IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 27, 2019, accepted October 4, 2019, date of publication October 15, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947484

A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection

SEHRISH QUMMAR[1], FIAZ GUL KHAN[1], SAJID SHAH[1], AHMAD KHAN[1], SHAHABODDIN SHAMSHIRBAND[2,3], ZIA UR REHMAN[1], IFTIKHAR AHMED KHAN[1], AND WAQAS JADOON[1]

[1]Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22010, Pakistan
[2]Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[3]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Corresponding author: Shahaboddin Shamshirband (shahaboddin.shamshirband@tdtu.edu.vn)

This work was supported in part by the Nvidia Corporation in the form of hardware donation, i.e., Tesla K40 GPU for this project

ABSTRACT Diabetic Retinopathy (DR) is an ophthalmic disease that damages retinal blood vessels. DR causes impaired vision and may even lead to blindness if it is not diagnosed in early stages. DR has five stages or classes, namely normal, mild, moderate, severe and PDR (Proliferative Diabetic Retinopathy). Normally, highly trained experts examine the colored fundus images to diagnose this fatal disease. This manual diagnosis of this condition (by clinicians) is tedious and error-prone. Therefore, various computer vision-based techniques have been proposed to automatically detect DR and its different stages from retina images. However, these methods are unable to encode the underlying complicated features and can only classify DR's different stages with very low accuracy particularly, for the early stages. In this research, we used the publicly available Kaggle dataset of retina images to train an ensemble of five deep Convolution Neural Network (CNN) models (Resnet50, Inceptionv3, Xception, Dense121, Dense169) to encode the rich features and improve the classification for different stages of DR. The experimental results show that the proposed model detects all the stages of DR unlike the current methods and performs better compared to state-of-the-art methods on the same Kaggle dataset.

INDEX TERMS CNN, diabetic retinopathy, deep learning, ensemble model, fundus images, medical image analysis.

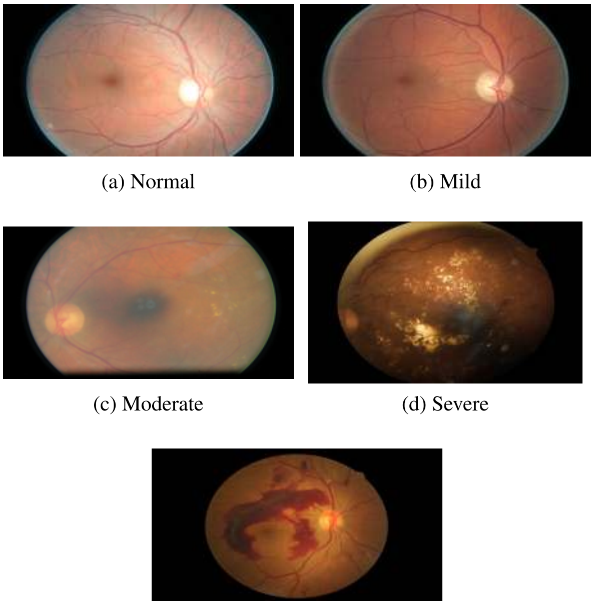
I. INTRODUCTION

Diabetic Retinopathy (DR) is one of the major causes of blindness. DR mutilate the retinal blood vessels of a patient having diabetes. The DR has two major types: the Non-Proliferative Diabetic Retinopathy (NPDR) and Proliferative Diabetic Retinopathy (PDR) [1]. The DR in the early stages is called NPDR which is further divided into Mild, Moderate, and Severe stages. Where the mild stage has one micro-aneurysm (MA), a small circular red dot at the end of blood vessels. In the Moderate stage the MAs rapture into deeper layers and form a flame-shaped hemorrhage in the retina. The severe stage contains more than 20 intraretinal hemorrhages in each of the four quadrants, having definite venous bleeding with prominent intraretinal microvascular abnormalities. PDR is the advanced stage of DR which leads to neovascularization, a natural formation of new blood vessels in the form of functional microvascular networks that grow on the inside surface of the retina [2]. The figure, 1 visually presents the different stages of DR. It is clear from the given figure the Normal and Mild stage looks visually similar. Hence, it is difficult to detect the Mild stage.

Globally, the number of DR patients is expected to increase from 382 million to 592 million by 2025 [3]. A survey [3] conducted in the province of Khyber Pakhtunkhwa (KPK), Pakistan, report 30% of diabetes patients are affected by DR in which 5.6% succumbs to blindness. Over time, the mild NPDR develops into PDR if not controlled in the early stages. Another survey [1], conducted in Sindh, Pakistan, observed 130 patients with DR symptoms. It is reported that 23.85% of the total observed patients were DR in which 25.8% were diagnosed as PDR patients.

In the early stages of the DR the patients are asymptomatic but in advanced stages, it leads to floaters, blurred vision,

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


(a) Normal

(b) Mild

(c) Moderate

(d) Severe

(e) PDR

FIGURE 1. The different stages of DR.

distortions, and progressive visual acuity loss. Hence it is difficult but utmost important to detect the DR in early stages to avoid the worse effect of latter stages.

As explained in the previous section, the color fundus images are used for the diagnosis of DR. The manual analysis can only be done by highly trained domain experts and is, therefore, expensive in terms of time and cost. Therefore, it is important to use computer vision methods to automatically analyze the fundus images and assist the physicians/radiologists. The computer vision-based methods are divided into hand-on engineering [4]–[7] and end-to-end learning [8]–[10]. The hand-on engineering methods extract features using traditional approaches such as HoG [11], SIFT [12], LBP [13], Gabor filters [14] and etc which fails to encode the variations in scale, rotation, and illumination. The end-to-end leaning automatically learns the hidden rich features and thus performs better classification. Many hand-on engineering and end-to-end learning-based approaches [15]–[17] are used to detect the DR in Kaggle dataset[1] but no approach is able to detect the Mild stage. The detection of the mild stage is important for the early control of this fatal disease. This study focuses to detect all the stages of DR (including the mild stage) using end-to-end deep ensemble networks. The results show that the proposed approach outperforms state-of-the-art methods.

