Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 10 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i10.48603

PAPER

e-LSTM: EfficientNet and Long Short-Term Memory Model for Detection of Glaucoma Diseases

Wiharto[1](✉), Wimas Tri Harjoko[1], Esti Suryani[2]

[1]Department of Informatics, Universitas Sebelas Maret, Surakarta, Indonesia

[2]Department of Data Science, Universitas Sebelas Maret, Surakarta, Indonesia

wiharto@staff.uns.ac.id

ABSTRACT

Glaucoma is an eye disease that often has no symptoms until it is advanced. According to the World Health Organization (WHO), after cataracts, glaucoma is the second-leading cause of permanent blindness globally and is expected to affect 111.8 million patients by 2040. Early detection of glaucoma is important to reduce the risk of permanent blindness. Detection is achieved by structural measurement of early thinning of the retinal nerve fiber layer (RNFL). The RNFL is the portion of the retina located outside the optic nerve head (ONH) and can be observed in fundus images of the retina. Analysis of retinal fundus images can be performed with computer assistance using machine learning, especially deep learning. This study proposes a deep learning-based model, a convolutional neural network (CNN) using the EfficientNet architecture combined with long short-term memory (LSTM), for laucoma detection. Using ACRIMA, DRISHTI-GS, and RIM-ONE DL datasets with k-fold cross-validation, the model achieved high performance on the ACRIMA dataset: accuracy 0.9799, loss 0.0596, precision 0.9802, sensitivity 0.9799, specificity 0.9771, and F1score 0.9799. This EfficientNet and LSTM combination (e-LSTM) outperformed previous studies, offering a promising alternative for evaluating retinal fundus images in glaucoma detection.

KEYWORDS

glaucoma, retina, deep learning, EfficientNet, long short-term memory (LSTM)

1 INTRODUCTION

The eye is one of the five sensory organs in humans that is very vital. In general, the eye has a function to recognize an object that is around. Eye disorders often occur without any symptoms. One visual disorder that tends to have no symptoms until it reaches an advanced stage is glaucoma [1].

Glaucoma is a neurodegenerative disorder of the eye caused by increased intraocular pressure on the optic nerve [2]. Data from the World Health Organization (WHO) suggests that glaucoma cases in recent years have increased and become

Wiharto, Harjoko, W.T., Suryani, E. (2024). e-LSTM: EfficientNet and Long Short-Term Memory Model for Detection of Glaucoma Diseases. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(10), pp. 64–85. https://doi.org/10.3991/ijoe.v20i10.48603

Article submitted 2024-02-17. Revision uploaded 2024-04-25. Final acceptance 2024-04-26.

the second cause of permanent blindness in the world after cataracts [3]. In 2020, glaucoma patients worldwide are expected to increase from 64 million to 76 million people [4] and are predicted to continue to increase to 111.8 million people in 2040 [5]. From this data, it is undeniable that eye examinations are needed for the early detection of glaucoma. Detection is done by taking structural measurements in the form of retinal nerve fiber layer (RNFL) thinning in the early stages. The RNFL is the part of the retina that lies on the outside of the optic nerve head (ONH), which can be observed in retinal fundus images. According to [6], developing an automatic glaucoma detection system using retinal fundus images is considered to be able to save costs when compared to retinal imaging technologies that are quite expensive, such as Heidelberg retina tomography (HRT) and optical coherence tomography.

Extensive research has been carried out on the detection of various diseases by analyzing retinal fundus images, and it has been largely computer-based, especially by using machine learning techniques. For instance, studies [7] and [8] used fundus images to detect diabetic retinopathy, which is an eye disease caused by diabetes. In another study, researchers [9] and [10] detected glaucoma by calculating the cup disc ratio (CDR) on the retinal fundus image, which refers to the ISNT rule (inferior, superior, nasal, temporal). The research conducted [11] classifies glaucoma by segmenting the optic disk before it is classified. Detection of glaucoma using CDR, ISNT rules, or pre-processing data such as segmentation can indeed be done, but it takes a lot of time and is not efficient [12].

In [13], a system capable of classifying glaucoma and non-glaucoma based on deep learning CNN with an accuracy of 87.6% was built and was able to outperform the accuracy of ophthalmologists and traditional methods such as advanced glaucoma intervention (AGIS) and glaucoma staging system 2 (GSS2), which only resulted in an accuracy of 45.9% and 52.3%, respectively. Research conducted by [14] using image processing and classification methods with pre-trained Deep CNN architecture models (GoogleNet, VGG, and ResNet), used to detect glaucoma, was able to achieve accuracy of 83.40%, 83.73%, and 85.56%. On the other hand, the EfficientNet deep learning model is used in the classification of other medical imaging problems, such as in research [15], to diagnose COVID-19 and pneumonia through X-ray images, with an accuracy value of 96.7%.

EfficientNet can produce promising performance, as seen in research conducted by Toptaş and Hanbay [16] and Marques et al. [15]. By utilizing compound scaling techniques, EfficientNet can overcome important factors in deep learning, namely computational efficiency and model performance [17]. The efficiency is obtained by balancing scaling in the dimensions of width, depth, and resolution. Therefore, the model size of EfficientNet is relatively smaller, with fewer parameters but good model performance.

Deep learning algorithms can significantly improve the performance of image classification in the feature extraction process. Research [18] combined deep learning CNN and LSTM algorithms in the case of coronavirus detection from X-ray images, which resulted in an accuracy of 99.4%. Research conducted in [19] related to brain tumor classification also uses the deep learning CNN model VGG-16 in the feature extraction process, which is then incorporated into LSTM with 100 units to help learn high-level features from the data. In addition, LSTM is also able to cover the shortcomings of the fully connected layer, where the network is fully connected and the nodes between the layers only process on one input, while LSTM can connect the nodes in a graph, which is considered an input [20].

Therefore, the combination of deep learning and LSTM can improve the classification performance of the system [21]. Referring to several previous studies, this study proposes a deep learning CNN model with efficientNet architecture combined with LSTM, hereinafter referred to as e-LSTM, for glaucoma detection. The e-LSTM model aims to improve the fully connected layer in the classification process and help learn high-level features from the data. Retinal fundus photos from several publicly accessible online datasets were utilized as the data to be tested on the proposed model.

2 MATERIALS AND METHODS

The research method used in this study can be seen in Figure 1. It shows many stages, including preprocessing, building the e-LSTM model architecture, training and tuning the hyperparameters, and evaluating model performance.

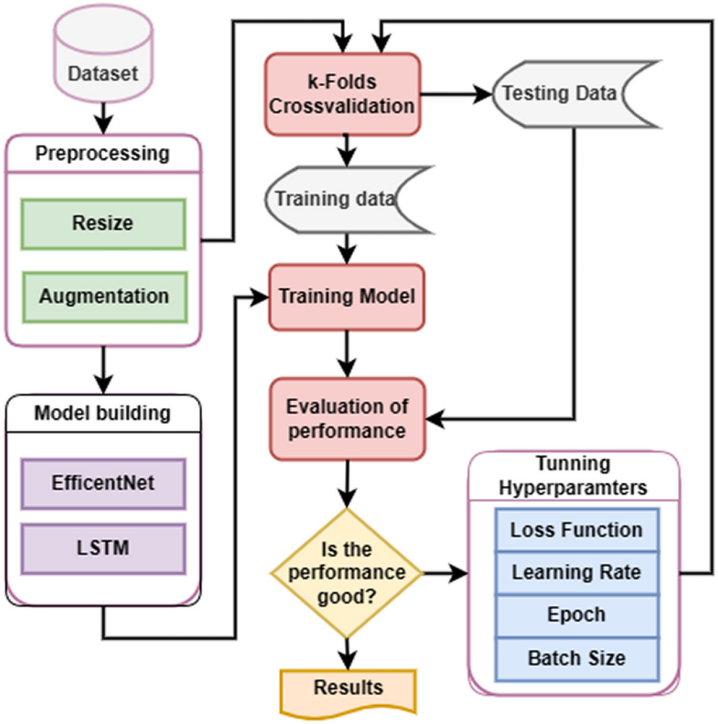

Fig. 1. Research flow

2.1 Image fundus retina dataset

Some of the datasets that will be used in this study are publicly available. The data are retinal fundus images labeled as glaucoma and normal, which can be used to evaluate the glaucoma disease detection model. Figure 2 is a sample of fundus image data from each dataset consisting of two classes. The retinal fundus images were obtained through three datasets available online, namely ACRIMA and DHRISTI-GS, obtained through https://www.kaggle.com/datasets/sshikamaru/glaucoma-detection, while the RIM-ONE DL dataset was obtained through https://github.com/miag-ull/rim-one-dl.

