ResNet-34/DR: A Residual Convolutional Neural Network for the Diagnosis of Diabetic Retinopathy

Noor M. Al-Moosawi
College of Computer Science & Information Technology, University of Basrah, Iraq
E-mail: almoosawinoor2@gmail.com

Raidah S. Khudeyer
College of Computer Science & Information Technology, University of Basrah, Iraq
E-mail: raidah.khudayer@uobasrah.edu.iq

Keywords: Convolutional Neural Networks (CNN), Deep Learning (DL), Diabetic Retinopathy (DR), ResNet-34, Transfer Learning (TL)

Received: October 10, 2021

Diabetic retinopathy (DR) is an eye complication associated with diabetes, resulting in blurred vision or blindness. The early diagnosis and treatment of DR can decrease the risk of vision loss dramatically. However, such diagnosis is a tedious and complicated task due to the variability of retinal changes across the stages of the diseases, and due to the high number of undiagnosed and untreated DR cases. In this paper, we develop a computationally efficient and scalable deep learning model using convolutional neural networks (CNN), for diagnosing DR automatically. Various preprocessing algorithms are utilized to improve accuracy, and a transfer learning strategy is adopted to speed up the process. Our experiment used the fundus image set available on online Kaggle datasets. As an ultimate conclusion of applicable performance metrics, our computational simulation achieved a relatively-high F1 score of 93.2% for stage-based DR classification.

Povzetek: Opisana je metoda globokih nevronskih mrež za diagnozo težav vida zaradi sladkorne bolezni.

1 Introduction

Diabetic retinopathy (DR) is a disease that affects the eye as a complication of diabetes, causing impaired vision as a result of damage to the retina, the light-sensitive tissues at the bottom of the eye that are required for vision [1-2]. Diabetes harms blood vessels in the retina. The longer a person gets diabetes, the more likely such person is to develop DR. According to the World Health Organization (WHO), the global population of DR patients is expected to increase to 592 million by 2025 [1]. Diabetic retinopathy (DR) develops through many stages with increasing severity, which could if left untreated, lead to blindness [3]. DR is mainly classified into no proliferative (NPDR) and proliferative (PDR). Furthermore, NPDR can be classified as mild, moderate, or severe. Figure 1 shows examples of different stages. DR stages are as follows [3-5]: a) No DR: The eye is healthy. b) Mild NPDR: Small swellings appear in retina blood vessels. c) Moderate NPDR: As the disease progresses, some retina blood vessels become blocked. d) Severe NPDR: More blood vessels are blocked, depriving the retina of oxygen and nutrients. e) PDR: In this stage, the growth of new blood vessels is stimulated proliferative. However, such new blood vessels have an abnormal appearance and very thin and fragile walls. When these vessels bleed, they can cause severe vision loss and even blindness.

The early detection of the disease helps avoid complications and improves chances of recovery. More than 90% of patients can avoid vision loss by early detection and treatment [3]. Typically; an ophthalmologist diagnoses DR by manually interpreting and analyzing fundus photographs. However, DR diagnosis is a tedious and complicated task due to the variability of retinal changes across the stages of the diseases, and due to the high number of undiagnosed and untreated DR cases. Human competency is prone to error and novel computational techniques are being pursued in an attempt to overcome this problem.

Diagnosis can be more reliable if it is based on extracted highly discriminative features and resistant to specific conditions, such as lighting changes. Deep learning and CNN are the most current methods for extracting features. CNN's extracted features have a high discriminative capacity [6].

The previous methods, which depended on a deep CNN for DR diagnosis using a very deep CNN model, GoogLeNet, VggNet, and ResNet, achieved good

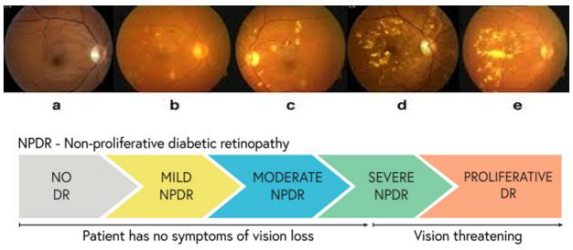


Figure 1: Examples of fundus images for DR stages according to disease severity.

accuracy rates. However, it is still possible to further optimize outputs by making some improvements to the models as follows:

a) We proposed an efficient CNN model that is based on ResNet-34 for transfer learning for DR diagnosis with higher accuracy.

b) The current APTOS 2019 and IDRID datasets are being analyzed to evaluate the performance of the proposed ResNet-34/DR model.

c) Transfer learning with pre-trained deep CNNs and hyperparameter tuning are critical components of the training process and have been highly beneficial in medical image analysis. We use weights from the ImageNet dataset to initialize the weights instead of initializing the weights randomly.

d) By comparison with the accuracy rates of the modified GoogLeNet and VggNet models, ResNet-34/DR yielded better classification performance.

This paper is structured into several sections as follows. Section 2 provides an overview of deep learning, CNN for image classification, and the need for Transfer Learning (TL) in our task. The methodology used in this paper will be described in detail in Section 3. Section 4 explains preprocessing for the dataset, the training process with hyper_parameters for all experiments. The final results of the experiments are summarized in Section 5. In Section 6, the performance of our work is discussed against previous studies. Finally, section 7 concludes this paper with the baseline for the future.

2 Background

Numerous previous studies employed a variety of methods to handle the problem of diagnosing DR. We will highlight them in this section. Table 1 summarizes the previous studies discussed in this section.

2.1 Deep learning methods

Recent developments in artificial intelligence (AI) have paved the road for big advances in the field of automatic diagnosis in various medical fields as compared with manual methods. Computer-Aided Diagnosis (CAD) systems could provide features, such as the reduction of human error, supporting medical decisions, and improved patient care, as the diagnosis of DR is essentially made in reliance on image processing techniques, using the latest AI technologies, particularly, machine learning (ML) and deep learning (DL), whereas DL is a special type of ML, involving a deeper level of data analysis, and hence, deeper learning [7]. DL has quickly established itself as a valuable technique for the analysis and classification of medical images [3-5].

