IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 July 2024, accepted 21 July 2024, date of publication 24 July 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432867

RESEARCH ARTICLE

Enhancing Early-Stage Diabetic Retinopathy Detection Using a Weighted Ensemble of Deep Neural Networks

KINZA NAZIR[1], JISOO KIM[2], AND YUNG-CHEOL BYUN[3]
[1]Department of Electronic Engineering, Institute of Information Science and Technology, Jeju National University, Jeju-si 63243, South Korea
[2]Faculty of Software (Artificial Intelligence Major), College of Engineering, Jeju National University, Jeju-si 63243, South Korea
[3]Department of Computer Engineering (Major of Electronic Engineering), Institute of Information Science and Technology, Jeju National University, Jeju-si 63243, South Korea

Corresponding author: Yung-Cheol Byun (ycb@jejunu.ac.kr)

This work was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2023RIS-009.

ABSTRACT Diabetic Retinopathy (DR) is one of the biggest reasons for vision loss. It is a fatal eye disease damaging the retina, which is the light-sensitive tissue in the rear of the eye. Ophthalmologists use fundus images to capture retinal inner structures to find broken blood vessels and scars. To detect DR on time, early diagnosis is very important which is often not possible due to complex procedures. Therefore, automation of DR detection can solve this problem. Accessibility to regular examinations and specialized eye care remains a challenge, especially in underserved areas, due to late diagnosis, a lack of healthcare infrastructure, and other factors. Although automated detection and grading of diabetic retinopathy from retinal images has shown promising results, challenges arise to accomplish high accuracy, particularly in hidden or early-stage DR situations. One of the major limitations in developing a detection model is the lack of imaging datasets as it requires a large number of images to train the model more accurately. Deep transfer learning-based models have shown promising results especially when datasets are not very large. This study used a weighted average ensemble approach to combine three different deep learning models: inception-v3, VGG16, and a custom-built convolutional neural network. The proposed weighted average ensemble approach achieved an accuracy of 95.06%, a precision of 87.88%, a recall of 83.78%, f1-score of 85.69%, and a 98.10% area under the curve which is higher compared to other pre-trained models. A comprehensive comparative analysis is done to compare the proposed approach with other state-of-the-art methods. The proposed system is efficient, considerably accurate, and can aid as a clinical assistant to detect and grade diabetic retinopathy.

INDEX TERMS Diabetic retinopathy, retinal fundus images, multi-class classification, deep learning, weighted average ensemble.

I. INTRODUCTION

A significant contribution of computer-aided diagnosis (CAD) in medical applications is the detection and grading of retinal illness. Retinal macular eye disease has a significant impact on the patient's life because it causes permanent vision loss. CAD techniques have been used to investigate the fundamental behavioral characteristics of biological models for over ten years. The primary focus of automated detection systems is visual assessment using optical coherence tomography (OCT) images. Detecting retinal diseases used to be a laborious, manual process that required a doctor to examine and accurately diagnose the condition. Automated detection methods are faster, easier, more effective, and require less time.

Diabetic retinopathy is a disease which is caused by diabetes that results in visual impairment or permanent vision loss. DR is caused by broken or clotted blood vessels

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

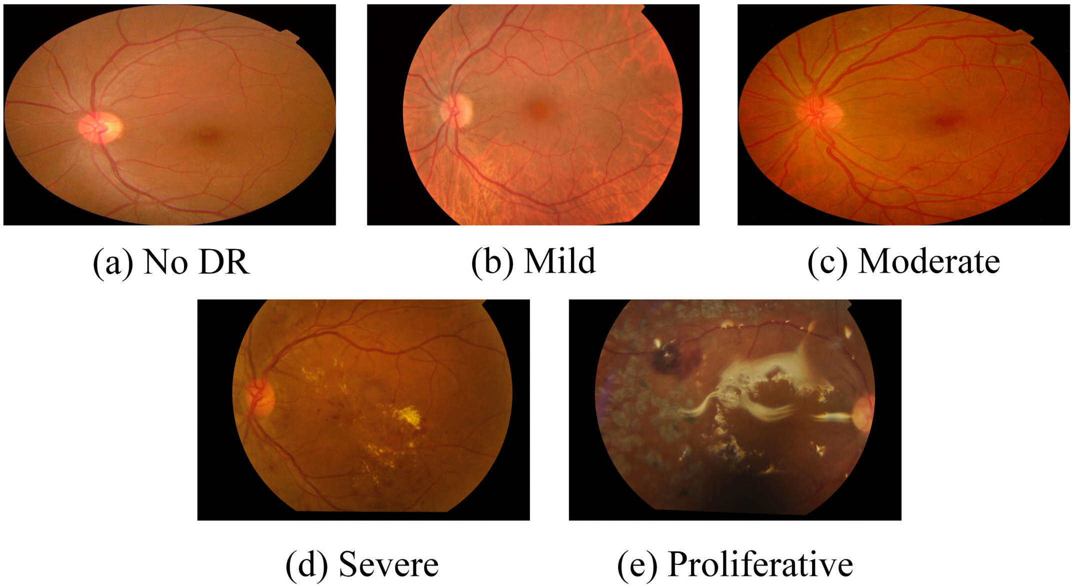


FIGURE 1. Different diabetic retinopathy stages in fundus images.

in the photosensitive retina. According to the international diabetes federation, the total number of diabetic patients is approximately 415 million. 145 million patients out of these 415 million are suffering from some stage of diabetic retinopathy. By 2030, the number of DR patients is expected to rise to 224 million [1]. It is very important to diagnose DR in its early stages because if detected early, treatment is possible. Medical research shows that if there are abnormal lesions in the retina, it means that the person is in one of the five stages of diabetic retinopathy and needs immediate treatment [2], [3]. There are five stages of diabetic retinopathy. The first stage is Mild DR also called background retinopathy. In this stage, a very small area of the retinal blood vessel is swollen. The slight bulges called microaneurysms are formed in blood vessels. This is the first stage and if detected on time, the patient can be treated for visual impairment. The second stage is Moderate. In this stage, a part of the retinal blood vessel is clotted and no longer moves blood. These changes could result in diabetic macular edema (DME). The third stage is called severe stage. In this stage, a huge amount of blood vessels become clotted and blocked [4]. This suggests that the retinas get a lot less blood. Scar tissue forms as a result. In reaction to a lack of blood, the retinas get a signal to produce new blood vessels. If the patient reaches this stage, there is a very high risk that they will lose their vision. With treatment, vision loss may be halted [5]. In the fourth stage called proliferative, the new retinal blood vessels are affected and there is heavy blood leakage in the retina and the gel-like fluid fills the eyeballs. This eventually causes blindness. No DR is when the patient is healthy and has no diabetic retinopathy [6].

DR is diagnosed using fundus images, which require highly skilled experts to manually analyze. When a medical specialist manually examines each fundus photograph, there is a chance of misdiagnosis. Computer-assisted disease detection can provide accurate results in a timely and efficient manner [7]. Despite the alarming statistics, the main issue is that diabetics have no symptoms or warning signs. To provide early diagnosis and care, experienced ophthalmologists are required for the diagnosis of diabetic retinopathy. In recent years, retinal imaging has grown in importance for detecting eye problems. It is necessary to be able to externally observe the retina, as well as brain tissues, blood vessels, and retinal problems, to diagnose ocular disorders. Fluorescein angiography imaging, fundus imaging, and spatial domain optical coherence tomography images (OCT) are currently used to diagnose DR and other retinal conditions [8], [9]. Many detection methods have been created during the past few years using a variety of deep learning and machine learning techniques. Studies have shown that several machine-learning approaches classify retinal pictures into DR-affected images and normal images using a hand-crafted feature. For the creation of the diagnostic system, researchers use the machine learning (ML) techniques K-nearest neighbors [10], histogram orient gradient descriptors [11], Support vector machine, principal component analysis [12], and local binary pattern.

Traditional methods for classifying diabetic retinopathy have significant limitations, prompting researchers to look into more advanced techniques like machine learning and deep learning. Existing methods frequently rely on ophthalmologists manually examining and interpreting retinal images, which is time-consuming and may need to be more scalable to efficiently handle massive amounts of patient data, problems in early detection, it becomes more difficult to handle and evaluate large collections of retinal images;

frequently need more robustness and accuracy. Because of these drawbacks, the literature demonstrates that researchers' attention has switched to the deep learning detection system that is automated. The principal methods of deep learning used in previous studies are of two types: morphological approaches and neural networks. Deep learning is becoming very popular in medical image classification problems. Deep convolution neural networks DNNs are performing significantly well in the computer vision field. In the past decade, many researchers have come up with novel CNN architectures for image classification. Some of the popular models include MobileNet, VGGNet, GoogleNet, InceptionNet, and XceptionNet among many others. These architectures have proved to make immense progress in many areas such as cancer detection, glaucoma screening, and pneumonia detection among others [13]. This study aims to develop a fully automated diabetic retinopathy severity level grading system using a novel ensemble learning technique on the Kaggle APTOS 2019 dataset. Following are the major contributions of the suggested model:

- This study evaluates different deep learning models to determine the top three models for the diabetic retinopathy classification task.
- A weighted average ensemble model is proposed combining VGG16, inception-v3, and a custom convolutional neural network as base learners, leading to high model performance on diabetic retinopathy severity level classification.
- The proposed study applied several pre-processing steps to improve the quality of fundus images and compared the weighted average ensemble model with other state-of-the-art models.
- The automated detection system will help in the early diagnosis of the disease and may reduce the number of vision loss in diabetes patients.
- The proposed system is a robust, efficient, and considerably accurate system and it will be a significant contribution to the global healthcare field.

