Jurnal Infotel
Vol. 16, No. 1, February 2024, pp. 136–155.
doi:10.20895/infotel.v16i1.999

Research Article

Weighted Voting Ensemble for Enchanced Diabetic Retinopathy Classification using CNN Architectures

Anita Desiani[1,*], Rifkie Primartha[2], Herlina Hanum[3], Siti Rusdiana Puspa Dewi[4], Bambang Suprihatin[5], Muhammad Gibran Al-Filambany[6], and Muhammad Suedarmin[7]

[1,3,5,6,7]Mathematics Department, Universitas Sriwijaya, Sumatera Selatan 30862, Indonesia
[2]Informatics Engineering Department, Universitas Sriwijaya, Sumatera Selatan 30862, Indonesia
[4]Dentistry Department, Universitas Sriwijaya, Sumatera Selatan 30862, Indonesia

*Corresponding email: anita_desiani@unsri.ac.id

Received: July 17, 2023; Revised: November 18, 2023; Accepted: February 2, 2024.

Abstract: Diabetes can cause an eye disorder known as diabetic retinopathy (DR). DR disorders can be recognized through retinal images. The process of assisting retinal images can be done by applying deep learning-based methods, one of which is the convolutional neural network (CNN). CNN has many architectures that can perform image classification processes, namely ResNet-50, MobileNet, and EfficientNet. Weaknesses of each architecture can be overcome through ensemble learning methods that can add up the performance results of each classification method. The study applies the ensemble learning method to improve the performance of the ResNet-50, MobileNet, and EfficientNet architectures in paying for DR disease on the retina by weighted voting. The data used are the APTOS and EyePACS datasets. The method in this research is data collection, training, testing, and evaluation of each architecture and ensemble learning. The results of the superior ensemble learning performance in the value of accuracy, F1-Score, and Cohens Kappa were obtained respectively 93.3 %, 93.42 %, and 0.866. The best specificity value was obtained by Resnet-50 at 99.78 % and the highest sensitivity value was obtained by EfficientNet at 96.2 %. Based on the classification results of each architectural and ensemble learning, it can be interpreted that the proposed ensemble learning method is excellent for performing image classification for Diabetic Retinopathy.

Keywords: diabetic retinopathy, classification, ensemble learning, MobileNet, ResNet-50, EfficientNet

1 Introduction

Diabetic retinopathy (DR) is a complication of diabetes mellitus. This complication causes damage to the retinal blood vessels, especially the parts that are sensitive to light [1]. DR detection is commonly performed using conventional methods. The most commonly used conventional methods are support vector machines (SVM) and Naive Bayes. However, the conventional method is considered less effective because it takes a long time and requires significant costs [2]. In addition, the conventional method allows for errors in classification. Limitations of conventional methods have led to the development of automated diagnostics using computers to classify retinal images [3]. Image classification is one of the applications of deep learning. Deep learning is popularly used in image classification, including identifying DR disease in retinal images. One deep learning method that can solve such problems is the convolutional neural network (CNN). The method consists of convolutional layers and pooling layers that are interconnected so that it can extract features and identify objects in images [4]. In some cases, the training process using CNN can be time-consuming, especially when dealing with large amounts of data. The using of pre-trained CNN architectures can be an appropriate solution to accelerate classification performance. In this way, the process of identifying DR disease can become more efficient [5].

One common issue encountered during CNN training is the vanishing gradient. The issue of vanishing gradients can arise when a model has a very deep layer [6]. The gradient of the loss function to the parameter in the training process can become smaller because the network becomes unresponsive while training the data [7]. The problem of vanishing gradients can be solved by applying the Residual Network-50 (ResNet-50) using skip connections [8]. skip connection is a type of shortcut that connects an output layer to another input layer that may not be nearby. Skip connection can maintain the stability of the gradient during the training process, as it provides a shortcut to pass several layers. This allows the gradient to flow smoothly through the network, preventing the issue of vanishing gradients [9]. DR classification has been applied by various studies using ResNet. Heisler *et al.* [10] applied the DR classification on the OCTA dataset using the ResNet architecture. This study obtained an accuracy of 90.71 % and did not count any other performance. Qummar *et al.* [11] applied the DR classification using ResNet architecture on EyePACS datasets. This study achieved an accuracy of 80.8 %. However, the sensitivity value is still below 80 % and did not count other performances. Jinfeng *et al.* [12] applied the DR classification using ResNet architecture on the EyePACS dataset. This study obtained a specificity score of 84.8 %. However, the accuracy value is still below 75 % and did not count other performances. ResNet-50 used residual networks to deal with the problem of vanishing gradients, but ResNet-50 often overfitted due to a large number of parameters [13]. Reducing the layer of convolution layers on ResNet can decrease both the model's parameters and performance [14].

One of the CNN architectures that can perform feature extraction to reduce computing complexity is the Mobile Neural Network (MobileNet) [15]. MobileNet has a smaller structure than any other CNN architecture. MobileNet applies a sharing parameter, where each filter is distributed across the input. This can reduce the complexity of the model so as to overcome the increasing number of parameters. MobileNet used depthwise separable convolutions for efficient feature extraction while maintaining good accuracy [15]. DR classification has been applied by various studies using MobileNet. Sheikh and Qidwai [16] applied DR classification on the Eyepacs dataset using MobileNet. This study obtained an accuracy of 90 %. However, the F1-score value is still below 70 % and did not count other performances. Sejal and Thiruthuvanthan [17] applied DR classification on the same dataset using MobileNet. This study got an accuracy of 90 %. However, the F1-Score value is still below 70 % and did not count other performances. Suriyal et al [18] applied DR classification using MobileNet on Eyepacs datasets. This study obtained an accuracy of 73 % and did not count other performances. MobileNet has limited feature learning due to its low parameter count, which can result in underfitting [18, 19].

The underfitting problems can be solved using the EfficientNet architecture. EfficientNet solved underfitting issues using compound scaling [20]. Compound scaling is a technique used in EfficientNet that involves uniform adjustment to all dimensions of the model, which include depth, width, and resolution [21]. Model performance can be improved with this technique by finding an optimal balance between these dimensions, making it more efficient [22]. DR classification has been applied by various studies using EfficientNet. Liu *et al.* [23] applied DR classification using EfficientNet on EyePACS and APTOS datasets. This study obtained an accuracy of 85.44 %. However, the specificity value is still below 75 % and did not count other performances. Momeni *et al.* [24] applied DR classification using EfficientNet architecture on Messidor and IDRiD datasets. This study has achieved an accuracy of 94.5 %. However, this study did not count other performances. EfficientNet's high network complexity can cause overfitting during training [22, 25].

