

[1,2,3] *Computer Science Master's Study Program, Nusa Mandiri University, Jakarta, Indonesia*
[4] *Department of Information System, Nusa Mandiri University, Jakarta, Indonesia*

ARTICLE HISTORY

Received: 05 January 24
Final Revision: 31 January 24
Accepted: 01 February 24
Online Publication: 30 March 24

KEYWORDS

Cataract, Fundus, EfficientNet, Convolutional Neural Network

CORRESPONDING AUTHOR

14220031@nusamandiri.ac.id

DOI

10.37034/medinftech.v2i1.25

A B S T R A C T

Turbidity of the lens of the eyeball that causes blindness or loss of vision is known as a cataract. By diagnosing the causes and symptoms of cataracts, early detection helps patients in prevention and treatment. The purpose of the research was to classify the image of the fundus into two classes: normal and cataract. The study also looked at how the optimizers for stochastic gradient descent, adaptive moment estimation, root mean square propagation, adaptive gradient algorithm, adaptive delta, and Nesterov-accelerated adaptive moment estimation stacked up against each other. We used the EfficientNet architecture in CNN and preprocessed the normal fundus and cataract fundus images by dividing each into training data (N = 80) and validation data (N = 20) from the Kaggle repository. We added test data from the normal fondus image (N =20) to see the accuracy of the results. We get 100% accuracy of training data, 87% and 77% validation data, and 100% and 95% test data.

1. Introduction

A cataract is a medical disorder characterized by the opacification of the lens in the eye, resulting in a gradual deterioration in visual acuity. Aging commonly causes cataracts as oxidative stress can make the eye's lens opaque, resulting in blurred vision [1]. For millennia, A group of well-known illnesses that are characterized by lens opacities has been recognized [2]. Cataract is the most prevalent cause of blindness on a global scale, affecting 94 million individuals who are blind or visually compromised. According to data from 2020, cataracts caused around 15.2 million instances of blindness in those aged 50 and above, in addition to 78.8 million instances of mild to severe vision impairment [3]. Approximately 1.6 million individuals in Indonesia are blind, while 8 million have moderate to severe visual impairment [4].

Over the past few years, deep learning (DL) has become a powerful tool across many imaging domains, including classification, prediction, detection, segmentation, diagnosis, interpretation, and reconstruction, among others. Physicians will be empowered, and clinical decision-making will be accelerated by the diminishing capacity of DL to diagnose diseases [5]. Convolutional neural network (CNN) are algorithms for deep learning that are widely implemented [6]. Cetiner [7] suggested that the MobileNet V3 model attains the best level of accuracy in accuracy for both validation and training. At the 20th epoch, the training accuracy rate achieved a level of 98.31%, but the validation accuracy rate stood at 96.62%. Simanjuntak et al. [8] proposed CNN model achieves the highest level of accuracy (0.93) when the Adam optimizer is applied to a learning rate of 0.001. The model achieves an accuracy of 0.92 for test data and 0.93 for validation data. Firdaus et al. [9] reported The CNN approach with the RMSprop optimizer gave excellent results in the cataract test. The training accuracy was 99.74%, the validation accuracy was 91.18%, and the testing accuracy was 93.33%. Junayed et al. [10] suggested using a breakthrough deep neural network, CataractNet, to automatically identify cataracts in fundus images. Adjustments are made to the activation and loss functions in order to train the network with a reduced number of layers, training parameters, and kernels. The Adam optimizer is applied to optimize the proposed network. The experimental findings provide evidence that the proposed method exhibits an above-average accuracy of 99.13% when compared to the most recent advancements in cataract detection. Cahya et al. [11] accomplished The CNN AlexNet model achieves a remarkable accuracy of

98.37%, making it the most precise. The utilization of the Adam optimizer and feature extraction in three layers - the convolutional layer, the pooling layer, and the fully connected layer - achieves this. Syarifah et al. [12] applied CNN The Lookahead optimizer on SGD and Adam integrated into the AlexNet architecture improved optimizer SGD by 2.5% and increased Adam's precision by 20%. By implementing Lookahead optimization, Adam achieved a training and validation accuracy of 97.5%.

