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

Classification of Myopia Levels using Deep Learning Methods on Fundus Image

Waeisul Bismi[1] and Jufriadif Na`am[2]*

[1,2] *University of Nusa Mandiri Jakarta, Indonesia*

ARTICLE HISTORY

Received: 08 June 23

Final Revision: 10 June 23

Accepted: 20 June 23

Online Publication: 30 June 23

KEYWORDS

Myopia, Eye Disease, Classification, Augmentation, Deep Learning

CORRESPONDING AUTHOR

jufriadifnaam@nusamandiri.ac.id

DOI

10.37034/medinftech.v1i2.8

A B S T R A C T

Disorders of the eye or also known as eye disease is a condition that can affect vision for some people in their lifetime. There are 40 types of eye disorders or eye diseases, one of which is Myopia. Myopia is a visual disturbance that causes objects that are far away to appear blurry, but there is no problem seeing objects that are near. Myopia or nearsightedness is also known as minus eye. From this description, it is very important to conduct research in detecting eye diseases before the increase in eye minus and blindness. This study aims to classify myopic eye disease using the Deep Learning method with several different architectures, namely the VGG16, VGG19 and InceptionV3V3 models. Where the first is to distinguish normal and abnormal while the other is to classify with Augmented myopia image dataset and non augmented myopia image dataset obtained from the Retinal Fundus Multi-Disease Image Dataset (RFMID). In the implementation of the Deep Learning method using 20 Epochs. The results of the accuracy of the classification of eye diseases using the non augmented myopia image dataset are 66.0% for the VGG16 architectural model, then 95.99% for the VGG19 architectural model and 93.99% for the InceptionV3 architectural model and the accuracy results using the Augmented myopia image dataset are 97.53% for the VGG16 architectural model, 97.53% for the VGG19 architectural model and 99.50% for the InceptionV3 architecture model.

1. Introduction

Disorders of the eye or also called eye disease is a condition that can affect vision for some people in their lifetime [1]. Therefore the eye is one of the five sense organs that is very important in human life to see. If the eye has an eye disorder or disease, it will be fatal for human life, namely blindness. Eye disease is a disease with the number of sufferers increasing every year in Indonesia. The prevalence of myopia is increasing globally being at 1% to 4% of the leading causes of blindness [2] WHO data estimate that 246 million people worldwide are at an alarming rate, with a significant increase in the risk of impaired vision from pathological conditions associated with high myopia, including retinal damage 43%, cataract 33%, and glaucoma 2% [3]. Myopia has been considered a public health problem in the world, although most people with myopia generally maintain good vision with glasses or contact lenses [4].

Judging from the previous description, it is very important to detect myopic eye disease or eye disorders before there is an increase in eye minus or potentially blinding complications [5]. Research on eye diseases has been done quite a lot, such as this cataract detection research can also be done using digital images [1] then Identification of Retinal Disease with Segmentation Techniques on Diabetic Retinopathy on diabetic retinopathy images [6] and then classifying eye disease using optical cup segmentation and cup-to-disc ratio [7] The identification carried out on an image has been developed for quite a long time, one of which is by distinguishing the texture of the image. Image textures can be distinguished by several factors, including four

texture features used, namely contrast, correlation, energy and homogeneity [8]. In image processing, texture is defined as a measure of the surface variation of the pixel brightness intensity [9]. Texture is the main term used to define the object or concept of an image as well as the sight of the human eye. Texture analysis plays an important role in computer vision cases such as object recognition, surface flaw detection, pattern recognition, medical image analysis [10].

Classification method is used for workload distribution and detailed activity planning [11]. Traditionally, the image classification method is carried out with extracted image features and allows the learned features for the classification of one class to be used in the classification of another class as well as the most commonly used performance metric for the classification method is accuracy, the percentage of samples that are classified correctly [12].

Deep learning has shown proven advantages over traditional methods that rely on handcrafted features that include similar images in different categories and different images in the same category as for medical image classification [13]. Some of the advantages of deep learning models are the reduced need for feature production. and is considered to be one of the most time-reducing parts of the machine learning approach. In the case of very large data and images it can easily recognize, understand, troubleshoot and work more efficiently. By using deep learning methods, it is possible to avoid the common problem of consuming too much time. Image classification is an important area and it is a big challenge for images to be classified with medical expert knowledge [14]. By Therefore this study will carry out a classification process where the first one is to distinguish normal and abnormal while the other is to classify with Augmented myopia image dataset and non Augmented myopia image dataset obtained from the Retinal Fundus Multi-Disease Image Dataset (RFMID) [15] using the Deep Learning method and the architectural models used are VGG16, VGG19, and InceptionV3. This is because the Deep Learning Model works by trying to imitate an image recognition system in the human visual cortex so that it has the ability to process image information like humans. The benefit of this research is that it becomes more efficient and facilitates the classification process as expected, especially in the health sector, namely the classification of eye diseases. Classification can also be obtained with a precise time and higher accuracy using the method proposed in this study.