The ensemble Learning overcomes the weaknesses of each architecture by combining the strengths of each architecture. Ensemble learning is a process of combining output values from several classification models into one predictive model [11]. Some decision-making techniques in ensemble learning that are commonly used are bagging, stacking, boosting, voting, and averaging [26]. Voting is a popular decision-making technique in ensembles [27]. Weighted voting is one of the most commonly used techniques for voting. Weighted voting is a decision-making process that takes the best weight from the final prediction of a single classification [28]. DR classification has been applied by various studies using ensemble learning. Antal and Hadju [29] applied DR classification using ensemble learning on forward and backward search. The study obtained an accuracy of 87 %. However, they did not evaluate the other performances. Jiang *et al.* [30] applied DR classification using ensemble learning on InceptionV3, InceptionResNetV2, and ResNet152. This study obtained an accuracy of 88 %. However, the sensitivity and specificity values are still below 85 %. Saleh *et al.* [31] applied DR classification using ensemble learning on dominance-based rough set fuzzy random forest. This study obtained an accuracy of 80 %. However, the sensitivity and specificity values are still below 80 %. In previous studies, ensemble learning was only applied to the final results of testing data. Previous studies did not apply ensemble learning to the training process. It could not be recognized how ensemble learning performance on training data and validation data.

This study aims to optimize the image classification from the performance results of the ResNet-50, MobileNet, and EfficientNet architectures by applying the ensemble learning method. In this study, decision-making will use voting techniques. The ensemble is applied to the training process, and the performance results of the training and validation data are weighted at each epoch. Ensemble in this study is performed to check whether the weights are overfitting or not. Weighted voting is performed to ensure that the weight in the final stage of training is the most optimal weight for classification on testing data. By applying the ensemble learning method, the study is expected to increase the performance results in the Diabetic Retinopathy classification based on retinal images. The retina image used in this study was categorized into two labels (Normal and Abnormal). The Normal label referred to retina images that did not display any DR symptoms. The Abnormal label was assigned to retinal images that showed signs of Diabetic Retinopathy. To evaluate the resulting method, the study compares the results of every single classifier performance(ResNet, MobileNet, and EfficientNet) and the results of ensemble learning, based on accuracy, sensitivity, specificity, F1-Score, and Cohen's Kappa.

2 Research Method

This section we discusses data description, image pre-processing, single classification architecture, ensemble learning, and confusion matrix.

2.1 Data Description

This study uses the Eyepacs Dataset and APTOS dataset. Aptos dataset consists of 3,662 retinal eye fundus data, with 1,805 normal eyes and 1,857 eyes infected with DR. The eyepacs dataset consists of 35,126 eye fundus images, with 25,810 normal retinas and 9,316 retinas infected with DR. In this study, only normal and DR-infected eyes were classified, by taking 9,000 normal eyes and 9,000 DR-infected eyes to maintain the balance of the dataset.

2.2 Image Pre-Processing

Several stages contained in pre-processing include improving image quality, increasing image contrast, and filtering noise [18].

2.2.1 Green channel

The green channel shows better background contrast than the red channel and the blue channel has low contrast and noise. The red channel has the highest image illumination and the blue channel has the lowest image illumination [14].

2.2.2 Contrast limited adative histogram equalization (CLAHE)

CLAHE is a way to increase image contrast [19]. CLAHE is commonly used to partition the image into contextual regions and then apply an equalization histogram. CLAHE spreads the intensity distribution and adjusts the intensity of the original shadow [20]. The CLAHE calculation technique is based on the histogram clip limit, which is determined by (1).

$$\beta = \frac{M}{N}\left(1 + \frac{\alpha}{100}(S_{max} - 1\right) \tag{1}$$

In (1), M is the region's area size, N denoted grayscale value, and α denotes the clipping factor that adds to the histogram limit, ranging from 0 to 100.

2.3 Single classification architecture

2.3.1 ResNet-50

ResNet-50 is a conceptualization of residual blocks within a residual learning network [32]. The residual block connects the first block's input to the second block's output. This addition operation allows the residual block to learn the residual function and avoid parameter explosion. ResNet50 design contains 50 residual block layers, including a convolutional layer, 48 residual blocks, and a classifier layer with small filter sizes of 1 × 1 and 3 × 3 [33]. Figure 1 illustrates the ResNet architecture that will be employed in this study.

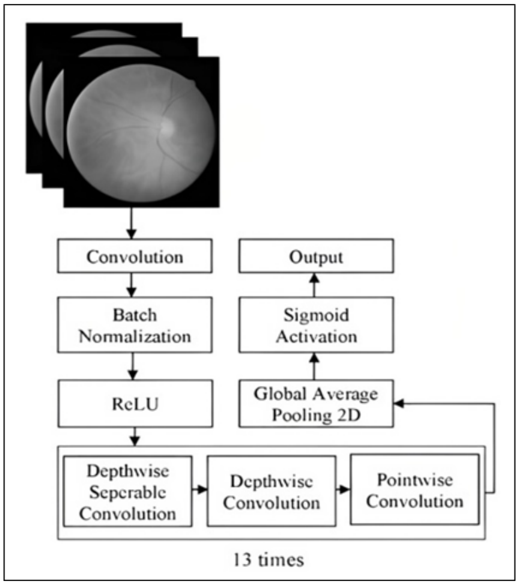

Figure 1: ResNet-50 architecture.

The ResNet-50 architecture in Figure 1 contains block Convolution, Max Pooling, ConvBlock, Identity, Global Average Pooling 2D, and Dense Sigmoid Prediction. The steps for the ResNet50 training process are as follows:

a. The input data will go through the convolution process following (2) to (3). After that, the process continues with max pooling.

$$C^l_p = \sigma(l * k^l_{p,q} + b^l_p) \tag{2}$$

with * denote the convolution operation.

$$l * k^l_{p,q} = \sum_{m=1}^{M} l^{0j}_{i+m-1} k^l_{p,q} \tag{3}$$

Where C^l_p represents the value p-th feature map in the l-th layer, p denoted the number of filters, M denoted the kernel size height, σ denoted the activation function, I is the input and $k^l_{p,q}$ represents the kernel layer up to the l-th layer in the feature map, specifically referring to the entries in the p-row and q-column.

b. The process continues with ConvBlock, which comprises three convolutional networks and one residual convolution network.

c. The next step is carried out with Identity Block, Identity Block consists of 3 convolution networks and 1 residual network from the previous output.

d. The ResNet process uses 1 ConvBlock, followed by 2 Identity Blocks, then 1 ConvBlock, followed by 3 Identity Blocks, then 1 ConvBlock, 5 Identity Blocks, 1 ConvBlock, 2 Identity Blocks, ending with Global Avarage Max Pooling and dense sigmoid.

e. The calculation of the loss (error) value for each epoch should use (4) for the training data.