2. Research Method

In this study, we used desktop computers to analyze the data. The system employs the Windows 11 Pro 64-bit operating system, powered by a 12th generation Intel Core i7-12700H processor with 20 CPUs, running at around 2.3 GHz, and equipped with 32768MB of RAM. The Jupyter Notebook version used is 6.5.2, the Python version is 3.10.9, the Scikit-Learn version is 1.1.2, and the TensorFlow version is 2.8.0. This study utilized fundus images, which were categorized into two groups: normal fundus (label = 0) and cataract fundus (label = 1). The images were divided into training data (N = 80), validation data (N = 20), and test data (N = 20), as shown in Table 1.

Table 1. Segmentation of fundus images

Class	Label	Training Data	Validation Data
Normal	0	80	20
Cataract	1	80	20
Total Data		160	40

The approach applied to CNN with the EfficientNet architecture can be seen in Figure 1 through the addition of additional components or layers into the CNN design. The study examined various neural network design approaches for classification, including Stochastic Gradient Descent, Adaptive Moment Estimation, Root Mean Square Propagation, the adaptive gradient algorithm, adaptive delta, and Nesterov-accelerated adaptive moment estimation.

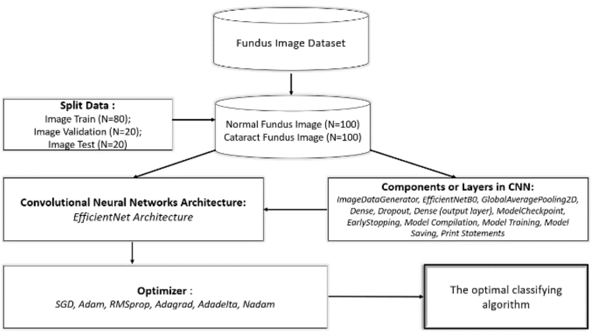

Figure 1. Identify the optimal process flow diagram for the classification method.

2.1. Dataset

The fundus image dataset was sourced from the Kaggle dataset [13]. Images are processed using the RGB format, which stands for red, green, and blue. Images are read using OpenCV's cv2.imread function and resized. This method is applied to RGB images by OpenCV's cv2.imread function. The parameter initialization involves assigning a value of 32 to the 'batch_size' variable, indicating that the model will process 32 images during each training iteration. To provide the dimensions of the input images for the model, we assign the value (224, 224) to the image_size variable.

The data collection procedure commences by establishing the number of samples required for training, validation, and testing in each category, namely 20 for validation, 20 for testing, and 80 for training. We provide and prepare the image labels, which identify normal (0) and cataract (1), for utilization in training classification models aimed at detecting cataracts in medical images. Figure 2 illustrates the disparity between regular fundus and cataract fundus photos in RGB format, whereas Figure 3 demonstrates the contrast between normal fundus and cataract fundus images in grayscale format.

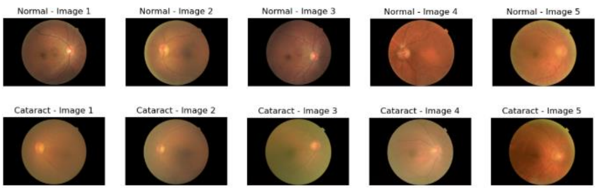

Figure 2. Normal and cataract fundus images in RGB format.

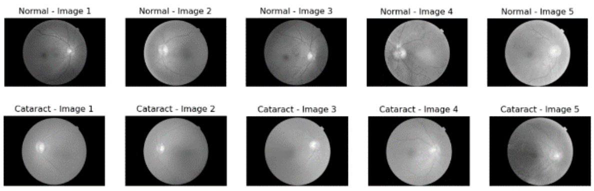

Figure 3. Normal and cataract fundus images in grayscale form.

2.2. Classification using EfficientNet architecture

CNN is a special artificial neural network for image processing. CNN emulates the manner in which nerve cells transmit information to linked neurons. CNN represents each neuron in two dimensions, while Multilayer Perceptron represents each neuron as one-dimensional [14]. Deep CNN architectures sometimes include an excessive number of parameters. This is a result of the network's increasing depth, breadth, and number of convolutional layers. As a result, The effectiveness of the network decreases with the addition of more convolutional layers and increased network breadth and depth. increasing the computational cost of the system. There exists a compromise between the effectiveness of a network and the level of precision it can achieve. Although deep networks may demonstrate strong generalizations based on test results, their effectiveness keeps improving regarding network parameters, dimensions of the model, floating-point operations per second (flops), and inference speed.