2. Research Method

Using Deep Learning to be able to classify eye diseases. The method proposed in this study can be described as in Figure 1.

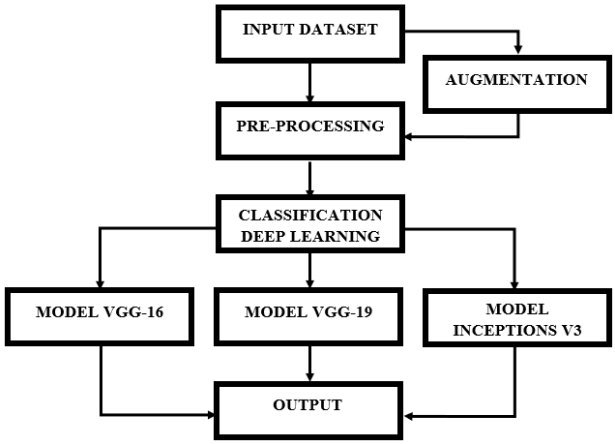


Figure 1. Research Methods

2.1. Dataset

The dataset used in this study was obtained from [15]. This dataset is a collection of images of retinal fundus multi-diseases so that researchers want to try to develop it. This study describes the disease myopia. This dataset consists of 1920 datasets which are divided into 40 classes. Then the researcher made a selection among the datasets obtained by dividing them into 2 classes that explained the classification of the differences between normal eyes and myopia.

Where the purpose of this classification will be very useful and help detect eye diseases precisely and accurately. An example of the image used can be seen in Figure 2.

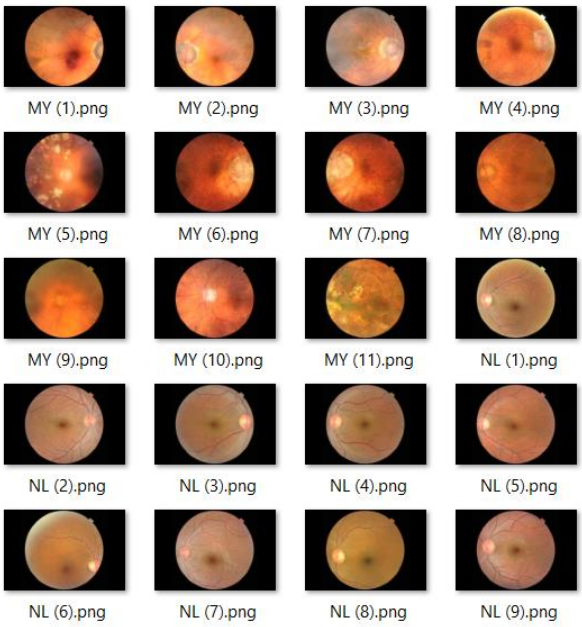


Figure 2. Example Dataset For Each Class

2.2. Augmentation

Image augmentation is one of the effective training strategies to grow image collections for Deep Learning

models which also does not include sufficient image data. Augmentation of image data is needed because it is needed to achieve better test accuracy and assist in producing balanced data [16]. At this stage, image data augmentation is carried out, because the original is generated from ground-truth data naturally. The techniques implemented on image data include Rotation, Zoom, and Flipping. The following is an example of an image that has been implemented with augmentation techniques, which can be seen in Figure 3.