The remaining article is organized as follows. Section II discusses previous related work for diabetic retinopathy detection and grading. Section III and IV presents material and methods including the proposed methodology, dataset acquisition, pre-processing techniques, and proposed model framework. Section V presents the results and proposed model comparison with prior related work and discussions of the study. Section VI concludes the research study.

II. RELATED WORK

Several studies have been made for diabetic retinopathy severity level classification. Many researches explored CNN based techniques to classify DR images. Studies shows that the deep learning techniques performs relatively well and give better accuracy since it is difficult to find large medical datasets. There are many CNN based automated detection methods. The authors proposed a multistage deep learning method. Conventional CNN was used with a feature extractor and decoder [22]. The CNN trained on ImageNet was used as encoder initialization. The three decoders were the classification head, ordinal regression head, and regression head. The datasets used were a combination of Messidor, IDRiD, and APTOS 2019. The binary classification was performed to achieve the best results with a quadratic kappa score of 92.5. In [23] a shallow neural network with nine blocks of layers was proposed. According to the results, the model performed well in frequent classes but worst in the least frequent classes. For the second approach transfer learning with Efficient-B3 was used which performed better on the least frequent classes as compared to the shallow neural network. The shallow network achieved an accuracy of 69.03% while Efficient-B3 scored 77.87% accuracy. Researchers in [24] proposed a hybrid machine-learning architecture that contained two blocks. The first block was the detection algorithm and the second block was the grading algorithm. The detection algorithm discussed whether the patient has DR and the grading algorithm told the severity level. In [25] a convolutional neural network with 18 convolutional layers along with fully connected layers was proposed. Several pre-processing and data augmentation techniques were used to maximize the performance of the model. The model achieved a validation accuracy of 88% when evaluated on the Kaggle dataset APTOS2019. The researchers in [26] used CNN512 which gave an accuracy of 88.6% and 84.1% for DDR and APTOS respectively. The second model used in this study is YOLOv3. For the final step, both models were fused together to classify the images obtaining 6 accuracies of 89% and 89% respectively. The authors in [27] used a composite DNN. The main technique used was the gated-attention mechanism. The model was trained and tested on APTOS 2019. The model was compared with some other models and recorded an accuracy of 82.54%.

Several studies explored ensemble learning to detect lesions in retinal fundus images. The researchers in [28] proposed a novel approach. The information extracted from colored fundus images was fed to the ensemble learning algorithm. In the first step, all the noisy images were discarded followed by image crop, resizing, and augmentation. In the next step image preprocessing was applied followed by feature extraction using an image histogram. For ensemble learning the ET classifier was used. The dataset used for model evaluation was APTOS2019. The results showed 91.07% accuracy.

Deep learning has become immensely popular among researchers for the past decade due to the lack of large datasets in medical imaging. It is hard to find large fundus image datasets as it requires ophthalmologists to carefully read and label the images. In one of the early studies on APTOS 2019, [29] the author proposed a model which comprised two main steps. In the first step, the features were extracted from DenseNet 100. In the second step, the classification was performed using CenterNet. The architecture was evaluated on APTOS 2019 and IDRiD. The model performed well achieving promising results. The authors in [30] used

TABLE 1. Prior work on DR classification based on different ML techniques.

Ref	Year	Model	Highlights	Limitations
[14]	2019	Random forest feature selector + PCA	The model reduced the dimensionality by asking subsets of input features from highest to lowest.	Features are extracted and classified from single location within an eye. AUC is an only metric used for model evaluation.
[15]	2023	VGG16, VGG19, Resnet50, Densenet	Different variants of pretrained models were used and results were compared.	No significant pre-processing technique is used and the models achieved poor accuracies.
[16]	2022	CNNs with multiple optimizers	The model used Adagrad, RMSPROP with momentum and Adam optimizer and compared their performance with proposed model.	The model performed well on binary class classification but its effectiveness on multi-class classification is not defined.
[17]	2023	A classifier with wrapper-based feature selection approach including k-Nearest Neighbour technique and genetic algorithm	2D wavelength transform is applied to extract features than high level features are selected with wrapper approach. The model achieved low computational cost with average accuracy.	the model's reliance on entropy-based features can limit its application in broader context.
[18]	2021	SVM and random forest	Pre-processing was done through histogram equalization then DR images were segmented via k-mean clustering.	The dataset contained only 89 images with 84 images belonging to same class. manual segmentation and pre-processing may introduce potential biases. Also lack of evaluation metrics may result in overfitting and generalization.
[19]	2024	Cat swarm optimization with explainable AI framework	OptiDex model utilized enhanced cat swarm algorithm and explainable AI to maximize model performance.	The model needs computational extensive resources for data training.
[20]	2024	HPLBO_DMN	High level pre-processing and wavelet transform along with HPLBO_DMN.	Computational comlexity is not mentioned, which can hinder the model's practical use on smart devices.
[21]	2020	CNNs with SVM, Adaboost, J48, Random forest and Naive Bayes	Features were extracted via novel CNN model and given to five different classifiers as an input.	While the suggested model demonstrated computational efficiency, it lacks comparison with other advanced techniques. Additionally, its ability to generalize in real-world scenarios remains uncertain.

the modified version of Xception network. The authors compared the result of modified Xception with InceptionV3, ResNet50, MobileNet, and the basic Xception model. Tuning of hyperparameters was also applied in the last step to further improve the performance of the proposed model. The model was evaluated on the APTOS2019 dataset. The modified Xception network scored a total accuracy of 83.09%.

Researchers in [31] used InceptionV3 and Xception for the classification task. The Gaussian blur was used as a preprocessing technique with several augmentation methods. InceptionV3 and Xception achieved 91.90% and 93.10% accuracy respectively. Reference [32] used Hinge Attention Network with VGG16 as the base model. The model was validated against benchmark datasets APTOS 2019 and IDRiD. The model achieved 85.54% and 66.41% accuracy for APTOS and IDRiD datasets respectively. The researchers in [33] proposed a hybrid model by using Inception-ResNet-v2 and adding a custom CNN block. They used Messidor-1 and APTOS 2019 as validation datasets and scored an accuracy of 72.33% and 82.18% on Messidor-1 and APTOS 2019.

In [34] VGG16 is used as a base model and then several other fusion and pooling methods were used for activation filter values extraction. The model was trained and tested on APTOS and achieved an accuracy of 84.31% and 97 AUC. In [19] the authors used a modified Densenet121 architecture. The dataset used in this study was 8 APTOS 2019. The model was tested against multi-labeled classification and binary classification. For multi-label, the model achieved 96.51% accuracy while for binary classification it scored 94.44% of accuracy. The Xception model [35] has been used to extract multi-level features fed to multi-level perceptron (MLP) for diabetic retinopathy grading. They compared the model with Inceptionv3, MobileNet, ResNet50, and simple Xception. The dataset used to validate the model was APTOS 2019. The model achieved 83.09% accuracy.

There are some limitations in existing approaches that are needed to be improved [36] such as high computation and low efficiency. Another hindrance is the fact that CNN models need large training samples to train the model which is hard to obtain from an actual domain. Consequently, transfer learning (TL) achieved a substantial potential to apply DL architectures that have already been trained on imageNet for DR classification. Pre-trained models, including AlexNet, mobileNet, ResNet, and others do have certain drawbacks such as intricate designs [37] with multiple additional hyper-parameters and low computational overhead efficiency.

Therefore, in this study, a novel ensemble model has been proposed to classify the DR fundus images into 4 stages namely No DR, moderate DR, severe and proliferative DR. Additionally the proposed ensemble model has been compared with existing classification techniques to show the efficiency of our model.

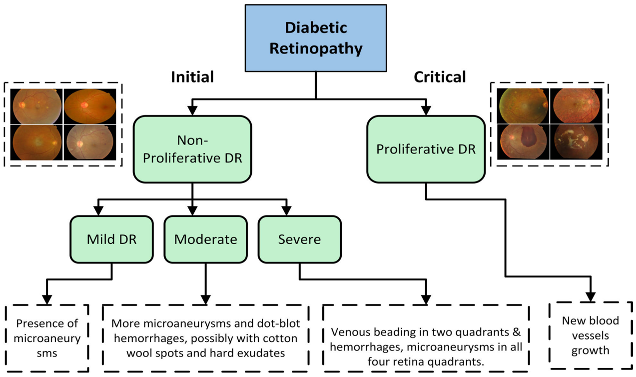


FIGURE 2. Illustration of the various forms of Diabetic Retinopathy (DR) and their corresponding clinical manifestations.

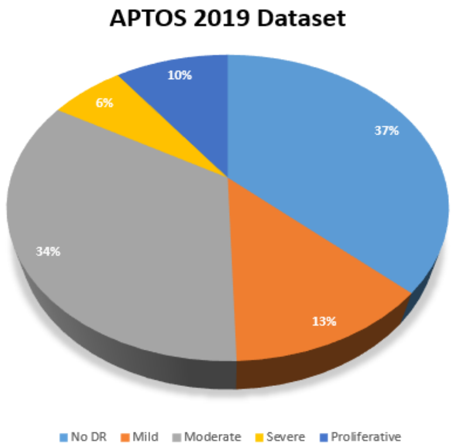


FIGURE 3. Graphical representation of image distribution of APTOS 2019 dataset.