Fig. 2. The first row is normal fundus images and the second row is fundus images with glaucoma from ACRIMA, DRISHTI-GS, and RIM-ONE DL datasets

2.2 Data preprocessing

In the previous stage, we collected retinal fundus image data that had different sizes. To build a model using EfficientNet-B0, we need to standardize the input size to 224 × 224 pixels. Therefore, we will equalize the size of the fundus image to facilitate the model-building stage. This will ensure that the pixel size is consistent throughout the model, making it easier to analyze the data.

Deep learning is more successful in big data, but limited datasets require data augmentation. Furthermore, augmentation is added to increase the training images and minimize overfitting. Augmentation is applied in the form of rotation by 90 degrees clockwise and counterclockwise, then a rotation of 180 degrees and mirroring vertically and horizontally. The amount of data after augmentation is shown in Table 1, while the following illustration of augmentation can be seen in Figure 3.

Table 1. The number of images after augmentation

Dataset		Original	Data Augmentation
ACRIMA	Glaucoma	396	2376
	Normal	309	1854
DRISHTI-GS	Glaucoma	70	420
	Normal	31	186
RIM-ONE DL	Glaucoma	172	1032
	Normal	313	1818

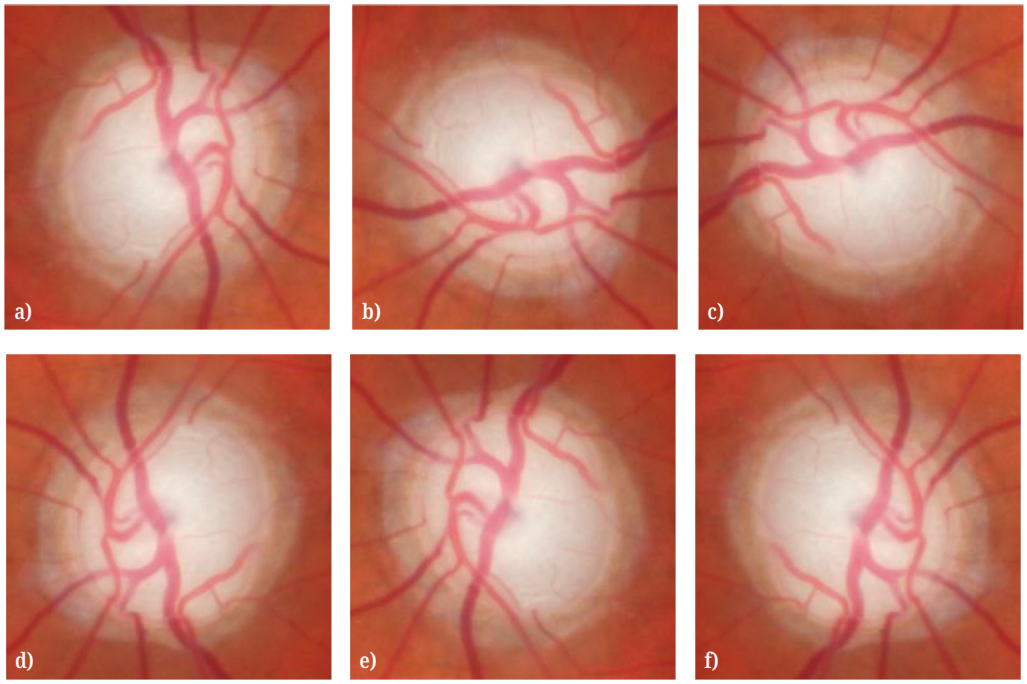

Fig. 3. Augmentation illustration (a) Original image (b) Rotation 90° to the Right (c) Rotation 90° to the Left (d) 180° Rotation (e) Vertical flip (f) Horizontal flip

2.3 Model building

In this stage, a model is created that can identify glaucoma disease from the retinal fundus image that has been collected. The model will be compiled using one of the CNN architectures, EfficientNet, combined with LSTM. In the process, the EfficientNet used is EfficientNet-B0. EfficientNet-B0 is used to extract complex features owned by each fundus image. The next stage is LSTM, with a size of 100 units, which corresponds to the number of features useful for assisting the fully connected layer in the classification process. The use of LSTM can provide support in providing memory to store the features that have been obtained. Figure 4 is an illustration of the architecture created.

1. **EfficientNet:** EfficientNet is a popular CNN architecture that is widely utilized for tasks such as image classification and object recognition. By applying compound scaling techniques, EfficientNet can balance two important factors in deep learning, namely computational efficiency and model accuracy [17]. Compound scaling is a technique to balance the three scaling dimensions of width, depth, and resolution. The mathematical formula for compound scaling is given in Equations 1–3.

$$w = \beta^{\phi}, d = \alpha^{\phi}, r = \gamma^{\phi} \tag{1}$$

$$s.t\ \alpha.\ \beta^2.\ \gamma^2 \approx 2 \tag{2}$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{3}$$