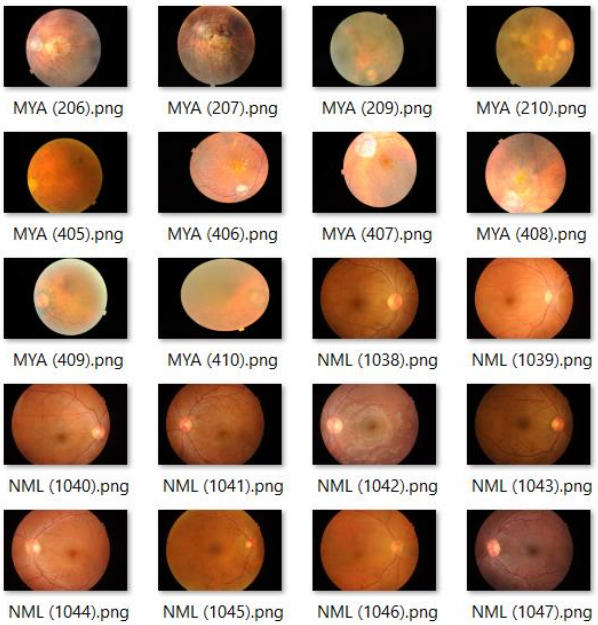


Figure 3. Example of an Augmented Dataset

2.3. Pre-Processing

Pre-processing is a universal technique that can be widely used in deep learning i.e. the process is executed once for the entire dataset (which can then be stored in preprocessed form and used multiple times for training/testing phases) [17], The preprocessing stage is used to remove noise/variations in the retinal fundus image and improve image quality and contrast. Apart from contrast enhancement and noise reduction, the preprocessing step can be used for image normalization and non-uniform intensity correction to remove artifacts and increase the accuracy of the processing steps [18]. At this stage an analysis of the pre-processed input data is carried out before entering the classification stage. The pre-processing stages carried out are Resize, RGB, and Label Encode.

2.4. Deep Learning

Deep learning is rapidly becoming an art leading to performance improvement in various medical applications such as image review, detection of anatomical and cellular structures, tissue segmentation, computer-aided diagnosis and prognosis of disease and so on [19]. This concept lies at the base of many deep learning algorithms: a model (network) made up of many layers that converts input data (e.g. images) into outputs (e.g. disease present/absent) while learning progressively higher-level features. The most successful type of model for image analysis to date is convolutional neural networks (CNNs) [20]. With the rapid development of Deep Learning technology and increasing computational capabilities, deep learning has been widely used in the field of image classification [21]. In general, Deep learning methods often contain many parameters that can be trained and require a large number of labeled samples to achieve optimal performance, especially convolutional neural network (CNN) analysis [22]. CNN was first developed under the name NeoCognitron by Kunihiko Fukushima, a researcher from the NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan [1].

The CNN Architecture model used in this study, namely the VGG16, VGG19 and Inceptionv3 architecture can be seen in Figure 4.

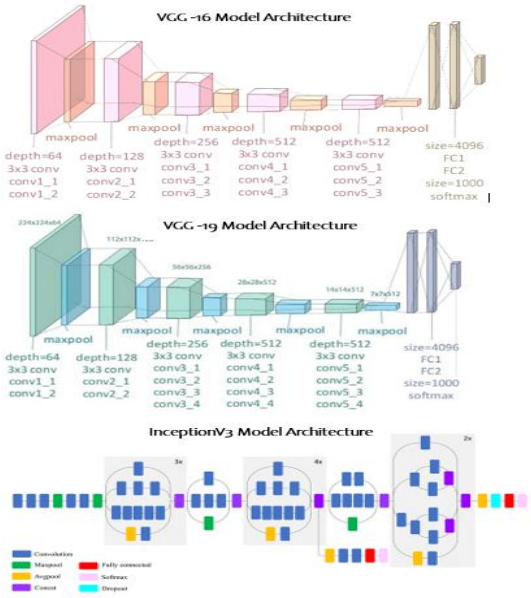


Figure. 4 Deep Learning Architectural Models

2.5. The Consept of Deep Learning

The concept of a deep learning algorithm is similar to the multilayer perceptron (MLP) model, in that both models are trained with a version of the back-propagation algorithm, and both consist of input, output, and hidden layers. reused a propagation algorithm [23] which consists of several neurons (also known as nodes) connected together to form a complex network. Neurons receive multiple input signals and produce outputs [24]. Each neuron in one layer is connected to every neuron in the next layer [25].

However in CNN, the hidden layer consists of several convolutional layers, rectified linear unit (RELU) layers, pooling-layers, fully connected layers, and

normalization layers. The main difference between CNN from MLP is that, in the case of modeling problems with a high number of input variables, it is not possible to connect a neuron to all neurons in the previous layer because not only would such an architecture not take into account the spatial structure of the data set, but would also greatly increase the duration of the process. training. The following Figure 5 is related to the MLP Architecture and the Convulsion process on CNN.