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) \tag{4}$$

where N displays multiple architectures. y_i is the actual class label. If the value is 1 for the DR class and 0 for the normal class. p_i is the predicted value. L is the binary cross entropy loss function.

f. Save weights if the error in the validation data is smaller than the previous epoch. If the weight bigger than previous epoch then the weights will be updated for the next epoch.

g. Repeat all same steps from step (b) to (f) until the last epoch.

2.3.2 MobileNet

MobileNet is an architecture that is designed using depthwise separable convolution referred to as depthwise separable convolution [18]. MobileNet can build a lightweight CNN model and can reduce computation time [34]. MobileNet can show strength in performance even using small parameters. The MobileNet architecture that will be used in this study is depicted in Figure 2.

In Figure 2, MobileNet architecture utilizes depth-separable convolution that consists of two convolutions, namely depth convolution and point convolution. The training process at MobileNet is as follows:

a. Input the image, then do the convolution layer process using (2) to (3).

b. Do batch normalization which can be calculated from (5) to (7).

$$\hat{c}_{l,m} = \frac{c_{l,m} - \mu_m}{\sqrt{\sigma_m^2 + \epsilon}} \tag{5}$$

with

$$\mu_m = \frac{1}{n}\sum_{l=1} x_{l,m} \tag{6}$$

$$\sigma_m^2 = \frac{1}{n}\sum_{l=1}^{n}(c_{l,m} - \mu_m)^2 \tag{7}$$

where l is a row, m is a column, μ_m is the mean value across each m-th column, m is the amount of data in one mini-batch (row), $c_{l,m}$ is the matrix entry resulting from convolution c in the row entry of the l-th and m-th columns, σ_m^2 is the variance in the hidden layer m-th column, and ϵ is the minimum positive constant.

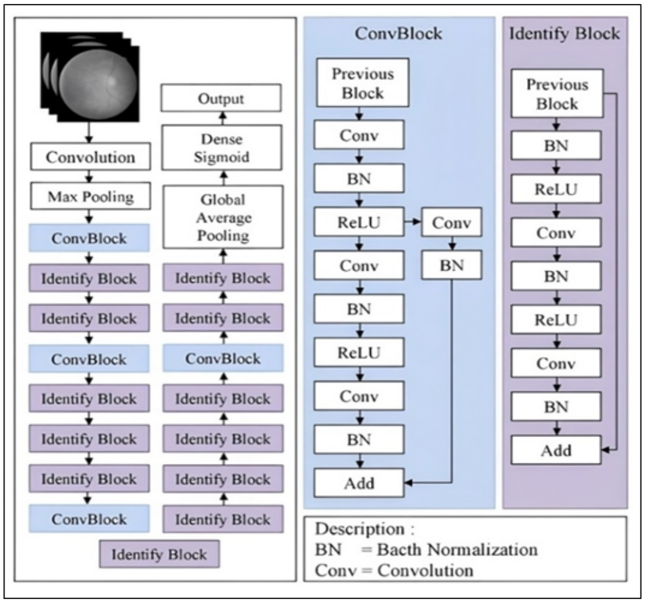

Figure 2: MobileNet architecture.

c. Use the ReLU activation function which can be calculated in (8).

$$f_{l,m} = r(\hat{c}_{l,m}) = max(\hat{c}_{l,m}, 0) = \begin{cases} \hat{c}_{l,m} & \text{if } \hat{c}_{l,m} \geq 0, \\ 0 & \hat{c}_{l,m} < 0 \end{cases} \tag{8}$$

where $f_{l,m}$ is the outcome of the ReLU activation function and $\hat{c}_{l,m}$ is the entry value of the convolution matrix c which has been normalized in l-row and m-column with $\hat{c}_{l,m} \in (-\infty, +\infty)$.

d. Carry out the Depthwise Separable Convolutions process which includes depthwise convolution, batch normalization which can be seen as step (c), the ReLU activation function as step (c), and pointwise convolution. This step will be repeated 13 times.

e. Do global average pooling and the sigmoid function which can be seen in (9).

$$h_{l,m} = \sigma(f_{l,m}) = \frac{1}{1 + e^{-f_{l,m}}} \tag{9}$$

where, $h_{l,m}$ is the outcome of the sigmoid activation function with $h \in (0, 1)$ and $f_{l,m}$ is the matrix entry of the ReLu activation function in the l-row and m-column entry where e is the functions of sigmoid activation.

f. Calculate the loss (error) for each epoch using (4) and accuracy for the training data.

g. Save weights if the error in the validation data is lower than the previous epoch, otherwise, the weight will be upgraded for the next epoch.

h. Repeat all same steps from step (b) to (g) until the last epoch.

2.3.3 EfficientNet

EfficientNet is a development of a mobile-size baseline that is used to make the feature enhancement method more effective. The EfficientNet architecture was designed by Tan and Le [21]. Uniformly, EfficientNet improves all dimensions of depth, width, and resolution of a CNN by using simple but highly effective composite coefficients. EfficientNet achieves much better accuracy and efficiency than ConvNets [23]. Each EfficientNet model refers to a variant with more parameters and more accuracy. EfficientNet uses transfer learning to save time and computational power so EfficientNet gives higher accuracy than other models [35]. The EfficientNet architecture can be seen in Figure 3.