In 2019, the Google AI research team published a collection of EfficientNet models.; these models include EfficientNetB0 and EfficientNetB7 [15]. This series has demonstrated superior contrast, performance in segmentation, more transfer learning-based tasks, and image classification using ImageNet to a number of cutting-edge deep CNN-based architectures, including DenseNet, Inception-V3, ResNet50, and Inception-ResNetV2. To scale up the CNN architecture, EfficientNet employed uniform compound scaling and fixed scaling coefficients [16].

Table 2. Sequential Neural Network Architecture

Layer (type)	Output Shape	Param #
efficientnetb0 (Functional)	(None, 7, 7, 1280	4049571
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dense (Dense)	(None, 512)	655872
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513
Total params: 4,705,956		
Trainable params: 4,663,933		
Non-trainable params: 42,023		

The model presented in Table 2 is a sequential neural network design that integrates EfficientNetB0 as a functional layer. Feature extraction is carried out by the EfficientNetB0 layer, which generates a tensor with the following dimensions: None, 7, 7, 1280, with a total of 4,049,571 parameters. Subsequently, a GlobalAveragePooling2D layer is utilized to calculate the mean value of each characteristic in the 7x7x1280 matrix, yielding a feature vector with a length of 1280. Following that, a dense layer consisting of 512 neurons processes this feature vector, resulting in the introduction of 655,872 parameters. Subsequently, a dropout layer is implemented with a dropout rate of 0, denoting the absence of dropout during the training process, in order to alleviate the issue of overfitting. The model ends with an additional dense layer that has only one neuron. This layer generates a binary classification output (None, 1) and has a total of 513 parameters.

The model consists of a total of 4,705,956 parameters. Out of these parameters, 4,663,933 are trainable, meaning they can be modified during the training process. By comparison, there are 42,023 parameters classified as non-trainable, which remain constant during the training process. Non-trainable parameters often encompass elements such as batch normalization statistics or fixed components within the architecture. These parameters play a role in enhancing the stability and efficiency of the model during training.

As illustrated in Figure 4, the general structure of EfficientNet contains a stem block, ten blocks, and a final layer.

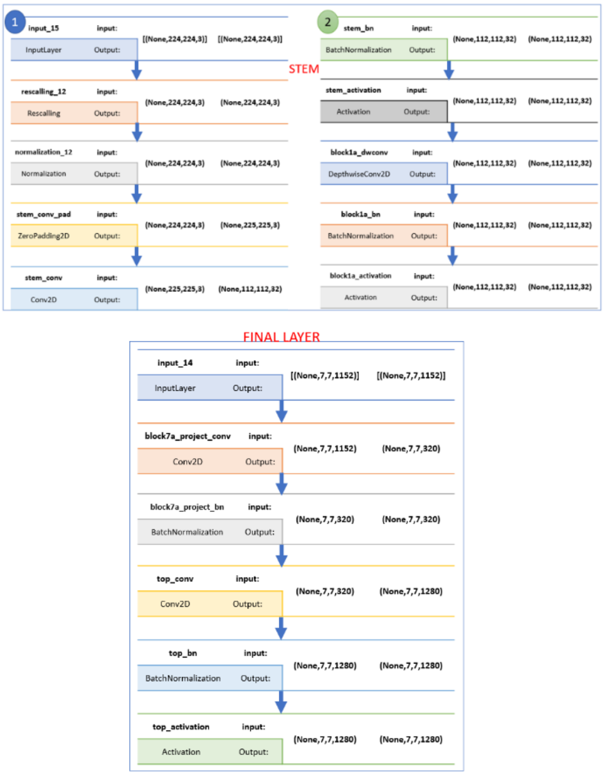

Figure 4. Stem and Final Layer in EfficientNet.