Previous studies that relied on machine learning methods and feature extraction have produced excellent work for the diagnosis of DR, as characterized by solid secretions, red lesions, micro aneurysms, and blood vessels [8]. The classifiers that are used to accomplish the task include neural networks, random forest, sparse representation classifiers, linear discriminant analysis (LDA), support vector machine (SVM), K-nearest neighbors (KNN) algorithm [3][9]. Such techniques assemble healthy and infected eye fundus images for the analysis thereof.

DL methods for DR diagnosis were being used with a possibility to variate the features which the corresponding architecture deems to constitute diagnostic indicators, which help identify the most significant areas in the images by researchers [10], by the addition of a global average pooling layer to the CNN instead of a fully-connected layer. Convolutional neural networks (CNNs) are being discussed in the next section as a new inspiring DL method for providing a more accurate and more detailed, and hence, more useful, diagnosis of DR.

2.2 CNN overview

The major limitations of the majority of the aforementioned techniques are that: a) they merely give a binary result, yielding: “DR” or “no DR”, becoming, practically, a mere detection rather than a full-scale classification. b) Most models have been trained by researchers on small samples, limiting the generalizability of their findings. Therefore, such automatic diagnosis systems are limited [11]. The development of CNN layers has provided a greater ability to classify images and detect patterns, objects, and other distinguishing features in a picture [12]. These are multiple computational layers that involve the application of image analysis filters in the form of convolutions. [13]. By convoluting multiple filters over an image within a layer, a feature map is generated to be used, as an input, to the next layer, enabling the processing of images as pixels for such input in order to generate the required classification (in our case, diagnosis), as an output. Such a classification approach within a single classifier replaces multiple steps of the previous methods of image analysis [14], and thus, enabling a faster and more efficient image interpretation process.

CNNs have been used a lot in the fields of computer vision, in general, and medical imaging, in particular, thanks to their great ability to handle and process images. They have become a state-of-art technique in various medical fields. CNNs generally consist of three types of layers [15]: i) Convolutional layers, where a number of filters are applied to identify a certain feature or pattern in the inputted image. A stack of filters is used in order to extract various features. The values of such filters are tuned by training to be consistent with the extraction of the attributes that are associated with the disease (disease indicators). ii) Pooling layers to reduce the feature map extracted by the filters in order to reduce the necessary calculations while retaining the best values of the attributes resulting from the convolutional layers. iii) Fully-connected (FC) layers, where the final classification process takes place, as every neuron is associated with the neurons of the preceding layer. In addition to such layers, an activation function is employed. The number and sequence of layers vary depending on the complexity of the corresponding problem.

Since 2015, researchers have relied on CNNs as a powerful tool in the field of computer-aided diagnosis. For

example [16], CNN was utilized for the diagnostic of DR in fundus images as one of two classes: normal and abnormal, as such proposed architecture relied on linking three stacks of convolutional filters in parallel, whereby the output is an outcome of global max pooling. The RGB layers of an image were being isolated in order to use the green layer only, as it is the layer that demonstrates the attributes of the disease in the clearest and most distinguishable manner. This architecture has helped reduce the number of parameters and avoid overfitting, yielding a final accuracy of 81% when experimented on 12,000 images.

In another work [12], a methodology was proposed for the further classification of the image database (Aptos 2019) into three stages: a) No DR. b) Moderate DR. c) Severe DR. Architecture was built to consist of 18 convolutional layers and 3 fully-connected layers, in addition to max pooling and the use of the preprocessing image techniques of image resizing and data augmentation. This architecture yielded an accuracy of 88%. DL methods for DR diagnosis were being used with a possibility to variate the features which the corresponding architecture deems to constitute diagnostic indicators, which help identify the most significant area in the images by researchers [8], by the addition of a global average pooling layer to the CNN instead of a fully-connected layer.

2.3 The need for Transfer Learning

As implied hereinbefore, DL requires a huge quantity of data for the efficient training of a CNN. This is not usually possible in the field of ophthalmology, as the available real data are relatively limited and unbalanced. Therefore, researchers are intensively relying on transfer learning (TL) to overcome the obstacles of computational time and the need for ongoing training. TL is a method of overcoming the limitedness of data by leveraging knowledge from another domain [15].

CNNs and TL are the main two methods of automatic DR diagnosis using DL techniques [9], [12], [14]. TL has proven itself as a very effective technique, especially when handling domains of limited data [16]. Instead of completely training a blank network from scratch, a feed forward approach can be used to fix weights in the lower levels that have already been optimized in order to identify the structures that can be generally detected in images, and

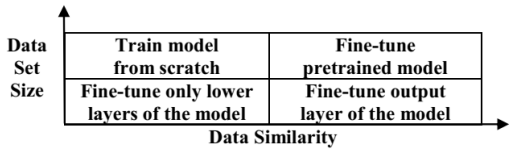


Figure 2: The relationship between data similarity, data size, and network tuning.

retain the weights of the upper levels with backpropagation, enabling the model to identify the unique features of a given set of images, such as fundus images, with much less time, training examples, and computational power [14][19]. Data are analyzed in different methods based on the complexity of the problem and the similarity to, or difference from, the data on which the neural network has been trained. Figure 2 presents the relationship between data similarity, data size, and required tuning.

