Biomedical Signal Processing and Control 62 (2020) 102115

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

A lightweight CNN for Diabetic Retinopathy classification from fundus images

Gayathri S. *, Varun P. Gopi, P. Palanisamy

National Institute of Technology Tiruchirapalli, Tamilnadu, India

ARTICLE INFO

Keywords:
DR binary and multi class classification
Retinal fundus images
CNN feature extraction
Classifiers
10-fold cross-validation

ABSTRACT

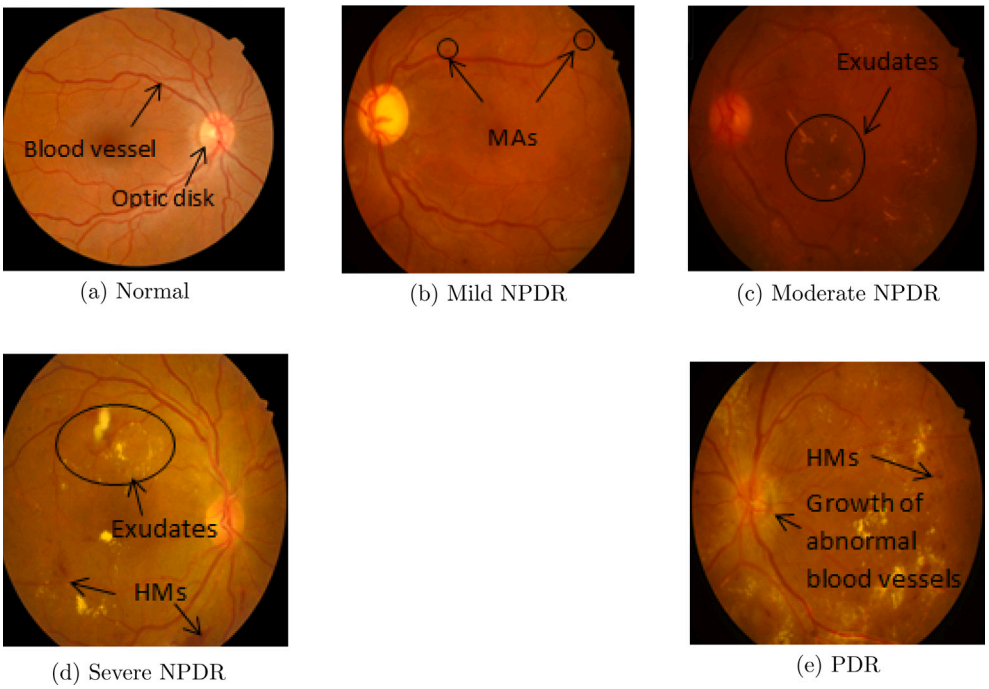
Diabetic Retinopathy (DR) is a complication of diabetes mellitus that damages blood vessel networks in the retina. This is a serious vision-threatening issue in most diabetic subjects. The DR diagnosis by color fundus images involves skilled clinicians to recognize the presence of lesions in the image that can be used to detect the disease properly, making it a time-consuming process. Effective automated detection of DR is a challenging task. The feature extraction plays an excellent role in effective automated disease detection. Convolutional Neural Networks (CNN) have superior image classification efficiency in the present scenario compared to earlier handcrafted feature-based image classification techniques. This work presents a novel CNN model to extract features from retinal fundus images for better classification performance. The CNN output features are used as input for different machine learning classifiers in the suggested system. The model is evaluated through various classifiers (Support Vector Machine, AdaBoost, Naive Bayes, Random Forest, and J48) by using images from generic IDRiD, MESSIDOR, and KAGGLE datasets. The efficacy of the classifier is evaluated by comparing the specificity, precision, recall, False Positive Rate (FPR), Kappa-score, and accuracy values for each classifier. The evaluation results indicate that the proposed feature extraction technique along with the J48 classifier outperforms all the other classifiers for MESSIDOR, IDRiD, and KAGGLE datasets with an average accuracy of 99.89% for binary classification and 99.59% for multiclass classification. Furthermore, for the J48 classifier, the average Kappa-score (K-score) is 0.994 for binary classification and 0.994 for multi-class classification.