2.3. The optimizer used in the EfficientNet architecture

The CNN model applies optimizers to minimize loss and maximize production efficiency, utilizing momentum to expedite the optimization process. Keras has many optimizers, including stochastic gradient descent (SGD), adaptive moment estimation (Adam), root mean square propagation (RMSprop), adaptive gradient algorithm (AdaGrad), adaptive delta (AdaDelta), and Nesterov-accelerated adaptive moment estimation (Nadam). You can find more information about these optimizers on the Keras website: https://keras.io/api/optimizers/.

In this study, we examine the formulations of SGD, Adam, RMSprop, AdaGrad, AdaDelta, and Nadam [17].

Based on the equation provided below, you can change the parameters of the CNN using SGD.

$$W_{new} = W_{old} - \eta \nabla L = (W_{old}, x_i, y_i)$$

The updated weight, W_{new}, is calculated using the previous weight value, W_{old}, the learning rate, η, and the gradient of the loss function $L = (W_{old}, x_i, y_i)$ is$\nabla L = (W_{old}, x_i, y_i)$.

Based on the equation provided below, you can change the parameters of the CNN using Adam.

$$W_{new} = W_{old} + \Delta W$$

$$\Delta W = -\eta \frac{\hat{m}_t}{\sqrt{\hat{u}_t} + \varepsilon}$$

To ensure numerical stability, we employ a tiny constant, ε, set to 10^{-8}.

Based on the equation provided below, you can change the parameters of the CNN using RMSprop.$W_{new} = W_{old} - \frac{\eta}{\sqrt{MeanSquare(W,t)}} \nabla L(W_{old})$

$$MeanSquare(W,t) = \rho MeanSquare(W, t-1) + (1 - \rho)(\nabla L(W))^2$$

Therefore, ρ represents the forgetting factor, which has a value of 0.9, and t represents the current time step.

Based on the equation provided below, you can change the parameters of the CNN using AdaGrad.

$$W_{t+1,i} = W_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \cdot g_{t,i}$$

The partial derivative of the loss function is $g_{t,i}$. $g_{t,ii}$ is a diagonal matrix with i diagonal elements. A smoothing factor, ε, prevents zero division.

Based on the equation provided below, you can change the parameters of the CNN using AdaDelta.

$$W_{t+1} = W_t - \frac{RMS[\Delta W]_{t-1}}{RMS[g]_t} g_t$$

RMS refers to the root mean square error.

Based on the equation provided below, you can change the parameters of the CNN using Nadam.

$$W_{new} = W_{old} - \frac{\eta}{\sqrt{\sqrt{\hat{u}_t} + \varepsilon}} \widehat{m}_t$$

3. Result and Discussion

A total of 240 fundus images were used in this study, applying the EfficientNet architecture. The performance of various optimizers, including SGD, Adam, RMSprop, AdaGrad, AdaDelta, and Nadam, was evaluated to determine the optimal classification approach. Epoch Number set the total of 100, as seen in Table 3.

Table 3. Classification Results of EfficientNet Using Different Optimisations

Optimizer	Epoch Number	Best Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Testing Loss	Testing Accuracy	Elapsed Time (seconds)
SGD	100	54	0.1225	0.95	0.6356	0.8	0.3628	0.9	2575.16
Adam	100	13	0.0046	1	1.2824	0.775	0.0002	1	5315.72
RMSprop	100	87	0.0569	0.9812	1.9877	0.9	0.8647	0.9	2518.24
AdaGrad	100	2	0.0303	1	0.2965	0.875	0.0465	0.95	4935.94
AdaDelta	100	68	0.6143	0.6687	0.6266	0.7	0.4908	0.9	3570.92
Nadam	100	44	0.0355	0.9937	1.0153	0.775	0.19	0.95	7812.87

The model trained with Adam Optimizer achieved its best epoch at 13, demonstrating a remarkable training accuracy of 1.0000. However, the validation loss and accuracy at the end of training were 1.2824 and 0.7750, respectively. Meanwhile, the testing loss was impressively low at 0.0002, with the testing accuracy matching the training accuracy at 1.0000. The training and testing process took a total of 5315.72 seconds. On the other hand, the Adagrad optimizer, in its best epoch at 2, resulted in a slightly higher training loss of 0.0303 compared to Adam. Nevertheless, the training accuracy remained excellent at 1.0000. The validation loss and accuracy for Adagrad were 0.2965 and 0.8750, respectively. The testing loss and accuracy were 0.0465 and 0.9500, respectively, indicating a robust generalization to unseen data. Notably, the total elapsed time for training and testing with Adagrad was slightly shorter, recorded at 4935.94 seconds.