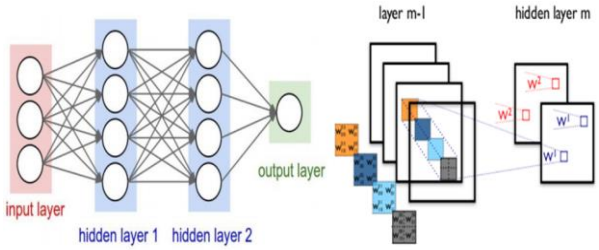


Figure 5. MLP Architecture and Convolution Process

2.6. Deep Learning Architecture

CNN architecture is usually created using the input layer, convolution layer, pooling layer, fully connected layer, and output layer. Different models are created with different layouts of these layers [26]. Based on the architecture, there are four main layers in a CNN, only three layers are applied in this study:

a. Convolution Layer Convolution Layer performs convolution operations on the output of the previous layer. This layer is the main process that underlies a CNN. The weights in this layer specify the convolution kernel used, so that the convolution kernel can be trained based on the input to the CNN. The purpose of convoluting image data is to extract features from the input image. Convolution can produce a linear transformation of the input data according to the spatial information in the data. The weight of the layer specifies the convolution kernel used, so that the convolution kernel can be trained based on the input to the CNN.

b. Subsampling Layer Subsampling is the process of reducing the size of an image data. In image processing, subsampling aims to increase the position invariance of features. Max pooling divides the output from the convolution layer to produce several small grids then takes the maximum value from each grid to compose the reduced image matrix as shown in Figure 5. Then the results of the process can be seen in the grid collection to the right. CNN for the purpose of reducing the size of the image so that it can be easily replaced with a convolution layer with the same stride as the relevant pooling layer.

c. Fully Connected Layer The Fully Layer is a layer that is usually used in the application of MLP and aims to transform the data dimensions so that they can be classified linearly. Each neuron in the convolution layer needs to be transformed into one-dimensional data before it can be entered into a fully connected layer. With it causing the data to lose its spatial information and become irreversible, the fully connected layer can only be implemented at the network end. A convolution layer with a kernel size of 1 x 1 performs the same function as a fully connected layer but retains the spatial character of the data.

2.7. Trining Process

The training process is the stage where CNN is trained to obtain higher accuracy from the classification carried out. At this stage there is a feed forward process and a backpropagation process. To carry out the feedforward process, it is necessary to have the number and size of the layers to be formed, the size of the subsampling, and the vector image obtained. The results of the feedforward process are in the form of weights that will be used to evaluate the neural network process.

a. Feedforward Process The feed forward process is the first stage in the training process. This process will produce several layers used to classify image data which uses updated weights and biases from the backpropagation process.

b. Backpropagation Process The backpropagation process is the second stage of the training process. At this stage, as described above, the results of the feedforward process are traced to errors from the output layer to the first layer.

c. Gradient Calculation The gradient process for the convolution network is a process to generate new bias and weight values and will be needed during training.

2.8. Testing Process

The testing process is a classification process that uses bias and weights from the results of the training process. So that the end of this process results in the accuracy of the classification carried out, the data that failed to be classified, the number of images that failed to be classified, and the form of the network formed from the feedforward process. The output layer is fully connected with the existing labels.

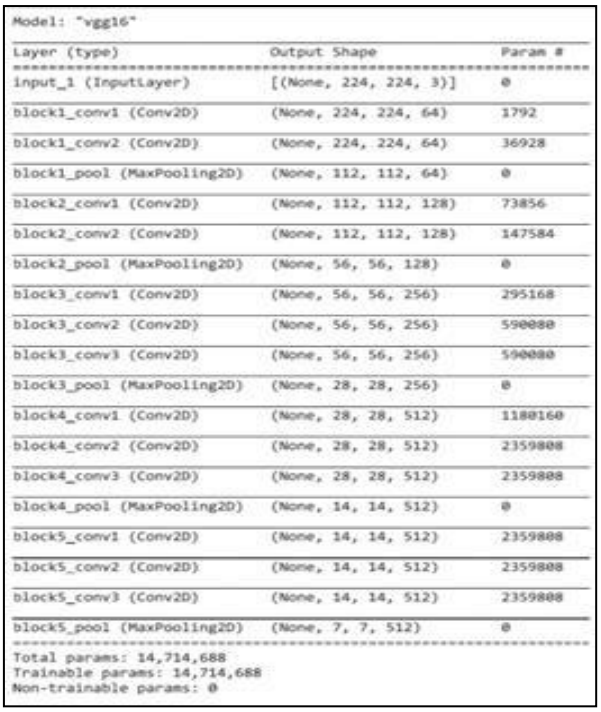
2.9. Validation Process

The data validation process is used to obtain a level of confidence in the results of image interpretation [27] in the recognition of myopic eye disease in retinal fundus images. Validation is done by using image data, namely the Retinal Fundus Multi-Disease Image Dataset (RFMID) [15] which is divided into training data.