1. Introduction

Diabetic Retinopathy (DR) occurs predominantly in people with diabetes Mellitus history. The high blood glucose level triggers the leakage of blood and other fluids from the blood vessels in the retina. It is a cause for loss of vision in diabetic subjects. The main stages involved in DR are Non-Proliferative DR (NPDR) and Proliferative DR (PDR). The major lesions that we are considering for grading [1] are MicroAneurysms (MA), Blood vessels, Haemorrhage, and Exudates. The DR stage with any of these lesions is considered as NPDR and the advanced phase with neovascularization is called PDR. The retina with the presence of these lesions is depicted in Fig. 1. MA is a small swelling form in the wall of tiny blood vessels. In patients with DR, these minute swelling MAs are considered as the earliest visible symptom of DR. They appear in the retina as small red dots [2]. As the disease progress, its size increases. Retinal haemorrhage is another disorder in the retina which is caused by DR. The other reasons for haemorrhage (HM) may be hypertension and retinal vein occlusion. If they are very small, then it resembles MAs. When there are lipid and protein residues in the leaked blood from the damaged capillaries, it forms yellow flicks in the retina called exudates. Normally the DR is graded into different categories: no DR, mild NPDR, moderate NPDR, severe NPDR, and PDR. Severe NDPR is diagnosed based on the "4-2-1" rule. In this, Intra Retinal Microvascular Abnormalities (IRMA) is one of the important factors. IRMA is the abnormal branching or expansion of retinal blood vessels. The rules in grading DR are mentioned in Table 1. DR in its serious phase is hard to heal. It will result in complete vision loss. Reducing its prevalence worldwide is very crucial. Several approaches have been implemented to detect the symptoms of DR. But the challenging task is the methods of feature extraction. Making the feature extraction for DR classification as precise as possible and also reducing computational costs and time is very important. By considering all these facts, the proposed work is trying to present a simple CNN for feature extraction and combining it with Machine Learning (ML) Classifiers for better classification.

* Corresponding author.
E-mail address: gsgayathriunnithan@gmail.com (Gayathri S.).

https://doi.org/10.1016/j.bspc.2020.102115
Received 4 November 2019; Received in revised form 7 July 2020; Accepted 1 August 2020
Available online 11 August 2020



(a) Normal (b) Mild NPDR (c) Moderate NPDR

(d) Severe NPDR (e) PDR

Fig. 1. Different stages of DR in fundus images.

Table 1
Grading of diabetic retinopathy.

Lesion detection	Grading level
Not observed any abnormalities	No DR
Only observed MAs	Mild NPDR
A small number of MAs with or without cotton-wool spots, venous beading, or presence of IRMA which is less than the 4-2-1 guidelines ≥ mild but ≤ severe	Moderate NPDR
Presence of any of the 4-2-1 guidelines • All the four quadrants contains dot blot hemorrhage • venous beading in atleast two quadrants • Prominent IRMA in atleast one quadrant	Severe NPDR
Presence of either Preretinal hemorrhage or neovascularization	PDR

Automated grading has two main issues. The first one is achieving a desirable precision offset and specificity. In this work we are considering five class grading, again it is significantly harder to achieve the same. Second is overfitting in the neural network. These two are particularly related to the training algorithm in the system. Convolutional Neural networks (CNN) are state-of-the-art deep learning techniques that have led to many breakthroughs in the identification and detection of objects in many areas including medical imaging. The task here is to solve the issues regarding the feature extraction stage with a well-defined CNN architecture. The availability of a large database is a crucial stage in deep learning algorithms. To manage this issue, machine learning classifiers can be utilized for classification and CNN for extracting minute features from fundus images for accurate disease classification. Thus handcrafted feature extraction methods can be replaced with efficient and accurate automated methods.