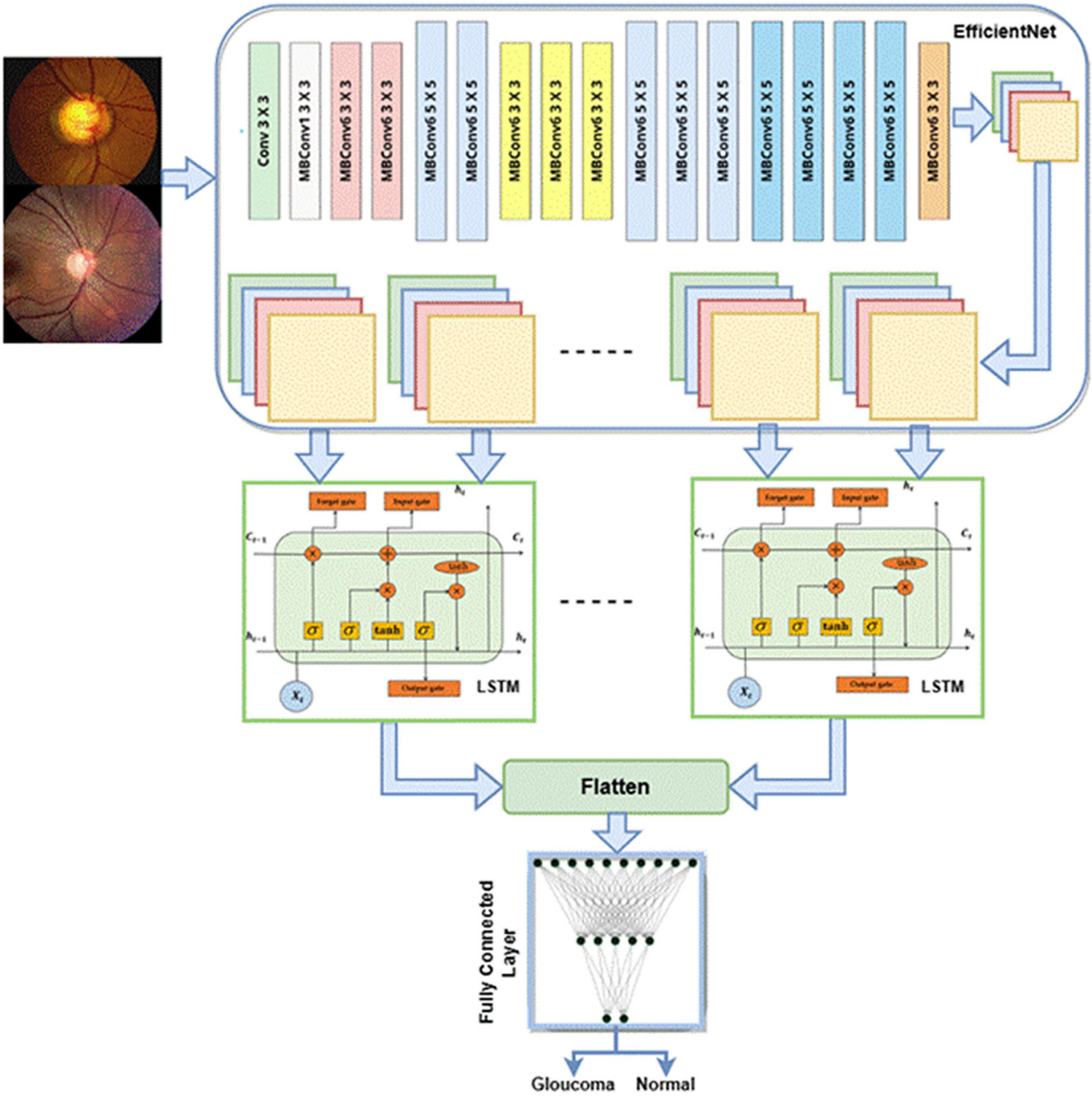

Fig. 4. e-LSTM architecture model

The optimal combination of width, depth, and resolution scaling is obtained by conducting a systematic grid search. This search is denoted by 'phi' (φ), which is a compound coefficient. The value of φ is determined by the user to scale the overall model dimensions. While α, β, and γ are constants that represent the dimensions of width, depth, and resolution, the width scaling of compound scaling refers to the number of channels in each layer of the neural network. The depth scaling is related to the total number of layers in the network. Resolution scaling involves adjusting the size of the input image that will be used in the model. Table 2 is the basic network of EfficientNet.

Table 2. EfficientNet baseline network

Step	Operator	Resolution	Channels	Layers
1	Conv3×3	224×224	32	1
2	MBConv1, k3×3	112×112	16	1
3	MBConv6, k3×3	112×112	24	2
4	MBConv6, k5×5	56×56	40	2
5	MBConv6, k3×3	28×28	80	3
6	MBConv6, k5×5	14×14	112	3
7	MBConv6, k5×5	14×14	192	4
8	MBConv6, k3×3	7×7	320	1
9	Conv1×1 and Pooling and FC	7×7	1280	1

In Table 2, the main underlying network of EfficientNet is composed of a mobile inverted bottleneck (MBConv) [22], which is also a module used in the MobileNetV2 architecture. The MBConv layer combines depthwise separable convolution and inverted residual blocks that are optimized using squeeze-and-excitation (SE) to enhance the performance of the model. Figure 5 illustrates the architecture of a mobile inverted bottleneck.

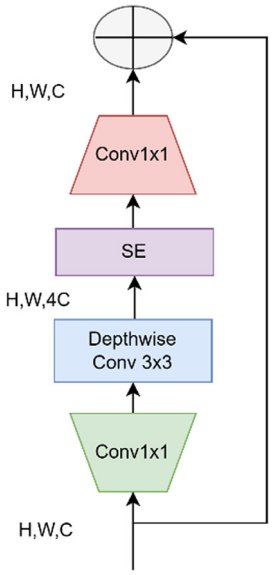

Fig. 5. MBConv architecture

Depthwise separable convolution is a type of convolution that reduces computation and the number of parameters in the model [23]. This technique is more efficient than traditional convolution, with a computational amount that is 8–9 times lighter. Figure 6 illustrates depthwise separable convolution.

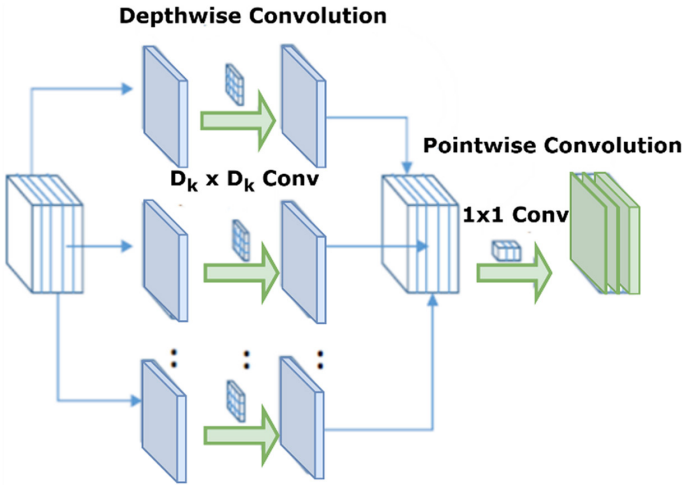

Fig. 6. Depthwise separable convolution

Inspired by the residual block in [24], the inverted residual block is a kind of block used in neural networks. The residual block in ResNet is designed to map inputs with a wide range of channels, first narrowing them in the inner layer and then widening them again at the output layer. However, the inverted residual block takes the opposite approach: it starts with a narrow input channel, then widens it in the inner layer, and finally narrows it again at the output layer. As a result, the inverted residual block has fewer parameters than the regular residual block. The differences between the two blocks can be observed in Figure 7.

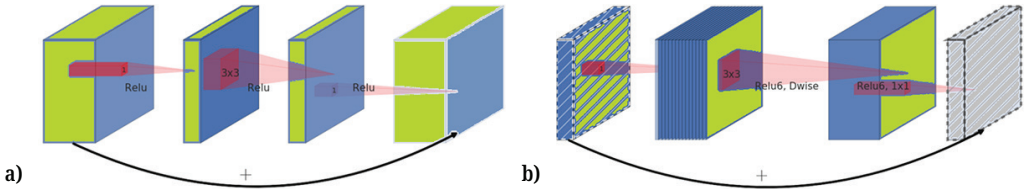

Fig. 7. (a) Residual block (b) Inverted residual block

2. **Long short-term memory:** Long short-term memory is an architecture developed from a recurrent neural network (RNN). This architecture is intended to overcome the vanishing gradient problem that causes the difficulty of processing a lot of data, commonly referred to as long-term dependencies [25]. LSTM can store and connect information that has been obtained in previous data with data obtained at this time [26]. The architecture of LSTM adds three gates to the cell that is built, namely the input gate, output gate, and forget gate. Figure 4 shows the structure of a long short-term memory.

   The addition of a forget gate in LSTM allows for resetting of the internal memory once the stored information is deemed unnecessary. This reduces the load the memory. Equations 7–8 outline the working principle of long short-term memory.

$$ft = \sigma(W(f)xt + V(f)ht - 1 + bf) \tag{4}$$

$$it = \sigma(W(i)xt + V(i)ht - 1 + bi) \tag{5}$$

$$ot = \sigma(W(o)xt + V(o)ht - 1 + bo) \tag{6}$$

$$ct = ft \otimes ct - 1 + it \otimes tanh(W(c)xt + V(c)ht - 1 + bc) \quad (7)$$

$$ht = ot \otimes tanh(ct) \quad (8)$$

To update information about the cell state, the process includes several steps. These steps are: (1) discarding any irrelevant information obtained from the previous step's state; (2) extracting important information and adding it to the state; (3) calculating the state unit; and (4) calculating the output of the current step.

2.4 Tuning hyperparameter

To find good training conditions, hyperparameter tuning is carried out on the model created. Usually, the hyperparameter tuning used is the value of batch size, learning rate, epoch, activation function, loss function, etc. In this study, we will find the best hyperparameters used in the model for batch size, learning rate, loss function, and epoch. In addition, a 10-fold cross-validation technique is also applied, where the data will be cross-tested as a reference to determine the best hyperparameters.

2.5 Training model

After undergoing preprocessing, the data will move on to the model training stage. Training will be run using Google Collab with GPU. Four training schemes will be conducted, where each dataset will be used as input to the model. In addition, ACRIMA, DHRISTI-GS, and RIM-ONE DL data will also be combined to compare the performance of the model. Model training is useful for extracting characteristics from each data point with each label. Training will store the weight values that will be used in the classification stage.