3. Result and Discussion

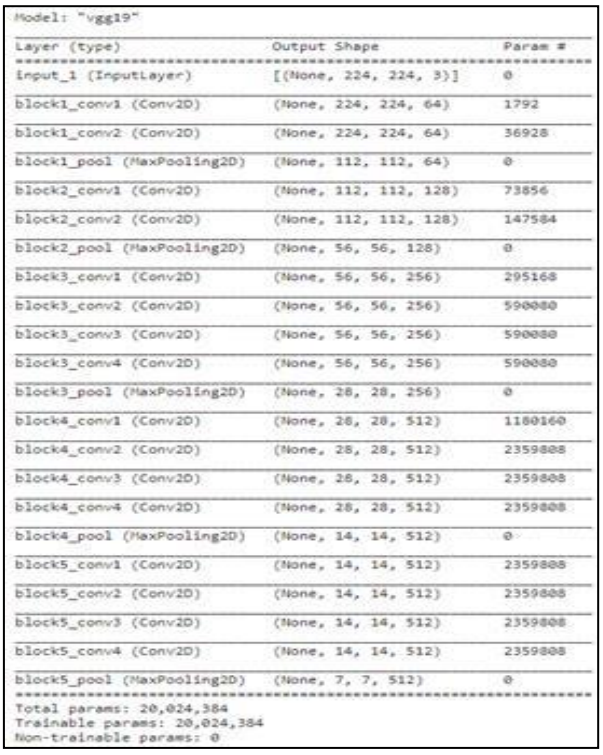
3.1. Deep Learning Model Architecture

The deep learning model architecture or CNN used in this research is the Vgg16, Vgg19 and InceptionV3 architecture with the applied models can be seen in Figure 6, Figure 7 and Figure 8.



```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Figure 6. VGG-16 Architecture Model