The remaining sections are organized as follows: Section **??** reviews recent literature to detect and classify diabetic retinopathy. The proposed methodology is presented in Section III while Section IV contain the results and discussion followed by conclusion in Section V.

II. RELATED WORK

The classification of DR has been extensively studied in the literature. Gondal *et al.* [18] proposed a CNN model for the referable Diabetic Retinopathy (RDR). They used two publicly available datasets Kaggle and DiaretDB1, where the Kaggle dataset is used for training and DiaretDB1 is used for testing. They are doing binary classification as normal and mild stages are considered as non-referable DR where the rest of the three stages are used as referable DR. The performance of the CNN model is evaluated based on binary classification resulting in sensitivity 93.6% and specificity 97.6% on DiaretDB1. Wang *et al.* [19] proposed a novel architecture that classifies the images as normal/abnormal, referable/non-referable DR and gets the high AUC on a normal and referable DR task 0.978 and 0.960 respectively and specificity is 0.5. Their proposed method uses three networks: main, attention and crop. The main network uses the Inception model that is trained on ImageNet where the attention network highlights different types of lesions in the images and crop network's crop the high attention image. Quelle *et al.* [10] proposed three CNN models for binary classification and detected DR lesions. They also used the Kaggle and DiaretDb1 dataset for training and testing respectively. Diabetic retinopathy has five (5) stages to classify the occurrence of diseases. The stage-wise classification is discussed by Chanrakumar and Kathirvel [20] introduced the CNN model with a dropout regularization technique trained on the Kaggle dataset and tested on DRIVE and STARE dataset. The accuracy achieved by their model is 94%. They manually performed an augmentation and preprocessing steps by using an image editing tool. Moreover, CNN architecture is applied to the Kaggle dataset proposed by Memon *et al.* [1]. The preprocessing is done on the dataset and they used nonlocal mean denoising and added a delta value to get the equal level of brightness of the images. For evaluation, the overall kappa score accuracy is 0.74, for the validation purpose, 10% of the images were used. Pratt *et al.* [21] proposed a CNN architecture used for classifying five stages but could not classify the mild stage accurately, due to the nature of architecture. Another limitation is that they used the skewed dataset of Kaggle that led to the high specificity with the tradeoff in low sensitivity. Yang *et al.* [22] proposed DCNN (Deep Convolution Neural Network) for two stages of DR (normal and NPDR). The preprocessed data is given as input to the two networks (local and global). Lesions are highlighted and sent to the global network for grading. They used class weight and kappa scores for evaluation of the model. However, the PDR stage was not considered in their work.

In [16], [23], the authors checked the performance of the Kaggle dataset over different CNN models. Garcia *et al.* [23] proposed a method of using the right and left eye images separately and applied CNN (Alexnet, VGGnet16, etc.). The preprocessing and augmentation phases were performed on the dataset to improve the contrast of images. They achieve the best results on VGG16 with no fully connected layer and achieved 93.65% specificity, 54.47% sensitivity, and

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

TABLE 1. Dataset: distribution of different classes.

	Class-0 (Normal)	Class-1 (Mild)	Class-2 (Moderate)	Class-3 (Severe)	Class-4 (PDR)	Total
Original	25810	2443	5292	873	708	35126
Up Sample	25810	25810	25810	25810	25810	129050
Down Sample	708	708	708	708	708	3540

83.68% accuracy. However, DR stages were not explicitly classified in their work. Dutta *et al.* [16] used Kaggle dataset with three deep learning models (Feed Forward Neural Network (FNN), Deep Neural Network (DNN), and Convolutional Neural Network (CNN). They used 2000 images out of 35128 images with a 7:3 validation split. They applied many preprocessing steps (median, mean, Std deviation, etc.) and then trained their model on the training dataset. The Best training accuracy of 89.6% was obtained on DNN.

We can summarize all these works of DR classification problem into two groups. The first one is the binary classification of DR i.e. either the patient has DR or not. The problem in this method is we can not identify the severeness of the disease even after knowing that a patient has DR. The solution to this problem is multi-class classification. In multi-class classification, we classify DR into five different classes or stages as discussed in the introduction section. But most of the related work is unable to properly classify all the stages of DR, especially the early stages. It is important to predict the DR at early stages for the cure as in later stages it is difficult to cure the disease and can lead to blindness. To the best of our knowledge, no other work has detected the mild stages of DR, by using the Kaggle dataset which we have used for our research. Our model can detect the mild stage and performs better than the current state of the art. Moreover, in the related work, no one has shown the effect of a balanced dataset. The imbalanced dataset can lead to bias in the classification's accuracy. If the samples in the classes are equally distributed as in the case of a balanced dataset then the network can learn the features properly, but in case of unequal distribution network outperforms for highly sampled class. Moreover, the current CNN architectures for DR detection lacks to analyze the effect of different hyperparameters tuning (meta-learning) and its implications.

III. PROPOSED METHOD

A. PREPROCESSING

The different preprocessing steps that we perform on input dataset before giving it to the model are shown in Figure 2

We use the Kaggle[2] dataset which contains 35126 color fundus images, each is of size 3888 × 2951. It contains the images from five different classes based on the severity of diabetic retinopathy (DR). Table 1, shows the distribution of sample images in different classes of the Kaggle dataset The distribution of different classes is shown in the first row of Table 1, which is perfectly imbalanced. Training of deep networks with imbalance data leads to classification biasness. In the first preprocessing step, we resize each input image shown in Figure 2 (a) to 786 × 512 shown in Figure 2 (b) by maintaining the aspect ratio to reduce the training overhead of deep networks. Moreover, for balancing the dataset we performed up-sampling and down-sampling [24]. The up-sampling (The Table 1, second row) is performed with augmentation of minority classes by randomly cropping patches, of size 512 × 512 as shown in Figure 2 (c), followed by flipping and 90^o rotation to balance the samples of different classes, enrich dataset and avoid overfilling as shown in Figure 2 (e). In down-sampling (Table 1, third row) extra instances of majority classes are removed to meet the cardinality of the smallest class. In the resultant distributions, before flipping and rotation, each image is mean normalized to avoid features biasness and speed-up training time, shown in Figure 2 (d). The dataset is divided into three parts: training, testing, and validation sets with ratio 64% and 20% and 16% respectively. During training, the validation set is used to check and reduce the over-fitting.

