



Classification of Diabetic Retinopathy Disease Using Convolutional Neural Network

Agus Eko Minarno ^a, Mochammad Hazmi Cokro Mandiri ^a, Yufis Azhar ^{a,*}, Fitri Bimantoro ^b,
Hanung Adi Nugroho ^c, Zaidah Ibrahim ^d

^a Faculty of Engineering, Universitas Muhammadiyah Malang, Indonesia

^b Faculty of Engineering, Universitas Mataram, Indonesia

^c Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia

^d Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia

Corresponding author: *yufis@umm.ac.id

Abstract— Diabetic Retinopathy (DR) is a disease that causes visual impairment and blindness in patients with it. Diabetic Retinopathy disease appears characterized by a condition of swelling and leakage in the blood vessels located at the back of the retina of the eye. Early detection through the retinal fundus image of the eye could take time and requires an experienced ophthalmologist. This study proposed a deep learning method, the Efficientnet-b7 model to identify diabetic retinopathy disease automatically. This study applies three preprocessing techniques that could be implemented in the dataset "APTOS 2019 Blindness Detection". In preprocessing technique trial scenarios, Usuyama preprocessing technique obtained the best results with accuracy of 89% of train data and 84% in test data compared to Harikrishnan preprocessing technique which has 82% accuracy in test data, and Ben Graham preprocessing has 81% accuracy in test data. In this study, Hyperparameter tuning was conducted to find the best parameters for use on the EfficientNet-B7 Model. In this study, we tested the Efficientnet-B7 model with an augmentation process that can reduce the occurrence of overfitting compared to models without augmentation. Preprocessing techniques and augmentation techniques can influence the proposed EfficientNet-B7 model in terms of performance results and reduce the overfitting of models.

Keywords— Image; classification; diabetic retinopathy; CNN; APTOS.

Manuscript received 27 Jan. 2022; revised 22 Feb. 2022; accepted 8 Mar. 2022. Date of publication 31 Mar. 2022.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Diabetes is a disease that causes high sugar levels in the blood due to the pancreatic organ's difficulty producing the hormone insulin so that the body does not absorb nutrients. Diabetic Retinopathy is a disease of the eye's retina arising from the effects of diabetes. Diabetic Retinopathy is characterized by one of the conditions in the damage to the blood vessels at the back of the retina [1]. Patients with diabetes for a long period may develop disease of Diabetic Retinopathy. The development of the disease can be prevented by early diagnosis through fundus examination is the most effective method to detect abnormal signs in the eye condition, to minimize the occurrence of eye blindness in patients [2].

Diabetic Retinopathy disease can be classified into five classes, among others no DR (normal), Mild DR (mild), Moderate DR (moderate), Severe DR (severe), and

Proliferative DR. Mild DR is a type of symptom in the retinal blood vessels of the eye in the form of small red spots round like balloons. Moderate DR is a symptom of blockage in one of the blood vessels resulting in bleeding in the eye's retina, characterized by the appearance of spots such as cotton wool. Severe DR is a symptom that occurs due to numerous blockages in blood vessels so that the blood supply does not reach the retina. The symptoms are characterized by dilated red spots on the eye's retina [3]. Mild DR, Moderate DR, and Severe DR classes are included as Non-Proliferative Diabetic Retinopathy (NPDR) levels. At the NPDR stage, blood vessels occur blockages so that new blood vessels cannot grow properly. Proliferative DR is caused by the growth of new blood vessels that are thin and brittle so that when blood vessels leak could result in vision loss and even blindness [4], [5].

Early diagnosis of Diabetic Retinopathy is an early stage of preventing complications in the eye's retina. Diagnosis aims to minimize patients with diabetes from the appearance of

symptoms of red spots and blockage of blood vessels in the retina area. These symptoms are an early sign of Diabetic Retinopathy, and a timely diagnosis can reduce the risk of developing Diabetic Retinopathy towards serious symptoms[6]. Detection of Diabetic Retinopathy can be done manually by an ophthalmologist and automatically by the system. In the manual system, an ophthalmologist requires an ophthalmologist to analyze and explain fundus images on the retina, but the costs incurred are very expensive. However, automated systems with artificial intelligence called Deep Learning can have a role in the early detection of Diabetic Retinopathy [7].