```
Model: "vgg19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv4 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv4 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 20,024,384
Trainable params: 20,024,384
Non-trainable params: 0
```

Figure. 7 VGG-19 Architecture Model

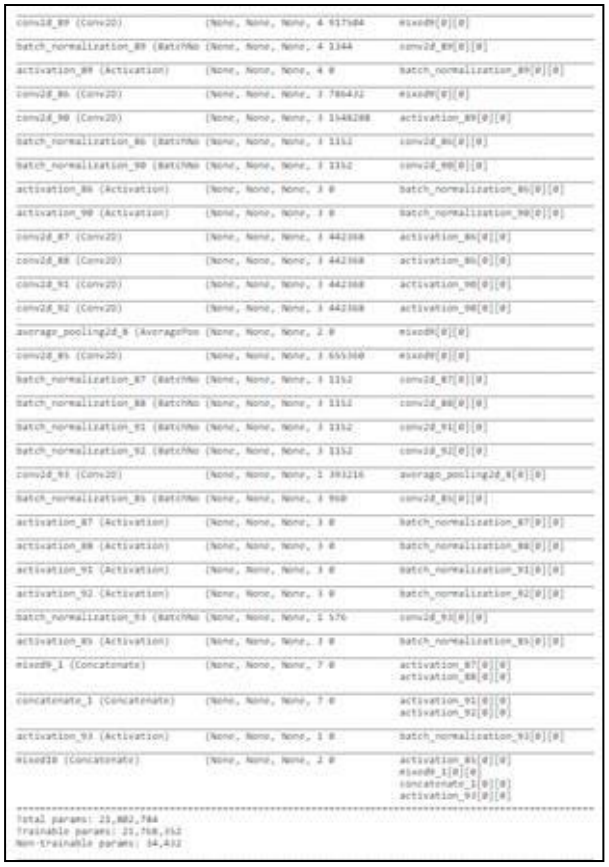


Figure. 8 InceptionV3 Architecture Model

3.2. Distribution of Training, Testing, and Valid Data

In the pre-processing stage, the image is resized to 224x224px. with a uniform image size for ease of computation. In addition, a uniform image size can also facilitate the recognition stage. Next, enter the data distribution which can be seen in Table 1.

Table 1. Distribution Of Train, Test And Valid Data

No.	Dataset	Train	Test	Val	Total
1	Dataset Original	395	50	50	495
2	Dataset Augmentation	1619	203	203	2025

Based on Table 1 the dataset is divided into 2 parts, namely the original dataset and the augmentation dataset, the training and validation data are used in the training, tuning, and evaluation process of the CNN model, while the test data is used to test the performance of the training model. After getting an image with a uniform size and distribution of image data, then enter the classification stage. At this stage the researcher uses the parameters as shown in Table 2.

Table 2. Parameter

Size (Pixel)	Epoch	Batch_size	Optimizer
224x224	20 Epoch	32	Adam

Based on the parameters used with the Vgg16, Vgg19 and IncepstionV3 architecture and the model applied, the results of the loss and accuracy graphs of the original dataset are obtained and the results of the loss

and accuracy graphs of the augmentation datasets as shown in Figure 9 and Figure 10.

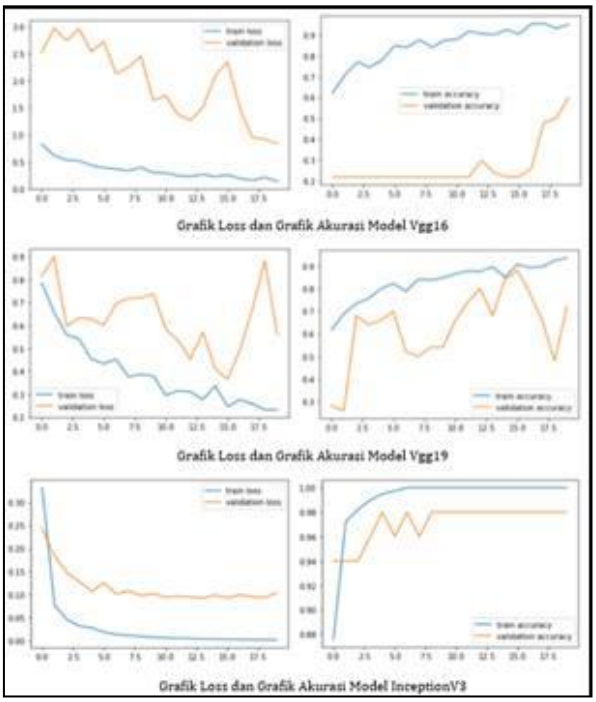


Figure 9. Graph Results of Each Original Dataset Model

Based on the graph results, each model can classify eye diseases using Deep Learning with several models from the Original Dataset and Augmentation Dataset which produces the accuracy shown in Table 3.

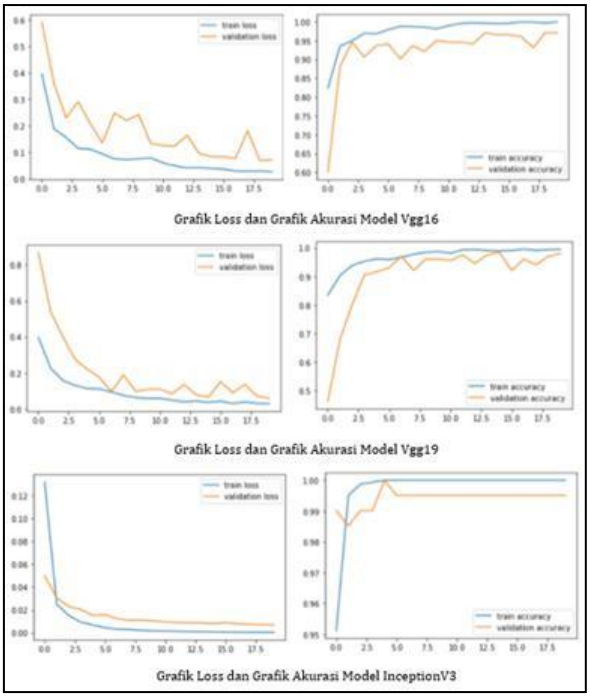


Figure 10. Graph Results of Each Augmentation Dataset Model

Tabel. 3 Accuracy And Kappa Score

Dataset Original

Model	Accuracy (%)	Kappa Score (%)
Vgg-16	66,00	39,20
Vgg-19	95,99	91,10
Inceptionv3	93,99	85,98

Dataset Augmentation

Model	Accuracy (%)	Kappa Score (%)
Vgg-16	97,53	94, 96
Vgg-19	97,54	94,94
Inceptionv3	99,50	98,98

To find out the classification performance, an evaluation of the confusion matrix is carried out as can be seen in Figure 11.

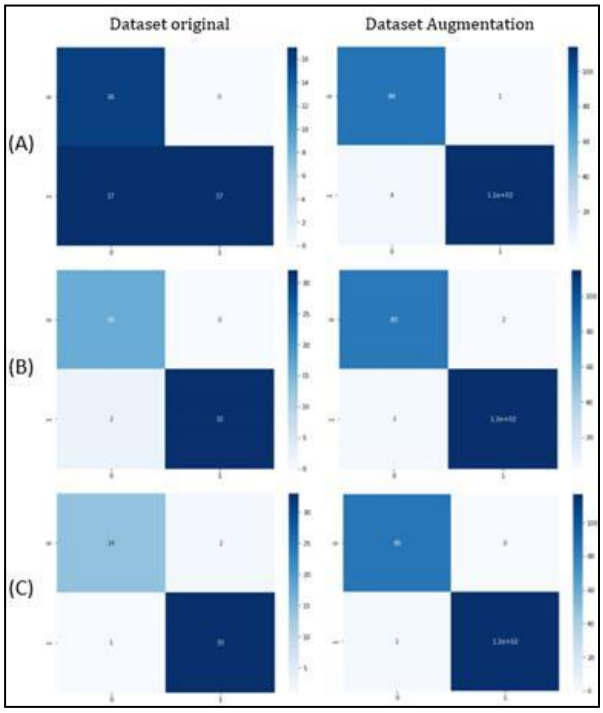


Figure. 11 Confusion Matrix (A) VGG-16, (B) VGG-19, (C) InceptionV3

4. Conclusion