2. Related works

The automatic severity level measurement of DR using Artificial Neural Network (ANN) is explained in [3]. Lesions such as MAs, Blood vessels, hemorrhage, and exudates are derived from the retinal fundus image. The lesions then fed into the multilayer feed-forward neural network to classify DR into mild, moderate, and severe. The method explained in [4] deals with the two-stage CNN for detecting the abnormal lesions in the fundus image. Here, the lesion type and its location in the image was pointed out and graded the DR. In [5], features like area, perimeter, and count from the four lesions extracted and the classification is performed using ANN and graded DR into mild, moderate or severe. The automatic grading in [6] is obtained from the results of a validated red lesion detection method. Assessment is performed on a public database by leave one out validation method. They tried to prove the feasibility of automatic DR screening. Diabetic or non-diabetic classification of the retinal fundus is carried out in [7]. The image was divided into four sub-images. After extracting the features, they applied Haar wavelet transformation. To select a better feature, they used Principal Component Analysis (PCA). Then backpropagation neural network and one rule classifier were used for the classification process. The work in [8] is based on the screening of fundus images with different illumination and fields of view. They graded the DR using different classifiers such as k-Nearest Neighbor (k-NN), AdaBoost, Support Vector Machines (SVM), and Gaussian mixture model (GMM). According to their performance analysis, GMM and k-NN performed well than others. The main task was to implement the feature ranking method to reduce the number of features for lesion classification. A combined feature extraction method using ADTCWT(Anisotropic Dual-Tree Complex Wavelet Transform) and Haralick is introduced in [9]. Then the selected features are fed into the classifier for DR grading. A three-stage (preprocessing, image analysis, and classification) model for grading was proposed in [10]. They used a hybrid classifier (a combination of GMM and m-medoids classifier) to improve the classification efficiency. A genetic algorithm was used to learn the weights of the classifier. The results showed that the system detected all the NPDR lesions and graded the severity at some accuracy.

In [11], A CNN method is proposed for the diagnosis and accurate severity classification of DR from fundus images. It obtained an accuracy of 75% on 5000 validation images. In [12], they extracted hard exudates area, blood vessels area, texture, entropies, and bifurcation points. The work used a combination of texture and morphological changes for classification. The parameter(σ) of the Probabilistic Neural Network (PNN) classifier is tuned using genetic algorithm and particle

swarm optimization. The five class grading of DR was also one of the highlights of this work. Authors of [13] explained how to handle the blurred retinal images for the detection of DR. To enhance the system performance, they used a regularized filter deblurring algorithm. The areas of the blood vessel, MAs, exudates are computed and fed into the ANN classifier. In [14], modified Alexnet architecture is employed to categorize the input fundus images. A high level of accuracy CNN with the application of suitable pooling was proposed to classify DR fundus images into the severity of the disease. In [15], a siamese neural network architecture named binocular neural network is proposed for DR classification. They used the Inception-v3 CNN model in the binocular structure. Local features of retinal images are extracted using Local Binary Pattern (LBP) in [16]. Then it is evaluated across ANN, Random Forest, and SVM for the detection task. The sparse coding technique with linear SVM for retinal image classification is proposed in [17]. They used the BoVW technique for feature extraction. According to their evaluation, a dictionary size of 100 achieves better sensitivity and specificity. A CNN using fractional max pooling is developed in [18] for DR classification. They used the network for feature extraction and SVM with Teaching Learning Based Optimization (TLBO) for binary as well as multiclass classification. In [19], Inception-v3 architecture is used for DR grading and their studies included the different size of the input to the network that can be able to improve the classification ability. A modified inception-v3 network (named as inception@4)for DR grading is demonstrated in [20] and the system is evaluated using their database which achieves an accuracy of 88.72%. In [21], a multichannel CNN is proposed for DR classification. The network is evaluated across EYEPACS images and obtained an accuracy of 97.08%.

From the review of literature, it can be seen that the multi-class classification using deep neural networks use heavy and large networks (pre-trained networks and pre-trained networks with transfer learning), which may not be suitable for real-time deployment. Thus, it is necessary to develop methods that can boost accuracy while keeping the network size as small as possible.

3. Materials and methods

The proposed work presents a CNN architecture to extract the features from retinal fundus images for binary and multi-class classification of DR. It is necessary to develop automated methods to improve the accuracy of diagnosis and classify the subjects into various stages of DR. The proposed work presents a method that can minimize the computational complexity and provide better classification performance. Through the layer by layer stacking of convolution, pooling, non-linear activation function mapping, CNN extracts symbolic information from the input data and makes the layer by layer abstraction possible. This process is termed as feed-forward operation. By calculating the error between the predicted value and true value, the detected error is fed back from the last layer by the backpropagation algorithm. The extracted features are then fed into the classifier for binary as well as multi-class classification as shown in Fig. 2. The proposed method is implemented in the Keras framework in python using Tensorflow.

3.1. Feature extraction

The feature extraction is one of the predominant stages in an automated classification system, as perfect and minute features are required for precise prediction. CNN has the ability to extract minute features from the image that are sufficient for good classification. To get the features from the CNN model, it is required to train the CNN network up to the last dense layer with respect to the target variable. To improve the latency of the automated system, it is better to use CNN for feature extraction, followed by a simple classifier.