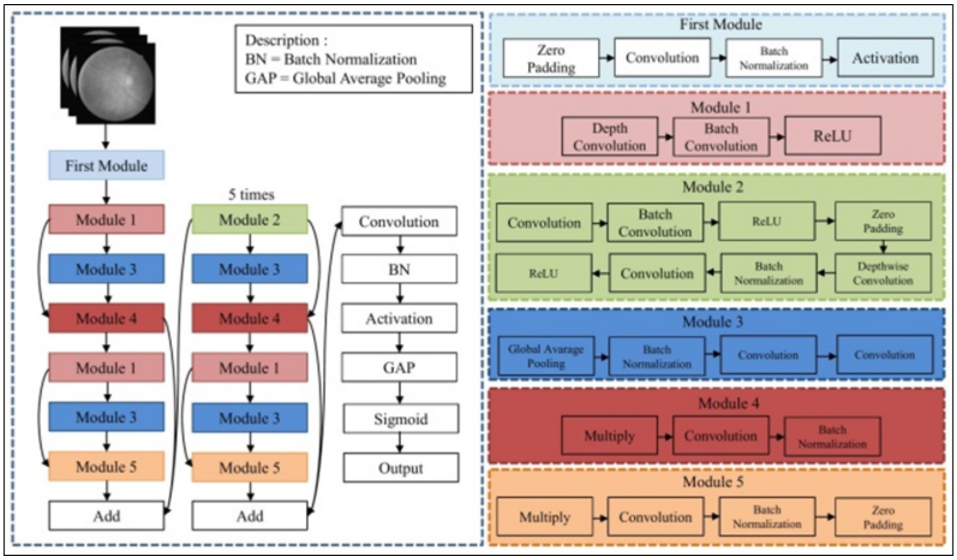

Figure 3: EfficientNet architecture.

In Figure 3, the EfficientNet contains Module 1, Module 2, Module 3, Module 4, Module 5, Add, Convolution, Batch Normalization, Global Average Pooling 2D, and Sigmoid Activation. The steps are as follows:

a. Entry the input data into the convolution process according to (2) to (3). After that, the process continues with maxpooling.

b. Do training with the EfficientNet network, the Module 1, Module 2, and Module 3 on EfficientNet network contain of convolution, batch normalization, and ReLU activation, while Module 4 and Module 5 have residual networks.

c. Calculate the loss (error) value for each epoch using (4) and the accuracy for the training data.

d. Save the weights. The weight will be stored if the error of the validation data is lower than the previous epoch, otherwise, the weight will be upgraded for next epoch.

e. Repeat same steps from step (b) to the last epoch.

f. Save the model and the output weights of the model.

2.4 Ensemble Learning

Ensemble learning uses several classification methods to get better predictive performance by combining the decision results from several methods into one powerful ensemble method [36]. At this stage, the final weight of each architecture is taken. The study uses 50 training repetitions (epochs), and 600 iterations in one epoch. The purpose of this step is to train the training data. The data training process for ensemble learning is depicted in Figure 4.

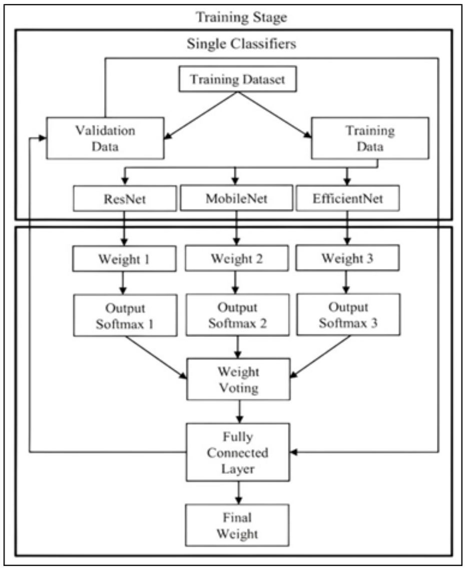

Figure 4: Illustration of ensemble learning.

Figure 4 illustrates the steps of the ensemble learning training process, along with the explanation.

a. Input data will be entered into the three architectures, in the training ensemble the three architectures will not change the weight, the three architectures will exclude disease probabilities from the input data.

b. Do weighted voting, and the three probability outputs will be entered into the weighted voting layer. Weighted voting is used to combine the predicted results from all models into one layer using weight training. To calculate the final prediction results, it can be used Equation (10) [37].

$$f(x_i) = \sum_{i=1}^{n} x_i p(w_i) \tag{10}$$

where $f(x_i)$ represent the output prediction result, x_i denoted the probability prediction result of each architecture, w_i denoted the weight in the ensemble layer, and $p(w_i)$ is the softmax function.

c. Calculate the loss (error) for each epoch using (4) and the accuracy for the training data.

d. Save the weights. The weight will be stored if the error of the validation data is lower than the previous epoch, otherwise, the weight will be upgraded for next epoch.

e. Repeat all steps (b) throught (e) until the last epoch.

2.5 Confusion Matrix

Confusion matrix is a performance measure in a classification problem where the output is two or more classes [38]. The performance is measured using a confusion matrix, which includes accuracy, sensitivity, specificity, F1-score, and Cohen's kappa. Equation (11) through (15) show calculations of accuracy, sensitivity, specificity, F1-score, and Cohen's Kappa values [39].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{11}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100\% \tag{12}$$

$$\text{Spesitivity} = \frac{TN}{TN + FP} \times 100\% \tag{13}$$

$$\text{F1-score} = \frac{2TP}{2TP + FP + FN} \times 100\% \tag{14}$$

$$\kappa = \frac{2(TP \times TN - FN \times FP)}{(TP + FP)(FP + TN)(TP + FN)(FN + TN)} \tag{15}$$

where TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative. Equation (11) through (15) will compare the results of single classification performanceand the results of the proposed methods.

3 Results

In this study, each architecture has three stages including preprocessing, training and testing, and evaluation.

3.1 Resnet50

The ResNet50 model uses 18,000 EYEPACS training data. The training process consists of 30 epochs. Accuracy and loss are used as evaluation values in the training process. The accuracy training process in 1st epoch obtained 0.7501 and a loss value obtained 0.5133 for training data, while the accuracy value data validation obtained 0.67898 and a loss value obtained 0.6032. Based on validation loss values, the weights are stored and used in the 2nd epoch. The validation loss value continuously updated a better validation loss value is achieved. The ResNet50 architecture model training process is carried out until the 30th epoch to get the best weights to use for the testing process. The results of accuracy and loss graphs for training and validation data on the ResNet50 model are illustrated in Figure 5.

In Figure 5(a) the accuracy and val_accuracy graph in ResNet50 for data training obtained during the training process is unstable, accuracy value increases and decreases from the 1st epoch until the 30th epoch. The accuracy graph is overfitting, the graph on the accuracy value and the accuracy value on the validation data should have increased. The graph of the loss value for the validation data in the ResNet50 architecture during the training process is shown in Figure 5(b), where it increases and decreases from the first to the thirtyth epoch. On the graph, the loss value for training data has decreased from the first epoch to the 30th epoch, which is equal to 0.49. Based on Figure 5, the model is experiencing overfitting, the graph of the validation data value should decrease to close to 0. The loss value for training data on the graph has dropped to 0.49 by the 30th epoch from the 1st epoch. The model is overfitting, as seen in Figure 5 and the validation data value graph approaches 0.