TL methods include feature extraction, copying the architecture of a pre-trans model, and freezing some layers while training the others. An experiment was conducted on three TL models [20], being namely, (Vgg-16, Vgg-19, InceptionV3) in addition to the techniques of data augmentation and local average coloring for the removal of camera noise. Data are classified binarily once and then quinarily once again. Such an experiment demonstrated that increasing data accuracy is directly associated with the number of convolution and pooling layers in the model. Vgg-19 achieved an accuracy of up to 80% and 76.9% in binary and quinary classification, respectively.

Architecture [21] has been deployed for DR diagnosis using Messidor-1 datasets and GoogLeNet, AlexNet pre-trained architecture. GoogLeNet achieved the highest accuracy of 66% on the said dataset. In continuation of such efforts, we have sought herein to devise an effective transfer learning algorithm for processing fundus images for providing a faster and more accurate identification of the distinguishing pathological features of every eye image.

2.4 ResNet-34

As neural networks are inspired by the human brain and how it thinks, it is quite natural that the solution of complex problems that require deeper thinking is, thus, simulated by deeper networks for the solution of such problems. The main problem that is facing deep networks is the problem of vanishing gradients [22]. ResNet

Research	Year	DR-stages	Method	Accuracy
[21]	2018	Four-stage	TL with GoogLeNet with used adam optimizer and dropout regulation technique; CLAHE filtering used for preprocessing.	66%
[16]	2019	Binary (normal/abnormal)	CNN architecture with three stack convolutional filters in parallel in Conv layer and GMP layer. With several preprocessing steps.	81%
[20]	2019	Binary; and Five-stage	TL models Vgg-16, Vgg-19, InceptionV3 with data augmentation and local average color.	80% 76.9%
[12]	2020	Three-stage (normal/moderate/severe)	CNN with 18 Conv layer and 3 FC layer with preprocessing and data augmentation techniques.	88%

Table 1: A summary of previous research on the diagnosis of DR using various methods.

(residual network) [23] is a type of neural network that alleviates this problem of training deep learning networks by using skip-connections to "skip" a number of convolutional layers in every basic block in the network, a thing which provides alternative paths for original and derived data, rendering training faster and more possible. Such skip connections add the outputs of the prior blocks to the following ones, as expressed by the following equation:

$$y = F(x) + x \quad (1)$$

Where x is input, y is output, and F is the residual function). Each basic block consists of 2 convolution layers and a pooling layer (3x3 size), following by a (ReLU) activation function and batch normalization (BN). Figure 3 shows a learning block of residual learning. Using ResNet has greatly improved the performance of neural networks, where such networks are stacked with more layers for the creation of a deeper architecture, and hence, deeper learning, in contrast with shallower learning.

ResNet-34 [23] (ResNet with 34 layers) consists of 33 convolution layers and a max-pooling layer (3x3 size) and an average pooling layer, followed by a fully connected layer. Table 2 shows the architecture of ResNet-34.

3 Methodology

In order to develop the best model of optimum performance, we have used pre-trained CNN models that were trained and tested on the ImageNet dataset. Based on our dataset, each VggNet (Vgg-19), GoogLeNet (Xception), and ResNet (ResNet-34) has been trained and tested with a number of refinements for each model. The hyper parameter has been tuned to enhance the networks' ability to capture complex patterns in DR images. Several preprocessing and data augmentation techniques were applied on the dataset fundus images uniformly for all experiments in order to obtain comparable results.

Layer Name	Output Size	34-Layer
Conv1	112 x 112	7 x 7, 64, stride 2 3 x 3 max pool, stride 2
Conv2-x	56 x 56	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 3$
Conv3-x	28 x 28	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 4$
Conv4-x	14 x 14	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 6$
Conv5-x	7 x 7	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 3$
	1 x 1	Average pool,1000 fc, Softmax

Table 2: ResNet-34 architecture.

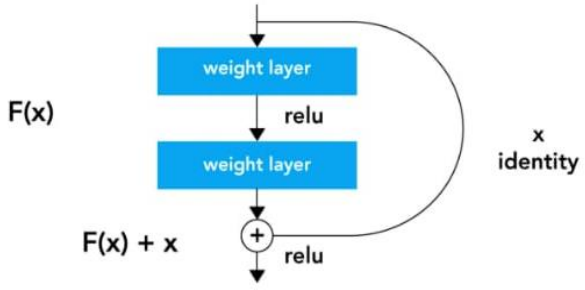


Figure 3: A learning block of residual learning.

3.1 Relevant approaches

Vgg-19 The Visual Geometry Group (VGG) model presented in [24], consisting of 19 weighted layers, divided into 5 blocks, was the first one that was trained. Each block consists of 2-4 convolution layers (Conv layers), followed by a pooling layer for the reduction of dimensions. The top of the architecture includes 3 fully connected layers (FC layers). Figure 4 shows a proposed modified Vgg-19 model. The FC layers were omitted from the original architecture and replaced with our custom classifier. Global average pooling (GAP) was added to convert the output from the convolutional layers (n x n x d) into a 1-dimension vector (1 x 1 x d) as an input to the required FC layers. Dropout (0.25) regularization technique was used to reduce overfitting. ReLU (y = max (0, x)) was used as activation function in the FC layer. The prediction layer is being used with 4 nodes and a Softmax activation function for predicting the four stages of DR:

$$F(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}} \quad (2)$$

Xception Model

The next model used is the Xception model [25], which is a CNN architecture that relies on depth-wise separable convolutions that contribute effectively to reducing computational cost and required memory size. This CNN model uses depth-wise separable convolution, which is an independent spatial convolution for each channel, followed by a pointwise convolution (1 x 1) across the channels. This can be thought of as looking firstly for correlations in a 2D space, and then looking for correlations in a 1D space. This 2D + 1D mapping appears to be easier to learn than a complete 3D mapping. This model, as shown in Figure 5, mainly consists of 36 Conv layers, distributed within 14 units, including linear residual connection. It was also used as the feature extraction, while a fully connected layer replaced the top of the architecture. GAP was added to receive the output of the Conv layers, and dropout was used as a regularization technique. FC Layer with (nod= 4) Is being used instead of (node = 1000) in the original architecture. The activation function is Softmax.