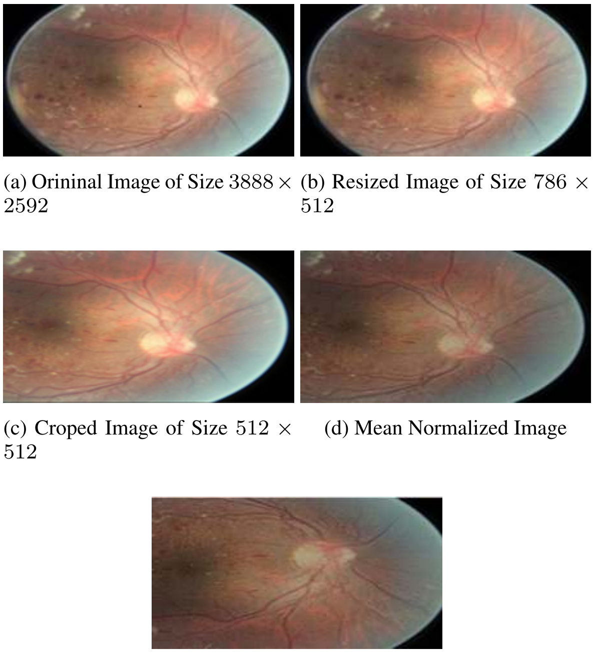

(a) Orininal Image of Size 3888 × 2592
(b) Resized Image of Size 786 × 512
(c) Croped Image of Size 512 × 512
(d) Mean Normalized Image
(e) Rotated Image

FIGURE 2. The different preprocessing steps.

B. ENSEMBLE MODEL

Ensemble method is a meta-algorithms that combine several machine learning techniques into one predictive model.

[2] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

Algorithm 1 Proposed Algorithm

```
Require: Fundus Images $(X, Y)$; where $Y = \{y/y \in \{Normal, Mild, Moderate, Severe, PDR\}\}$
Output : The trained model that classifies the fundus image $x \in X$
Perform Preprocessing:
    • Resize the image to dimension $786 \times 512$
    • Perform augmentation: Randomly crop five patches, of size $512 \times 512$, of each image and perform flip flop and $90^o$
      rotation
    • Mean normalize the each image
Import a set of pre-trained models $\mathcal{H} = \{Resnet50, Inceptionv3, Xception, Dense121, Dense169\}$.
Replace the last fully connected layer of each model by a layer of $(5 \times 1)$ dimension.
foreach $\forall h \in \mathcal{H}$ do
    $\alpha = 0.01$
    for epochs=1 to 50 do
        foreach mini batch $(X_i, Y_i) \in (X_{train}, Y_{train})$ do
            Update the parameters of the model $h(\cdot)$ using Eq.(2)
            if the validation error is not improving for five epochs then
                $\alpha = \alpha \times 0.01$
            end
        end
    end
end
foreach $x \in X_{test}$ do
    Ensemble the output of all models, $h \in \mathcal{H}$, using Eq.(3)
end
```

It can be used for different objectives such as to decrease variance (Bagging), bias (boosting), or improve predictions (stacking). Stacking is a model used to combine information from multiple predictive models to generate a new model. The stacked approach often outperform individual models due to its soothing nature. Stacking highlights each base model where it performs best and discredits each base model where it performs poorly. For this reason, stacking is most effective when the base models are significantly different. For this reason, we used stacking to improve the prediction of our model, which is evident from our results

The proposed approach ensembles the five deep CNN models Resnet50 [25], Inceptionv3 [26], Xception [27], Dense - 121 [28], Dense169 [28]. Algorithm 1 presents the proposed model in detail. Let $\mathcal{H} = \{Resnet50, Inceptionv3, Xception, Dense121, Dense169\}$ be the set of pre-trained models. Each model is fine tuned with the Fundus Images dataset (X, Y); where X the set of N images, each of size, 512×512, and Y contain the corresponding labels, $Y = \{y/y \in \{Normal, Mild, Moderate, Severe, \mathcal{P}DR\}\}$. We divide the training set (X_{train}, Y_{train}) into mini batches, each of size $n = 8$, such that $(X_i, Y_i) \in (X_{train}, \mathcal{Y}_{train})$, $i = 1, 2, \ldots \frac{N}{n}$ and iteratively optimizes (fine tuning) the CNN model $h \in \mathcal{H}$ to reduce the empirical loss:

$$L(w, X_i) = \frac{1}{n} \sum_{x \in X_i, y \in Y_i} l(h(x, w), y) \tag{1}$$

where $h(x, w)$ is the CNN model that predicts class y for input x given w and $l(\cdot)$ is the categorical cross-entropy loss penalty function. The Nesterov-accelerated Adaptive Moment Estimation [29] is used to update the learning parameters:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}} + \epsilon}\left(\beta_1 \hat{m}_t + \frac{(1-\beta_1)\frac{\partial}{\partial w_t} L(w_t, X_i)}{1-\beta_1^t}\right) \tag{2}$$

where α, $\hat{m}$ and $\hat{v}$ are the learning rate, first-order moment and a second-order moment of the gradient respectively. While β_1 and β_2 represent the decarov Momentum hy rates which are initially set to 0.9. The Nestelps to know the direction of the next step and avoid the fluctuations. In the start w_t, $t = 0$, is initialized to the learned weights of the model $h \in \mathcal{H}$ using transfer learning [30]. The output layer of each model, $h \in \mathcal{H}$, uses *SoftMax* as an activation function which generates the probabilities that how much the input belongs to the set of different classes $\{Normal, Mild, Moderate, Severe, PDR\}$. The learning rate, α, is initially set to 10^{-2} and decreased by a factor of 0.1 to 10^{-5}. We use 50 epochs for training with early stopping if the model starts over-fitting.