Research related to the classification of Diabetic Retinopathy was conducted to propose a deep convolutional neural network (CNN) method to classify Diabetic Retinopathy into four classes, namely normal, moderate, heavy, and severe. The datasets used in the study amounted to 4476 images and were sourced from hospitals in China's Sichuan Province. In this study, researchers trained a modified model of the pre-trained inceptionV3 model with the help of the data processing and data augmentation stage. The proposed result achieved an accuracy of 88.72% [8].

Further research related to Diabetic Retinopathy disease was conducted by Kwasigroch et al. [9] that propose the Deep Convolutional Neural Network (CNN) method for detecting and classifying Diabetic Retinopathy. The datasets used in this study were sourced from EYEPACKS organizations with five classes showing the severity of the disease (0 - no disease, 5 - the most severe cases). The results showed 82% accuracy for detecting Diabetic Retinopathy disease. In a subsequent study conducted by Eman Abdel Maksoud et al. in 2020, this study proposed the deep learning (DL) method with the EfficientNet architecture model. The dataset used in this study was sourced from Indian Diabetic Retinopathy Image Dataset (IDRiD) with five classes showing the level of diabetic retinopathy disease class. In this study, the preprocessing technique was simply to resize the image and implement augmentation. This study showed an accuracy value by the model of 86% [10].

Research related to Diabetic Retinopathy disease was conducted by Hari Krishnan et al. [11] that propose the Recurrent Neural Network (RNN) method for classifying diabetic retinopathy disease stage. The dataset used in this study is sourced from Kaggle's website released for the APTOS 2019 competition. Dataset consists of five classes: no DR, mild DR, moderate DR, severe DR, and proliferative DR. In this study, preprocessing techniques were to resize the image, gaussian blur technique with a standard deviation of 10 as well as augmentation process. The results showed 85% training accuracy and 82% test accuracy [11]. In addition to the research conducted by K. Shankar et al., the hyperparameter adjustment technique helps select the correct parameter values and better classification performance [12].

Based on the description of previous research [9], [10], [11], this study proposes the classification of Diabetic Retinopathy disease with CNN model architecture Efficientnet-B7. The EfficientNet-B7 Architecture Model is proposed because it has optimal performance compared to other architectures. In the preprocessing stage, the data is done augmentation process on the image using augmentation technique from the hard library.

In this study, the EfficientNet-B7 architecture model was implemented, then combined several preprocessing techniques and the hyperparameter Tuning method to obtain the results of the right combination of parameters used in the CNN method with EfficientNet-B7 architecture model.

II. MATERIAL AND METHOD

At this stage, we present the stages of research to classify Diabetic Retinopathy disease in the eyes. Here is an overview of the stages of research that can be seen in Fig. 1.

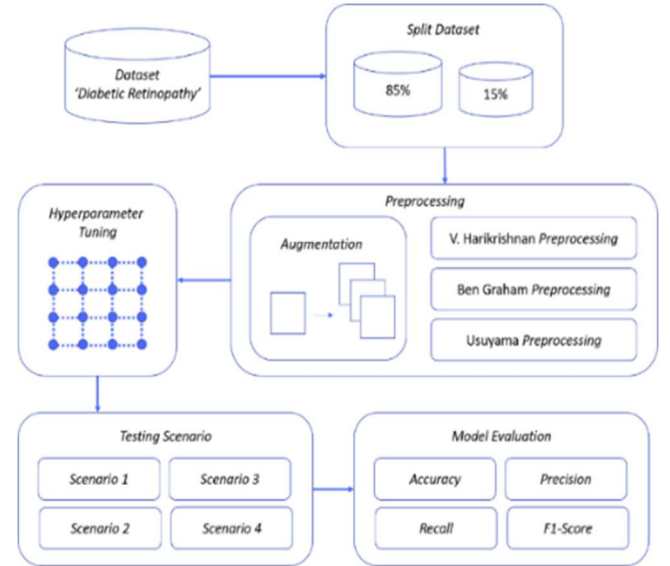


Fig. 1 Research Stage Diagram

This section includes the stages of works and the proposed method to provide solutions to calcifying the research data. This study was conducted with several scenarios to determine the influence of different preprocessing stages and the implementation of augmentation techniques. This research could be conducted on datasets on diabetic retinopathy disease.