TABLE 2. APTOS 2019 DR image distribution count.

Diabetic Retinopathy Stage	Image Count
No DR	1085
Mild NPDR	370
Moderate NPDR	999
Severe NPDR	192
Proliferative DR	295

III. MATERIALS

This section presents the materials utilized in our study, comprising the dataset description and preprocessing techniques. The dataset for this study is collected from the Kaggle repository. Due to the varied lighting conditions and exposure settings under which these images were captured, preprocessing is essential to standardize the dataset. Additionally, data augmentation is employed to increase the diversity of the training set and mitigate overfitting. These processed images are then used to train our base learners, facilitating more robust and generalized model performance.

A. DATASET

The dataset used to implement, validate, and test the proposed model is APTOS 2019. It is a Kaggle competition dataset that was issued by the Asia Pacific Tele-Ophthalmology Society in 2019 [38]. The images were then examined by a team of experts and categorized into five diabetic retinopathy stages [39]. These stages include No-DR, mild, moderate, severe and proliferative DR. Fig.2 shows the various types of diabetic retinopathy and their corresponding clinical manifestations. The dataset consists of 3662 training images and 1992 test images. The labels for test images are publically unavailable as it is a competition dataset so labels are reserved for final assessment and thus not used in this research. The images are of different sizes and contain noise and other artifacts as they were captured under different lighting conditions over a certain period. Table 2 shows the distribution of images in each class.

B. PRE-PROCESSING

According to [40] the APTOS 2019 dataset is gathered from various rural regions under various lighting and exposure conditions. The dataset contains noise as well as other unneeded artifacts. There is a skewed relationship between resolution, zoom level, image crop, and so on. As a result, the pre-processing step is critical. We applied number of pre-processing steps to improve the quality of raw images. The photographs are taken from various patients and under various lighting conditions. Some images are extremely dark, while others are extremely bright. To improve the visualisation of the images, we used Ben Graham's [41] technique, which he used in the 2015 Kaggle competition for diabetic retinopathy detection on the EyePACS dataset. Ben Graham's technique aids in the resolution of various lighting issues. We subtract the intensity of each image from the weighted mean of its neighbor pixels' intensities or pixel values. The images are then grey-scaled to 50% and the local color is removed. For the next step, we used OpenCV Gaussian blur to smoothen the images around the corners. The value of kernel standard deviation along the X-axis (sigmaX) is set to 10. All the images were converted into RGB as theData Generator takes input in RGB. The black area around the retina image can cause inaccurate results so we cropped the images to get rid of black areas. Fig.4 shows the input images after the final pre-processing steps.

When we deal with deep learning, we need to process a large amount of data to find the best results. Data augmentation stretches the original dataset by applying multiple transformations such as rotation, zooming, flipping the images etc. The APTOS 2019 dataset is imbalanced since the majority of images are in class 0 as normal images with no DR. To address this problem and avoid overfitting, data augmentation is used. For data augmentation, we used a built-in Keras function called ImageDataGenerator. Random horizontal and vertical flips are used to bring diversity in training data. The rotation range is set to 360 which rotates the

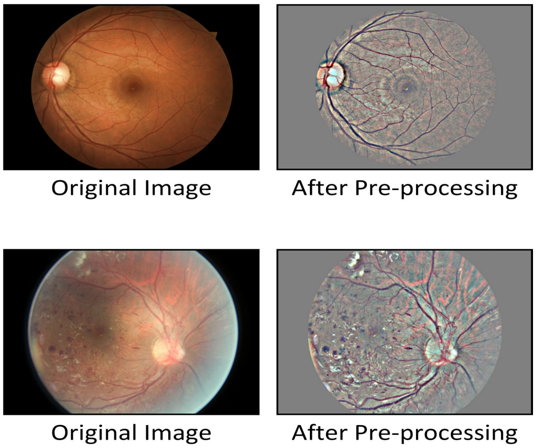


FIGURE 4. **Fundus images before and after pre-processing.**

TABLE 3. **Detailed description of data augmentation.**

Transformation	Description
Random Flip	Vertical and Horizontal
Rotation Range	360
Shear Range	0.1
Zoom Range	0.2

data randomly between 0 to 360 while training the model. The zoom range value is set to 0.2 which magnifies the images. Shear range fixes one axis and tilts the image. We set the shear range value to 0.1. The vertical and horizontal flip ensures the randomized flipping during data training. In addition, every image has square padding to prevent aspect ratio loss. The image resolutions are lowered to the default input sizes of the model architecture during the posterior resizing, which are 224 × 224 pixels for CNN and VGG16 and 299 × 299 pixels for Inception-v3. Table 3 shows the details about data augmentation techniques applied to address the class imbalance problem.

IV. METHODS

This section presents the detailed methodology of our proposed model. An overview of the proposed scheme is shown in Fig.5. Our model is comprised of two stages. In the first stage, the DR images are pre-processed via Ben Graham's pre-processing. We then cropped the images to get rid of black areas and applied Gaussian blur. In the next step, data augmentation is applied to tackle the data imbalance problem. In the second stage, we trained three models CNN, Inception-v3 and VGG16 on training images. By stacking convolution, pooling, and non-linear activation functions, our CNN model processes the data, extracts the features, and enables layer-by-layer subtraction. The output is fed to the J48 classifier for the final prediction. We used Inception-v3 and VGG16 as our second and third models before feeding the extracted features to softmax classifiers for each model. Finally based on the weights given to the three models, a weighted average ensemble is trained to produce final predictions.

A. BASE LEARNERS SELECTION

We tested a wide variety of pre-trained models, such as Inception-v3, Xception, ResNet50, VGG16, MobileNet, and DenseNet201, in addition to a custom CNN model integrated with a J48 classifier, in order to choose base learners for our ensemble model. We thoroughly evaluated each model's performance through training and careful performance evaluation on our DR dataset. Notably, the most promising choices were Inception-v3, VGG16, and our custom CNN model, which showed better prediction accuracy and robustness than the others. Therefore, we opted to use these three models as the foundational learners in our ensemble, taking advantage of their better performance to enhance the overall predictive capability of our model.

B. CUSTOM CNN ARCHITECTURE

We employed Convolutional Neural Networks (CNNs) as a fundamental part of our ensemble learning strategy for the classification of diabetic retinopathy. Our custom-created CNN architecture is tuned to extract complex patterns that point to the condition's accompanying retinal problems. When combined with Inception-v3 and VGG16 in our ensemble framework, the CNN improves feature extraction and helps achieve better classification performance.

Table 4 shows the trainable parameters for layer-wise feature extraction of the CNN architecture which is shown in Fig. 6. The network used six convolutional layers and two fully-connected layers. Thirty-two pertinent features are extracted from the final fully linked layer and used in the classification process. Before importing the images into CNN, they must be properly resized. The input fundus image is enlarged in the proposed work to 227 × 227 × 3 pixels, which correspond to the input fundus image's height, width, and three color channels that describe its depth. The CNN's job is to simplify the incoming visual data without sacrificing any important characteristics in order to produce an accurate forecast. Below is a description of each layer's purpose in the CNN [42].

- The output of neurons is calculated by the convolutional layer as a dot product of a tiny area of the image with the relevant weights. This technique is performed all the way across the length and width. The parameter sharing mechanism is used by these layers to control the quantity of parameters.
- Here, the most basic non-linear activation function is used in the Rectified Linear Unit (ReLU) layer. This layer applies the function ***f(k) = max(0,k)***, where the input from the neuron is k to introduce non-linearity into the system and replaces any negative activations with 0.
- By reducing the amount of inputs to the subsequent feature extraction layer, the pooling layer enables us to create many more distinct feature maps. One technique for discretization based on samples is max pooling. The