In the case of testing, to predict the class label of the unseen example, we use stacking to combine the results of all different models and generate a unified output. The ensemble approach combines the strengths of individual models and leads to better performance. The proposed stacking ensemble is illustrated in Figure 3. Let x_{test} be a new test sample, then the ensemble output is given by:

$$m^* = \arg\max_m \frac{\sum_{\forall h \in \mathcal{H}} h(w, x_{test})}{|\mathcal{H}|} \tag{3}$$

TABLE 2. The distribution of samples in test datasets.

	Class-0 (Normal)	Class-1 (Mild)	Class-2 (Moderate)	Class-3 (Severe)	Class-4 (PDR)	Total
Imbalance	4119	391	845	140	113	5608
Up	4119	4119	4119	4119	4119	20595
Down	113	113	113	113	113	565

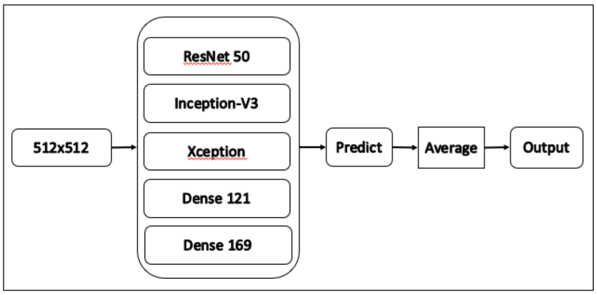


FIGURE 3. Proposed ensemble model.

where $h(\cdot)$ is the fine tuned model, $|\mathcal{H}|$ is the cardinality of the models and m represents the different modalities such that $m \in \{Normal, Mild, Moderate, Severe, PDR\}$.

In the case of imbalanced training data, the accuracy tends to bias towards majority classes [31].

IV. RESULTS AND DISCUSSION

In this section results of the proposed model are compared with state-of-the-artwork. The proposed model is trained on a high-end Graphics Processing Unit (GPU). The GPU used (NVIDIA Tesla k40) contains 2880 CUDA core and comes with NVIDIA CUDA Deep Neural Network library (CuDNN) for GPU learning. The deep learning package Keras[3] was used with the[4] TensorFlow machine learning back end library.

A. PERFORMANCE PARAMETERS

To quantitatively evaluate the proposed model we use accuracy, sensitivity, specificity, precision, F1 measures, AUC [32] and ROC [33] as performance metrics.

Accuracy: The accuracy can be calculated in terms of positive and negative classes:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where TP (True Positives) is the number of correctly classified instances of the class under observation, TN (True Negatives) is the number of correctly classified instances of rest of the classes, FP (False Positives) is the number of miss-classified instances of rest of the classes and FN (False Negatives) is the number of miss-classified instances of the class under observation.

TABLE 3. Imbalanced confusion matrix.

		Predicted Class Label				
		0	1	2	3	4
Actual Class Label	Class-0	4010	22	78	1	8
	Class-1	320	35	35	0	1
	Class-2	413	11	350	60	11
	Class-3	12	0	49	72	7
	Class-4	6	0	27	16	64

Recall/Sensitivity: it is the ratio of TP and $TP + FN$

$$Sensitivity = \frac{TP}{TP + FN}. \tag{5}$$

Specificity: it is the ratio of TN and $TN + FP$ Highlight

$$Specificity = \frac{TN}{TN + FP}. \tag{6}$$

Precision: it is the ratio of TP and $TP + FP$ Highlight

$$Precision = \frac{TP}{TP + FP}. \tag{7}$$

F1-Score: it is the weighted harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{8}$$

It return score between 0 and 1, where 1 means best score and 0 is the worst.

Receiver Operating Curve (ROC) [33]: plots the true positive rate (TPR) against the false positive rate (FPR).

Area Under The Curve (AUC) [32]: It represents the degree or measure of separability of different classes. The higher the AUC score means the better the model and vice versa.

To show the effect of the imbalanced dataset we have used three different datasets that are i) Imbalanced ii) Up Sample and iii) Down Sample Dataset. In the end, we also have shown the effect of different hyper-parameters on the overall performance of the model. The distribution of Test dataset samples is given in Table 2.

Imbalanced Dataset: The distribution of imbalanced dataset test samples of different classes is given in the first row of Table 2. However, accuracy is a misleading metric when the dataset is highly imbalanced [31], it gives bias results. In our case, the accuracy achieved is biased towards the negative class which is class-0 (normal). So along with accuracy, we have also used other parameters such as Recall, Precision, Specificity, F1-score, and ROC-curve to provide unbiased results. The achieved accuracy, recall, specificity,

[3] http://keras.io/

[4] https://www.tensorflow.org/

TABLE 4. Comparison of the proposed method with Pratt et. al [21].

	Recall		Precision		Specificity		F1-Score	
Class	Our Model	Pratt et. al[22]	Our Model	Pratt et. al[22]	Our v	Pratt et. al [22]	Our Model	Pratt et. al[22]
0	**0.97**	0.95	**0.84**	0.78	**0.40**	0.19	**0.90**	0.85
1	**0.80**	0.00	**0.51**	0.00	**0.99**	1.00	**0.15**	0.00
2	**0.41**	0.23	**0.65**	0.40	**0.95**	0.93	**0.50**	0.29
3	**0.51**	0.78	**0.48**	0.52	**0.98**	0.99	**0.49**	0.10
4	**0.56**	0.44	**0.69**	0.32	**0.99**	0.97	**0.62**	0.37

precision, and F1-score are 80.8%, 51.5%, 86.72%, 63.85% and 53.74% respectively. Table 3 shows the confusion matrix of a final ensemble model. Where recall, specificity, precision and F1-score of each class is shown in Table 4 along with the comparison of Pratt *et al.* [21] results. Class 0 give maximum recall due to the more number of negative samples which also provides better accuracy. Class 1 gives minimum recall due to a very minute characteristic and feature which make it difficult to detect DR stage.