From the results of research on eye disease classification using Deep Learning CNN method with model architecture Vgg16, Vgg19, and InceptionV3 with an update in the form of using 2 classes, namely normal and abnormal and using the original dataset and augmented dataset obtained from the Retinal Fundus Multi-Disease Image Dataset (RFMID) dataset. then proceed with the pre-processing stage to change the image size to 224x224px. The next step is feature extraction with 3 layers, namely the convutional layer, pooling layer, and fully connected layer according to the proposed architecture. At the CNN implementation stage using 20 epochs, the highest accuracy results from the myopia eye disease classification study using the Deep learning method from the original dataset of 95.99% for the vgg19 model and from the augmentation dataset of 99.50% for the InceptionV3

model. and for future work, implementing Generative Adversarial Networks (GANs) to Improve Convolutional Neural Network (CNN) Performance.

References

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

[2] J. Ruiz-Medrano, E. Almazan-Alonso, I. Flores-Moreno, M. Puertas, M. García-Zamora, and J. M. Ruiz-Moreno, "The relationship between myopic choroidal neovascularization activity and perforating scleral vessels in high myopia," Retina, vol. Publish Ah, pp. 204–209, 2021, doi: 10.1097/iae.0000000000003290.

[3] World Health Organization, The Impact of Myopia and High Myopia: report of the joint World Health Organisation-Brian Holden Vision Institute Global Scientific Meeting on Myopia, no. March. 2015.

[4] R. Du et al., "Deep Learning Approach for Automated Detection of Myopic Maculopathy and Pathologic Myopia in Fundus Images," Ophthalmol. Retin., pp. 1–10, 2021, doi: 10.1016/j.oret.2021.02.006.

[5] N. Széll et al., "Myopia-26, the female-limited form of early-onset high myopia, occurring in a European family," Orphanet J. Rare Dis., vol. 16, no. 1, pp. 1–13, 2021, doi: 10.1186/s13023-021-01673-z.

[6] P. Powar and C. R. Jadhav, "Retinal Disease Identification by Segmentation Techniques in Diabetic Retinopathy," Lect. Notes Networks Syst., vol. 10, pp. 255–265, 2018, doi: 10.1007/978-981-10-3920-1_26.

[7] B. B. Naik and R. Mariappan, "Classification of Eye Diseases Using Optic Cup Segmentation and Optic Disc Ratio," IOSR J. Comput. Eng., vol. 18, no. 05, pp. 87–94, 2016, doi: 10.9790/0661-1805038794.

[8] F. S. Ni'mah, T. Sutojo, and D. R. I. M. Setiadi, "Identification of Herbal Medicinal Plants Based on Leaf Image Using Gray Level Co-occurence Matrix and K-Nearest Neighbor Algorithms," J. Teknol. dan Sist. Komput., vol. 6, no. 2, pp. 51–56, 2018, doi: 10.14710/jtsiskom.6.2.2018.51-56.

[9] B. A. Varghese, S. Y. Cen, D. H. Hwang, and V. A. Duddalwar, "Texture Analysis of Imaging: What Radiologists Need to Know," Ajr, no. 212, pp. 1–9, 2019.

[10] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods - A review," vol. 2, no. 1, pp. 1–29, 2019, [Online]. Available: http://arxiv.org/abs/1904.06554.

[11] G. Bruno and D. Antonelli, "Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells," Int. J. Adv. Manuf. Technol., vol. 98, no. 9–12, pp. 2415–2427, 2018, doi: 10.1007/s00170-018-2400-4.

[12] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, "A defect classification methodology for sewer image sets with convolutional neural networks," Autom. Constr., vol. 104, no. December 2018, pp. 281–298, 2019, doi: 10.1016/j.autcon.2019.04.013.