2.6 Evaluation result

The next process is the evaluation of the built model. The evaluation is carried out to compare the classification of the EfficientNet model added by LSTM with the EfficientNet model alone. The assessment of this evaluation will consider the performance of the model in terms of average accuracy, precision, sensitivity, specificity, and the F1 score obtained from 10-fold cross-validation. The performance parameters are calculated by referring to the confusion matrix. A confusion matrix is a table that displays the predicted and actual data shown in Figure 8. It is used to measure the accuracy, precision, sensitivity, specificity, and F1 score of a model, which consists of the following components:

a) True positive (TP): If the fact is positive for glaucoma, the e-LSTM model is also detected as positive for glaucoma.
b) True negative (TN): If the patient is negative for glaucoma, the e-LSTM model also detects negative.
c) False positive (FP): If the patient has negative glaucoma, the e-LSTM model detects positive glaucoma.
d) False negative (FN): If the patient has positive glaucoma, the e-LSTM model detects negative glaucoma.

Referring to the values of *TP*, *TN*, *FP*, and *FN*, the parameters used for evaluating the performance of the e-LSTM model can be written in a formula, as shown in Equations 9–13.

$$Accuracy = ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{9}$$

$$Specificity = SPE = \frac{TN}{(TN + FP)} \tag{10}$$

$$Sensitivity = SEN = \frac{TP}{(TP + FN)} \tag{11}$$

$$Precision = PRE = \frac{TP}{(TP + FP)} \tag{12}$$

$$F1 - Score = F1 = \frac{2 \times TP}{(2 \times TP + FP + FN)} \tag{13}$$

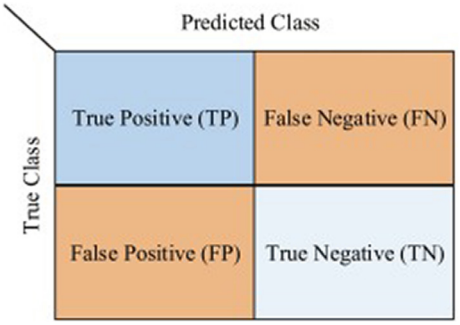

Fig. 8. Confusion matrix

The accuracy of a model refers to its ability to make correct predictions from all available data. However, the accuracy value may not be valid if the tested data is unbalanced. To test for unbalanced data, precision, sensitivity, and specificity metrics are used. Precision measures the level of TP predictions in comparison to all predicted positive data. Sensitivity measures the accuracy of TP predictions in comparison to all actual positive data. Specificity, on the other hand, is the measure of TNs in comparison to all actual negative data. To calculate the accuracy level of the model, the F1 score is used by combining precision and sensitivity with the ideal.

3 RESULTS

This study is built using the Python 3 programming language and the TensorFlow framework on the Google Collaboratory Pro platform with a GPU accelerator. In addition, hyperparameter tuning will be done to get the best performance from the built model. Then, the results obtained from the experiments will be analyzed.

3.1 Loss function experiments

In the first experiment, we will test the effect of the loss function on the performance of the EfficientNet-B0 LSTM model. The values that will be used as a reference are accuracy, loss, precision, sensitivity, specificity, and F1 score. This experiment will test three loss functions, namely binary cross entropy (BCE), mean squared error (MSE), and mean absolute error (MAE), by applying the 10-fold cross-validation testing technique to take the average of the performance matrix. Table 3 shows the hyperparameters used in this test.

Table 3. Hyperparameter for loss function experiments

Hyperparameter	Value
Batch Size	16
Optimizer	Adam
Learning Rate	Reduce Learning Rate
Loss Function	**BCE, MSE, and MAE**
Activation Function	ReLu, Sigmoid
Unit LSTM	100
Epoch	50

The results shown in Table 4 indicate that using the BCE loss function achieves the best value in terms of average accuracy, precision, sensitivity, specificity, and F1 score. However, on the loss value, using the MSE or MAE loss function produces a relatively smaller value. Figures 9 and 10 are comparison graphs of the use of loss functions against accuracy and loss values, where the BCE loss function can achieve the highest accuracy when compared to other loss functions. But for the loss value, MSE produces the minimum value when compared to other loss function values. Therefore, the next experiment will use the BCE loss function since BCE can outperform MSE in other performance matrices such as precision, sensitivity, specificity, and F1 score.

Table 4. Loss function experiment results

Performance Matrix	Loss Function		
	BCE	MSE	MAE
Accuracy	**0.9709**	0.9662	0.9603
Loss	0.0857	**0.0257**	0.0422
Precision	**0.9715**	0.9668	0.9611
Sensitivity	**0.9709**	0.9662	0.9603
Specificity	**0.9659**	0.9599	0.9515
F1 Score	**0.9710**	0.9662	0.9604

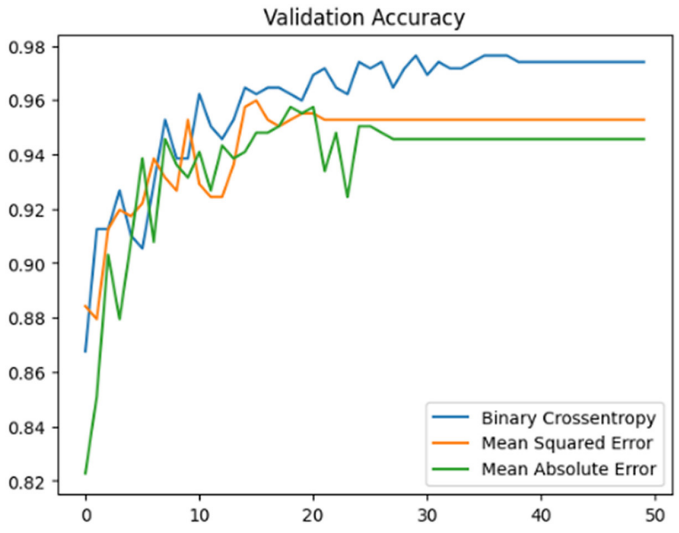

Fig. 9. Comparison graph of loss function on accuracy value

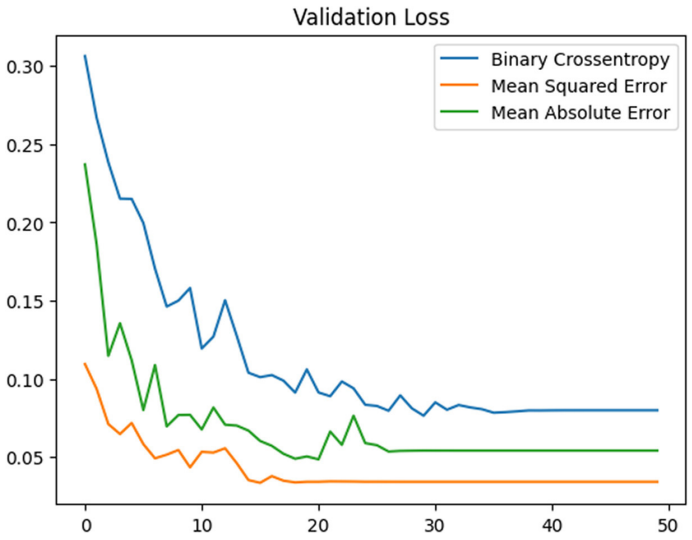

Fig. 10. Comparison graph of loss function on loss value

3.2 Learning rate and epoch experiments

In the previous experiment, a reduced learning rate was used, where the learning rate would decrease when the loss value did not change during training. This experiment will test the constant learning rate and epoch value on the performance of the EfficientNet-B0 LSTM model. The values that will be used as a reference are accuracy, loss, precision, sensitivity, specificity, and F1 score. This experiment will use the 10-fold cross-validation technique and compare several constant learning rate values, namely 0.001, 0.0001, and 0.00001, with the number of epochs 20, 35, 50, and 65. Table 5 is the hyperparameter used for the experiments conducted.