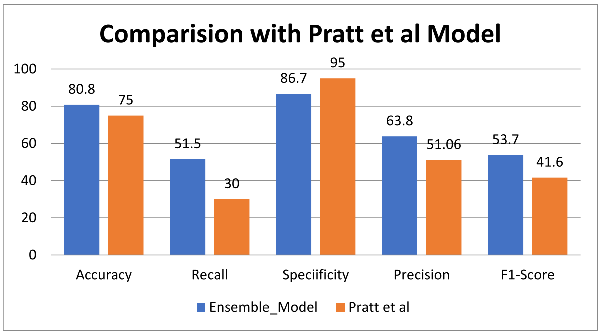


FIGURE 4. Comparison of our model with [21].

TABLE 5. Confusion of up sample dataset.

		Predicted Class Label				
		0	1	2	3	4
Actual Class Label	Class-0	4006	23	80	2	8
	Class-1	1009	2248	831	18	13
	Class-2	683	721	2095	490	130
	Class-3	83	48	1136	2540	312
	Class-4	33	2	750	879	2455

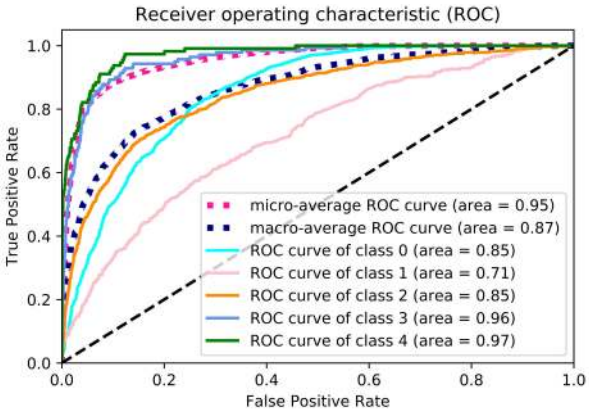


FIGURE 5. ROC-curve of imbalanced dataset.

TABLE 6. Performance measure of each class of up sample dataset.

	Recall	Precision	Specificity	F1-Score
Class-0	0.97	0.68	0.68	0.80
Class-1	0.54	0.73	0.73	0.62
Class-2	0.50	0.42	0.42	0.46
Class-3	0.61	0.64	0.64	0.63
Class-4	0.59	0.84	0.84	0.69

TABLE 7. Performance measure of each class of down sample dataset.

	Recall	Precision	Specificity	F1-Score
class-0	1.00	0.45	0.60	0.62
class-1	0.07	0.89	0.99	0.13
class-2	0.71	0.50	0.77	0.58
class-3	0.56	0.75	0.92	0.64
class-4	0.56	0.91	0.97	0.69

Figure 4 shows the summary of our results on an imbalanced dataset and provides a comparison with Pratt *et al.* [21]. The model proposed by Pratt is considered for comparing our results because it uses the same dataset and works on classifying the five stages of DR. However, they are unable to classify the mild stage. In the best of our knowledge, no other model classifies all the five stages of DR using Kaggle dataset. Our model outperforms the Pratt *et al.* [21] model in all performance parameters as shown in figure 4. The accuracy, sensitivity, specificity, precision, and F1-score of our model and Pratt et al are shown in Figure 4. As our model improves the accuracy, sensitivity, precision, F1-score and decreases the specificity. Moreover, our model detects and classifies all five stages of DR. Further for our model we also calculated the ROC (Receiver Operating Characteristics) shows in Figure 5. As the ROC curve shows how our model distinguishes among classes. The highest AUC of 0.97 is achieved by class 4 which has 113 samples. The ROC curve of class-0 is 0.85. The micro and macro average also shows the overall performance of the model. As, the micro-average ROC sum up the individual true positive, false positive and false negative and then map a value on a graph. Where macro-average takes the average of precision and recall and map a value on a graph. However, the micro-average ROC is considered when the dataset is highly imbalanced. The Result with Balanced Dataset: There are many techniques for balancing the datasets, such as penalized Models, Anomaly Detection, etc. [24], [34]. However, we used up and down-sampling because the network preferred to minimize the loss by just learning to classify the high-occurrence classes well and ignoring the low-occurrence ones.

TABLE 8. Configuration of Hyper-parameters in models.

No.	Model	Layers	Batch size	Momentum	Epoch	Learning Rate	Optimizer
1	ResNet50	168	8	0.9	50	0.01,0.001, 0.0001, 1e-05	Stochastic Gradient Descent (SGD)
	Inception-V3	159					
	Xception	126					
	Dense121	121					
	Dense169	169					
2	ResNet50	168	8	0.9	50	0.001, 0.0001, 1e-05	Adaptive Moment Estima-tion (Adam)
	Inception-V3	159					
	Xception	126					
	Dense121	121					
	Dense169	169					

TABLE 9. SGD and Adam with different learning rate on imbalanced, up sample and downsample dataset.