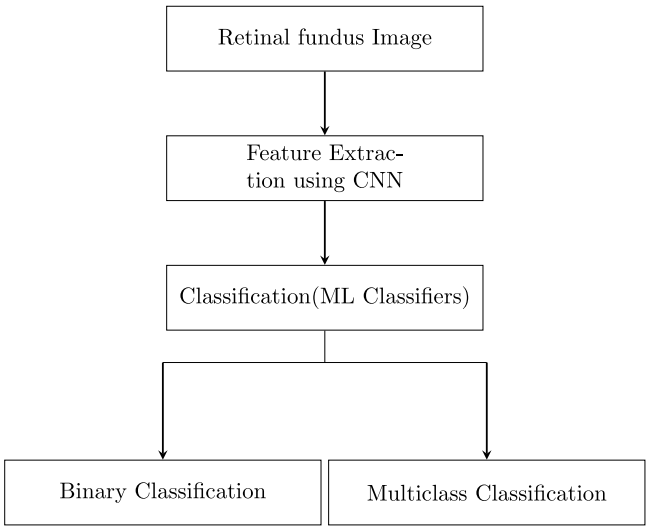


Fig. 2. Proposed work.

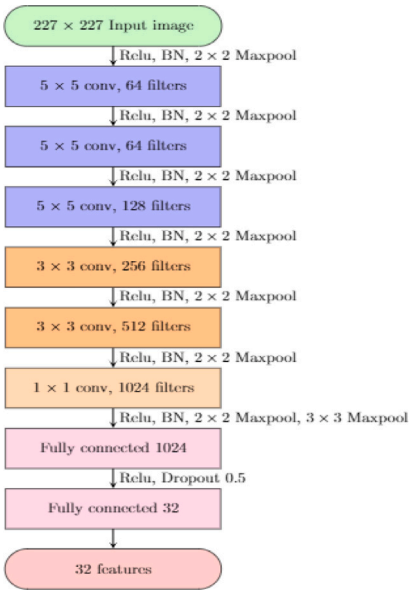


Fig. 3. Proposed CNN Architecture.

3.1.1. Proposed network architecture

The CNN architecture for feature extraction is demonstrated in Fig. 3 and the learnable parameters for each layer are listed in Table 2. In the designed network six convolutional layers and two fully connected layers are used. From the last fully connected layer 32 relevant features are obtained and used for classification.

It is essential to resize the image appropriately before feeding them into CNN. In the proposed work, the input fundus image is resized to a dimension of 227 × 227 × 3 pixels corresponding to the breadth, height, and the three color channels representing the depth of the input fundus image. The function of the CNN is to reduce the input images to a form that is simpler to process without losing critical features to obtain a good prediction. The function of each layer in the CNN [22] is narrated below.

- **Convolutional layer** calculates the output of neurons as a dot product of a small portion of the image with their corresponding weights. Along the length and breadth, this process is repeated. These layers use the parameter sharing system to regulate the number of parameters.

Table 2
Tensor sizes & number of parameters in the proposed architecture.

Layer name	Tensor size	No: of parameters
Input image	227 ×227 ×3	0
Conv-1	227 ×227 ×64	4864
Max pool-1	114 ×114 ×64	0
Conv-2	114 ×114 ×64	1,02,464
Max pool-2	57 ×57 ×64	0
Conv-3	57 ×57 ×128	2,04,928
Max pool-3	29 ×29 ×128	0
Conv-4	29 ×29 ×256	2,95,168
Max pool-4	15 ×15 ×256	0
Conv-5	15 ×15 ×512	11,80,160
Max pool-5	8 ×8 ×512	0
Conv-6	8 ×8 ×1024	5,25,312
Max pool-6	4 ×4 ×1024	0
Maxpool-7	2 ×2 ×1024	0
FC-1	1024 ×1	41,95,328
FC-2	32 ×1	32,800

Total learnable parameters: 65,41,024.