Cascaded CNN Model

Various CNN sub-models discover nonlinear discriminant features and semantic image descriptions from images at multiple levels of analysis [26]. In result, a cascaded CNN model will be extraordinarily generalized and helpful. In order to take advantage of CNN networks and their ability to extract features, the two aforementioned architectures (Vgg-19, Xception) have been concatenated as two different sources of knowledge in order to extract characteristics from an image in two different ways to enable models to achieve maximum learning of features from a given dataset. Thereafter, the outputs of each model are passed through GAP to reduce diminution. A merging

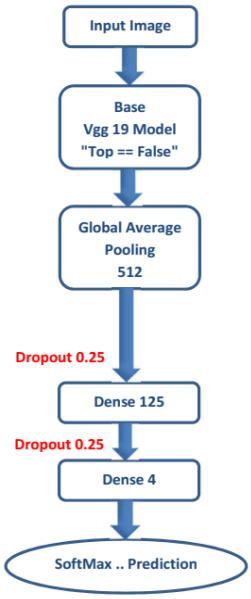


Figure 4: A proposed modified Vgg-19 model.

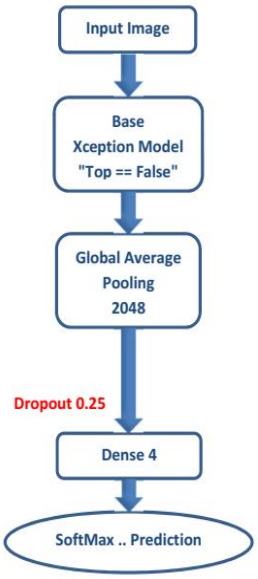


Figure 5: A proposed modified Xception model.

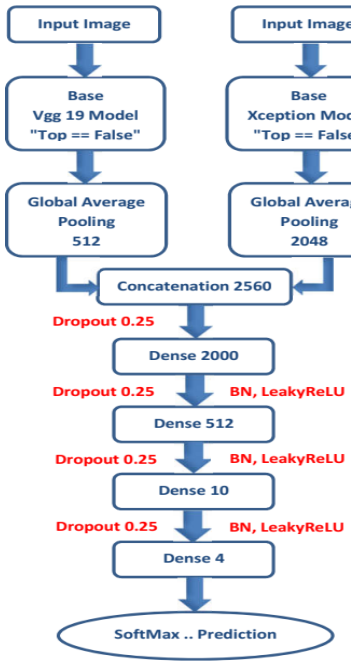


Figure 6: A Cascaded CNN Model.

layer is added to the top of each branch to combine the features that were deduced from various branches. Each branch's features are then concatenated and reformed into a vector:

$$x_{mrg} = Merge(x_1, x_2) \quad (3)$$

Where $x1$, $x2$ represent the outputs of the first and second branches, respectively.

The merged vector passes through four FC layers that are then added on the top of the merging layer with batch normalization and dropout in order to speed up processing and overcome the overfitting being greatly impaired by this network. The activation function used in FC layers is LeakyReLU. The topmost FC layer uses a Softmax activation function for prediction. Figure 6 illustrates a cascaded CNN model.

3.2 ResNet-34/DR

Architecture The proposed architecture is illustrated in Figure 7, where it can be divided into two parts: a features extraction part and a classifier part. ResNet-34 relied on the first part with the ability to handle trainable parameters. ResNet-34/DR consists of 16 basic units, with each unit consisting of 2 Conv layers (16 x 2 = 32 Conv layers in these blocks). ResNet-34/DR is composed of five convolutional groups in each group, where one or more Conv layer output passes through the BN layer and ReLU as a sequence (Conv→ BN→ ReLU) as demonstrated in section 2.4.

The first layer in ResNet-34/DR is a Conv layer with a (7 x 7) filter size that is flowed by a MaxPooling layer with (3 x 3) filters and a stride value of 2. Multiple identical residual units are Conv2-x, Conv3-x, Conv4-x, and Conv5-x, respectively, in the second to the fifth groups. ImageNet weights are used to initialize the first 33 layers (1+16 x 2 = 33 Conv layers in ResNet-34/DR. Then, the classifier part is being represented by the FC layer, followed by a Softmax activation function that is added to the ResNet-34/DR to conform to the DR Dataset's category label.

3.3 Selected Datasets

Our research has relied on APTOS 2019 blindness detection and the Indian Diabetic Retinopathy Image

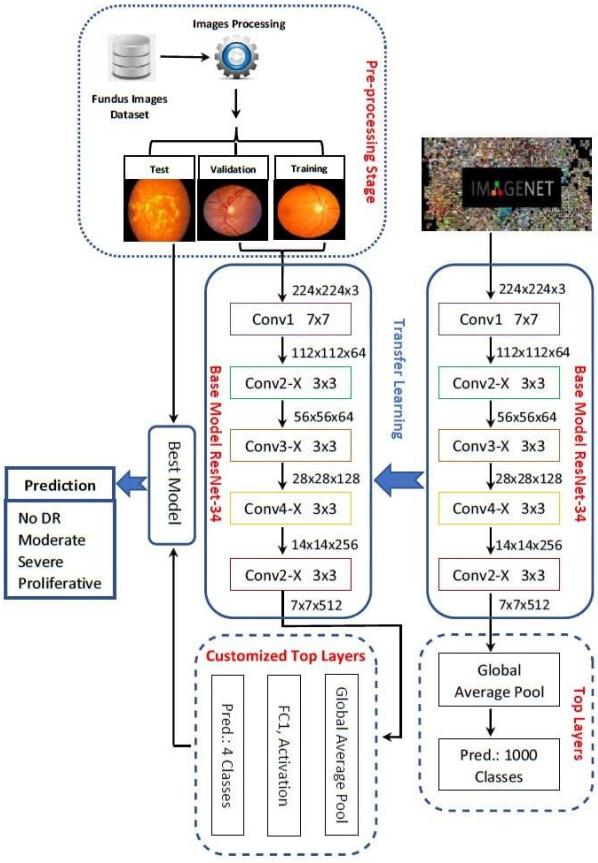


Figure 7: Proposed ResNet-34/DR architecture.