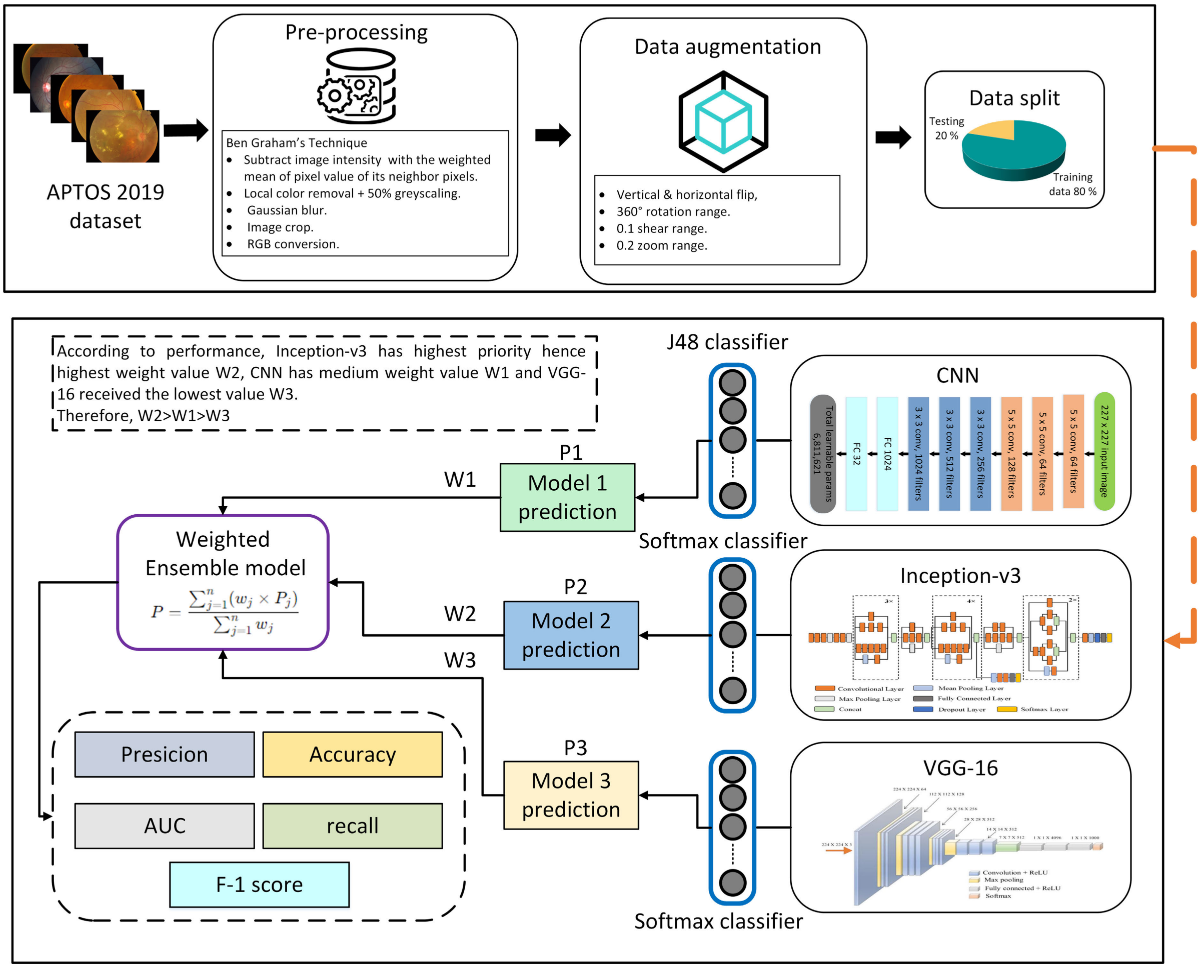


FIGURE 5. **Proposed model overview: A precise insight into model's core elements and methodology, offering a broader understanding of the workflow, models used and intended impact on the DR classification task.**

objective is to downsample an input representation in order to lower its dimensionality and make assumptions about the characteristics that are present in binned subregions. Applying a max filter to the non-overlapping subregions of the original representation are how max pooling is done.

- Regular neural networks display fully connected layer neurons, which have full connections to all previous layer activations. As a result, their activations can be computed by multiplying a matrix and adding a bias offset.

The parameters are trained using a backpropagation algorithm taking cross-entropy as the loss function. In addition to this, Batch normalization [43] dropout strategy, ***L1, L2*** regularization are included to avoid overfitting. The total number of learnable parameters in the proposed network architecture is 69,52,728.

TABLE 4. **Parameter count and tensor size in our custom CNN architecture.**

Layer name	Tensor size	Number of parameters
Input image	$227 \times 227 \times 3$	0
Conv2d_1	$227 \times 227 \times 64$	204928
maxpooling2d_1	$114 \times 114 \times 64$	0
Conv2d_2	$114 \times 114 \times 64$	295168
maxpooling2d_2	$57 \times 57 \times 64$	0
Conv2d_3	$57 \times 57 \times 128$	590080
maxpooling2d_3	$29 \times 29 \times 128$	0
Conv2d_4	$29 \times 29 \times 256$	1180160
maxpooling2d_4	$15 \times 15 \times 256$	0
Conv2d_5	$15 \times 15 \times 512$	2359808
maxpooling2d_5	$8 \times 8 \times 512$	0
Conv2d_6	$8 \times 8 \times 1024$	525312
maxpooling2d_6	$4 \times 4 \times 1024$	0
FC_1	1024×1	1049600
FC_2	32×1	32800

Trainable parameters: 6,811,621

1) J48 CLASSIFIER

J48 is the C4.5 Decision tree's open-source Java implementation. The primary purpose of this decision tree method is data mining. This selects each node test feature by evaluating the information gain ratio. We call this process feature selection. The feature that gains the most information while functioning will be chosen as the current node's test feature. Assume that

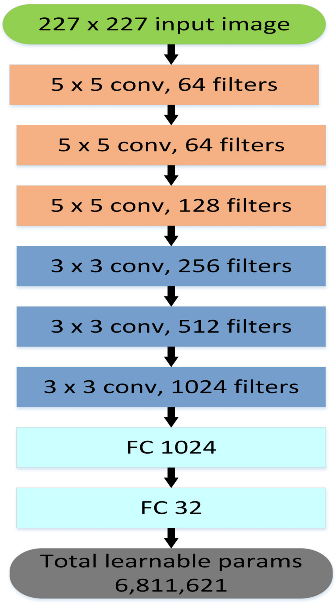


FIGURE 6. Proposed CNN model architecture.

F is a collection of input feature vectors that the classifier receives and that it comprises ***F***1, ***F***2,..., ***Fn*** instances.

Assume that each of the ***t*** separate classes ***Ci (where i = 1, 2, ..., n)*** has ***k*** distinct values. Next, using (1), the gain ratio ***GA*** of sub-attribute ***A*** in each attribute may be determined.

$$G_A = G(A)/S_A(F). \tag{1}$$

where, as indicated by (2), ***G(A)*** is the information gain of attribute ***A*** and may be found by taking the difference between the attribute information ***IA(D)*** and the total information ***I(D)***.

$$G(A) = I(D) - I_A(D) \tag{2}$$

(3) and (4) can be used to compute ***I(D)*** and ***IA(D)***, respectively, if ***Pi*** is the distinct class probability.

$$I(D) = \sigma P_i \log_2(P_i) \tag{3}$$

$$I_A(D) = -\Sigma \frac{|F_J|}{|F|} I\left(F_j\right) \tag{4}$$

The (5) shows the split information:

$$S_A(F) = -\Sigma \frac{|F_j|}{|F|} \log\left(\frac{|F_j|}{|F|}\right) \tag{5}$$

The weight of the ***jth*** partition is represented by the absolute value of the fraction |***Fj***|/|***F***|. These formulas are used to build the C4.5 decision tree, which produces appropriate categorization conditions. These conditions determine how the input feature vector is classed during testing.

C. FINE-TUNED INCEPTION-V3

Inception-v3 is the third version of InceptionNet, also known as GoogleNet. The Inception-v3 has proven to be effective for medical imaging issues. The popular Imagenet dataset is used to train the Inception-v3. On the ILSVRC 2012 benchmark, it outperforms GoogleNet [44]. Inception-v3 combines several convolutional filters into a single convolutional filter. This step aids in the reduction of trainable parameters. Inception-v3 basic architecture is depicted in Fig.8. We proposed a custom block of layers after the inception layers. A customized set of global average pooling layers, batch normalization layer, dense layer, dropout layer, and softmax layer is used after the Inception block. The batch normalization layer is the first layer. The batch normalization layer is used to standardize the Inception-v3 block's output. Because our dataset is imbalanced, batch normalization reduces the likelihood of model overfitting because it is used for regularization.

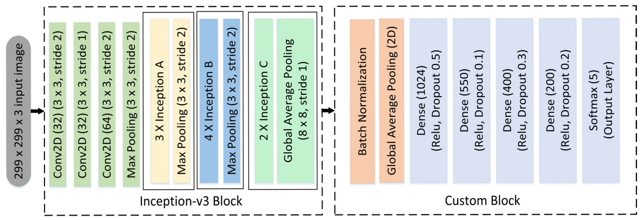


FIGURE 7. Fine-tuned Inception-v3 architecture.

Batch normalization provides each layer with an independent learning opportunity. The batch normalization layer normalized the output of the Inception block's previous layers. The next layer used is the global average pooling layer 2d, which down-sampled features extracted from pre-processed input images. We replaced the fully connected layers on the Inception block with the global average pooling 2d layer instead of the flattened layer. Following the global average pooling layer, 5 dense layers were used as fully connected layers. Each dense layer's neurons are linked to every neuron in the previous layer. The output size for the first dense layer is set to 1024 units. The rectified linear unit function ReLu is used in the model to add non-linearity. Following each dense layer, a dropout layer is used to turn off some of the neurons in order to increase the model learning rate. The dropout value for the first layer is set to 0.5. The output size for the second dense layer is set to 550 units. The dropout rate in the second dropout layer is 0.1. The output size for the third dense layer is 400 units, and the dropout layer value is 0.3. The output size in the fourth dense layer is 1024 units, with a dropout layer value of 0.4. The output parameter and dropout rate for the 5th dense and dropout layers are 200 and 0.2, respectively. The last layer is the softmax layer which is modified according to the number of classes in the APTOS 2019 dataset which is 5.