- In **Rectified Linear Unit (ReLU)** layer, simplest non-linear activation function is employed here. This layer replaces all the negative activations with 0 by introducing non-linearity to the system and by applying the function $f(k) = max(0, k)$. where, k is the neuron input.
- **Pooling layer** reduces the number of inputs to the next layer of feature extraction, thus allowing us to have many more different feature maps. Max pooling is a method of discretization based on samples. The goal is to down-sample an input representation that reduces its dimensionality and enables assumptions about features contained in binned sub-regions to be made. Max pooling is performed by applying a max filter to (generally) the initial representation's non-overlapping subregions.
- **Fully connected layer** neurons, as seen in regular Neural Networks, have complete links to all prior layer activations. Therefore, their activations can be calculated with a matrix multiplication accompanied by a bias offset.

The parameters are trained using a backpropagation algorithm taking cross-entropy as the loss function. In addition to this, Batch normalization [23,24] dropout strategy, L_1, L_2 regularization are included to avoid overfitting. The total number of learnable parameters in the proposed network architecture is 65,41,024.

3.2. Classifiers

3.2.1. Support vector machine (SVM)

Support vector machines [25] are supervised learning methods with associated learning algorithms. If the vectors are non linearly separable in a space, then the SVM helps to make it linearly separable in a higher-dimensional space. In this work, the SVM is implemented using a Radial Basis Function (RBF) [26] kernel with gamma value selected as the reciprocal of the product of the total number of features and the variance. The algorithm [26] for training the SVM is explained in Algorithm 1.

3.2.2. Random forest

Random Forest [27] is an ensemble model classifier, in which a group of trees is developed together with each has an independent random vector. i.e., the Kth tree generates a random vector Φ_K which is independent from previously generated random vectors $(\Phi_1, \Phi_2, \ldots, \Phi_{K-1})$ but have same distribution [28]. In this work, the number of trees used in the classifier is 100 [28]. The steps in the pseudo-code generation are as follows:

1. Select the features randomly from total features.

Algorithm 1 Pseudo code for SVM

Require: S and t load with labeled data for training
consider initially $\eta = 0$
START:

1. γ assume random value initially
2. **repeat**
3. **Do for:** $\{s_i, t_i\}, \{s_j, t_j\}$
4. find η_i and η_j and optimize
5. **end for**
6. **until** η and γ become unchanged

Ensure: Retain support vectors where $\eta_i > 0$

2. From the selected features find out the mother node for the tree using best split method
3. Again use the best split method to split the others into branches
4. Repeat steps 1–3 until form a root node with target as the leaf nodes
5. construct the forest by iteration (doing steps 1–4) for n times to create n trees

3.2.3. Multi layer perceptron (MLP)

MLP [29] is a multi-layer feed-forward network that maps inputs to outputs in a nonlinear manner. The MLP base structure contains an input layer, a hidden layer, and an output layer, with each node fully connected to the nodes in the next layer with appropriate weights, which is schematically represented in Fig. 4. MLP uses a backpropagation method for training, there might be a non-linear activation function that is not seen in other neural networks. In MLP, the sigmoid function is generally used, and it is described in Eq. (1). In the proposed work only one hidden layer is used by considering the advantages of single hidden layer MLP which is mentioned in [29].

$$y_i(s_i) = (1 + e^{-s_i})^{-1} \tag{1}$$

where, y_i depicts the ith node output and the weighted sum of the input synapses is denoted by s_i. In back propagation algorithm [30], the motive is to reduce the error propagated in the network by adjusting the weights at each node. The error $e_j(n)$ at the jth output node in the nth data point can be calculated using the actual output value $a_j(n)$ and predicted output value $y_j(n)$ as in Eq. (2).

$$e_j(n) = a_j(n) - y_j(n) \tag{2}$$

To minimize the error in the entire output, the corrections in weights at each node are done by Eq. (3) and the new weight for each node can be acquired from Eq. (4).

$$\sigma(n) = \frac{1}{2}\Sigma_j[e_j^2(n)] \tag{3}$$

$$\Delta W_{ji}(n) = -\alpha \frac{\partial \sigma(n)}{\partial_j(n)} y_i(n) \tag{4}$$

where, α is the learning rate, $y_i(n)$ is the previous node output. The iterative process continues until the error becomes unchangeable.