A. Dataset

The dataset used in this study is an image of Diabetic Retinopathy disease collected from Kaggle's website titled "APTOS 2019 Blindness Detection". This dataset has a total of 3662 images. The Diabetic Retinopathy dataset in this study had a total of five classes with details No Dr of 1805, Mild DR of 370, Moderate DR of 999, Severe DR of 193, and Proliferative DR of 295.

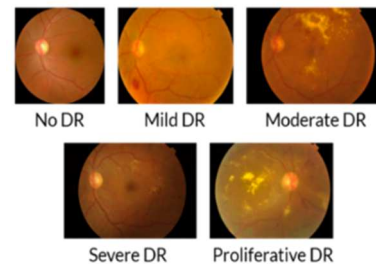


Fig. 2 Sample Data of Each Class Diabetic Retinopathy

Diabetic Retinopathy datasets total 3662 images. Datasets are divided into data train by 85% and test data by 15% [11]

[13]. The dataset could be shared using a library from Sklearn named `train_test_split`. The data train could be used to train the model to recognize the data that has been prepared, while the test data was used to test and validate the data of the trained model [14]. Here is a sample of data from each class of Diabetic Retinopathy disease can be seen in the Fig. 2.

B. Preprocessing Dataset

Preprocessing Datasets are a useful step for improving retinal image quality, as low-quality images can reduce model performance, and it is necessary to ensure that all images are consistent and image features can be improved [15]. Preprocessing data is intended to process images in datasets containing a lot of noise, under-focused images, overexposure, overexposure, and the presence of a black background in the image [16]. Three preprocessing techniques could be carried out in this study, among others:

1) *Harikrishnan Preprocessing technique* is a data preprocessing technique proposed by Harikrishnan et al. [11]. This technique applies to resizing an image to size 224x224 pixels, then carried out the implementation of gaussian blur by OpenCV library with a standard deviation value of 10. The following are the results of Harikrishnan preprocessing techniques can be seen in Fig. 3 below:



Fig. 3 (a) Original Image, (b) Result of Harikrishnan Preprocessing

2) *Graham [17] preprocessing technique*: is a data preprocessing technique proposed to improve Chatpatanasiri [18] technique. In this preprocessing, resize the image size, further reducing the average7 color of the image, then map to 50% gray—next, gaussian blur application with a standard deviation value of 30. Lastly, a crop circle is done so that the retinal image of the eye looks cut circularly and has an outer (circle border). The results of the Graham [17] preprocessing techniques can be seen in Fig. 4.



Fig. 4 (a) Original Image, (b) Result of Ben Graham Preprocessing

In this study, the authors resized the image size using a scale radius of 112 so that the image has a size of 224x224 pixels. Gaussian blur application with a standard deviation value of 5. Finally, a crop circle is done and remove the outer (circle border) around the retina [19]. The following results of Usuyama preprocessing techniques can be seen in Fig. 5.



Fig. 5 (a) Original Image, (b) Result of Usuyama Preprocessing

C. Augmentation

At this stage, the process of augmentation of data on diabetic retinopathy disease images aims to manipulate the number of images by providing several image augmentation techniques so that the image is recognized as a different image but still maintains the core of the image [20]. Some types of augmentation processes are implemented in preprocessing datasets. The augmentation process includes a rotation range of 90 degrees, zoom range = 0.2, shift range = 0.2, and horizontal and vertical flipping process. The augmentation process is intended to avoid or reduce the occurrence of overfitting on a small amount of data [21]. Here are the details of the augmentation process can be seen in Fig. 6.