D. FINE-TUNED VGG16

VGG16 won ILSVR ImageNet competition in 2014. It has 16 layers in total with their weights. Thirteen out of these 16 layers are convolutional layers, with five max pooling

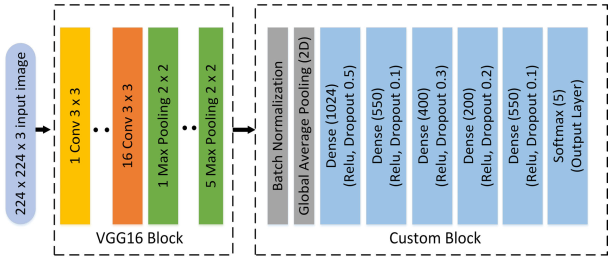


FIGURE 8. Fine-tuned VGG16 architecture.

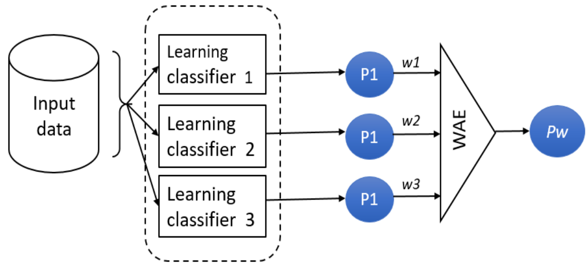


FIGURE 9. Weighted average ensemble (WAE) model structure.

layers. Then comes three dense layers. These layers are 21 in total but only 16 layers out of these have weights or in other terms, trainable parameters. It takes 224 × 224 input size along with 3 RGB channels. The convolutional layers in VGG16 are of 3 × 3 filter and stride equals to 1. The VGG16 model uses the same padding through each layer. The max pooling layer has 2 × 2 filter along with stride equals 2.

We proposed a custom block of layers following the VGG16 architecture. After the VGG16 block, a specially designed collection of layers is used, comprising batch normalization, global average pooling, dense, dropout, and softmax layers. Because our dataset is unbalanced, the batch normalization layer is positioned as the first layer to standardize the output of the VGG16 block, reducing the danger of overfitting. By normalizing the output of the layers that come before it in the VGG16 block, batch normalization enables separate learning chances for every layer. Next, features taken from pre-processed input images are down-sampled using the global average pooling layer 2D, which takes the place of the conventional fully connected layers in the VGG16 block. Five dense layers are integrated as fully connected layers after the global average pooling layer. The neurons in each dense layer are linked to the neurons in the layer above it. The first dense layer has 1024 units and introduces non-linearity using the rectified linear unit (ReLU) function. In order to improve model learning rates, dropout layers are merged after each dense layer, inhibiting certain neurons in the process. The dropout rate for the first dense layer is fixed at 0.5. The second dense layer's output size is then changed to 550 units with a 0.1 dropout rate. Next, the third dense layer's output size is set to 400 units, and its dropout rate is set to 0.3. The output size and dropout rate for the fourth dense layer are 1024 units and 0.4, respectively. Finally, 200 units and 0.2 are the output parameters and dropout rates that are customized for the fifth dense and dropout layers, respectively. The softmax layer, the last layer in the architecture, has been modified to support the APTOS 2019 dataset's unique class count of five.

E. ENSEMBLE LEARNING

By combining the predictions of multiple classifiers, ensemble learning has become a potent technique for image classification, providing better performance than individual classifiers. When it comes to classifying images, ensemble

Algorithm 1 Pseudocode for Training Ensemble Model

Input: Fundus images from APTOS 2019 dataset
$X = \{x_1, x_2, \ldots, x_n\}$ (images)
$Y = \{y_1, y_2, \ldots, y_n\}$ (labels)
Output: Trained ensemble model M
K: No. of epochs
N: Number of base learners
Begin
Get Dataset
Preprocess the data
Set n to 0
Do Model Train
Set Model
Set epoch to 0
while $epoch < K + 1$ **do**
Get Pre-processed image data
Apply Data Augmentation
Pass to Model
Get Validation Accuracy
repeat
end while
Save Model
Load Model
while $n < 3$ **do**
Set $P1$, $P2$, $P3$, WP to 0
Pass input to Models
Get score of $P1$, $P2$, $P3$
Set $w1$, $w2$, $w3$
Set $WP = ([w1, w2, w3].[P1, P2, P3])$
Get Weighted Average Ensemble Accuracy_score
end while
End = 0

approaches generally entail combining the results of several base classifiers, like support vector machines, decision trees, and neural networks. Ensemble models can achieve improved accuracy and robustness by utilising numerous classifiers that capture different features of the data and may have varied strengths and limitations [45]. To build ensemble classifiers, methods including bagging, boosting, and stacking are frequently used. By training each classifier on a bootstrapped sample of the training data, bagging produces a variety of models, whereas boosting emphasizes misclassified examples in order to repeatedly improve the performance of weak learners. Through the process of stacking [46], a meta-classifier gains knowledge of a higher-level representation of the data by using the predictions of base classifiers as inputs. Ensemble learning presents a possible path towards improving the state-of-the-art in image classification tasks by combining numerous classifiers. Furthermore, ensemble approaches are robust against noise and overfitting, which makes them especially useful for noisy and complicated image datasets. Furthermore, the intrinsic parallelizability of ensemble models allows for effective training and inference on large-scale image datasets. Consequently, ensemble learning has gained popularity in a wide range of computer vision applications, such as medical image analysis, and object recognition.

1) WEIGHTED AVERAGE ENSEMBLE MODEL (WAE)

In this work, we used an ensemble modeling technique to improve the diabetic retinopathy classification performance, which is a challenging multi-class classification task consisting of five different categories. Since there are five different classes in the dataset under investigation, a strong method is required to distinguish between them. To achieve this, we started a rigorous process of model selection and training. Initially, six pre-trained deep learning models were first trained, and their performance on the dataset was carefully assessed. The results showed that Inception-V3, VGG16, and our custom CNN were the most effective in capturing the complex features present in the retinal images.

Building upon this initial assessment, we developed an ensemble model that combines the predictive power of these three base classifiers. We selected weighted average ensemble which makes use of the diversity of multiple models to produce a predictive framework that is more reliable and accurate. Each base classifier in this ensemble paradigm contributes predictions that are weighted according to how well, it performs on a validation set. The ensemble's total performance is maximized by the weighted averaging procedure, which makes sure that classifiers with stronger predictive ability to have a bigger say in the final decision-making process.

Mathematically, the weighted average ensemble can be represented as:

$$\hat{y}_{\text{ensemble}} = \sum_{i=1}^{N} w_i \cdot \hat{y}_i \tag{6}$$

where $\hat{y}_{\text{ensemble}}$ represents the ensemble prediction, $\hat{y}_i$ denotes the prediction of the i^{th} base classifier, and w_i signifies the weight assigned to the i^{th} classifier's prediction. The weights w_i were established by evaluating each base classifier's performance on a validation set, with the goal of prioritising classifiers that exhibit better individual accuracies.

The weights are carefully adjusted by taking into account the accuracy that each base classifier achieved on the validation dataset. This results in the best possible combination of predictive strength. The weights of each classifier's prediction w_i are calculated by using (7):

$$w_i = \frac{\text{Acc}_i}{\sum_{j=1}^{N} \text{Acc}_j} \tag{7}$$

where Acc_i is the validation accuracy of the i^{th} classifier. Using the combined knowledge from several models, our ensemble method aims to overcome the drawbacks of single classifiers and improve the precision, consistency, and robustness of diabetic retinopathy classification.

TABLE 5. Summary of computing resources used in proposed study.

Computing Resources	Detail
CPU	Intel(R) Core(TM) i7-14700K 3.40 GHz
GPU	NVIDIA GeForce RTX 4080
RAM	64 GB
System	64-bit operating system, x64-based processor
Language	Python(3.9.19)
Framework	Keras(2.7.0) with TensorFlow(2.7.0) backend
Software	Anaconda

F. ENVIRONMENT SETUP

In order to guarantee accurate and effective research, we set up a dedicated workspace for this purpose. Table.4 provides a comprehensive overview of our environment configuration including all of the individual settings. This method improved the validity and dependability of our research findings by enabling us to perform a full investigation and analysis.

V. RESULTS AND DISCUSSION

A. EVALUATION METRICS

We used confusion matrix to classify and visualize our model performance. The actual and predicted classes are denoted by rows and columns.

1) ACCURACY

Accuracy measures the overall effectiveness of the model. Accuracy is the ratio of true positives and true negatives with the total cases examined.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

2) F1 SCORE

F1 score is helpful in cases where classes are imbalanced. It is a harmonic mean of recall and precision and provides one metric to balance out both.

$$\text{F1 Score} = \frac{Precision.Recall}{Precision + Recall} \tag{9}$$

3) RECALL

Sensitivity or recall is the fraction of true predicted values to actual positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

TABLE 6. Summary of deep learning models performance in term of accuracy, precision, recall, f1 score and AUC. It can be seen that inception-v3, VGG16 and CNN achieved high performance compared to other DL models.