3.2.4. J48 classifier

This is the open-source java implementation of C4.5 [31] Decision tree. This is a decision tree algorithm mainly designed for data mining. In this, the information gain ratio is evaluated to select each node test feature. This procedure is termed as feature selection. While operating, the attribute with the largest information gain will be selected as the test feature for the present node. Let us assume that F is a set of input feature vectors given to the classifier which contains $F_1, F_2, \ldots, F_n$ instances. Suppose there are t distinct values for t distinct classes

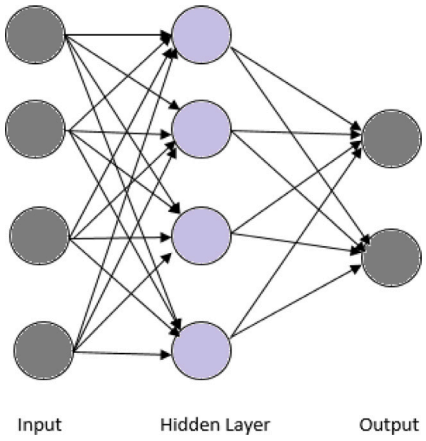


Fig. 4. Schematic of Multi Layer Perceptron.

$C_i(where, i = 1, 2, \ldots, n)$. Then the gain ratio G_A of sub-attribute A in each attribute can be calculated using Eq. (5).

$$G_A = G(A)/S_A(F) \tag{5}$$

Where, $G(A)$ is the information gain of attribute A which can be obtained by taking the difference between total information $I(D)$ and the attribute information $I_A(D)$ as shown in Eq. (6).

$$G(A) = I(D) - I_A(D) \tag{6}$$

If P_i is the distinct class probability, then $I(D)$ and $I_A(D)$ can be calculated using Eqs. (7) and (8) respectively.

$$I(D) = \Sigma P_i log_2(P_i) \tag{7}$$

$$I_A(D) = -\Sigma \frac{|F_J|}{|F|} I(F_j) \tag{8}$$

The split information value $S_A(F)$ of attribute can be formulates as:

$$S_A(F) = -\Sigma \frac{|F_j|}{|F|} log\left(\frac{|F_j|}{|F|}\right) \tag{9}$$

Actually, the fraction $\frac{|F_j|}{|F|}$ acts as the jth partition weight. By utilizing all these equations the C4.5 decision tree can be developed which forms appropriate conditions that can be used for classification. Then during testing, it will classify the input feature vector according to the conditions.

3.3. Performance analysis

The performance of the classifier is evaluated based on K-fold evaluation [32]. In this assessment methodology, the entire dataset available is divided into K-sub parts during the training itself (K=1,2,3, . . .). Then each subsection is treated as a validation set for each iteration. The general steps in K-fold validation are as follows:

1. Randomly shuffle the dataset
2. Data will split into K-sub groups(If K=10, then split the data into 10 groups)
3. The evaluation process is performed for each groups
   - Use one group as a test set
   - Use the remaining groups as the training dataset
   - Train the classifier with this dataset and evaluate the model with the test data
   - Retain the evaluation score and repeat the steps by selecting other group.

Table 3
Confusion matrix for binary classification.

		Predicted diagnosis	
		DR	NO DR
Actual diagnosis	DR	*J*	*K*
	NO DR	*M*	*N*

Table 4
Confusion matrix for severity grading.

		Predicted diagnosis			
		Normal	Mild	Moderate	Severe
Actual diagnosis	Normal	A_{11}	A_{12}	A_{13}	A_{14}
	Mild	A_{21}	A_{22}	A_{23}	A_{24}
	Moderate	A_{31}	A_{32}	A_{33}	A_{34}
	Severe	A_{41}	A_{42}	A_{43}	A_{44}

4. Summarize the model efficiency using the evaluation scores.

The findings are stored in the form of a confusion matrix [33]. The confusion matrix structure that depicts a binary classifier's characteristics is shown in Table 3. In that matrix, J, K, M, and N represents the number of True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) respectively. TP and TN give the results of correctly classified data while FP and FN give the incorrectly classified details. Similarly the confusion matrix for multiclass classification is demonstrated in Table 4. The number of TPs, FPs, FNs, TNs in multiclass classification can be easily acquired through formulas in Eq. (10) for each actual class i by taking p predicted classes. Using these values, we can calculate the accuracy, F-score, specificity, precision, and recall of the classifier to examine system efficiency.