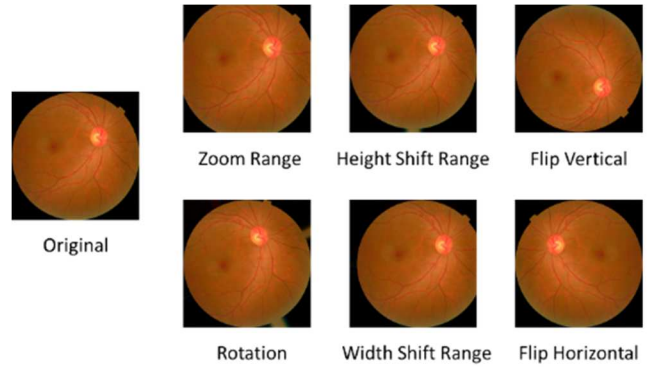


Fig. 6 Detail Type Augmentation Image

D. Hyperparameter Tuning

Hyperparameter Tuning is a technique for determining the architecture of layers in feature extraction on the Convolutional Neural Network (CNN), specifically in the fully connected layer section [22]. Hyperparameter Tuning is performed to search for parameters on a CNN model called EfficientNet-B7 optimally. On the model, hyperparameter tuning is adjusted using the random search algorithm of Keras Tuner, which is an optimization method to get the optimal parameters for the model. Some tuning parameters include optimizer for data training, dense value, and dropout on a fully connected layer. Hyperparameter Tuning could provide the best combination of parameters against the model and provide optimal results [12].

There are two hyperparameter tuning methods. The first method could be focused on finding optimizers that can develop with diabetic retinopathy datasets. Optimizers to be tested include SGD, Adam, Adamax, and Rmprop. The optimizer parameters are proposed based on research conducted by Maksoud et al. [10], and Harikrishnan et al. [11]. Some optimizers were used to train models on diabetic retinopathy datasets. In this study, selecting the right optimizer could make the EfficientNet-B7 model optimal in conducting data training. Then, the second method could be focused on finding the best parameters for dense value and dropout on the fully connected layer with the best optimizer that has been obtained from the first method trial. The selection of the value of dense value and the arrangement on the fully connected layer containing 2 dense and 2 dropouts is based on research conducted by Harikrishnan et al. [11]. Table 1 contains details of the parameters for use in the hyperparameter tuning process:

TABLE I
DETAIL PARAMETER FOR HYPERPARAMETER TUNING PROCESS

No	Parameters	Value
1	Optimizer	Adam, Adamax, SGD, RMSprop
2	Dropout	0.25, 0.5
3	Dense Layer	1024, 2048

E. Purposed Architecture Model

At this stage, the proposed architect model design is the EfficientNet-B7 model. In the implementation stage, the initial layer is an input layer with a size of 224x224 pixels that has been adapted to the EfficientNet-B7 model. The second layer is filled by the EfficientNet-B7 model with a 'Noisy-Student' weight. The 'Noisy-Student' weight is a form of development rather than the 'ImageNet' weight and achieves a 2% better result [23]. The next layer implemented Global Average Pooling to overcome overfitting, then continued by a fully connected layer with 2 dense layers, 2 dropout layers, and an output layer with SoftMax activation for five classification classes. Hyperparameter Tuning is implemented to provide the best parameters for optimizers, dense layers, and dropout layers. In the design of this architecture model dense layer value, the dropout layer is determined based on the results obtained from hyperparameter tuning depending on the location of each layer, as shown in Fig. 7.

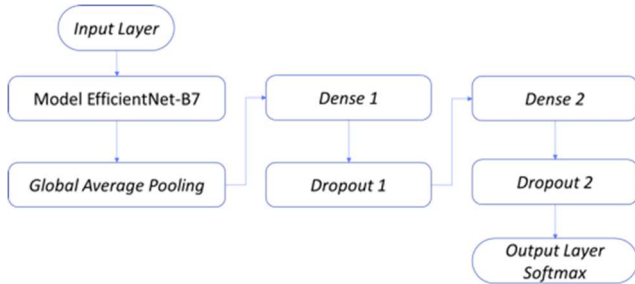


Fig. 7 Model Architecture Design Details

III. RESULT AND DISCUSSION

At this stage, testing is conducted using three scenarios to determine the effect of preprocessing techniques and augmentation techniques for the proposed EfficientNet-B7 model. Testing in this scenario was tested with diabetic retinopathy dataset with optimizer obtained from hyperparameter tuning process, training epoch of 50, batch size of 12, a learning rate of 1e-3 (0.001), and loss in the form of "categorical_crossentropy". In the fourth test, scenarios were also implemented Learning Rate Scheduler and ReduceLRonPlateau, aiming to reduce overfitting by stabilizing the learning rate condition at the time of training. The test results of each scenario could be compared in terms of accuracy, loss, precision, recall, and f1-score.