Model	Accuracy	Precision	Recall	F1 Score	AUC
DenseNet201	90.31	76.09	71.82	71.96	91.51
MobileNet	88.50	71.37	70.97	71.17	86.91
ResNet50	86.69	75.48	49.58	59.32	87.87
Xception	89.28	73.39	72.78	73.07	90.08
Inception-v3	93.08	84.48	80.14	82.21	97.20
CNN	90.56	81.88	67.78	73.86	93.82
VGG16	91.29	84.63	71.82	79.09	96.72

4) PRECISION

Precision tells how accurate the model made the positive predictions. It is the fraction correctly predictive true positive and total predictive positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

5) AREA UNDER THE CURVE (AUC)

The area under the Receiver Operating Characteristic (ROC) is also known AUC. The diagnostic capability of a classifier as its discrimination threshold is represented by ROC curve. Higher AUC value indicates better model performance. AUC value ranges from 0 to 1.

$$\text{AUC} = \int_{1}^{0} TPRd(FPR) \tag{12}$$

where:

- TPR is true positive rate plotted on y-axis.
- FPR is false positive rate plotted on x-axis.

B. RESULTS

We conducted several experiments on the APTOS dataset. The dataset is split into 2 subsets, 80% for training and 20% for testing. We first utilized six pre-trained models DenseNet201, MobileNet, VGG16, ResNet50, In-ceptionv3, Xception and a custom CNN then evaluated all the models against six performance matrices accuracy, loss, precision, recall, F-1 score, and AUC to select base learners for our ensemble model. We compared all 7 DL models and found that Inception-v3, VGG16 and our custom CNN with J49 classifier outperformed other pre-trained models. Notably, inception-v3 achieved 93.08% accuracy, 84.48% precision, 80.14% Recall, 82.21% F1-score and 97.20% AUC. VGG16 scored second place with 91.29% accuracy, 84.63% precision, 71.82% Recall, 79.09% F1-score and 96.72% AUC followed by custom CNN model. Our CNN model ranked third among all DL models with 90.56% accuracy, 81.88% precision, 67.78% Recall, 73.86% F1-score and 93.82% AUC. Fig.13 shows the training and validation accuracy graph for CNN model. The comparison among all models is summarised in Table 6.

As the Fig.10 shows, Inception-v3 scored highest accuracy followed by VGG16 and CNN model respectively. These three models are selected as base learners for our ensemble model.

Top-3 models are selected as base learners for our weighted average ensemble. The weights ***w1, w2, w3*** are calculated using Eq. 7 that is ***w1*** for Inception-v3 since it has the highest accuracy score. ***w2*** is given to VGG16 and ***w3*** for our custom CNN model. The weighted average ensemble (WAE) merged the strengths of top-3 base which results in higher accuracy and overall model performance compare to the standalone models. The proposed ensemble approach achieved model's robustness by assigning optimal weights to each base learner's prediction. The optimal weights assignment is based on their respective accuracies. Table 7 shows the performance analysis of base learners and our weighted average ensemble model. Fig. 11 shows the confusion matrix for the base learner CNN. Confusion matrix provides performance overview of the model across different classes. Fig.12 and Fig.15 presents the confusion matrix for inception-v3 and VGG16. The diagonal values are number

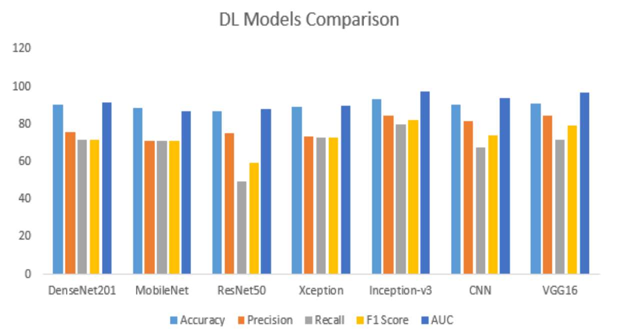


FIGURE 10. Comparison of various deep learning models trained on APTOS 2019 dataset. The bar chart shows that inception-v3, VGG16 and CNN has the highest performance among other pre-trained models.

TABLE 7. Comparative analysis of base learners and our ensemble model.

Model	Accuracy	Precision	Recall	F1 Score	AUC
Inception-v3	93.08	84.48	80.14	82.21	97.20
CNN	90.56	81.88	67.78	73.86	93.82
VGG16	91.29	84.63	71.82	79.09	96.72
Proposed WAE	95.13	89.50	85.69	87.58	98.60

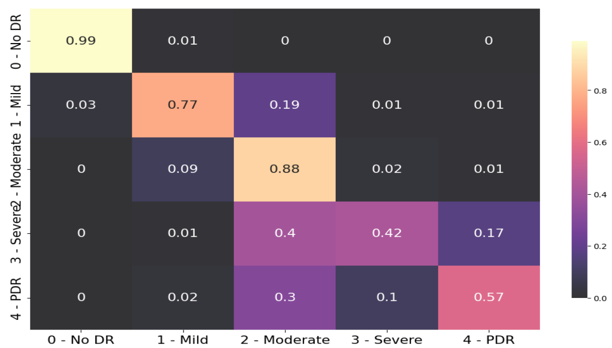


FIGURE 11. Confusion matrix of convolutional neural network for diabetic retinopathy classification.

FIGURE 12. Confusion matrix of fine-tuned inception-v3 trained on training subset.

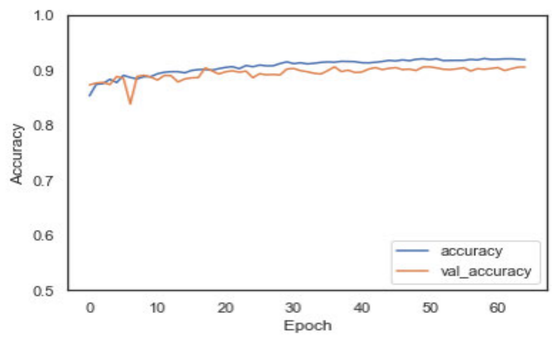


FIGURE 13. Convolutional neural network performance graph in term of training and validation accuracy.

of correct predictions indicating when the model accurately predict the data.

Our proposed weighted average ensemble model performed significantly well with high testing accuracy as shown in Table.7. Fig.16 shows the performance of our ensemble model across difference classes. The model correctly predicted 99%, 77%, 88% and 42% of the cases for class-1, class-2, class-3 and class-4 respectively. The model correctly predict 78% of the times for class-2 but 3% misclassified as class-1, 19% for class-2 and 1% for class-3 and class-4 respectively. For severe cases, the model gave correct prediction 42% of the time but heavily misclassified as class-2 (40%) and 17% class-4. For the last stage of DR, the model gave 30% misclassification into class-3, 10% into class-2 ans 2% for class-4 simultaneously.

We performed the pre-processing steps mentioned in section III-B on training data before feeding it to base learners. Table.8 shows a comparative analysis of the model's performance with pre-processing and without pre-processing. The results indicate that the model's performance significantly improved after pre-processing steps were applied. This indicates the potency of pre-processing in image quality, increasing the predictivity of the proposed model. Therefore, pre-processing is a crucial step in medical image analysis pipeline, leading to a more accurate outcome of the model.

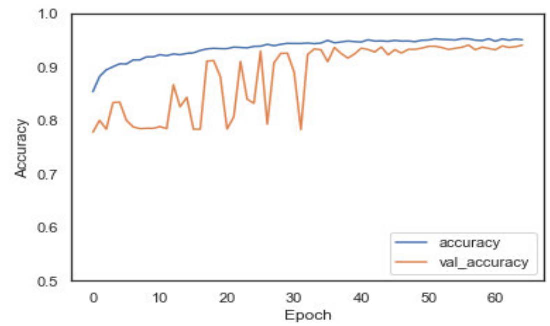


FIGURE 14. Inception-v3 performance graph in term of training and validation accuracy.

FIGURE 15. Confusion matrix of fine-tuned VGG16 trained on training subset.

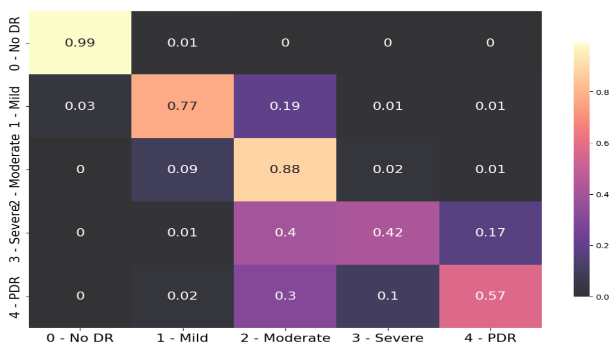


FIGURE 16. Confusion matrix of proposed weighted average ensemble model for diabetic retinopathy classification.

TABLE 8. Performance evaluation of proposed ensemble model prior to pre-processing vs after pre-processing.

Model	Accuracy	Precision	Recall	F1 Score	AUC
WAE without pre-processing	94.06	86.88	84.78	84.69	97.10
WAE with pre-processing	95.06	87.88	83.78	85.69	98.10

C. DISCUSSION

The results of this study emphasise on the efficiency of our proposed weighted average ensemble approach for classifying the diabetic retinopathy images into 5 classes. By utilizing the strength of various high performing DL

TABLE 9. Summary of comparative analysis of proposed ensemble model with prior literature.