$$\begin{aligned} Number\ of\ TPs, \quad & J_i = A_{ii} \\ Number\ of\ FNs, \quad & K_i = \Sigma_{j=1}^{p} A_{ij} - J_i \\ Number\ of\ FPs, \quad & M_i = \Sigma_{j=1}^{p} A_{ji} - J_i \\ Number\ of\ TNs, \quad & N_i = \Sigma_{j=1}^{p} \Sigma_{k=1}^{n} A_{ik} - J_i - M_i - K_i \end{aligned} \tag{10}$$

Accuracy defines the overall power of the system. For binary class and multi class, it can be obtained from the confusion matrix using the formula in Eqs. (11) and (12) respectively. For binary classification,

$$Accuracy = \frac{J + N}{J + K + M + N} \tag{11}$$

For multi class classification, each class accuracy:

$$Accuracy_i = \frac{J_i}{J_i + K_i + M_i + N_i} \tag{12}$$

False Positive Rate(FPR) gives the rate of incorrect positive predictions. The best FPR rate for a good classifier is 0.0.

For binary classification,

$$FPR = \frac{M}{M + N} \tag{13}$$

For multi class classification, each class FPR:

$$FPR_i = \frac{M_i}{M_i + N_i} \tag{14}$$

Precision gives the positive prediction value. This value provides the information on how efficiently our system avoids FPs. It can be measured as,

For binary classification,

$$Precision = \frac{J}{J + M} \tag{15}$$

For multi class classification, each class Precision:

$$Precision_i = \frac{J_i}{J_i + M_i} \tag{16}$$

Table 5
Confusion matrix for the evaluation of each classifier using IDRiD Database for binary classification.

	DR	No DR
DR	269	10
No DR	3	131

(a) SVM

	DR	No DR
DR	279	0
No DR	5	129

(b) Random forest

	DR	No DR
DR	278	1
No DR	4	130

(c) MLP

	DR	No DR
DR	278	1
No DR	0	134

(d) J48

Table 6
Confusion matrix for the evaluation of each classifier using MESSIDOR Database for binary classification.

	DR	No DR
DR	650	4
No DR	525	21

(a) SVM

	DR	No DR
DR	653	1
No DR	1	545

(b) Random forest

	DR	No DR
DR	647	7
No DR	6	540

(c) MLP

	DR	No DR
DR	654	0
No DR	1	545

(d) J48

Table 7
Confusion matrix for the evaluation of each classifier using KAGGLE database for binary classification.

	DR	No DR
DR	9221	95
No DR	90	25720

(a) SVM

	DR	No DR
DR	9315	1
No DR	0	25810

(b) Random forest

	DR	No DR
DR	9293	23
No DR	49	25761

(c) MLP

	DR	No DR
DR	9315	1
No DR	0	25810

(d) J48

Recall, also called as sensitivity gives the information about how efficiently the model reduces FNs. This can be calculated as,

For binary classification,

$$Recall = \frac{J}{J+K} \tag{17}$$

For multi class classification, each class recall:

$$Recall_i = \frac{J_i}{J_i + K_i} \tag{18}$$

F1-score determines the model's accuracy. This score gives the harmonic mean of precision and recall.

For binary classification,

$$F1 - score = \frac{2J}{2J + M + K} \tag{19}$$

For multi class classification, each class F1-score:

$$(F1 - score)_i = \frac{2J_i}{2J_i + M_i + K_i} \tag{20}$$

High values of these measures except False positive rate indicates the good performance of the classifier.

Specificity quantifies how efficiently the false positives(FPs) are reduced in a model. The sum of FPR and Specificity gives 1.

For binary classification,

$$Specificity = \frac{N}{M+N} \tag{21}$$

Table 8
Detailed efficiency measures of different classifiers using IDRiD Database for binary classification.

Classifier	FP rate	Specificity	Precision	Recall	F1 Score	Class
SVM	0.022	0.978	0.989	0.964	0.976	DR
	0.036	0.964	0.929	0.978	0.953	Normal
Random forest	0.037	0.963	0.982	1.00	0.991	DR
	0.00	1.00	1.00	0.963	0.981	Normal
MLP	0.030	0.97	0.986	0.996	0.991	DR
	0.004	0.996	0.992	0.970	0.981	Normal
J48	**0.00**	**1.00**	**1.00**	**0.996**	**0.998**	**DR**
	0.004	**0.996**	**0.993**	**1.00**	**0.996**	**Normal**

For multi class classification, each class Specificity:

$$Specificity_i = \frac{N_i}{M_i + N_i} \tag{22}$$

The overall accuracy, FPR, Precision, Recall, F1-score and specificity can be achieved in multiclass classification by taking the mean values acquired for each class using Eqs. (12), (14), (16), (18), (20) and (22) respectively.