Table 5. Hyperparameter for learning rate and epoch experiments

Hyperparameter	Value
Batch Size	16
Optimizer	Adam
Learning Rate	**0.001, 0.0001, and 0.00001**
Loss Function	BCE
Activation Function	ReLu, Sigmoid
Unit LSTM	100
Epoch	**20, 35, 50, and 65**

Table 6 shows the results of the experiments that have been conducted. A constant learning rate value of 0.0001 with several epochs of 50 produces an optimal average performance matrix of accuracy, loss, precision, sensitivity, specificity, and F1 score when compared to other combinations of learning rate and epoch. Figures 11 and 12 are the average accuracy and loss comparison graphs for each learning rate used.

Table 6. Learning rate and epoch experiment results

Performance Matrix	Learning Rate		
	0.001	0.0001	0.00001
Accuracy	0.9667	**0.9780**	0.9721
Loss	0.0966	**0.0641**	0.0706
Precision	0.9673	**0.9784**	0.9728
Sensitivity	0.9667	**0.9780**	0.9721
Specificity	0.9671	**0.9740**	0.9687
F1 Score	0.9667	**0.9780**	0.9721

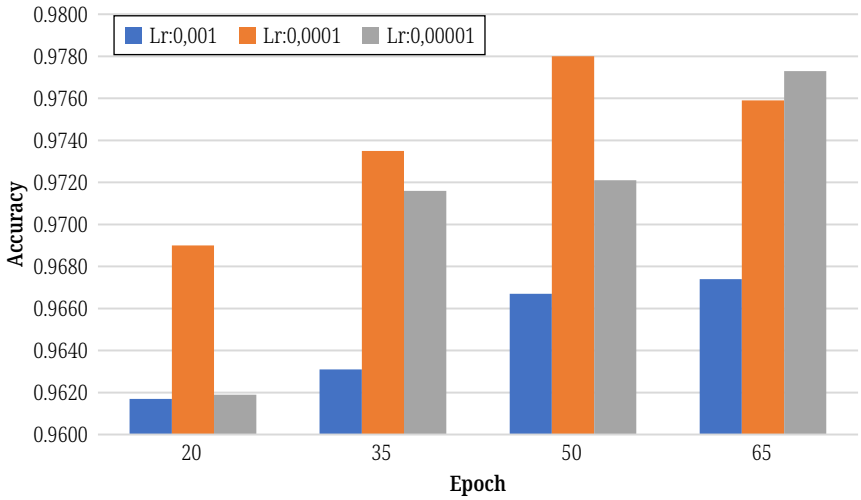

Fig. 11. Comparison of learning rate on accuracy value

Figure 13 is a comparison graph of the average accuracy and loss of the reduced learning rate and constant learning rate of 0.0001 with the loss function BCE and epoch 50. The comparison of the reduced learning rate and constant learning rate with a value of 0.0001 in Figure 9 shows that the use of these two types of learning rates does not increase significantly in terms of average accuracy and loss. However, in the next experiment, a constant learning rate with a value of 0.0001 will be used.

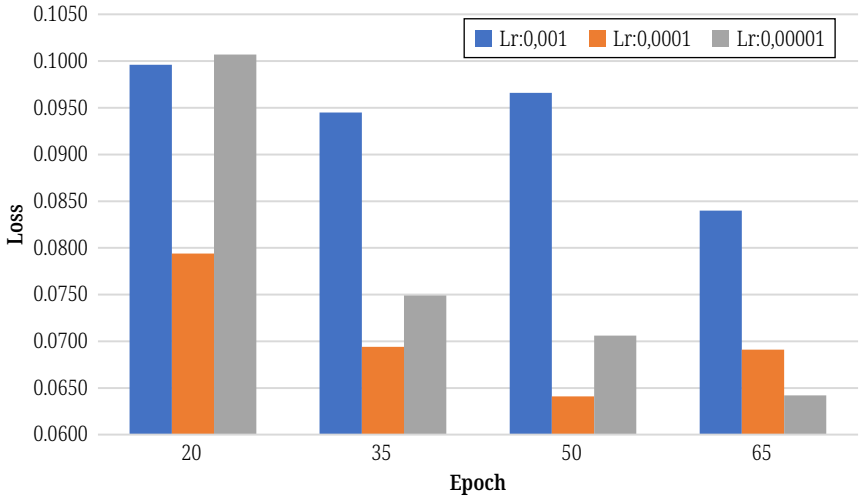

Fig. 12. Comparison of learning rate on loss value

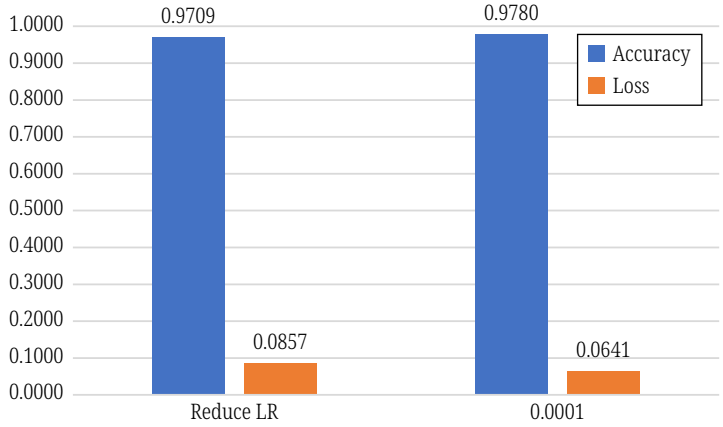

Fig. 13. Comparison of reduced learning rate and constant learning rate 0.0001 on accuracy and loss value

3.3 Batch size experiments

In the next experiment, we will test the effect of the batch size value on the performance of the EfficientNet-B0 LSTM model. The values that will be used as a reference are accuracy, loss, precision, sensitivity, specificity, and F1 score. This experiment will use the 10-fold cross-validation technique and several batch size values, namely 16, 32, and 64. Table 7 below shows the hyperparameters used for the experiments conducted.

Table 7. Hyperparameters for batch size experiments

Hyperparameter	Value
Batch Size	**16, 32, and 64**
Optimizer	Adam
Learning Rate	0.0001
Loss Function	BCE
Activation Function	ReLu, Sigmoid
Unit LSTM	100
Epoch	50

From the experimental results shown in Table 8 and Figure 14, it can be seen that the use of batch size with a value of 32 produces the most optimal results on the average value of accuracy, loss, precision, sensitivity, specificity, and F1 score compared to other batch size values.

Table 8. Batch size experiment results

Performance Matrix	Batch Size		
	16	32	64
Accuracy	0.9740	**0.9799**	0.9688
Loss	0.0748	**0.0596**	0.0842
Precision	0.9748	**0.9802**	0.9704
Sensitivity	0.9740	**0.9799**	0.9688
Specificity	0.9743	**0.9771**	0.9581
F1 Score	0.9740	**0.9779**	0.9689

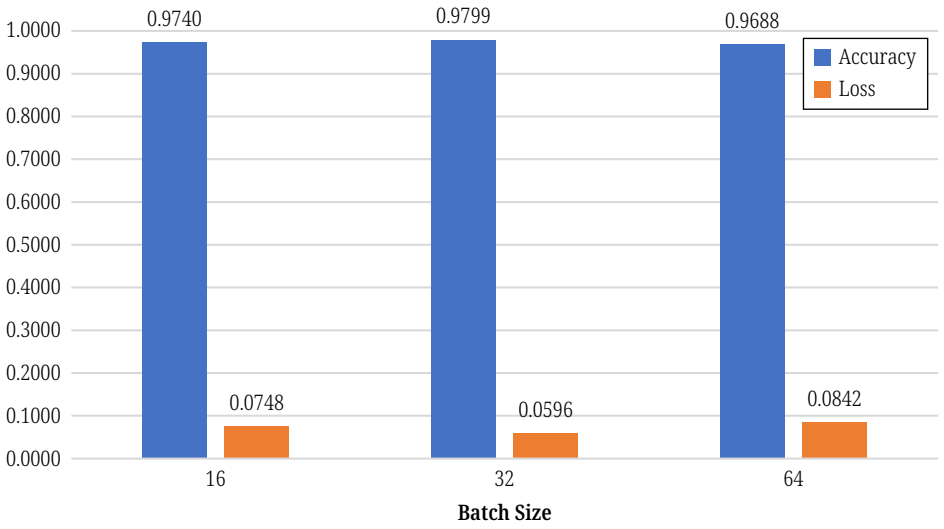

Fig. 14. Comparison of batch size on accuracy and loss value

3.4 Summary of experiment results

All the experiments that have been conducted previously obtained a combination of several hyperparameters, namely loss function, learning rate, epoch, and batch size, to achieve the optimal average performance matrix. The hyperparameter combinations used can be seen in Table 9.