Figure 5 and Figure 6 show the graphical depiction of the loss function and model accuracy after applying data augmentation.

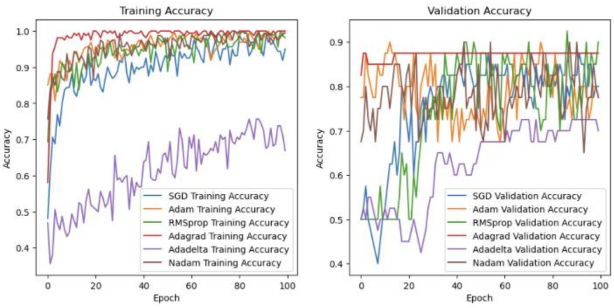

Figure 5 displays the accuracy of model training and validation when data augmentation is implemented.

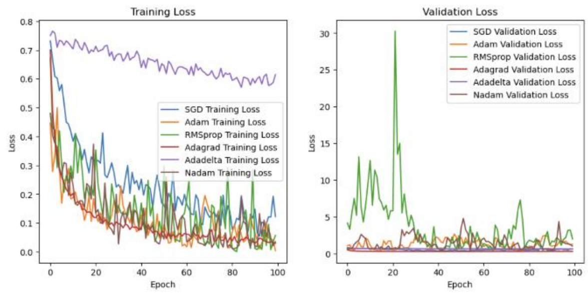

Figure 6 displays the loss of model training and validation when data augmentation is applied.

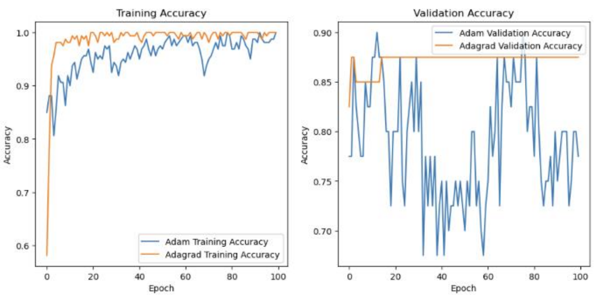

Figure 7. Data augmentation accuracy for Adam and Adagrad model training and validation.

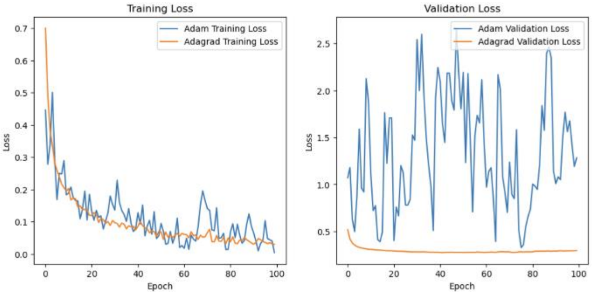

Figure 8. Data augmentation loss for Adam and Adagrad model training and validation.

4. Conclusion

The study examined the categorization of cataracts using fundus images by evaluating several Optimizer in EfficientNet architectures, including SGD, Adam, RMSprop, AdaGrad, AdaDelta, and Nadam, with RGB input. In summary, while Adam exhibited superior training accuracy, Adagrad demonstrated faster convergence and strong generalization performance on the validation and testing sets. Depending on the particulars of the job and the intended trade-offs between training speed and model generalization, one can select one of these optimizers. This study hopes to assist medical personnel in the early detection of cataracts while mitigating the potential concerns associated with cataracts and allowing the treatment of suitable medical interventions. Our goal is to increase the quantity of datasets in order to enhance the classification accuracy of the cataract detection system in the future.

References

[1] Imelda, Eva, Rinaldi Idroes, Khairan Khairan, Rodiah Rahmawaty Lubis, Abdul Hawil Abas, Ade John Nursalim, Mohamad Rafi, and Trina Ekawati Tallei, "Natural Antioxidant Activities of Plants in Preventing Cataractogenesis," Antioxidants, vol. 11, no. 7, p. 1285, 2022. DOI: 10.3390/antiox11071285.