Model	Year	Dataset	Accuracy (%)	Precision (%)	F1 Score (%)	AUC (%)
CNN with custom weighted loss function [47]	2022	MESSIDOR-2	92.49	92.01	93.6	-
Modified EfficientNet	2021	APTOS2019	79.0	-	-	-
EfficientNetB0, Resnet101, Densenet201 [48]	2023	DR database	93.42	91.75	91.06	-
Xception model with a contrastive loss function [49]	2022	APTOS 2019, MESSIDOR-2	84.36	-	-	93.81
BCNN [22]	2022	IDRiD	92.0	-	-	-
Multi-scale feature fusion with Mobilenet-v2 [50]	2021	MESSIDOR-2	85.32	92.69	-	97.0
DRGCNN [51]	2024	EyePACS, MESSIDOR-2	-	-	-	-
Composite neural network with gated attention [52]	2022	APTOS 2019	82.54	81.87	79.25	-
CTNet [53]	2024	APTOS 2019, IDRiD	-	-	-	98.7
Proposed WAE model	-	**APTOS 2019**	**95.06**	**87.88**	**85.69**	**98.10**

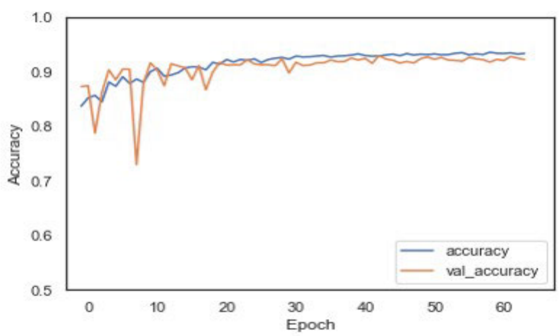


FIGURE 17. VGG16 performance graph in term of training and validation accuracy.

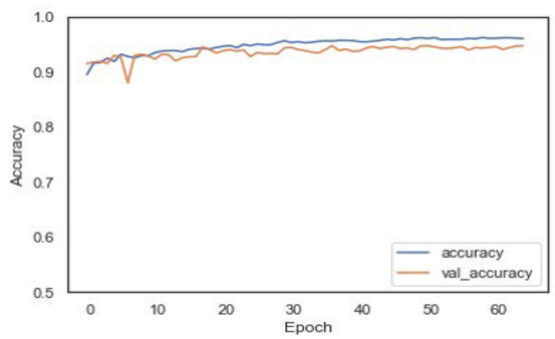


FIGURE 18. Weighted average ensemble model performance graph in term of training and validation accuracy.

models, our ensemble model achieved significantly high accuracy compared to other state-of-the art models. Initially, we trained 6 pre-trained models along with a custom CNN on our APTOS dataset. We selected these models due to their popularity and proven performance in medical image analysis. After a detailed evaluation, we subsequently selected inception-v3, VGG16 and CNN as our top learners. The weighted average ensemble was deployed to merge the predictions of selected learners. Different weights were assigned to the base learners based on their accuracy, resulting in high overall training accuracy of the ensemble. We optimized the weight while training to maximize the model performance. Table.9 demonstrates the comparative analysis of our ensemble approach with existing classification techniques. Recent studies have shown the remarkable advancements in diabetic retinopathy classification and detection task. A recent study [54] proposed a novel hybrid approach while merging different wavelet-based models achieving decent accuracy and overall performance. Another study [55] revised the ResNet50 architecture to improve model standardization hence, improving the model predictability.

The proposed weighted average ensemble model, combining the Inception-v3, VGG16, and a custom CNN, significantly improves the accuracy and robustness by utilizing the strengths of each base learner. This method achieves high performance including 95.06% accuracy and 98.10% AUC. However, its intricacy and computational cost may limit its feasibility in resource-constrained environments. Furthermore, the interpretability of the ensemble is reduced, making it difficult to discern how each model contributed to the ultimate decision. Careful weights adjustment and optimization strategies are essential to balance these strengths and weaknesses.

VI. CONCLUSION AND FUTURE WORK

Visual impairment is a serious global concern because it affects a person's eyesight and quality of life. Diabetes retinopathy is receiving a lot of attention from researchers because it causes irreversible vision loss. The lack of early warning signs or minor visual issues that eventually lead to blindness makes it a difficult disease to treat for ophthalmologists. The creation of an automated diagnostic system may have significant implications for the field of medical healthcare. Ensemble learning generally outperforms a single base classifier because it combines several independent learning algorithms. Consequently, it has gained popularity and proven an effective machine-learning method. One of the most significant issues is finding a way to combine the most accurate base classifiers. To solve this problem, We first utilized six pre-trained models DenseNet201, MobileNet, VGG16, ResNet50, Inceptionv3, Xception and a custom CNN then evaluated all the models. Inception-v3, VGG16 and our custom-built CNN emerged as top performers. A weighted average ensemble learning approach is employed to refine the final prediction. Our ensemble method achieved impressive accuracy and outperformed popular DL models in terms of other evaluation metrics. Our ensemble approach leveraged the strengths of each individual base learner while indemnifying their weaknesses resulting in high model performance. Combining the models using an ensemble approach can provide the advantage of existing DL approaches increasing the quality of diagnosis. In future work, we will focus on applying our ensemble approach to other domain such as smart farming and plant disease detection and classification to investigate how our proposed method can enhance agricultural decision-making and plant disease diagnostics. This cross-domain application will aid in understanding the adaptability and robustness of our model. Also the additional fine-tuning could result in even better performance in diverse settings.

REFERENCES

[1] 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. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 2045–2048.

[2] S. S. Mondal, N. Mandal, K. K. Singh, A. Singh, and I. Izonin, "EDLDR: An ensemble deep learning technique for detection and classification of diabetic retinopathy," *Diagnostics*, vol. 13, no. 1, p. 124, Dec. 2022.

[3] Z. Shen, Q. Wu, Z. Wang, G. Chen, and B. Lin, "Diabetic retinopathy prediction by ensemble learning based on biochemical and physical data," *Sensors*, vol. 21, no. 11, p. 3663, May 2021.

[4] S. Dutta, B. Manideep, S. M. Basha, R. D. Caytiles, and N. Iyengar, "Classification of diabetic retinopathy images by using deep learning models," *Int. J. Grid Distrib. Comput.*, vol. 11, no. 1, pp. 89–106, 2018.

[5] K. Bhatia, S. Arora, and R. Tomar, "Diagnosis of diabetic retinopathy using machine learning classification algorithm," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Oct. 2016, pp. 347–351.

[6] M. A. Bravo and P. A. Arbeláez, "Automatic diabetic retinopathy classification," in *Proc. 13th Int. Conf. Med. Inf. Process. Anal.*, Nov. 2017, pp. 446–455.

[7] F. Nawaz, M. Ramzan, K. Mehmood, H. Ullah Khan, S. Hayat Khan, and M. Raheel Bhutta, "Early detection of diabetic retinopathy using machine intelligence through deep transfer and representational learning," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 1631–1645, 2021.

[8] M. Usman Akram, S. Khalid, A. Tariq, S. A. Khan, and F. Azam, "Detection and classification of retinal lesions for grading of diabetic retinopathy," *Comput. Biol. Med.*, vol. 45, pp. 161–171, Feb. 2014.

[9] S. Wan, Y. Liang, and Y. Zhang, "Deep convolutional neural networks for diabetic retinopathy detection by image classification," *Comput. Electr. Eng.*, vol. 72, pp. 274–282, Nov. 2018.

[10] B. Wu, W. Zhu, F. Shi, S. Zhu, and X. Chen, "Automatic detection of microaneurysms in retinal fundus images," *Computerized Med. Imag. Graph.*, vol. 55, pp. 106–112, Jan. 2017.

[11] D. Sarwinda, T. Siswantining, and A. Bustamam, "Classification of diabetic retinopathy stages using histogram of oriented gradients and shallow learning," in *Proc. Int. Conf. Comput., Control, Informat. Appl. (IC3INA)*, Nov. 2018, pp. 83–87.

[12] E. V. Carrera, A. González, and R. Carrera, "Automated detection of diabetic retinopathy using SVM," in *Proc. IEEE XXIV Int. Conf. Electron., Electr. Eng. Comput. (INTERCON)*, Aug. 2017, pp. 1–4.

[13] G. Mathew, S. S. Ramachandran, and V. Suchithra, "EdgeAI: Diabetic retinopathy detection in Intel architecture," in *Proc. IEEE / ITU Int. Conf. Artif. Intell. Good (AI4G)*, Sep. 2020, pp. 75–80.

[14] W. Cao, N. Czarnek, J. Shan, and L. Li, "Microaneurysm detection using principal component analysis and machine learning methods," *IEEE Trans. Nanobiosci.*, vol. 17, no. 3, pp. 191–198, Jul. 2018.

[15] A. S. Sathwik, R. Agarwal, A. Jubilson, and S. S. Basa, "Diabetic retinopathy classification using deep learning," *EAI Endorsed Trans. Pervasive Health Technol.*, vol. 9, no. 36, pp. 1–8, Nov. 2023.

[16] B. Anoop, "Binary classification of DR-diabetic retinopathy using CNN with fundus colour images," *Mater. Today: Proc.*, vol. 58, pp. 212–216, 2022.

[17] Y. B. Özçelik and A. Altan, "Classification of diabetic retinopathy by machine learning algorithm using entorpy-based features," in *Proc. ÇAnkaya Int. Congr. Sci. Res.* Adiyaman Province, Turkey: IKSAD Golbasi, 2023, pp. 10–12.

[18] A. Soni and A. Rai, "A novel approach for the early recognition of diabetic retinopathy using machine learning," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2021, pp. 1–5.