Learning rate	Performance Measure	Imbalanced		Up		Down	
		SGD	Adam	SGD	Adam	SGD	Adam
0.01	Accuracy	0.15	...	0.20	...	0.20	...
	Recall	0.20	...	0.20	...	0.20	
	Precision	0.00	...	0.00	...	0.00	...
	Specificity	0.80	...	0.80	...	0.80	...
	F1-Score	0.00	...	0.00	...	0.00	...
0.001	Accuracy	0.50	0.15	0.42	0.20	0.38	0.19
	Recall	0.39	0.20	0.43	0.20	0.38	0.47
	Precision	0.00	0.00	0.72	0.00	0.00	0.00
	Specificity	0.76	0.79	0.82	0.79	0.78	0.52
	F1-Score	0.00	0.00	0.38	0.00	0.00	0.00
0.0001	Accuracy	0.55	0.45	0.55	0.53	0.48	0.49
	Recall	0.47	0.47	0.55	0.53	0.48	0.49
	Precision	0.47	0.35	0.57	0.55	0.52	0.50
	Specificity	0.79	0.76	0.83	0.82	0.79	0.80
	F1-Score	0.39	0.35	0.52	0.53	0.46	0.49
1.00E-05	Accuracy	0.70	0.55	0.60	0.57	0.51	0.51
	Recall	0.49	0.49	0.60	0.57	0.51	0.51
	Precision	0.49	0.41	0.59	0.57	0.55	0.53
	Specificity	0.82	0.79	0.85	0.84	0.82	0.81
	F1-Score	0.45	0.40	0.57	0.55	0.48	0.50

The distribution of Up and Down samples dataset for training and test sets are shown in Table 1 and Table 2 respectively. The confusion matrix of Up Sample Dataset is shown in Table 5 and sensitivity, specificity, precision and F1-score of each class are given in Table 6.

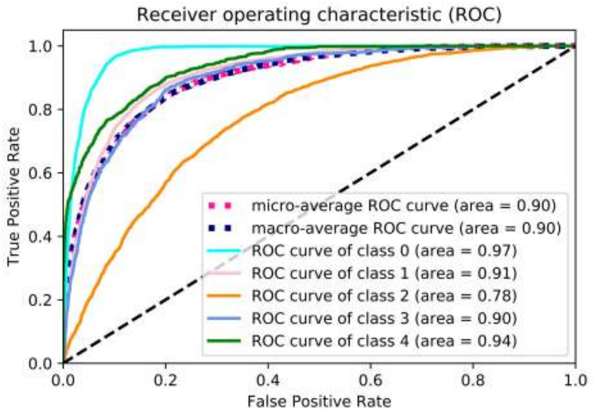


FIGURE 6. ROC-curve of up sample dataset.

The ROC curve of up sample dataset is shown in Figure 6. The maximum calculated AUC for class 0 is 0.97, which shows our model predicts well for this class. On a balanced dataset, micro and macro average ROC curves have the same value that is 0.90. For down sample datasets, to remove the biasness of accuracy we randomly consider ten different subsets of this dataset. Each subset contains 113 images of each class. All the performance measures for the downsampled dataset are averaged over the ten subsets. Results of the Down sampled Dataset is shown in Table 7. The accuracy, recall, precision, specificity, and F1-Score achieved by the down-sampling are 58.08%, 58.10%, 70.3%, 85.5%, and 53.64% respectively.

The results obtained for imbalanced and balanced (Random Up and Downsampling) datasets are shown in Table 9. The trends illustrated that if the learning rate reduces from 0.01 to 1e-05 the recall and accuracy improves, but the specificity is affected due to misclassification of the positive class. Table 10 and 11 shows how the change in learning rate affects the model in classifying the positive class images based on precision-recall and AUC. As AUC is better when its value is near to 1 but if the value is 50 or below 50 then a model is poor. With the learning rate value set to 0.01, the model train very fast, but misclassified the positive classes in both datasets (Imbalanced and Up). When a learning rate value is reduced to 0.0001 we get a minor improvement in the model. However, the learning rate value is again decreased

TABLE 10. SGD with imbalanced, up, and down dataset precision-recall and AUC.

		Learning Rates											
		Imbalanced				Up				Down			
		0.01	0.001	0.0001	1.00E-05	0.01	0.001	0.0001	1.00E-05	0.01	0.001	0.0001	1.00E-05
Recall	class-0	0.00	0.64	0.65	0.87	0.00	0.64	0.65	0.87	0.00	0.62	0.63	0.87
	class-1	0.00	0.02	0.46	0.20	0.00	0.19	0.84	0.71	0.00	0.02	0.48	0.21
	class-2	1.00	0.00	0.10	0.15	1.00	0.00	0.10	0.14	0.20	0.00	0.15	0.23
	class-3	0.00	0.97	0.72	0.71	0.00	0.97	0.73	0.73	0.00	0.97	0.74	0.73
	class-4	0.00	0.31	0.42	0.54	0.00	0.34	0.46	0.55	0.00	0.31	0.42	0.54
Precision	class-0	0.00	0.87	0.83	0.83	0.00	0.71	0.67	0.65	0.00	0.49	0.37	0.40
	class-1	0.00	0.09	0.10	0.12	0.00	0.71	0.43	0.52	0.00	0.46	0.36	0.41
	class-2	0.00	0.00	0.61	0.59	0.00	1.00	0.37	0.44	0.2	0.00	0.42	0.48
	class-3	0.00	0.06	0.34	0.33	0.00	0.28	0.57	0.56	0.00	0.29	0.59	0.60
	class-4	0.00	0.76	0.53	0.62	0.00	0.94	0.83	0.82	0.00	0.97	0.88	0.86
AUC	class-0	0.50	0.76	0.75	0.78	0.49	0.92	0.94	0.95	0.50	0.83	0.84	0.87
	class-1	0.51	0.56	0.64	0.63	0.51	0.85	0.88	0.88	0.51	0.77	0.80	0.79
	class-2	0.50	0 .62	0.77	0.81	0.51	0.65	0.70	0.73	0.51	0.64	0.74	0.77
	class-3	0.50	0.93	0.95	0.96	0.49	0.88	0.89	0.89	0.50	0.87	0.89	0.89
	class-4	0.51	0.94	0.96	0.97	0.50	0.89	0.92	0.92	0.51	0.89	0.91	0.92

TABLE 11. Adam with imbalanced, up, and down sample dataset precision-recall and AUC.