[2] Jiao, Xiaodong, Mariia Viswanathan, Nadiia Fedorivna Bobrova, Tatiana Viktorivna Romanova, and J. Fielding Hejtmancik, "Molecular Genetic Analysis of Ukrainian Families with Congenital Cataracts," Children, vol. 10, no. 1, p. 51, 2022. DOI: 10.3390/children10010051.

[3] Cicinelli, M. V., Buchan, J. C., Nicholson, M., Varadaraj, V., & Khanna, R. C., "Cataracts," The Lancet, vol. 401, no. 10374, pp. 377-389, 2023. DOI: 10.1016/S0140-6736(22)01839-6.

[4] Kementerian Kesehatan RI, "Katarak - Kebutaan yang Dapat Dicegah," [Online]. Available: https://yankes.kemkes.go.id/view_artikel/891/katarak-kebutaan-yang-dapat-dicegah.

[5] Y. Zhang, J. M. Gorriz, and Z. Dong, "Deep Learning in Medical Image Analysis," 2021. DOI: 10.3390/books978-3-0365-1470-3.

[6] Purwono, A. M., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. ul., "Understanding of Convolutional Neural Network (CNN): A Review," Int. J. Robotics Control Syst., vol. 2, no. 4, pp. 739-748, 2022. DOI: 10.31763/ijrcs.v2i4.888.

[7] ÇetiNer, H., "Cataract disease classification from fundus images with transfer learning-based deep learning model on two ocular disease datasets," Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2023. DOI: 10.17714/gumusfenbil.1168842.

[8] Simanjuntak, R. B. J., Fu'adah, Y., Magdalena, R., Saidah, S., Wiratama, A. B., & Ubaidah, I. D. S., "Cataract Classification Based on Fundus Images Using Convolutional Neural Network," JOIV: Int. J. Informatics Visualization, vol. 6, no. 1, p. 33, 2022. DOI: 10.30630/joiv.6.1.856.

[9] Firdaus, D. H., Imran, B., Bakti, L. D., & Suryadi, E., "Klasifikasi Penyakit Katarak pada Mata Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Web," J. Kecerdasan Buatan dan Teknologi Informasi (JKBTI), vol. 1, no. 3, pp. 18-26, 2022.

[10] Junayed, M. S., Islam, M. B., Sadeghzadeh, A., & Rahman, S., "CataractNet: An Automated Cataract Detection System Using Deep Learning for Fundus Images," IEEE Access, vol. 9, pp. 128799-128808, 2021. DOI: 10.1109/ACCESS.2021.3112938.

[11] Cahya, F. N., Hardi, N., Riana, D., & Hadiyanti, S., "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," SISTEMASI, vol. 10, no. 3, p. 618, 2021. DOI: 10.32520/stmsi.v10i3.1248.

[12] Syarifah, M. A., Bustamam, A., & Tampubolon, P. P., "Cataract classification based on fundus image using an optimized convolution neural network with lookahead optimizer," 2020. DOI: 10.1063/5.0030744.

[13] "Cataract Dataset," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/jr2ngb/cataractdataset.

[14] Eka Putra, W. S., "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," J. Teknik ITS, vol. 5, no. 1, 2016. DOI: 10.12962/j23373539.v5i1.15696.

[15] Tan, M., & Le, Q., "Efficientnet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Machine Learning, 2019, pp. 6105–6114.

[16] Raza, R., Zulfiqar, F., Khan, M. O., Arif, M., Alvi, A., Iftikhar, M. A., & Alam, T., "Lung-EffNet: Lung cancer classification using EfficientNet from CT-scan images," Eng. Appl. Artif. Intell., vol. 126, p. 106902, 2023. DOI: 10.1016/j.engappai.2023.106902.

[17] Uddin, M. J., Li, Y., Sattar, M. A., Nasrin, Z. M., & Lu, C., "Effects of learning rates and optimization algorithms on forecasting accuracy of hourly typhoon rainfall: Experiments with convolutional neural network," Earth Space Sci., vol. 9, p. e2021EA002168, 2022. DOI: 10.1029/2021EA002168.