Class Index (Type)	DR Stage	APTOS 2019 Dataset	IDRID Dataset	Our Dataset
0	No DR	1805	134	1939
1	Mild	370	20	390
2	Moderate	999	136	1135
3	Severe	193	74	267
4	Proliferative	295	49	344

Table 3: The distribution of images among classes within various datasets.

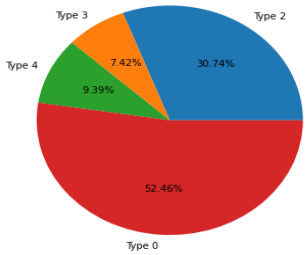


Figure 8: The unbalanced distribution of data within the categories, with the majority of data belonging to the (Normal) class.

Dataset (IDRID) that is available in the Kaggle dataset and used extensively by researchers. APTOS 2019, organized by the Asia Pacific Tele-Ophthalmology Society [18] provided a high-quality fundus image dataset (3662 images) taken by various cameras with various effects, such as camera flashing, low contrast, out of focus, etc. IDRID, being a part of a DR grading challenge [19], includes 512 fundus images. We have only used the training part of this dataset (413 images) because it contains the labels (the DR stage) we need for the classification process. All images have a resolution of 4288 x 2848 pixels.

Both datasets were combined in order to increase the volume of available data for training and because the categorized distribution of data was extremely unbalanced, as demonstrated in Figure 8. Both datasets contain fundus images accompanied by labels indicating one of the five different DR stages: (none (class-0), mild (class-1), moderate (class-2), severe (class-3), or proliferative (class4)).

The distribution of images among the classes is being shown in Table 3. As class-1 is almost healthy and its images are almost indistinguishable from those of class-0, and based on accustomed medical practice, the images of class-1 will be merged along with those of class-0 in the upcoming proceedings of our experiment in this paper.

4 Implementation

Preprocessing was carried out on a Python 3.7.9 environment. Deep-learning CNN models were trained on

Original Image | Gaussian Blurring | Adding Weight | Image Masking | Image Cropping

Figure 9: Preprocessing an eye fundus image (filtering).

Google Colab [19], which provides a free GPU Jupiter environment for implementation via the cloud, with the use of Keras and PyTorch deep learning frameworks and Scikit_learn, NumPy, Pandas, and MatPlotLib Python packages.

4.1 Image preprocessing

Numerous preprocessing steps have been used in our experiments to enhance and highlight disease-related features in fundus images and to configure the data for DL tasks, as follows:

Image Filtering

The images are raw data and they have been taken by different camera resolutions with different sizes, containing many effects. These observations were taken into account when dealing with the images to remove noises and increase robustness in our model. Specifically, we adopted the following preprocessing steps:
I) Gaussian blurring, II) add weight,
III) Masking, and IV) cropping & resizing. Firstly, the fundus images are blurred using the Gaussian function:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

Where σ indicates the distribution standard deviation. In our experiment, σ equals 30. This processing method was inspired and modified from Ben Graham's approach [30]. It is similar to medium filtering, but it employs a different kernel to generate a Gaussian (bell-shaped) hump. This is done to eliminate noise from such images.

In the next step, the output image from the previous step was combined with the original image using the equation:

$$I_c = \alpha I + \beta G(p) * I + \gamma \quad (5)$$

Where * denotes convolution, I denotes input images, G (p) denotes a Gaussian filter with a standard deviation, while α, β, γ are predefined parameters. Then, we took care to distinguish the fundus area from the background. We enriched the images with circle masks and a dark background. The last step was that about 10% of the image's outer borders on both sides are cut off, which does not include any helpful information. This sequence of preprocessing steps transformed every image in our dataset from a differently-sized image into a square-shaped image with a similar background and then resized

it to 500 x 500. After preprocessing a dataset, 10% of the data was being isolated for the testing set. The remaining data were randomly divided at a ratio of 75:25 for the training validation sets, including 2487, 829, and 368 images for training, validation, and testing phases, respectively. Figure 9 shows the preprocessing steps of an eye fundus image within the filtering stage.

Image Normalization This step is crucial in DL because it accelerates the convergence process on the gradient descent algorithm, thereby increasing the model's efficiency. A straightforward and effective method was used, which involved dividing each pixel in the image by 225.

Related Parameters	Vgg-19	Xc
Input Image Size	224 x 224	299
Batch Size	32	
Learning Rate	1×10^{-5}	5×1
Epochs	30	

Table 4: Hyper-pa

Data Augmentation We used the data augmentation technique to expand the training dataset artificially. Training deep learning models on additional data can result in more skilled models. Augmentation techniques can generate image variations that can improve the fit models' ability to generalize their learning to new images and avoid overfitting. Image augmentation generates artificial training images through various processing methods or combinations of multiple processing methods, such as random rotation, resizing, mirroring, shearing, and flipping.