		Learning Rates								
		Imbalanced			Up			Down		
		0.001	0.0001	1.00E-05	0.001	0.0001	1.00E-05	0.001	0.0001	1.00E-05
Recall	Class-0	0.00	0.49	0.64	0.00	0.49	0.64	0.00	0.49	0.62
	Class-1	0.00	0.49	0.40	0.00	0.73	0.77	0.00	0.49	0.41
	Class-2	1.00	0.23	0.22	0.99	0.30	0.17	0.99	0.30	0.29
	Class-3	0.00	0.65	0.69	0.00	0.58	0.71	0.00	0.67	0.71
	Class-4	0.00	0.54	0.56	0.00	0.55	0.57	0.00	0.54	0.55
Precision	Class-0	0.50	0.84	0.84	0.20	0.70	0.70	0.00	0.43	0.43
	Class-1	0.00	0.09	0.10	0.00	0.46	0.50	0.00	0.45	0.46
	Class-2	0.15	0.31	0.49	0.20	0.33	0.42	0.19	0.37	0.47
	Class-3	0.00	0.28	0.26	0.00	0.57	0.53	0.00	0.56	0.53
	Class-4	0.00	0.24	0.37	0.11	0.70	0.73	0.00	0.73	0.77
AUC	Class-0	0.50	0.72	0.76	0.50	0.92	0.94	0.50	0.81	0.84
	Class-1	0.50	0.60	0.61	0.50	0.85	0.87	0.50	0.80	0.82
	Class-2	0.50	0.69	0.73	0.50	0.68	0.72	0.50	0.71	0.74
	Class-3	0.50	0.92	0.94	0.50	0.86	0.88	0.50	0.85	0.89
	Class-4	0.50	0.91	0.96	0.50	0.87	0.91	0.50	0.86	0.90

to 1e-05, the result suggested the major improvement. This is due to setting the learning rate value very small, which helps a model to learn the minute features of images. Here, in an Imbalanced dataset, the AUC curve shows better results because the model detects only negative class were in up sample dataset, the value of the AUC curve is also good because samples are equally distributed. In both (imbalanced and up) datasets AUC curve shows better results. In Imbalanced dataset results, the sample distribution is unequal as a negative class has maximum images but in the Up sampled dataset, the sample distribution is equal, so the model predicts an accurate result.

Hyper-parameters Setting The parameters of the CNN which need to be set by the user prior to the filter learning are called hyper-parameter. Hyper-parameters are the variables related to the structure of the network (e.g. number of layers and number units in each layer) training (e.g. learning rate). These parameters are adjusted before training (before optimizing the weights and bias). In order to set the values of other hyper-parameter, we have adopted good practices from literature. For the learning rate, we have considered three different values while two optimizers are considered as shown in Table 8. Table 8 shows five architecture as mentioned above are trained with different hyper-parameters. After completion of training, all architectures are ensembled. Table 9 shows the accuracy, recall, precision, specificity, and F1-score of SGD and Adam optimizer with different learning rates. The learning rate is decreased from 0.01 to 1e-05. The performance of the model increases with a decrease in the learning rate. Also, it can be noted that most of the time SGD has better performances than Adam.

V. CONCLUSION

Diabetes is one of the fast-growing diseases in recent times. According to various surveys, a patient having diabetes has around 30% chances to get Diabetic Retinopathy (DR). DR has different stages from mild to severe and then PDR (Proliferative Diabetic Retinopathy). In the later stages of the diseases, it leads to floaters, blurred vision and finally can lead to blindness if it is not detected in the early stages. Manual diagnosis of these images requires highly trained experts and is time-consuming and difficult. Computer vision-based techniques for automatic detection of DR and its different stages have been proposed in the literature. In this paper, we focused to classify all the stages of DR, especially the early stages, which is the major shortcoming of existing models. We proposed a CNN ensemble-based framework to detect and classify the DR's different stages in color fundus images. We used the largest publicly available dataset of fundus images (Kaggle dataset) to train and evaluate our model. The results show that the proposed ensemble model performs better than other state-of-the-art methods and is also able to detect all the stages of DR. In future in order to further increase the accuracy of early-stage, we plan to train specific models for specific stages and then ensemble the outcome in order to increase the accuracy of early stages.

ACKNOWLEDGMENT

The authors would like to thank Nvidia Corporation for donating us a Telsa K-40 GPU for our project.

REFERENCES

[1] W. R. Memon, B. Lal, and A. A. Sahto, "Diabetic retinopathy," *The Prof. Med. J.*, vol. 24, no. 2, pp. 234–238, 2017.

[2] S. Haneda and H. Yamashita, "International clinical diabetic retinopathy disease severity scale," *Nihon Rinsho. Jpn. J. Clin. Med.*, vol. 68, pp. 228–235, Nov. 2010.

[3] S. Jan, I. Ahmad, S. Karim, Z. Hussain, M. Rehman, and M. A. Shah, "Status of diabetic retinopathy and its presentation patterns in diabetics at ophthalomogy clinics," *J. Postgraduate Med. Inst. (Peshawar-Pakistan)*, vol. 32, no. 1, pp. 24–27, Mar. 2018.

[4] 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.

[5] M. D. Abràmoff, P. T. Lavin, M. Birch, N. Shah, and J. C. Folk, "Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices," *Npj Digit. Med.*, vol. 1, no. 1, p. 39, Aug. 2018.

[6] L. Seoud, T. Hurtut, J. Chelbi, F. Cheriet, and J. M. P. Langlois, "Red lesion detection using dynamic shape features for diabetic retinopathy screening," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1116–1126, Apr. 2016.

[7] R. A. Welikala, M. M. Fraz, J. Dehmeshki, A. Hoppe, V. Tah, S. Mann, and T. H. Williamson, "Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy," *Computerized Med. Imag. Graph.*, vol. 43, pp. 64–77, Jul. 2015.

[8] R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," *Ophthalmology*, vol. 124, no. 7, pp. 962–969, 2017.

[9] C. González-Gonzalo, V. Sánchez-Gutiérrez, P. Hernández-Martínez, I. Contreras, Y. T. Lechanteur, A. Domanian, B. van Ginneken, and C. I. Sánchez, "Evaluation of a deep learning system for the joint automated detection of diabetic retinopathy and age-related macular degeneration," 2019, *arXiv:1903.09555*. [Online]. Available: https://arxiv.org/abs/1903.09555

[10] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, "Deep image mining for diabetic retinopathy screening," *Med. Image Anal.*, vol. 39, pp. 178–193, Jul. 2017.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, vol. 2, Sep. 1999, pp. 1150–1157.