A. Hyperparameter Tuning

In this process, hyperparameter tuning is performed using parameters that refer to Table 1. The first tuning uses hyperparameter tuning from the Tensorboard. This process is done to find the best optimizer to be used on the model later. In this process, testing was conducted for models with an

epoch of 5 and a batch size of 12. The result of the first hyperparameter tuning process is in the following Table 2:

TABLE II
HYPERPARAMETER TUNING RESULT IN TENSORBOARD VERSION

No	Optimizer	Accuracy	Loss
1	Adamax	0.6818	0.8564
2	Rmsprop	0.6745	1.1237
3	SGD	0.6200	0.9813
4	Adam	0.6091	1.0650

Based on Table 2 through Tensorboard hyperparameter tuning testing, Adamax accuracy and loss outperformed its 68.1% accuracy and 0.856 loss compared to RMSprop, SGD, and Adam optimizers. Adamax obtained better results when testing this EfficientNet-B7 model regarding accuracy and loss compared to other optimizers.

TABLE III
HYPERPARAMETER TUNING RESULT IN KERAS TUNER VERSION

Optimizer	Dense 1	Drop out 1	Dense 2	Drop out 2	Accuracy
Adamax	2048	0.5	1024	0.25	0.7745
Adamax	2048	0.5	2048	0.5	0.7618
Adamax	1024	0.25	2048	0.5	0.7527

Based on Table 3 through hyperparameter tuning testing of the Keras Tuner, the combination and follow-up parameters for the fully connected layer in the first rank with Adamax optimizer managed to get an accuracy of 77.45%. These parameters are obtained after performing the random search process from Keras Tuner. In the process, when the dense value is 2048, the working dropout value is 0.5, and vice versa; if the dense value is 1024, then the working dropout value is 0.25. In that result, larger dropouts (0.5) could also apply to larger denses, while smaller dropouts (0.25) could apply to smaller dense values. The combination and the best array of parameters in the first place could be implemented against the proposed EfficientNet-B7 model for testing scenarios. Combination and the best array of parameters in the first place could be implemented against the proposed EfficientNet-B7 model for testing scenarios.

B. Scenario 1: Proposed Model with Harikrishnan Preprocessing

In this first scenario, testing could be conducted using the proposed EfficientNet-B7 model. In this process, the dataset has been performed Harikrishnan preprocessing technique. In this scenario, the augmentation process described in Fig. 6 is implemented. The EfficientNet-B7 model is implemented with the parameters obtained from the tuning process in Table 3. The following plot graph results in the first test scenario can be seen in Fig. 8.

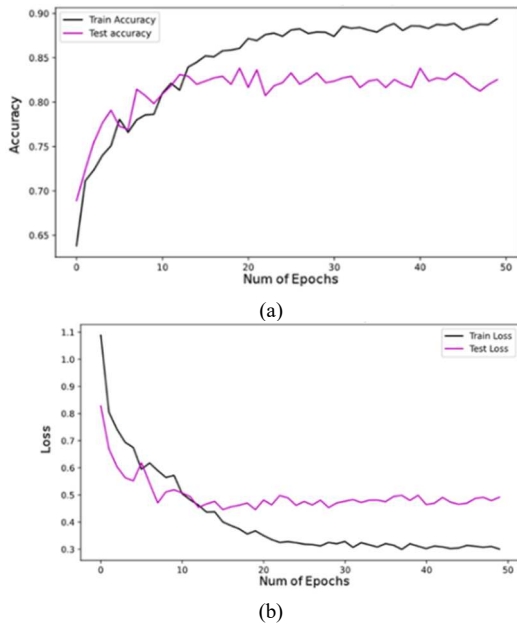


Fig. 8 Result of 1st Scenario with Harikrishnan Preprocessing (a) Graph of 1st Scenario Accuracy and (b) Graph of 1st Scenario Loss

C. Scenario 2: Proposed Model with Ben Graham Preprocessing

In this second scenario, testing could be conducted using the EfficientNet-B7 model with datasets that Graham [17] preprocessing techniques have performed. In this scenario, the augmentation process has been described in Fig. 6. The EfficientNet-B7 model is implemented with the parameters obtained from the hyperparameter tuning process in Table 3. The following plot graph results in the second test scenario can be seen in Fig. 9.