The EfficientNet LSTM model built using the hyperparameters in Table 9 was trained using the ACRIMA dataset, which has a total of 4,230 fundus images, with 3,807 images used for training and 423 images used for testing. The evaluation method used is 10-fold cross-validation. The data will be divided equally into 10 folds, and each fold can be used as testing data once.

Table 10 presents the results for tests conducted on each fold along with the performance matrix values in the form of accuracy, loss, precision, sensitivity, specificity, and F1 score. Tests conducted on fold one produce the best accuracy, loss, precision, sensitivity, and F1 score when compared to other folds, with values of 0.9905, 0.0337, 0.9906, 0.9905, and 0.9905. At the same time, the best specificity is generated in fold five, with a value of 0.9957. The resulting average accuracy is 0.9799. Then 0.0596, 0.9802, 0.9799, 0.9771, and 0.9799 are the average values generated for loss, precision, sensitivity, specificity, and F1 score.

Table 9. Hyperparameters for the best performance

Hyperparameter	Value
Batch Size	32
Optimizer	Adam
Learning Rate	0.0001
Loss Function	BCE
Activation Function	ReLu, Sigmoid
Unit LSTM	100
Epoch	50

3.5 Comparison of e-LSTM and EfficientNet

To evaluate the effectiveness of adding an LSTM layer to the EfficientNet-B0 architecture, we need to conduct performance testing. We will present the performance matrix results of e-LSTM tested using several datasets that have been collected, namely ACRIMA (D1), DRISHTI-GS (D2), RIM-ONE DL (D3), and the combination of the three (D4). The performance matrix that will be used as a reference is: accuracy, loss, precision, sensitivity, specificity, and the F1 score.

It can be seen in Table 11 that the performance matrix of the average F1 score on the four training schemes increased when EfficientNet added LSTM, with an increase in the ACRIMA dataset by 2.11%, the DRISGTI-GS dataset by 16.79%, the RIM-ONE DL by 1.23%, and the combined dataset by 2.73%. The average of the four tests' F1 score increases is 5.7%. Not only in the F1 score performance matrix, but the addition of LSTM can also increase other performance matrices such as accuracy, loss, precision sensitivity, and specificity. The second training scheme with the DRISHTI-GS dataset input experienced the most significant increase in the performance matrix when compared to the others.

Table 10. Experiment result for each fold of the ACRIMA dataset

Fold	ACC	LOSS	PRE	SEN	SPE	F1
1	**0.9905**	**0.0337**	**0.9906**	**0.9905**	0.9870	**0.9905**
2	0.9858	0.0421	0.9863	0.9858	0.9753	0.9858
3	0.9787	0.0935	0.9798	0.9787	0.9639	0.9788
4	0.9740	0.0726	0.9741	0.9740	0.9824	0.9740
5	0.9811	0.0403	0.9814	0.9811	**0.9957**	0.9811
6	0.9764	0.0597	0.9769	0.9764	0.9688	0.9764
7	0.9811	0.0450	0.9813	0.9811	0.9748	0.9811
8	0.9716	0.0880	0.9717	0.9716	0.9696	0.9716
9	0.9835	0.0600	0.9835	0.9835	0.9818	0.9835
10	0.9764	0.0606	0.9766	0.9764	0.9723	0.9764
Average	**0.9799**	**0.0596**	**0.9802**	**0.9799**	**0.9771**	**0.9799**

Table 11. Comparison performance of EfficientNet and e-LSTM

Data	Method	ACC	PRE	SEN	SPE	F1
D1	EfficientNet	0.9638	0.9539	0.9649	0.9638	0.9588
	e-LSTM	**0.9799**	**0.9802**	**0.9799**	**0.9771**	**0.9799**
D2	EfficientNet	0.8630	0.8465	0.6823	0.6823	0.7525
	e-LSTM	**0.9208**	**0.9251**	**0.9208**	**0.8679**	**0.9204**
D3	EfficientNet	0.8814	0.8896	0.9344	0.7847	0.9103
	e-LSTM	**0.9230**	**0.9255**	**0.9230**	**0.9444**	**0.9226**
D4	EfficientNet	0.9023	0.8866	0.9253	0.8785	0.9054
	e-LSTM	**0.9329**	**0.9353**	**0.9329**	**0.9520**	**0.9327**

Figures 15 and 16 are comparison graphs of the average accuracy and loss of EfficientNet LSTM tested using several datasets that have been collected. The ACRIMA dataset produces the best accuracy and loss when using the proposed method, with values of 0.9799 and 0.0596. Adding LSTM can improve the accuracy and loss of EfficientNet on each of the other datasets, namely DRISHTI-GS, RIM-ONE DL, and the combination of the three.

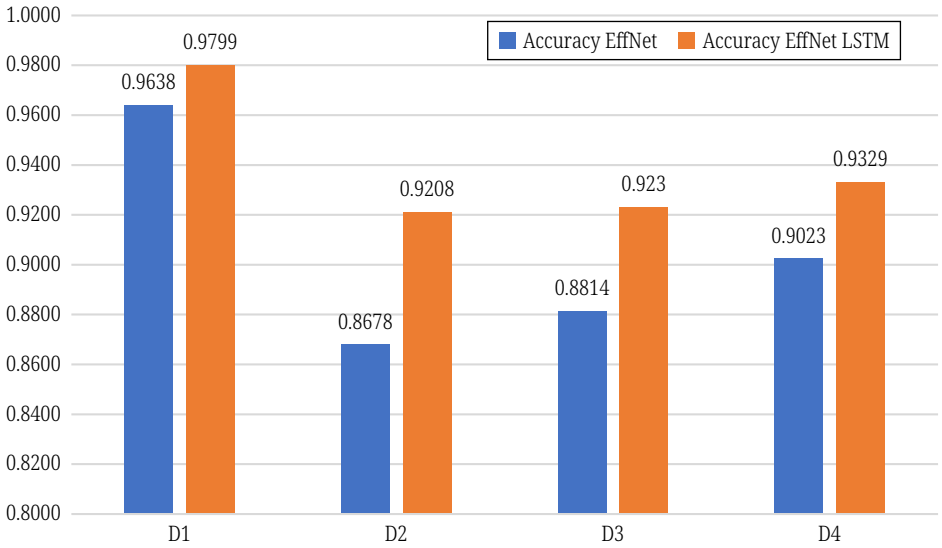

Fig. 15. Comparison of average accuracy for each dataset

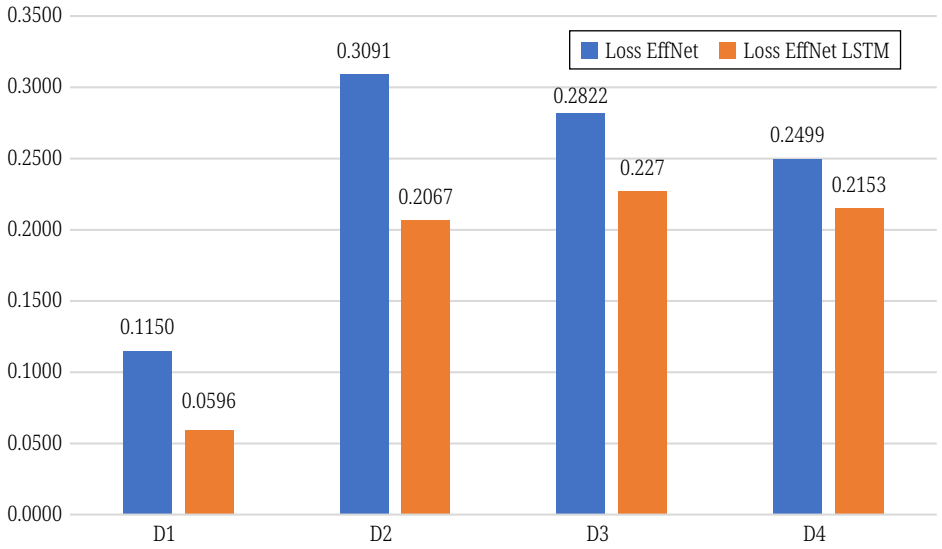

Fig. 16. Comparison of average loss for each dataset

3.6 Analysis of results

Analysis of the results is useful for obtaining information related to the feasibility of performance in calculating the success rate of a classifier model. This process is carried out using the confusion matrix shown in Figure 17. It can be seen that the model built made a prediction error of four images from a total of 423 images tested

on the ACRIMA dataset. The three images in the glaucoma class are predicted as a normal class, and one image for the normal class is assessed by the model in the glaucoma class.