[13] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[14] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *J. Neurophysiol.*, vol. 58, no. 6, pp. 1233–1258, Dec. 1987.

[15] R. Ghosh, K. Ghosh, and S. Maitra, "Automatic detection and classification of diabetic retinopathy stages using CNN," in *Proc. 4th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2017, pp. 550–554.

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

[17] N. Nida, A. Irtaza, A. Javed, M. H. Yousaf, and M. T. Mahmood, "Melanoma lesion detection and segmentation using deep region based convolutional neural network and fuzzy c-means clustering," *Int. J. Med. Informat.*, vol. 124, pp. 37–48, Apr. 2019.

[18] W. M. Gondal, J. M. Köhler, R. Grzeszick, G. A. Fink, and M. Hirsch, "Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2069–2073.

[19] Z. Wang, Y. Yin, J. Shi, W. Fang, H. Li, and X. Wang, "Zoom-in-net: Deep mining lesions for diabetic retinopathy detection," in *Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Berlin, Germany: Springer, 2017, pp. 267–275.

[20] T. Chandrakumar and R. Kathirvel, "Classifying diabetic retinopathy using deep learning architecture," *Int. J. Eng. Res. Technol.*, vol. 5, no. 6, pp. 19–24, Jun. 2016.

[21] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional neural networks for diabetic retinopathy," *Procedia Comput. Sci.*, vol. 90, pp. 200–205, Jan. 2016.

[22] Y. Yang, T. Li, W. Li, H. Wu, W. Fan, and W. Zhang, "Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2017, pp. 533–540.

[23] G. García, J. Gallardo, A. Mauricio, J. López, and C. Del Carpio, "Detection of diabetic retinopathy based on a convolutional neural network using retinal fundus images," in *Proc. Int. Conf. Artif. Neural Netw.* New York, NY, USA: Springer, 2017, pp. 635–642.

[24] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," KTH Roy. Inst. Technol., Stockholm, Sweden, Tech. Rep., 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[26] 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.*, Jun. 2016, pp. 2818–2826.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1251–1258.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[29] T. Dozat, "Incorporating nesterov momentum into adam," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR) Workshop Track*. San Juan, Puerto Rico: Caribe Hilton, May 2016, pp. 1–4.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[31] J. Akosa, "Predictive accuracy: A misleading performance measure for highly imbalanced data," in *Proc. SAS Global Forum*, Apr. 2017, pp. 2–5.

[32] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[33] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[34] A. Dalyac, M. Shanahan, and J. Kelly, *Tackling Class Imbalance With Deep Convolutional Neural Networks*. London, U.K.: Imperial College, Sep. 2014, pp. 30–35.

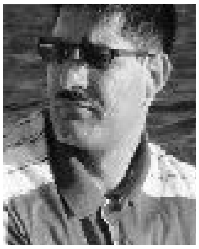
SEHRISH QUMMAR is currently pursuing the M.S. degree with COMSATS University Islamabad, Abbottabad Campus, and also doing research in Huanghai University, Zhumadian, China. Her area of research is medical image diagnosis using deep learning. Her research interests include image processing, machine learning, and data mining.

FIAZ GUL KHAN received the master's (specialization) and Ph.D. degrees from Politecnico di Torino, Italy, in 2013. He is currently an Assistant Professor with the Computer Science Department, COMSATS University Islamabad, Abbottabad Campus, Pakistan. His research interests include the field of concurrent computing, machine learning, artificial intelligence, and GPU computing.

SAJID SHAH received the M.S. and Ph.D. degrees from Politecnico di Torino, Italy. He is currently an Assistant Professor with COMSATS University Islamabad, Abbottabad campus. His research interests include data mining, text mining, machine learning, bioinformatics, and image processing.

AHMAD KHAN received the Ph.D. degree from the National University of Computer and Emerging Sciences (FAST- NU), Islamabad, Pakistan, in 2015. He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad (CUI), Abbottabad Campus. His research interests include computer vision, machine learning, and evolutionary algorithms.

SHAHABODDIN SHAMSHIRBAND received the M.Sc. degree in artificial intelligence from Iran and the Ph.D. degree in computer science from the University of Malaya (UM), Malaysia, in 2014. He was an Adjunct Assistant Professor with the Department of Computer Science, Iran University of Science and Technology. He was also a Senior Lecturer with UM, Malaysia, and with Islamic Azad University, Iran. He participated in many research programs within the Center of Big Data Analysis, IUST, and IAU. He has been associated with young researchers and elite club, since 2009. He supervised or co-supervised undergraduate and postgraduate students (master's and Ph.D.) by research and training. He has also authored or coauthored articles published in IF journals and attended to high-rank A and B conferences. He is a professional member of the ACM. He is an Associate Editor, a Guest Editor, and a Reviewer of high-quality journals and conferences.



ZIA UR REHMAN received the Ph.D. degree from Curtin University, Perth, WA, Australia, in 2014. He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad (CUI), Abbottabad Campus. His research interests include image processing and computational intelligence.

IFTIKHAR AHMED KHAN received the MCS degree in computer sciences from COMSATS, Abbottabad, in 2004, and the Ph.D. degree in human–computer interaction from Brunel University, West London, U.K., in 2009. He is currently an Assistant Professor with COMSATS University Islamabad, Abbottabad Campus. His research interests include HCI, affective computing, image processing, and architectural aspects of distributed systems. He has over 30 publications in international reputed journals and conferences.

WAQAS JADOON received the Ph.D. degree in computer science from Sichuan University, China, in 2014. He is currently an Assistant Professor with COMSATS University Islamabad, Abbottabad Campus, Pakistan, and is an approved Ph.D. Supervisor from Higher Education Commission (HEC), Pakistan. His research interests include pattern recognition, image processing, and theory and applications of machine learning.