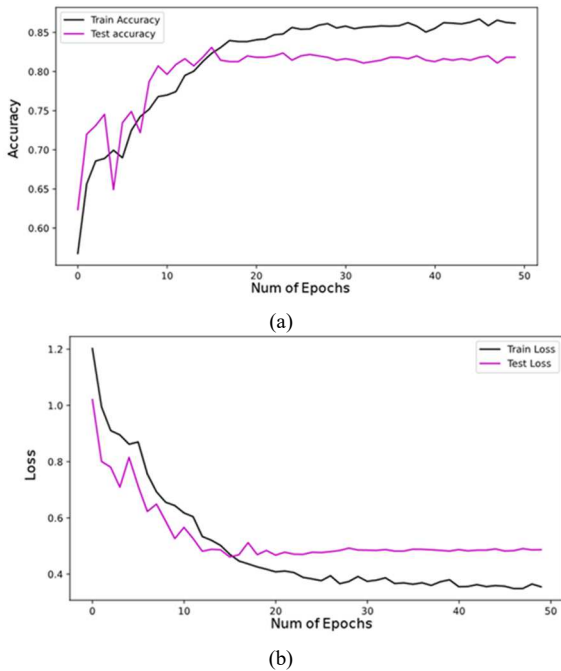


Fig. 9 Result of 2nd Scenario with Ben Graham Preprocessing (a) Graph of 2nd Scenario Accuracy and (b) Graph of 2nd Scenario Loss

D. Scenario 3: Proposed Model with Usuyama Preprocessing

In this third scenario, testing could be conducted using the EfficientNet-B7 model with a dataset that has been implemented Usuyama [19] preprocessing technique. In this scenario, the augmentation process has been described in Fig 6. The EfficientNet-B7 model is implemented with the parameters obtained from the hyperparameter tuning process in Table 3. The following plot graph results in the third test scenario can be seen in Fig. 10.

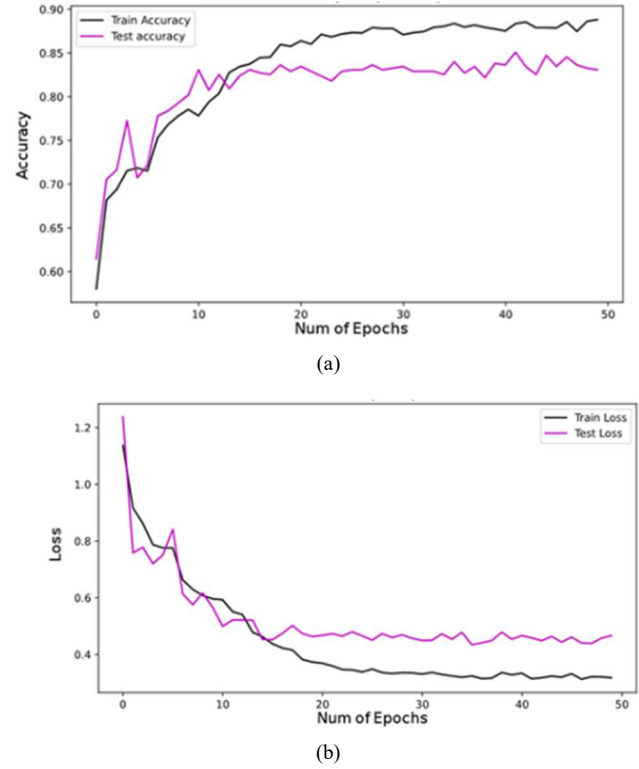


Fig. 10 Result of 3rd Scenario with Usuyama Preprocessing (a) Graph of 3rd Scenario Accuracy and (b) Graph of 3rd Scenario Loss

E. Compare and Analysis Model

This study proposed several scenario models with different preprocessing methods and the effect of implementing the augmentation process run on the EfficientNet-B7 model. The 3rd test scenario was the best compared to other scenarios in terms of accuracy of 84%, loss of 0.43, the precision of 83%, recall of 84%, and F1-score with a value of 83%. This scenario proves that the proposed model's use of Usuyama [19] preprocessing techniques and augmentation implementations can improve performance.