[19] B. Lalithadevi and S. Krishnaveni, "Diabetic retinopathy detection and severity classification using optimized deep learning with explainable AI technique," *Multimedia Tools Appl.*, vol. 83, pp. 1–65, Apr. 2024.

[20] V. Sagvekar, M. Joshi, M. Ramakrishnan, and A. Dudani, "Hybrid hunter-prey ladybug beetle optimization enabled deep learning for diabetic retinopathy classification," *Biomed. Signal Process. Control*, vol. 95, Sep. 2024, Art. no. 106346.

[21] S. Gayathri, V. P. Gopi, and P. Palanisamy, "A lightweight CNN for diabetic retinopathy classification from fundus images," *Biomed. Signal Process. Control*, vol. 62, Sep. 2020, Art. no. 102115.

[22] N. Khalifa, M. Loey, M. Taha, and H. Mohamed, "Deep transfer learning models for medical diabetic retinopathy detection," *Acta Inf. Medica*, vol. 27, no. 5, p. 327, 2019.

[23] G. Mushtaq and F. Siddiqui, "Detection of diabetic retinopathy using deep learning methodology," *IOP Conf. Series, Mater. Sci. Eng.*, vol. 1070, Feb. 2021, Art. no. 012049.

[24] A. K. Gangwar and V. Ravi, "Diabetic retinopathy detection using transfer learning and deep learning," in *Evolution in Computational Intelligence*, V. Bhateja, S.-L. Peng, S. C. Satapathy, and Y.-D. Zhang, Eds., Singapore: Springer, 2021, pp. 679–689.

[25] M. Chetoui and M. A. Akhloufi, "Explainable diabetic retinopathy using EfficientNET," in *Proc. 42nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2020, pp. 1966–1969.

[26] J. D. Bodapati, N. S. Shaik, and V. Naralasetti, "Composite deep neural network with gated-attention mechanism for diabetic retinopathy severity classification," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 9825–9839, Oct. 2021.

[27] J. Amin, M. Sharif, M. Yasmin, H. Ali, and S. L. Fernandes, "A method for the detection and classification of diabetic retinopathy using structural predictors of bright lesions," *J. Comput. Sci.*, vol. 19, pp. 153–164, Mar. 2017.

[28] A. Pak, A. Ziyaden, K. Tukeshev, A. Jaxylykova, and D. Abdullina, "Comparative analysis of deep learning methods of detection of diabetic retinopathy," *Cogent Eng.*, vol. 7, no. 1, Jan. 2020, Art. no. 1805144.

[29] T. Nazir, M. Nawaz, J. Rashid, R. Mahum, M. Masood, A. Mehmood, F. Ali, J. Kim, H.-Y. Kwon, and A. Hussain, "Detection of diabetic eye disease from retinal images using a deep learning based CenterNet model," *Sensors*, vol. 21, no. 16, p. 5283, Aug. 2021.

[30] 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 *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2019, pp. 1–6.

[31] S. Vinuja and R. Karthika, "Performance analysis of diabetic retinopathy classification using CNN," in *Proc. 3rd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Sep. 2021, pp. 823–828.

[32] S. Majumder and N. Kehtarnavaz, "Multitasking deep learning model for detection of five stages of diabetic retinopathy," *IEEE Access*, vol. 9, pp. 123220–123230, 2021.

[33] E. AbdelMaksoud, S. Barakat, and M. Elmogy, "A computer-aided diagnosis system for detecting various diabetic retinopathy grades based on a hybrid deep learning technique," *Med. Biol. Eng. Comput.*, vol. 60, no. 7, pp. 2015–2038, Jul. 2022.

[34] C. Lahmar and A. Idri, "On the value of deep learning for diagnosing diabetic retinopathy," *Health Technol.*, vol. 12, no. 1, pp. 89–105, Jan. 2022.

[35] G. Zhang, K. Li, Z. Chen, L. Sun, J. Zhang, and X. Pan, "Augmentation-consistent clustering network for diabetic retinopathy grading with fewer annotations," *J. Healthcare Eng.*, vol. 2022, pp. 1–10, Mar. 2022.

[36] A. Batool and Y.-C. Byun, "Lightweight EfficientNetB3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images," *IEEE Access*, vol. 11, pp. 37203–37215, 2023.

[37] F. J. Martinez-Murcia, A. Ortiz, J. Ramírez, J. M. Górriz, and R. Cruz, "Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy," *Neurocomputing*, vol. 452, pp. 424–434, Sep. 2021.

[38] S. D. Karthik and Maggie, "APTOS 2019 blindness detection," Asia Pacific Tele-Ophthalmol. Soc., Singapore, Tech. Rep. 931, 2019.

[39] T. S. Kern, "Contributions of inflammatory processes to the development of the early stages of diabetic retinopathy," *J. Diabetes Res.*, vol. 2007, no. 1, p. 95103, Jan. 2007.

[40] S. Das, S. Das, S. Debray, M. Mishra, and S. Majumder, "Automatic detection of diabetic retinopathy to avoid blindness," in *Applied Artificial Intelligence*. Boca Raton, FL, USA: CRC Press, 2023, pp. 221–234.

[41] B. Graham, "Kaggle diabetic retinopathy detection competition report," *Univ. Warwick*, vol. 22, no. 9, p. 17, Apr. 2015.

[42] J. Wu, "Introduction to convolutional neural networks," *Nat. Key Lab Novel Softw. Technol.*, vol. 5, no. 23, p. 495, 2017.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[45] S. Chatterjee and Y.-C. Byun, "Voting ensemble approach for enhancing Alzheimer's disease classification," *Sensors*, vol. 22, no. 19, p. 7661, Oct. 2022.

[46] A. Batool and Y.-C. Byun, "Toward improving breast cancer classification using an adaptive voting ensemble learning algorithm," *IEEE Access*, vol. 12, pp. 12869–12882, 2024.

[47] M. Masud, M. F. Alhamid, and Y. Zhang, "A convolutional neural network model using weighted loss function to detect diabetic retinopathy," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 1s, pp. 1–16, Feb. 2022.

[48] K. Kamal, R. A. Mohamed, A. Darwish, and A. E. Hassanien, "Deep learning with class imbalance for detecting and classifying diabetic retinopathy on fundus retina images," *Res. Square*, Sep. 2022.

[49] M. R. Islam, L. F. Abdulrazak, M. Nahiduzzaman, M. O. F. Goni, M. S. Anower, M. Ahsan, J. Haider, and M. Kowalski, "Applying supervised contrastive learning for the detection of diabetic retinopathy and its severity levels from fundus images," *Comput. Biol. Med.*, vol. 146, Jul. 2022, Art. no. 105602.

[50] R. Fan, Y. Liu, and R. Zhang, "Multi-scale feature fusion with adaptive weighting for diabetic retinopathy severity classification," *Electronics*, vol. 10, no. 12, p. 1369, Jun. 2021.

[51] Z. Hai, B. Zou, X. Xiao, Q. Peng, J. Yan, W. Zhang, and K. Yue, "A novel approach for intelligent diagnosis and grading of diabetic retinopathy," *Comput. Biol. Med.*, vol. 172, Apr. 2024, Art. no. 108246.

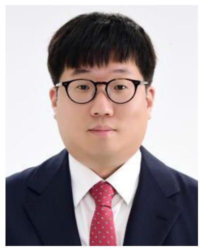
[52] M. D. Alahmadi, "Texture attention network for diabetic retinopathy classification," *IEEE Access*, vol. 10, pp. 55522–55532, 2022.

[53] R. Bala, A. Sharma, and N. Goel, "CTNet: Convolutional transformer network for diabetic retinopathy classification," *Neural Comput. Appl.*, vol. 36, no. 9, pp. 4787–4809, Mar. 2024.

[54] G. Alwakid, W. Gouda, and M. Humayun, "Deep learning-based prediction of diabetic retinopathy using CLAHE and ESRGAN for enhancement," *Healthcare*, vol. 11, no. 6, p. 863, Mar. 2023.

[55] C.-L. Lin and K.-C. Wu, "Development of revised ResNet-50 for diabetic retinopathy detection," *BMC Bioinf.*, vol. 24, no. 1, p. 157, Apr. 2023.

KINZA NAZIR received the B.S. and M.S. degrees in information technology from the University of Sargodha, Pakistan. She is currently pursuing the Ph.D. degree in electronic engineering with Jeju National University, South Korea. She is also a Research Fellow with the Machine Learning Laboratory, Jeju National University. Her research interests include machine learning, computer vision, deep learning, and medical image analysis.



JISOO KIM received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Seoul National University, South Korea, in 2009, 2011, and 2021, respectively. He is currently an Assistant Professor with the Faculty of Software (Artificial Intelligence Major), Jeju National University, South Korea. His research interests include deep learning applications and explainable artificial intelligence (XAI).

YUNG-CHEOL BYUN received the B.S. degree from Jeju National University, in 1993, and the M.S. and Ph.D. degrees from Yonsei University, in 1995 and 2001, respectively. He was a special Lecturer with Samsung Electronics and SDS, from 1998 to 2001. From 2001 to 2003, he was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI). He was promoted to join Jeju National University, as an Assistant Professor, in 2003, where he is currently a Full Professor with the Computer Engineering Department. His research interests include the areas of AI and machine learning, pattern recognition, blockchain and deep learning-based applications, big data and knowledge discovery, time-series data analysis and prediction, image processing, medical image applications, and recommendation systems.