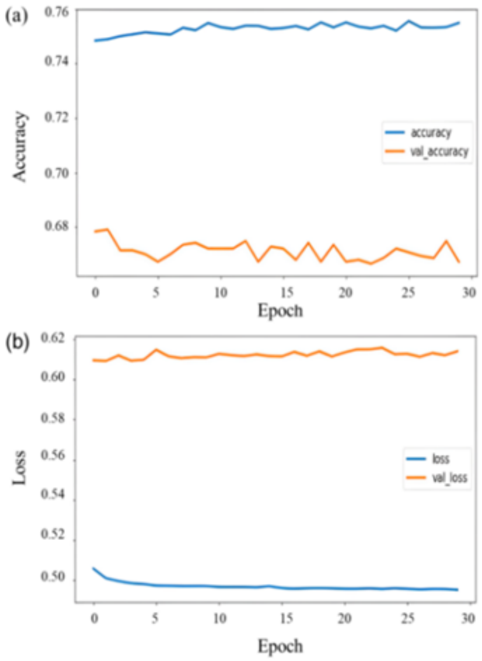

Figure 5: The results of the ResNet50 model training on the classification of diabetic retinopathy disorders based on retinal images (a) Accuracy graphs (b) Loss graphs.

The best weights will be saved when the training process is completed. The weights will be used in the testing process for predictions. Evaluate the performance model, can be done by comparing predicted values and actual values using a confusion matrix. The results obtained from the confusion matrix can be used to calculate performance based on accuracy, sensitivity, specificity, F1-score, and Cohen's kappa which can be calculated using Equations (11) to (15). The performance evaluation of Resnet50. The performance results of the Resnet50 architecture are an accuracy of 50.95 %, a sensitivity of 72.02 %, a specificity of 99.78 %, an F1-score of 1.42 %, and the Cohen's kappa of 0.005.

3.2 MobileNet

The MobileNet model used 35,126 image data. The data will be divided into 2 parts, namely 34,406 training data and 720 validation data. This process uses 30 epochs in training. The evaluation in the training process is accuracy and loss. The accuracy of the training process in 1st epoch was 0.5010 and loss obtained 0.6933. Then, the accuracy of validation data was 0.4972, and the loss obtained was 0.6932. Based on validation loss values, the weights are stored and used for 2nd epoch. The validation loss is continuously updated until the validation loss value is better. The MobileNet training process is carried out until the 30th epoch to obtain the best weight used for the testing data. Figure 6 is the accuracy graph of training and validation data on MobileNet.

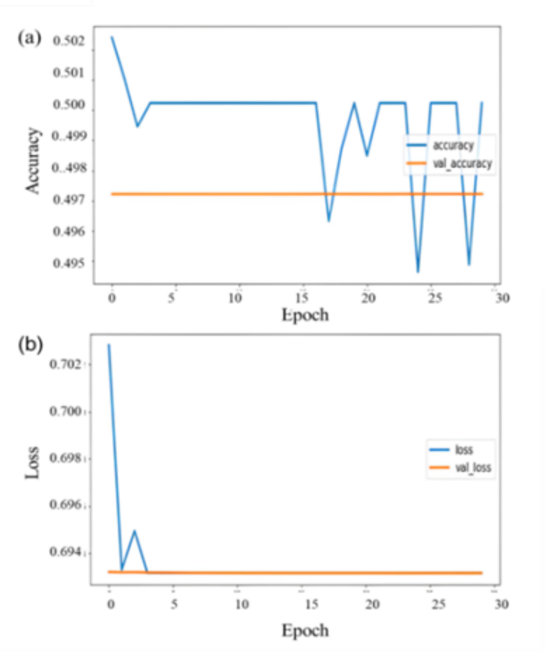

Figure 6: The (a) Accuracy and (b) Loss of the MobileNet on the classification of diabetic retinopathy disorders based on retinal images.

Figure 6(a) shows the unstable accuracy graph of the MobileNet model for the training data obtained during the training process. The accuracy value decreased from the 1st until the 3rd epoch, increased in the 4th, and remained constant until the 16th epoch. The accuracy value changed between during 17th and 30th epochs. The accuracy for validation data was stable from the beginning until the 30th epoch at a value of 0.497. The results of the accuracy graph and accuracy validation are still overfitting. Figure 6(b) shows the MobileNet loss graph for the training data decreased drastically in the 1st epoch then increased in the 2nd epoch, and the 3rd epoch decreased again until in the last epoch the loss was stable. The results of loss and loss validation are still underfitting.

After the training process is complete, the best weights obtained will be used in the testing process for predictions. Evaluation is carried out by comparing the predicted value and the actual value using a confusion matrix. The results of the confusion matrix can be used to calculate performance based on accuracy, sensitivity, specificity, F1-score, and Cohen's kappa which can be calculated using Equations (11) until Equation (15). The MobileNet performance results obtained accuracy, sensitivity, specificity, F1-score, and Cohen's kappa values of 84.79 %, 92.3 %, 55 %, 77.7 %, and 0.6996 respectively.

3.3 EfficientNet

In EfficientNet process used 30 epochs in training. The evaluation values generated in the training process are accuracy and loss. In the training process, accuracy in 1st epoch was 0.5010, and the loss value obtained was 0.6933 for training data. The accuracy in validation data was 0.4972 and the loss value obtained 0.6932. Based on validation loss values, the weights are stored and used for the second epoch. The validation loss values are continuously updated as a result of better validation loss values. The training process continues until the 30th epoch so that optimal weights can be determined for the testing process. Figure 7 shows the graphical results of the training and validation data accuracy of the EfficientNet model obtained during the training process.

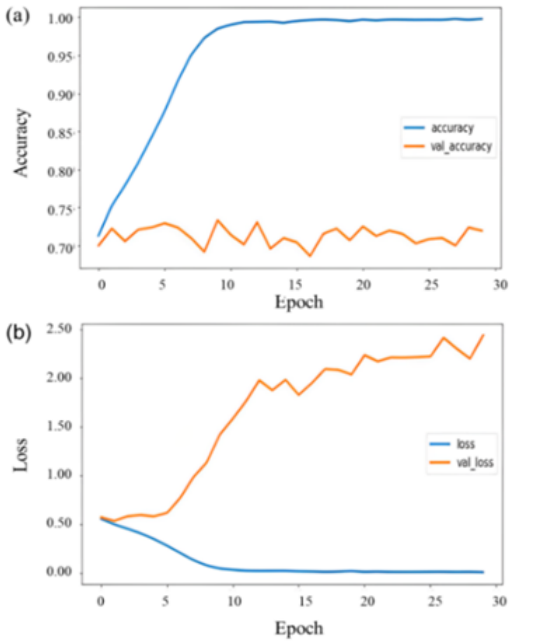

Figure 7: The (a) Accuracy and (b) Loss of the EfficientNet on the classification of diabetic retinopathy disorders based on retinal images.