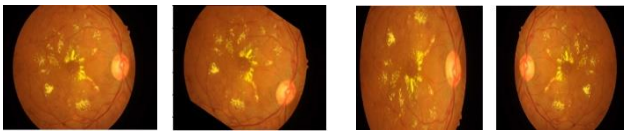
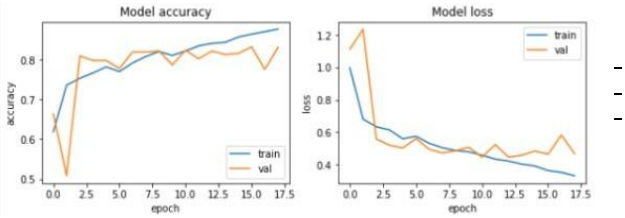
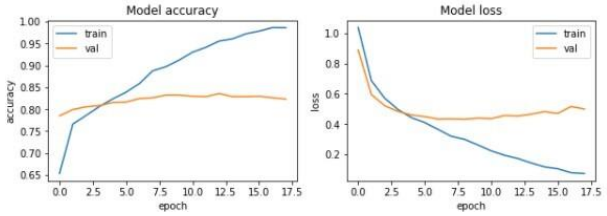


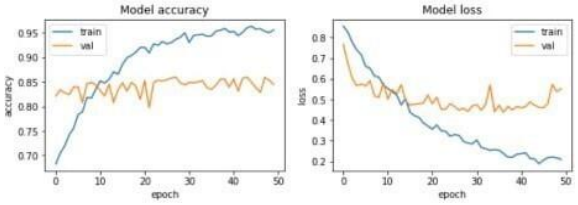
Figure 10: Our data augmentation techniques.



(a)Vgg19

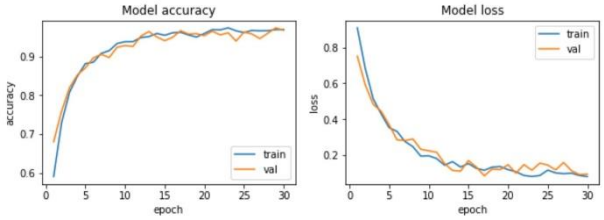


(b) Xception



(a) Cascaded CNN

Figure 11: Experimented CNN models' performance during the training phase.



(a) Training Accuracy **(b)** Training Loss

Figure 12: ResNet-34/DR learning curves during a training phase (epoch 30).

As depicted in Figure 10, several augmentation techniques are being applied to the training set, including zooming (10 degrees), horizontal flip, and random rotations between [-45, +45] degrees.

4.2 Training

After dividing the dataset during the training phase into training and validation sets (75:25), the validation set evaluates model performance improvement over time and selects the best parameter. Instead of generating random initial weights for all models, the advantages of transfer learning were relied upon, and the ImageNet pre trained weights were used as initial weights. This has significantly

contributed to speeding up the training process because the imported models have sufficient knowledge of images. The image size entered for each model was resized to reduce training time and avoid depleting resources and fitting to the input layer in each model. All models used Adam optimizer [31] to update the initial weights iteratively using a training set so that the model can adapt to the current problem area (classification DR). The error was calculated using categorical cross-entropy. An early stopping strategy has been used to stop training if training accuracy has not improved for ten consecutive cycles or worse performance in the validation set. This strategy also aids in reducing model overfitting. The optimal weights are saved in case training is halted prematurely and the validation error does not improve. The preprocessing parameters α, β, γ, p was sequentially set to -4, 30, 4, 128. Each model was trained with a different number of epochs, batch sizes, and learning rates during the training process to achieve the optimal result. The final hyper-parameters of our experiments are being presented in Table 4.

5 Results

5.1 Validation analysis

Models	Accuracy	Loss
Vgg-19	83%	0.44
Xception	83%	0.43
Cascaded CNN	85%	0.45
ResNet-34/DR	**95%**	**0.1**

Table 5: Models' performance comparison, with the best results, to be used for the testing phase, is being shown in bold format.

Several experiments were carried out in order to determine the solution with the best performance and to gain a better understanding of the performance of various networks. Plotting training curves for both training and validation data has enabled us to monitor the performance of the various networks during the training phase. Figure 11 clearly demonstrates the accuracy and loss performance of the three models (Vgg-19, Xception, cascaded CNN) when trained on the training and validation datasets. Our deep models were prone to overfitting, resulting in good training but poor validation performance. Over several epochs, models have reached their maximum degree of generalization, and validation loss has increased. In comparison, training loss continues to decrease over time.

Vgg-19 networks required an excessive amount of time to train, limiting the number of training epochs. Xception was faster to implement than its competitors. Our experiments with various networks are described in detail below. As previously stated, all models have high loss value, and neither model can detect and learn valuable patterns. The cascaded model was the most prone to overfitting, and attempts were made to mitigate it using batch normalization and dropout. However, we observed that increasing the complexity of the model did not produce a satisfactory result. Xception and cascaded

CNN models suffer from an overfitting problem in which the network's complexity is insufficient to capture the critical features of the landmarks.

The performance of the ResNet-34/DR model, which relies on the ResNet-34 architecture for feature extraction, was superior to that of the previous experiments as shown in Table 5 and Figure 12. As a result of the residual neural network's advantage, the architecture improved the model by providing a deeper network depth with a lower error rate, which contributed to the network's ability to extract more accurate patterns. Obviously, the best results were obtained using. Therefore, we have selected it as our final model for the testing phase.

5.2 Testing & evaluation

After obtaining the best model with the highest accuracy,

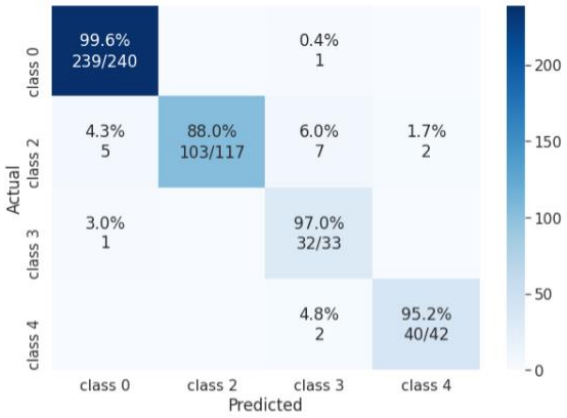


Figure 13: The confusion matrix of ResNet-34/DR testing phase.