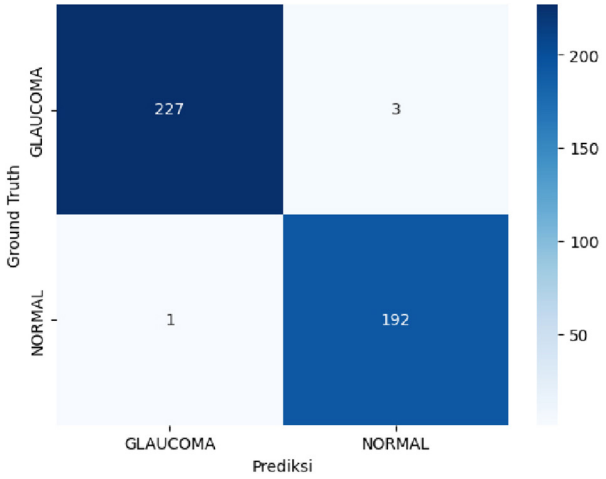

Fig. 17. Confusion matrix

Model errors in prediction can be caused by many factors, such as poor data quality, mislabeling of the dataset, or inadequate feature extraction. Figure 18a below is one of the fundus images that the model failed to classify. The image appears blurred in the optic cup and optic disc parts and is almost invisible, so the model has difficulty predicting the class of the image. Figure 18b is a fundus image that is visible in the optic cup (blue circle) and optic disc (red circle), so the image is successfully classified in the normal class by the model.

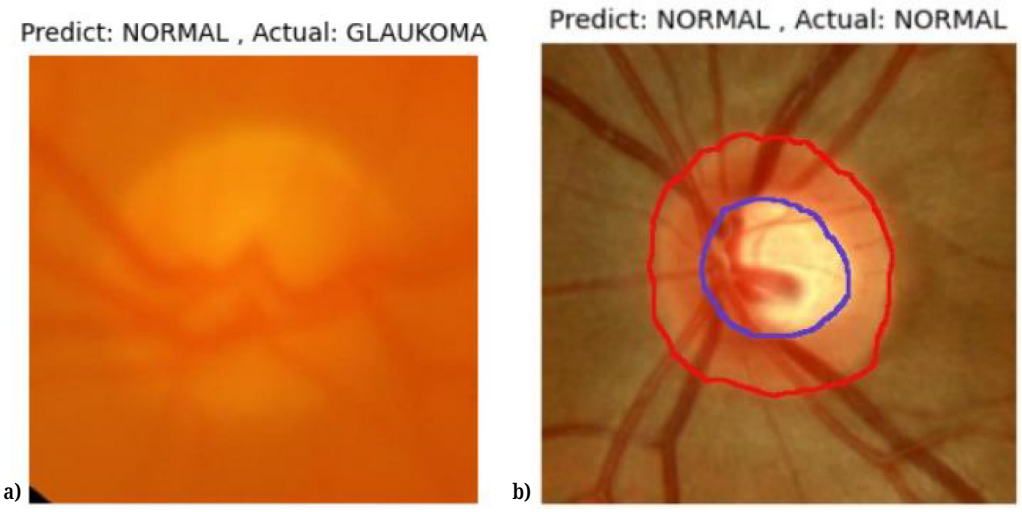

Fig. 18. Fundus image (a) Misclassified (b) Successfully classified

3.7 Comparison with previous research

To validate the performance of the proposed method, e-LSTM, we will compare the performance of the method with several previous methods on the same datasets, namely ACRIMA (D1), DRISHTI (D2), and RIM-ONE DL (D3).

Table 12. Comparison with the existing machine learning-based state-of-art methods of glaucoma classification

Data	Method	Evaluation Method	Performance
ACRIMA	Self-ONN [27]	10-Fold Cross Validation	ACC: 0.945 SEN: 0.945 SPE: 0.924 F1: 0.939
	EfficientNet-B0 [16]	Hold Out	ACC: 0.9775 PRE: 0.9945 SEN: 0.9577 SPE: 0.9953 F1: 9757
	e-LSTM (Proposed Method)	10-Fold Cross Validation	ACC: 0.9799 PRE: 0.9802 SEN: 0.9799 SPE: 0.9771 F1: 0.9799
DRISHTI-GS	DeeplabV3+Transfer learning [11]	Hold Out	ACC: 85.19
	e-LSTM (Proposed Method)	10-Fold Cross Validation	ACC: 0.9208 PRE: 0.9251 SEN: 0.9208 SPE: 0.8679 F1: 0.9204
RIM-ONE DL	AG-CNN [28]	Hold Out	ACC: 0.852 SEN: 0.848 SPE: 0.855 AUC: 0.916 F1: 0.837
	e-LSTM (Proposed Method)	10-Fold Cross Validation	ACC: 0.9230 PRE: 0.9255 SEN: 0.9230 SPE: 0.9444 F1: 0.9226

On the ACRIMA dataset, research conducted by Devecioglu et al. [27] using the Self-ONN method and the 10-fold cross-validation evaluation method for glaucoma classification cases resulted in accuracy, sensitivity, specificity, and an F1 score of 0.945, 0.945, 0.924, and 0.939. When compared to the results of the proposed method with the same dataset in Table 12, the proposed method can outperform all performance matrices.

Then Toptaş and Hanbay's research [16] using the EfficientNet method and the hold-out evaluation method with the ACRIMA dataset obtained results of accuracy of 0.9775, precision of 0.9945, sensitivity of 0.9577, specificity of 0.9953, and an F1 score of 0.9757. When compared to the results obtained by the proposed method in Table 12, the proposed method has decreased in precision and specificity. On the other hand, there is an insignificant increase in the accuracy and F1 score performance matrices. For sensitivity, the proposed method is superior, with a difference of approximately 2%. The ups and downs in the performance matrix are due to the use of different data preprocessing and evaluation methods. Toptaş and Hanbay's research applied contrast-limited adaptive histogram equalization (CLAHE) to improve image quality with the hold-out evaluation method. At the same time,

the proposed method does not use an algorithm for image enhancement and uses the 10-fold cross-validation evaluation method. The use of 10-fold cross-validation will improve the performance estimation of the model by calculating the average of 10 iterations with different training and testing data, resulting in a more stable and accurate performance estimate when compared to the hold-out method.

The results of the proposed method in Table 12 for the DRISHTI-GS dataset obtained better accuracy performance when compared to research conducted by Sreng et al. [11], which only obtained an accuracy of 0.8519. In addition, research conducted by Sreng et al. applied segmentation to the optic disc using DeepLabV3 from the dataset that had been obtained to be used as input to the model built. Whereas in the proposed method, the data used as input is raw data only, but it produces better performance, so it has better efficiency when compared to research conducted by Sreng et al. [11].

The results shown by the proposed method for the RIM-ONE DL dataset in Table 12 obtained better accuracy, sensitivity, specificity, and F1 score performance when compared to research conducted by Li et al. [28]. The research of Li et al. used the AG-CNN method and the hold-out evaluation method, with results of accuracy, sensitivity, specificity, and F1 scores of 0.852, 0.848, 0.855, and 0.837.