Figure 7(a) shows the graph of accuracy for the EfficientNet model in training data. the results obtained during the training process increase from the 1st epoch until the 8th epoch. For the next epoch until the last epoch is a stable value of 0.98. The accuracy value for the validation data has increased and decreased from the beginning to the 30th epoch. Based on the accuracy graph, the weights of accuracy are overfitting. The accuracy of validation data should have increased. Figure 7(b) shows the graph loss of EfficientNet in training data. The loss during the training process from the 1st until the 8th epoch has decreased. From the 9th until the 30th epoch, the loss is stable. The loss value in validation data has increased from the 1st epoch until the 30th epoch. The loss value in the validation data is 2.4. The overfitting of the EfficientNet model is shown in Figure 7, where the validation loss should decrease to close to 0.

After the training process is completed, The best weight obtained during the process of testing has been used to generate predictions when the training process is over. Using a confusion matrix, the predicted value and the actual value are able to be compared to conclude the architecture evaluation. Equations (11) to (15) are able to be used to compute architecture performance based on accuracy, sensitivity, specificity, F1-score, and Cohen's kappa, which are all determined by the confusion matrix results. The results of the EfficientNet evaluation in accuracy, sensitivity, specificity, F1-score, and Cohen's kappa obtained 87.72 %, 96.2 %, 79.6 %, 88.6 %, and 0.756 respectively.

3.4 Ensemble Learning

At this stage, the final weight of each architecture was taken, and then this study used the number of repetitions (epochs) in the training as much as 50 epochs. The evaluation from the training process is accuracy and loss. Figure 8 shows the accuracy and loss graphs for training and validation data in the ensemble model. Figure 8(a) shows the accuracy graph in the ensemble learning model. The training data obtained during the training process always increases in each epoch. In 1st epoch, the accuracy obtained was 0.5236 then until the next epoch the accuracy value increased continuously towards 0.995. The accuracy of validation data always increases in each epoch. The accuracy in 1st epoch obtained 0.50 and in subsequent epochs, it always increases up to the 30th epoch towards 0.9. Figure 8(b) shows the loss value and loss validation of ensemble learning for training data during the training process always decrease in each epoch. From the 1st epoch until the 30th epoch, the loss value in training data always decreases towards 0. In validation data, the loss value from the 1st epoch until the 30th epoch always goes to 0. Based on Figure 8, the ensemble learning model is not overfitting.

After the training process is completed, The best weights will be used for the testing process for predictions. Evaluation is carried out by comparing the predicted value and the actual value using a confusion matrix. The results from the confusion matrix are used to calculate performance based on accuracy, sensitivity, specificity, F1-score, and Cohen's kappa which can be calculated using Equations (11) to (15). The ensemble learning results in accuracy, sensitivity, specificity, F1-score, and Cohen's kappa obtained 93.3 %, 93.51 %, 93.32 %, 93.42 %, and 0.866 respectively.

4 Discussion

In this study, the classification of DR disease has been used the ensemble learning on the ResNet50, MobileNet, and EfficientNet architectures. The results of the method of ensemble learning will be contrasted with a single classification used to evaluate the effectiveness of the suggested model for DR classification. Figure 9 shows a comparison of the classi-

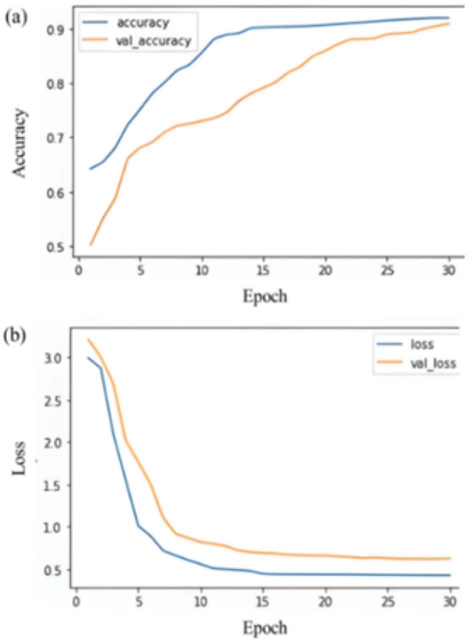

Figure 8: The (a) Accuracy and (b) Loss of ensemble learning model on the classification of diabetic retinopathy disorders based on retinal images.

fication results for single and multiclass classifications. In Figure 9, the accuracy results with the ensemble learning method are the greatest accuracy, which is 93.3 % better than the other three architectures. This means that the ensemble learning method has good performance in classifying DR. The sensitivity value of the ensemble learning method has a value of 93.51 % and the highest result is the EfficientNet architecture with a value of 96.2 %. The specificity value of the ensemble learning method has a value of 93.32 % and the highest result is the ResNet50 architecture with a value of 99.78 %. The F1 score on the ensemble learning method has a value of 93.42 %, this result is the highest compared to other architectures. The Cohens Kappa value in the ensemble learning method has a value of 86.6 %, and the highest result is in the EfficientNet architecture with a value of 89 %. The single classification ResNet50, MobileNet, and EfficientNet experience overfitting, but the ensemble method does not experience overfitting. The purpose of the comparison of results is to see how well the proposed method works. Of the 5 performance evaluation values, there are 3 highest scores from the ensemble learning method compared to the other 3 architectures. This proves that after doing ensemble learning is better than single classification. The overview of classification results with each single architecture and ensemble learning is shown in Figure 9 and the overview of the classification results of this study with other studies can be seen in Table 7.

Figure 9: Comparison of the results of the classification of each model.