In test 1st and 2nd scenarios, the results obtained have little different than the 3rd test scenario, but if tested with a higher epoch, the results are no better than the 3rd test scenario. Table 4 shows the comparison among previous research results with research conducted on the dataset diabetic retinopathy APTOS 2019.

TABLE II
SUMMARY OF RESEARCH COMPARISON ON DIABETIC RETINOPATHY DATASET

Method (Researcher)	Train		Test		Weighted Avg		
	Acc	Loss	Acc	Acc	Precision	Recall	F1-Score
RNN (Harikrishnan et al [11])	85		82				
Scenario 1 + Preprocessing (Harikrishnan et al. [11])	89.72	27.76	82.36	49.76	83.61	83.45	83.33
Scenario 2 + Preprocessing (Graham [17])	87.53	32.97	81.82	48.67	82.87	81.82	81.77
Scenario 3 + Preprocessing (Usuyama [19])	89.11	30.50	84.36	43.32	83.17	83.64	82.79

Harikrishnan et al. [11] conducted previous research using the diabetic Retinopathy APTOS 2019 dataset using RNN models. The research conducted by Harikrishnan et al. [11] uses the proposed preprocessing technique Gaussian blur method with a standard deviation of 10 without a crop circle. The results obtained in the study amounted to 85% for train data and 82% for test data.

The current research proposed several preprocessing scenarios to determine the effect of preprocessing with a proposed model named EfficientNet-B7. This study determines the effect of augmentation on the proposed model. In the 1st test, the scenario performed the same preprocessing technique by Harikrishnan et al. [11] with a different architectural model, Efficientnet-B7. The accuracy obtained exceeded, but in 82.36% accuracy. Previous studies did not achieve accuracy in the 2nd test scenario using Graham [17] preprocessing technique. In the 3rd Scenario testing by applying preprocessing technique, Usuyama [19] added that the augmentation process successfully exceeded the research conducted by Harikrishnan et al [11] with an accuracy of 89.11% in train data and 84.36% in test data. This proves that preprocessing techniques give different influences for models in classifying images. This study found that the Usuyama [19] preprocessing technique provides the best performance for the proposed model.

IV. CONCLUSION

This study proposed an EfficientNet-B7 model with a combination of parameters produced by hyperparameter tuning, preprocessing techniques, and augmentation techniques to classify diabetic retinopathy disease in the APTOS 2019 dataset. Not all preprocessing can impact or influence the proposed model. Based on the three preprocessing techniques performed, namely Harikrishnan et al. [11], Graham [17], and Usuyama [19], testing on Usuyama [19] preprocessing techniques provided the best performance compared to the other two preprocessing techniques. Usuyama preprocessing techniques provide better image input quality when trained using EfficientNet-B7 models and augmentation processes. The technique obtained an accuracy score of 89.1% on the data train and 84.36% on the test data. Further research is expected to propose other preprocessing techniques such as enlarged image size, image enhancement, or CLAHE implementation.

ACKNOWLEDGMENT

The authors thank the Department of Informatics of Muhammadiyah Malang University for supporting research on "Diabetic Retinopathy Classification."