Class (Type)	Specificity (%)	Precision (%)	Sensitivity (%)	$F1_{Score}$ (%)
Class 0	96.9	97	99.6	98
Class 2	100	100	88	93.6
Class 3	97.5	76.2	97	85.3
Class 4	99	95.2	95.2	95.2
Average	**98.5**	**92.0**	**95.0**	**93.2**

Table 6: ResNet-34/DR classification performance metrics.

ResNet-34/DR performance is being evaluated on the testing set (unseen data, constituting 10% of the entire dataset being used for this experiment) based on accuracy, sensitivity, specificity, precision, and F1 score as performance metrics (PMs). However, for a given number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), the following formulas represent performance metrics mathematically:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (7)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{F1}_{\text{Score}} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{10}$$

We know that our data is unbalanced, as shown in Figure 8. Therefore, accuracy could be deceptive and does not necessarily reflect the model's quality, as it already represents the correct evaluation of the total number of examples. In this case, precision, recall (i.e., sensitivity), and F1 score were chosen as the performance criteria for the model due to the high cost of false negatives and positives in medical diagnosis. F1 score is a combinational harmonic of the precision and recall metrics, describing the model's capability to detect class defects [32]. Thus, both precision and recall metrics contribute equally to the generation of the F1 score. Based on the calculations performed using equations (7), (8), (9), and (10) above, the confusion matrix, which summarizes our testing results, is being shown in Figure 13. Sensitivity (equation 9) indicates the proportion of fundus images with diabetic retinopathy which the model has identified as infected. Our model's average sensitivity rate was 0.95. This means that the model has correctly identified around 95% percent of infections in the testing data. Precision (equation 8) indicates the percentage of images identified as containing effects that include infected cases. For example, the value of 0.92 indicates that more than 92% of the images classified as infected are really infected. Finally, overall accuracy (equation 6) is 0.949, indicating that the model correctly recognizes more than 94% of all images (infected and uninfected). As shown in Table 6, ResNet-34/DR has achieved average accuracy, sensitivity; precision, specificity, and F1 score rates of 94.9%, 95.0%, 92.0%, 98.5%, and 93.2%, respectively.

6 Discussion

In this paper, the suitability and efficacy of our proposed work for the diagnosis of DR using fundus images were demonstrated. In comparison to other published research [12],[16],[20],[21] summarized in Table 1, our proposed model ResNet-34/DR achieved the highest accuracy 95%.

Our proposed work emphasized diagnosing DR at multiple stages, which is critical for detecting DR early and avoiding progression and vision loss. Early stages diagnosis contributes to reducing the limitations of human error and allows monitoring the development of the disease, which helps doctors improve medical treatment. In contrast, previous studies [16] and [20] relied solely on DR diagnosis to be normal or abnormal.

In addition to both studies in [12], [16] are constructed the CNN architecture from scratch. In contrast, our developed model relied on pre-trained TL models, which are more efficient and allow for the comparison of the performance of different networks to determine the best architecture for our problem. The author [21] paid little attention to data pre-processing, which could improve image quality and thus diagnostic accuracy, whereas our work evaluated these steps.

7 Conclusion

This paper has proposed the new ResNet-34/DR architecture, based on a deep CNN with the utilization of transfer learning, which can effectively classify Diabetic Retinopathy into four classes from publicly available Kaggle datasets (APTOS 2019, IDRID).

As initial attempts, two well-known architectures (Vgg-19 and Xception) were employed to classify DR stages. They suffered of high loss values due to overfitting, despite several attempts to reduce overfitting by adding dropout and batch normalization techniques and augmenting data. We believe the primary reason for this is that the data is highly skewed, with the vast majority of images falling within the healthy category. That is a significant impediment to networks extracting features, especially when the classes are few.

Transfer learning and fine-tuning on the pre-trained ResNet-34 network have proven to be extremely effective for our color fundus image dataset, yielding optimal performance metrics. Preprocessing provided a significant improvement to the color contrast of the input image. Data augmentation aided in increasing the training samples, especially for lower classes. The training technique employed by us in this paper has achieved a relative advancement in DR classification results.

As future work, we would look forward to compiling a dataset of our authentic images from Iraqi ophthalmologists. Additionally, we would like to build a new deep learning model from scratch and experiment it with modified pre-trained models. Finally, we might consider the utilization of different preprocessing techniques based on a semantic segmentation output that highlights the details of DR features and investigates how these changes affect the classification of DR stages, particularly, the early stages.

References

[1] W. H. O. and Others(2020), Diabetic retinopathy screening: a short guide. World Health Organization. Regional Office for Europe.

[2] M. T. Islam, H. R. H. Al-Absi, E. A. Ruagh, and T. Alam(2021), "DiaNet: A Deep Learning Based Architecture to Diagnose Diabetes Using Retinal Images only," IEEE Access, vol. 9, pp. 15686–15695,doi: 10.1109/ACCESS.2021.3052477.

[3] R. Sarki, K. Ahmed, H. Wang, and Y. Zhang(2020), "Automatic Detection of Diabetic Eye Disease through Deep Learning Using Fundus Images: A Survey," IEEE Access, vol. 8, pp. 151133–151149, doi: 10.1109/ACCESS.2020.3015258.

[4] J. J. Gómez-Valverde et al.(2019), "Automatic glaucoma classification using color fundus images based on convolutional neural networks and transfer learning," Biomed. Opt. Express, vol. 10, no. 2, p. 892, doi: 10.1364/boe.10.000892.