Table 1: Comparison of the classification results of this study with other studies

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	Cohen's Kappa
(InceptionV3 danQIY model) [40]	90	-	-	-	-
Fuzzy random forest and dominance-based rough set [31]	80.05	81.78	79.58	-	-
Ensemble learning (InceptionV3, Resnet152, inception-ResnetV2) [30]	88.21	85.57	88.41	-	-
Ensemble Learning (Resnet50, Inception-V3, Xception, Dense 121, Dense 169) [11]	80.8	-	86.7	53.7	-
Proposed Method	93.3	93.51	93.32	93.42	0.866

Based on Table 7, the study conducted [40] only calculates the accuracy value performance. The study [31], [30] only calculated performance evaluation values for accuracy, sensitivity, and specificity. The study [11] only calculated performance evaluation values for accuracy, specificity, and F-1 Score. In Table 7, the evaluation performance result in the proposed model, it can be concluded that the accuracy for performing DR classification is very good, as shown by values of sensitivity, specificity, accuracy, and F1-Score above 90 % and higher than the study [40], [31], [30], [11]. The ability of the model to classify all data correctly is above 90 %. Based on the F1 score, the model can describe the harmonic average of the sensitivity and specificity well above 90 %. The study [40], [31], [30], [11] has not been able to calculate the Cohen's kappa value. In this study, it was possible to calculate the Cohen's kappa. The resulting Cohen's kappa value is more than 0.5 and close to 1 means that the model has a good ability to measure the level of agreement between the predicted results and the actual label that determines the state of the retinal image on a nominal scale, the model has been consistent.

5 Conclusion

Based on the results and discussion, it can be concluded that the MobileNet and EfficientNet models show good performance in a single classification, with good accuracy and sensitivity. However, the ResNet-50 model needs improvement. Ensemble learning using these three models resulted in a significant increase in accuracy, with an increase of 28.37 % compared to a single classification. Ensemble learning can also overcome overfitting that occurs in a single model. By using ensemble learning, DR classification on retinal images can be performed very well, indicated by the accuracy, sensitivity, specificity, and F1-Score above 90 %. Ensemble learning can be an effective approach to improve performance and accuracy in DR classification.

Acknowledgments

We would like to thank the Rector of Universitas Sriwijaya for funding the study. The study of this article was funded by the DIPA of the Public Service Agency of Universitas Sriwijaya 2023. Nomor SP DIPA-023.11.2.6775151/2023, On November 30, 2022. In accordance with the Rector's Decree Number: 0188/UN9.3.1/SK/2023, On April 18, 2023.

References

[1] A. M. Mutawa, S. Alnajdi, and S. Sruthi, "Transfer learning for diabetic retinopathy detection: A study of dataset combination and model performance," *Appl. Sci.*, vol. 13, no. 9, 2023, doi: 10.3390/app13095685.

[2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan , "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[3] A. Kwasigroch, B. Jarzembinski, and M. Grochowski, "Deep CNN based decision support system for detection and assessing the stage of diabetic retinopathy," in *2018 Int. Interdiscip.*, vol. 8, no. 2, pp. 111–116, 2018, doi: 10.1109/IIPHDW. 2018.8388337.

[4] D. Erdem, A. Beke, and T. Kumbasar, "A deep learning-based pipeline for teaching control theory: Transforming feedback control systems on whiteboard into MATLAB," *IEEE Access*, vol. 8, no. 1, pp. 84631–84641, 2020, doi: 10.1109/ ACCESS.2020.2992614.

[5] I. Kandel and M. Castelli, "Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review," *Appl. Sci.*, vol. 10, no. 6, pp. 1–24, 2020, doi: 10.3390/app10062021.

[6] F. Chen and J. Y. Tsou, "Assessing the effects of convolutional neural network architectural factors on model performance for remote sensing image classification: An in-depth investigation," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 112, no. 1, p. 102865, 2022, doi: 10.1016/j.jag.2022.102865.

[7] S. Kilic, I. Askerzade, and Y. Kaya, "Using ResNet transfer deep learning methods in person identification according to physical actions," *IEEE Access*, vol. 8, no. 3, pp. 220364–220373, 2020, doi: 10.1109/ACCESS.2020.3040649.

[8] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An efficient brain tumor image segmentation based on deep residual networks (ResNets)," *J. King Saud Univ. - Eng. Sci.*, vol. 33, no. 6, pp. 404–412, 2020, doi: 10.1016/j.jksues.2020.06.001.

[9] Y. Yin, Z. Han, M. Jian, G.-G. Wang, L. Chen, and R. Wang, "AMSUnet: A neural network using atrous multi-scale convolution for medical image segmentation," *Comput. Biol. Med.*, vol. 162, no. 2, p. 107120, 2023, doi: https://doi.org/10.1016/j.compbiomed.2023.107120.

[10] M. Heisler, S. Karst, J. Lo, Z. Mammo, T. Yu, S. Warner, D. Maberley, M. F. Beg, E. V. Navajas, and M. V. Sarunic, "Ensemble deep learning for diabetic retinopathy detection using optical coherence tomography angiography," *Transl. Vis. Sci. Technol.*, vol. 9, no. 2, pp. 1–11, 2020, doi: https://doi.org/10.1167/tvst.9.2.20.

[11] S. Qummar, F. G. Khan, S. Shah, A. Khan, S. Shamshirband, Z. U. Rehman, I. A. Khan, and W. Jadoon, "A deep learning ensemble approach for diabetic retinopathy detection," *IEEE Access*, vol. 7, no. 4, pp. 150530–150539, 2019, doi: 10.1109/ACCESS.2019.2947484.

[12] G. Jinfeng, S. Qummar, Z. Junming, Y. Ruxian, and F. G. Khan, “Ensemble framework of deep CNNs for diabetic retinopathy detection,” *Comput. Intell. Neurosci.*, pp. 1–11, 2020, doi: 10.1155/2020/8864698.

[13] D. K. Elswah, A. A. Elnakib, and H. El-Din Moustafa, “Automated diabetic retinopathy grading using Resnet,” in *Natl. Radio Sci. Conf. NRSC, Proc.*, pp. 248–254, 2020, doi: 10.1109/NRSC49500.2020.9235098.

[14] A. Mustapha, L. Mohamed, H. Hamid, and K. Ali, “Diabetic retinopathy classification using ResNet50 and VGG-16 pretrained networks,” *International Journal of Computer Engineering and Data Science*, vol. 1, no. 1, pp. 1–7, 2021.

[15] W. Wang, Y. Hu, T. Zou, H. Liu, J. Wang, and X. Wang, “A new image classification approach via improved MobileNet models with local receptive field expansion in shallow layers,” *Comput. Intell. Neurosci.*, vol. 2020. pp. 1–10, 2020, doi: 10.1155/2020/8817849.

[16] 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.