[13] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," Med. Image Anal., vol. 54, pp. 10–19, 2019, doi: 10.1016/j.media.2019.02.010.

[14] R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, and K. Shankar, "Optimal feature selection-based medical image classification using deep learning model in internet of medical things," IEEE Access, vol. 8, pp. 58006–58017, 2020, doi: 10.1109/ACCESS.2020.2981337.

[15] S. Pachade et al., "Retinal Fundus Multi-disease Image Dataset (RFMiD)," 2020. doi: https://dx.doi.org/10.21227/s3g7-st65.

[16] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," Artif. Intell. Rev., no. 0123456789, 2021, doi: 10.1007/s10462-021-10066-4.

[17] P. Hurtik, V. Molek, and J. Hula, "Data Preprocessing Technique for Neural Networks Based on Image Represented by a Fuzzy Function," IEEE Trans. Fuzzy Syst., vol. 28, no. 7, pp. 1195–1204, 2020, doi: 10.1109/TFUZZ.2019.2911494.

[18] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, J. Ma, and K. Wang, "Image Preprocessing in Classification and Identification of Diabetic Eye Diseases," Data Sci. Eng., vol. 6, no. 4, pp. 455–471, 2021, doi: 10.1007/s41019-021-00167-z.

[19] Y. Zhang, J. M. Gorriz, and Z. Dong, "Deep learning in medical image analysis," J. Imaging, vol. 7, no. 4, p. NA, 2021, doi: 10.3390/jimaging7040074.

[20] G. Litjens et al., "A survey on deep learning in medical image analysis," Med. Image Anal., vol. 42, no. December 2012, pp. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.

[21] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and Y. Shiqi, "A survey: Deep learning for hyperspectral image classification with few labeled samples, Neurocomputing," Neurocomputing, vol. 448, pp. 179–204, 2021, doi: https://doi.org/10.1016/j.neucom.2021.03.035.

[22] V. Shah, R. Keniya, A. Shridharani, M. Punjabi, J. Shah, and N. Mehendale, "Diagnosis of COVID-19 using CT scan images and deep learning techniques," Emerg. Radiol., vol. 28, no. 3, pp. 497–505, 2021, doi: 10.1007/s10140-020-01886-y.

[23] N. Calik, M. A. Belen, and P. Mahouti, "Deep learning base modified MLP model for precise scattering parameter prediction of capacitive feed antenna," Int. J. Numer. Model. Electron. Networks, Devices Fields, vol. 33, no. 2, 2020, doi: 10.1002/jnm.2682.

[24] J. Sultana, M. Usha Rani, and M. A. H. Farquad, "Student's performance prediction using deep learning and data mining methods," Int. J. Recent Technol. Eng., vol. 8, no. 1 Special Issue 4, pp. 1018–1021, 2019.

[25] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo, "Fast flow field prediction over airfoils using deep learning approach," Phys. Fluids, vol. 31, no. 5, 2019, doi: 10.1063/1.5094943.

[26] A. Çinar and M. Yildirim, "Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture," Med. Hypotheses, vol. 139, no. February, p. 109684, 2020, doi: 10.1016/j.mehy.2020.109684.

[27] K. Arisudana et al., "Sistem pendeteksi kerusakan luar angkutan umum," pp. 175–187, 2020.