4 CONCLUSION

In some experiments conducted previously, it can be concluded that utilizing the LSTM algorithm can improve the performance of EfficientNet-B0 in the classification of glaucoma eye diseases. The addition of LSTM can improve the performance matrix F1 score in all four training schemes, with an average increase of 5.7%. The most optimal results were obtained with an average accuracy of 0.9799, loss of 0.0596, precision of 0.9802, sensitivity of 0.9799, specificity of 0.9771, and F1 score of 0.9799 in the training scheme with ACRIMA dataset input. Then, in another training scheme with the DRISHTI-GS dataset input, it obtained the most significant improvement in the performance matrix when added to long short-term memory.

The EfficientNet LSTM or e-LSTM model was built with 100 units in the LSTM layer, batch size 32, a learning rate of 0.0001, a loss function using binary cross-entropy, and a number of epochs as large as 50. These results are obtained using the 10-fold cross-validation method technique, where the model is tested with 10-fold different data, which will then take the average performance matrix obtained.

Based on the study conducted, the proposed method has shown satisfactory accuracy. Nevertheless, after analyzing the results in depth, the authors suggest further improvement by using image quality enhancement techniques such as histogram equalization (HE), adaptive histogram equalization (AHE), CLAHE, etc. Incorporating these techniques can substantially enhance the system's ability to detect glaucoma from blurry fundus images. However, this study's system only classifies the fundus image into two categories. The author suggests that in future research, glaucoma can be categorized based on its severity by comparing the cup-disc ratio to the fundus image.

5 ACKNOWLEDGEMENTS

We thank Universitas Sebelas Maret, Surakarta, Indonesia, for providing research funding through the **Group Research Grant scheme as stated in contract No. 228/UN27.22/PT.01.03/2023**.

6 REFERENCES

[1] A. K. Schuster, C. Erb, E. M. Hoffmann, T. Dietlein, and N. Pfeiffer, "The diagnosis and treatment of glaucoma," *Deutsches Ärzteblatt International*, pp. 225–234, 2020. https://doi.org/10.3238/arztebl.2020.0225

[2] O. Geyer and Y. Levo, "Glaucoma is an autoimmune disease," *Autoimmunity Reviews*, vol. 19, no. 6, p. 102535, 2020. https://doi.org/10.1016/j.autrev.2020.102535

[3] S. Kingman, "Glaucoma is second leading cause of blindness globally," *Bull World Health Organ*, vol. 82, no. 11, pp. 887–888, 2004.

[4] H. A. Quigley, "The number of people with glaucoma worldwide in 2010 and 2020," *British Journal of Ophthalmology*, vol. 90, no. 3, pp. 262–267, 2006. https://doi.org/10.1136/bjo.2005.081224

[5] Y. C. Tham, X. Li, T. Y. Wong, H. A. Quigley, T. Aung, and C. Y. Cheng, "Global prevalence of glaucoma and projections of glaucoma burden through 2040," *Ophthalmology*, vol. 121, no. 11, pp. 2081–2090, 2014. https://doi.org/10.1016/j.ophtha.2014.05.013

[6] Aziz-ur-Rehman *et al.*, "An ensemble framework based on Deep CNNs architecture for glaucoma classification using fundus photography," *MBE*, vol. 18, no. 5, pp. 5321–5346, 2021. https://doi.org/10.3934/mbe.2021270

[7] Jordan *et al.*, "A hybrid SVM Naïve-Bayes classifier for bright lesions recognition in eye fundus images," *IJEEI*, vol. 13, no. 3, pp. 530–545, 2021. https://doi.org/10.15676/ijeei.2021.13.3.2

[8] N. Gharaibeh, O. M. A. Hazaimeh, B. A. Naami, and K. M. O. Nahar, "An effective image processing method for detection of diabetic retinopathy diseases from retinal fundus images," *IJSISE*, vol. 11, no. 4, pp. 206–216, 2018. https://doi.org/10.1504/IJSISE.2018.093825

[9] H. Ahmad, A. Yamin, A. Shakeel, S. O. Gillani, and U. Ansari, "Detection of glaucoma using retinal fundus images," in *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, Islamabad, Pakistan: IEEE, 2014, pp. 321–324. https://doi.org/10.1109/iCREATE.2014.6828388

[10] A. Azeroual *et al.*, "Convolutional neural network for segmentation and classification of glaucoma," *Int. J. Onl. Eng.*, vol. 19, no. 17, pp. 19–32, 2023. https://doi.org/10.3991/ijoe.v19i17.43029

[11] S. Sreng, N. Maneerat, K. Hamamoto, and K. Y. Win, "Deep learning for optic disc segmentation and glaucoma diagnosis on retinal images," *Applied Sciences*, vol. 10, no. 14, p. 4916, 2020. https://doi.org/10.3390/app10144916

[12] V. K. Velpula and L. D. Sharma, "Multi-stage glaucoma classification using pre-trained convolutional neural networks and voting-based classifier fusion," *Frontiers in Physiology*, vol. 14, 2023. https://www.frontiersin.org/articles/10.3389/fphys.2023.1175881

[13] F. Li *et al.*, "Automatic differentiation of glaucoma visual field from non-glaucoma visual filed using deep convolutional neural network," *BMC Med Imaging*, vol. 18, no. 1, p. 35, 2018. https://doi.org/10.1186/s12880-018-0273-5

[14] S. Joshi, B. Partibane, W. A. Hatamleh, H. Tarazi, C. S. Yadav, and D. Krah, "Glaucoma detection using image processing and supervised learning for classification," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–12, 2022. https://doi.org/10.1155/2022/2988262

[15] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing*, vol. 96, p. 106691, 2020. https://doi.org/10.1016/j.asoc.2020.106691

[16] B. Toptaş and D. Hanbay, "Separating of glaucoma and non-glaucoma fundus images using EfficientNet-B0," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 11, no. 4, pp. 1084–1092, 2022. https://doi.org/10.17798/bitlisfen.1174512

[17] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv*, 2020. https://doi.org/10.48550/arXiv.1905.11946

[18] Md. Z. Islam, Md. M. Islam, and A. Asraf, "A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images," *Informatics in Medicine Unlocked*, vol. 20, p. 100412, 2020. https://doi.org/10.1016/j.imu.2020.100412

[19] I. Shahzadi, T. B. Tang, F. Meriadeau, and A. Quyyum, "CNN-LSTM: Cascaded framework for brain tumour classification," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, Sarawak, Malaysia: IEEE, 2018, pp. 633–637. https://doi.org/10.1109/IECBES.2018.8626704

[20] Y. Wang, Q. Wu, N. Dey, S. Fong, and A. S. Ashour, "Deep back propagation–long short-term memory network based upper-limb sEMG signal classification for automated rehabilitation," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 987–1001, 2020. https://doi.org/10.1016/j.bbe.2020.05.003

[21] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, 2021. https://doi.org/10.3390/s21082852

[22] M. Tan *et al.*, "MnasNet: Platform-Aware neural architecture search for mobile," *arXiv*, 2019. https://doi.org/10.48550/arXiv.1807.11626

[23] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017. https://doi.org/10.48550/arXiv.1704.04861

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[26] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation," *arXiv*, 2018. http://arxiv.org/abs/1610.02583

[27] O. C. Devecioglu, J. Malik, T. Ince, S. Kiranyaz, E. Atalay, and M. Gabbouj, "Real-time glaucoma detection from digital fundus images using Self-ONNs," *IEEE Access*, vol. 9, pp. 140031–140041, 2021. https://doi.org/10.1109/ACCESS.2021.3118102

[28] L. Li *et al.*, "A large-scale database and a CNN model for attention-based glaucoma detection," *IEEE Trans. Med. Imaging*, vol. 39, no. 2, pp. 413–424, 2020. https://doi.org/10.1109/TMI.2019.2927226

7 AUTHORS

Wiharto currently works as a lecturer at Universitas Sebelas Maret, Surakarta, in the Department of Informatics in Indonesia. His research is primarily focused on the biomedical informatics field (E-mail: wiharto@staff.uns.ac.id).

Wimas Tri Harjoko completed his bachelor's degree in Computer Science from Universitas Sebelas Maret, Surakarta, Indonesia, in 2023.

Esti Suryani is a lecturer in the Department of Data Science at Universitas Sebelas Maret, Surakarta, Indonesia. Her research interests are primarily focused on biomedical data.