REFERENCES

- [1] S. Sridhar and S. Sanagavarapu, "Detection and Prognosis Evaluation of Diabetic Retinopathy using Ensemble Deep Convolutional Neural Networks," *IES 2020 - Int. Electron. Symp. Role Auton. Intell. Syst. Hum. Life Conf.*, pp. 78–85, 2020, doi: 10.1109/IES50839.2020.9231789.
- [2] S. Sheikh and U. Qidwai, "Using MobileNetV2 to Classify the Severity of Diabetic Retinopathy," *Int. J. Simul. Syst. Sci. Technol.*, pp. 1–6, 2020, doi: 10.5013/ijssst.a.21.02.16.
- [3] H. Liu, K. Yue, S. Cheng, C. Pan, J. Sun, and W. Li, "Hybrid model structure for diabetic retinopathy classification," *J. Healthc. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/8840174.
- [4] B. Tymchenko, "Deep Learning Approach to Diabetic Retinopathy Detection," 2019.
- [5] W. Zhang *et al.*, "Automated identification and grading system of diabetic retinopathy using deep neural networks," *Knowledge-Based Syst.*, vol. 175, pp. 12–25, 2019, doi: 10.1016/j.knosys.2019.03.016.
- [6] R. Agarwal, A. Mahamuni, N. Gautam, P. Awachar, and P. Sagar, "Deep Learning-Based Grading of Diabetic Retinopathy Using Semantic Segmentation," no. 4, pp. 4–6, 2020.
- [7] L. Qiao, Y. Zhu, and H. Zhou, "Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms," *IEEE Access*, vol. 8, pp. 104292–104302, 2020, doi: 10.1109/ACCESS.2020.2993937.
- [8] Z. Gao, J. Li, J. Guo, Y. Chen, Z. Yi, and J. Zhong, "Diagnosis of Diabetic Retinopathy Using Deep Neural Networks," *IEEE Access*, vol. 7, pp. 3360–3370, 2019, doi: 10.1109/ACCESS.2018.2888639.
- [9] A. Kwasigroch, B. Jarzembinski, and M. Grochowski, "Deep CNN based decision support system for detection and assessing the stage of diabetic retinopathy," *2018 Int. Interdiscip. PhD Work. IIPHDW 2018*, pp. 111–116, 2018, doi: 10.1109/IIPHDW.2018.8388337.
- [10] E. Abdelmaksoud, S. Barakat, and M. Elmogy, "Diabetic retinopathy grading system based on transfer learning," *Int. J. Adv. Comput. Res.*, vol. 11, no. 52, pp. 1–12, 2021, doi: 10.19101/ijacr.2020.1048117.
- [11] V. K. Harikrishnan, M. Vijarana, and A. Gambhir, *Diabetic retinopathy identification using autoML*. Elsevier Inc., 2020.
- [12] K. Shankar, Y. Zhang, Y. Liu, L. Wu, and C. H. Chen, "Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification," *IEEE Access*, vol. 8, pp. 118164–118173, 2020, doi: 10.1109/ACCESS.2020.3005152.
- [13] S. S. Chaturvedi, K. Gupta, V. Ninawe, and P. S. Prasad, "Automated Diabetic Retinopathy Identification Using Convolutional Neural Network," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 3, pp. 1872–1882, 2021, doi: 10.30534/ijatcse/2021/541032021.
- [14] S. Jayabalan, P. S. Pratheeksha, N. S. Bolar, and N. L. Malavika, "Prediction of Diabetic Retinopathy Using SVM Algorithm," vol. 7, no. 14, pp. 1702–1711, 2020.
- [15] W. L. Alyoubi, M. F. Abulkhair, and W. M. Shalash, "Diabetic retinopathy fundus image classification and lesions localization system using deep learning," *Sensors*, vol. 21, no. 11, pp. 1–22, 2021, doi: 10.3390/s21113704.
- [16] G. Mushtaq and F. Siddiqui, "Detection of diabetic retinopathy using deep learning methodology," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1070, p. 012049, 2021, doi: 10.1088/1757-899x/1070/1/012049.
- [17] B. Graham, "Kaggle Diabetic Retinopathy Detection competition report," pp. 1–9, 2015.
- [18] Rattthachat Chatpatanasiri, "APTOS: Eye Preprocessing in Diabetic Retinopathy." <https://www.kaggle.com/rattthachat/aptos-eye-preprocessing-in-diabetic-retinopathy> (accessed Jul. 17, 2021).
- [19] J. L. Naoto Usuyama, "Medical image processing using Microsoft Deep Learning Framework (CNTK)," pp. 1–68, 2017.

- [20] C. Bhardwaj, S. Jain, and M. Sood, "Diabetic retinopathy severity grading employing quadrant-based Inception-V3 convolution neural network architecture," *Int. J. Imaging Syst. Technol.*, no. September, pp. 1–17, 2020, doi: 10.1002/ima.22510.
- [21] 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, doi: 10.3390/molecules22122054.
- [22] W. Y. Lee, S. M. Park, and K. B. Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," *Optik (Stuttg.)*, vol. 172, no. July, pp. 359–367, 2018, doi: 10.1016/j.ijleo.2018.07.044.
- [23] Q. Xie, M. T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 10684–10695, 2020, doi: 10.1109/CVPR42600.2020.01070.