[5] X. Ma et al.(2021), “Understanding adversarial attacks on deep learning based medical image analysis systems,” Pattern Recognit., vol. 110, doi: 10.1016/j.patcog.2020.107332.

[6] Feizi, A. (2019), "Convolutional gating network for object tracking." International Journal of Engineering 32.7: 931-939.

[7] Sezavar, A., H. Farsi, and Sajad Mohamadzadeh(2019). "A modified grasshopper optimization algorithm combined with cnn for content based image retrieval." International Journal of Engineering 32.7: 924-930.

[8] S. Long, J. Chen, A. Hu, H. Liu, Z. Chen, and D. Zheng(2020), “Microaneurysms detection in color fundus images using machine learning based on directional local contrast,” Biomed. Eng. Online, vol. 19, no. 1, pp. 1–23, doi: 10.1186/s12938-020-00766-3.

[9] Y. Tong, W. Lu, Y. Yu, and Y. Shen(2020), “Application of machine learning in ophthalmic imaging modalities,” Eye Vis., vol. 7, no. 1, pp. 1–15, doi: 10.1186/s40662-020-00183-6.

[10] Z. Wang and J. Yang(2017), “Diabetic Retinopathy Detection via Deep Convolutional Networks for Discriminative Localization and Visual Explanation,” [Online]. Available: http://arxiv.org/abs/1703.10757.

[11] M. T. Hagos and S. Kant(2019), “Transfer learning based detection of Diabetic Retinopathy from small dataset,” arXiv,.

[12] M. Shaban et al(2020)., “A convolutional neural network for the screening and staging of diabetic retinopathy,” PLoS One, vol. 15, no. 6 June, pp. 1–13, doi: 10.1371/journal.pone.0233514.

[13] I. Namatēvs(2018), “Deep Convolutional Neural Networks: Structure, Feature Extraction and Training,” Inf. Technol. Manag. Sci., vol. 20, no. 1, pp. 40–47, doi: 10.1515/itms-2017-0007.

[14] M. Al-Smadi, M. Hammad, Q. B. Baker, and S. A. Al-Zboon(2021), “A transfer learning with deep neural network approach for diabetic retinopathy classification,” Int. J. Electr. Comput. Eng., vol. 11, no. 4, pp. 3492–3501, doi: 10.11591/ijece.v11i4.pp3492-3501.

[15] B. K. Triwijoyo, B. S. Sabarguna, W. Budiharto, and E. Abdurachman(2020), "Deep learning approach for classification of eye diseases based on color fundus images". Elsevier Inc.

[16] S. N. Chakravarthy, H. Singhal, and N. R. P. Yadav(2019), “DR-NET: A Stacked Convolutional Classifier Framework for Detection of Diabetic Retinopathy,” in Proceedings of the International Joint Conference on Neural Networks, vol. 2019-July, no. 1, doi: 10.1109/IJCNN.2019.8852011.

[17] P. K. V. Warkar(2021), “A Survey on Multiclass Image Classification based on Inception-v3 Transfer Learning Model,” Int. J. Res. Appl. Sci. Eng. Technol., vol. 9, no. 2, pp. 169–172, doi: 10.22214/ijraset.2021.33018.

[18] M. A. Morid, A. Borjali, and G. Del Fiol(2021), “A scoping review of transfer learning research on medical image analysis using ImageNet,” Comput. Biol. Med., vol. 128, no. 408, 2021, doi: 10.1016/j.compbiomed.2020.104115.

[19] A. K. Gangwar and V. Ravi(2021), Diabetic Retinopathy Detection Using Transfer Learning and Deep Learning, vol. 1176. Springer Singapore.

[20] A. Jain, A. Jalui, J. Jasani, Y. Lahoti, and R. Karani(2019), “Deep Learning for Detection and Severity Classification of Diabetic Retinopathy,” Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. ICIICT 2019, pp. 1–6, doi: 10.1109/ICIICT1.2019.8741456.

[21] Lam, C., Yi, D., Guo, M., Lindsey(2018), "T.: Automated detection of diabetic retinopathy using deep learning". AMIA Summits Transl. Sci. Proc. 2017, 147 .

[22] M. Gao, D. Qi, H. Mu, and J. Chen(2021), “A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects”.

[23] S. Wu, S. Zhong, and Y. Liu(2017), “Deep residual learning for image Recognition,” Multimed. Tools Appl., pp. 1–17, doi: 10.1007/s11042-017-4440-4.

[24] K. Simonyan, A. Zisserman(2014), “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556.

[25] F. Chollet(2017), ‘‘Xception: Deep learning with depthwise separable convolutions,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. , pp. 1251–1258.

[26] Y. Chai, H. Liu, and J. Xu(2017), “Glaucoma diagnosis based on both hidden features and domain knowledge through deep learning models” Knowledge-Based Syst., vol. 161, no. December 2017, pp. 147–156, 2018, doi: 10.1016/j.knosys.2018.07.043.

[27] “APTOS 2019 Blindness Detection.” [Online]. Available: https://www.kaggle.com/c/aptos2019-blindness-detection/.

[28] P. Prasanna, P. Samiksha, K. Ravi, K. Manesh, D. Girish, S. Vivek, and M. Fabrice. (2018). Indian Diabetic Retinopathy Image Dataset (IDRiD), doi: 10.21227/H25W98.

[29] colab.research.google.com/notebooks/welcome.ipynb#.

[30] B. Graham(2015), “Kaggle Diabetic Retinopathy Detection competition report,” pp. 1–9.

[31] D. P. Kingma and J. L. Ba(2015), “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” pp. 1–15.

[32] Konovalenko, I., Maruschak, P., Brevus, V., & Prentkovskis, O. (2021). "Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network". Metals, 11(4), 549.