[17] A. Y. Sejal and M. M. Thiruthuvanathan, “Diabetic retinopathy diagnosis using retinal fundus images through MobileNetV3,” in *2023 IEEE Int. Conf. Contemp. Comput. Commun.*, vol. 1, no. 2, pp. 1–5, 2023, [Online]. URL: https://api.semanticscholar.org/CorpusID:263231104.

[18] S. Suriyal, C. Druzgalski, and K. Gautam, “Mobile assisted diabetic retinopathy detection using deep neural network,” in *2018 Glob. Med. Eng. Phys. Exch. Am. Heal. Care Exch. GMEPE/PAHCE 2018*, vol. 562, no. 2, pp. 1–4, 2018, doi: 10.1109/ GMEPE-PAHCE.2018.8400760.

[19] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, “Classification of remote sensing images using fficientNet-B3 CNN model with attention,” *IEEE Access*, vol. 9, no. 2, pp. 14078–14094, 2021, doi: 10.1109/ACCESS.2021. 3051085.

[20] F. Directions, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” *IEEE Access*, vol. 4, no. 2, pp. 1–8, 2023.

[21] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, 6105–6114.

[22] J. Wang, L. Yang, Z. Huo, W. He, and J. Luo, “Multi-label classification of fundus images with EfficientNet,” *IEEE Access*, vol. 8, no. 1, pp. 212499–212508, 2020, doi: 10.1109/ACCESS. 2020.3040275.

[23] H. Liu, K. Yue, S. Cheng, C. Pan, J. Sun, and W. Li, “Hybrid model structure for diabetic retinopathy classification,” *J. Healthc. Eng.*, vol. 20, no. 2, pp. 1–9, 2020, doi: 10.1155/2020/8840174.

[24] A. M. Pour, H. Seyedarabi, S. H. A. Jahromi, and A. Javadzadeh, “Automatic detection and monitoring of diabetic retinopathy using efficient convolutional neural networks and contrast limited adaptive histogram equalization,” *IEEE Access*, vol. 8, no. 2, pp. 136668–136673, 2020, doi: 10.1109/ACCESS.2020.3005044.

[25] T. Lawrence and L. Zhang, “IoTNet: An efficient and accurate convolutional neural network for IoT devices,” *Sensors*, vol. 19, no. 24, 2019, doi: 10.3390/ s19245541.

[26] V. C. Osamor and A. F. Okezie, “Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis,” *Sci. Rep.*, vol. 11, 1, pp. 1–12, 2021, doi: 10.1038/ s41598-021-94347-6.

[27] A. Desiani, E. S. Kresnawati, M. Arhami, Y. Resti, N. Eliyati, S. Yahdin, T. J. Charissa, and M. Nawawi, “Majority voting as ensemble classifier for cervical cancer classification,” *Sci. Technol. Indones.*, vol. 8, no. 1, pp. 84–92, 2023, doi: 10.26554/sti.2023.8.1.84-92.

[28] V. Deepa, C. S. Kumar, and T. Cherian, “Ensemble of multi-stage deep convolutional neural networks for automated grading of diabetic retinopathy using image patches,” *J. King Saud Univ. - Comput. Inf. Sci.*, pp. 1–11, 2021, doi: 10.1016/j.jksuci.2021.05.009.

[29] B. Antal and A. Hajdu, ‘An ensemble-based system for automatic screening of diabetic retinopathy,” *Knowledge-Based Syst.*, vol. 60, no. 2, pp. 20–27, 2014, doi: 10.1016/j.knosys. 2013.12.023.

[30] H. Jiang, K. Yang, M. Gao, D. Zhang, H. Ma, and W. Qian, “An interpretable ensemble deep learning model for diabetic retinopathy disease classification,” in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 2045–2048, 2019, doi: 10.1109/EMBC.2019.8857160.

[31] E. Saleh, J. Błaszczyński, A. Moreno, A. Valls, P. Romero-Aroca, S. D. L. Riva-Fernández, and R. Słowiński, “Learning ensemble classifiers for diabetic retinopathy assessment,” *Artif. Intell. Med.*, vol. 85, no. 1, pp. 50–63, 2018, doi: 10.1016/j.artmed.2017.09.006.

[32] M. B. Hossain, S. M. H. Sazzad, M. M. Islam, M. N. Akhtar, and I. H. Sarker“Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images,” *Informatics in Medicine Unlocked*, vol. 2, no. 1, pp. 1-10, 2020.

[33] S. H. Kassani, P. H. Kassani, R. Khazaeinezhad, M. J. Wesolowski, K. A. Schneider, and R. Deters, “Diabetic retinopathy classification using a modified xception architecture,” in *2019 IEEE 19th Int. Symp. Signal Process. Inf. Technol. ISSPIT 2019*, pp. 0–5, 2019, doi: 10.1109/ ISSPIT47144.2019.9001846.

[34] S. Phiphiphatphaisit and O. Surinta, “Food image classification with improved MobileNet architecture and data augmentation,” in *ACM International Conference Proceeding Series*, 2020, vol. 2, no. 4, pp. 51–56, doi: 10.1145/338 8176.3388179.

[35] G. Marques, D. Agarwal, and I. de la Torre Díez, “Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network,” *Appl. Soft Comput. J.*, vol. 96, no. 2, p. 106691, 2020, doi: 10.1016/j.asoc.2020. 106691.

[36] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively pruned deep learning ensembles for COVID-19 detection in chest x-rays," *IEEE Access*, vol. 8, no. 2, pp. 115041–115050, 2020, doi: 10.1109/ACCESS.2020.3003810.

[37] R. Yin, Z. Luo, P. Zhuang, Z. Lin, and C. K. Kwoh, "VirPreNet: A weighted ensemble convolutional neural network for the virulence prediction of influenza A virus using all eight segments," *Bioinformatics*, vol. 37, no. 2, pp. 737–743, 2021, doi: 10.1093/bioinformatics/btaa901.

[38] S. Bharati, P. Podder, R. Mondal, A. Mahmood, and M. Raihan-Al-Masud, "Comparative performance analysis of dsifferent classification algorithm for the purpose of prediction of lung cancer," *Advances in Intelligent Systems and Computing*, vol. 941, no. 2, pp. 1-8, 2020.

[39] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews correlation coefficient (MCC) is more informative than Cohen's kappa and Brier score in binary classification assessment," *IEEE Access*, vol. 9, no. 1, pp. 78368–78381, 2021, doi: 10.1109/ACCESS.2021.3084050.

[40] M. T. Hagos and S. Kant, "Transfer learning based detection of diabetic retinopathy from small dataset," 2019, doi: https://doi.org/10.48550/arXiv.1905.07203.