To summarize the performance of the system, the weighted average of each class performance measures are required. If P_1 and P_2 denotes the performance measures obtained for class 1 (C_1) and class 2 (C_2) respectively then the weighted average of performance measure W_{PM} can be calculated using the following equation:

$$W_{PM} = \frac{(P_1 * |C_1|) + (P_2 * |C_2|)}{|C_1| + |C_2|} \tag{23}$$

The Kappa statistic (K-score) is a quality metric of the classifier that assesses the interrater reliability. It is a measure relating an Observed Accuracy (A_O) to an Expected Accuracy (A_E). It can be computed as:

$$K - score = \frac{(A_O - A_E)}{(1 - A_E)} \tag{24}$$

4. Experimental results and discussions

The experimental analysis of the proposed method is conducted using a PC with Nvidia GeForce, RTX2080 11 GB GPU. The entire network is trained using backpropagation by automatic differentiation algorithm using stochastic mini-batch gradient descent to update the model parameters. The best learning rate has been found experimentally to be 0.003 for a mini-batch of 64. A momentum factor of 0.9 is used to make the training less noisy and converge faster to the objective. Initially, there was heavy overfitting of the data. Using a dropout factor of 0.5 and L_2 regularization improved validation accuracy. For others, the default parameter settings are used. The results obtained and discussions on the work are described in this section.

4.1. Datasets

The performance of the proposed system is assessed individually using IDRiD, MESSIDOR, KAGGLE databases that have 413, 1200, and 35126 retinal fundus images respectively. For binary classification, 279 DR and 134 normal images are used from IDRiD, 654 DR and 546 images are used from MESSIDOR, 9316 DR and 25810 normal images from KAGGLE is used. For multi-class classification: 134 normal, 20 mild NPDR, 136 moderate NPDR, 74 severe NPDR and 49 PDR from IDRiD database; 546 normal, 153 mild DR, 247 moderate DR and 254 severe DR images from MESSIDOR database; 25810 normal, 2443 mild NPDR, 5292 moderate NPDR, 873 severe NPDR, and 708 PDR images from KAGGLE database is used.

Table 9
Detailed efficiency measures of different classifiers using MESSIDOR Database for binary classification.

Classifier	FP Rate	Specificity	Precision	Recall	F1 Score	Class
SVM	0.02	0.98	0.983	0.976	0.979	DR
	0.038	0.962	0.969	0.994	0.981	Normal
Random forest	0.002	0.998	0.998	0.998	0.998	DR
	0.002	0.998	0.998	0.998	0.998	Normal
MLP	0.011	0.989	0.991	0.989	0.990	DR
	0.011	0.989	0.987	0.989	0.988	Normal
J48	**0.002**	**0.998**	**0.998**	**1.00**	**0.999**	**DR**
	0.00	**1.00**	**1.00**	**0.998**	**0.999**	**Normal**

Table 10
Detailed efficiency measures of different classifiers using KAGGLE Database for binary classification.

Classifier	FP Rate	Specificity	Precision	Recall	F1 Score	Class
SVM	0.003	0.997	0.990	0.990	0.990	DR
	0.01	0.99	0.996	0.997	0.996	Normal
Random forest	0.00	1.00	0.999	0.999	0.999	DR
	0.002	0.998	0.999	1.00	0.999	Normal
MLP	0.002	0.998	0.995	0.998	0.996	DR
	0.002	0.998	0.999	0.998	0.999	Normal
J48	**0.00**	**1.00**	**1.00**	**1.00**	**1.00**	**DR**
	0.00	**1.00**	**1.00**	**1.00**	**1.00**	**Normal**

Table 11
Weighted average values for each performance measures for binary classification.

Database	Classifier	FP rate	Specificity	Precision	Recall	F1 Score
IDRiD [34]	SVM	0.027	0.973	0.97	0.969	0.969
	Random forest	0.025	0.975	0.988	0.988	0.988
	MLP	0.021	0.979	0.988	0.988	0.988
	J48	**0.001**	**0.999**	**0.998**	**0.998**	**0.998**
MESSIDOR [35]	SVM	0.024	0.976	0.980	0.979	0.979
	Random forest	0.002	0.998	0.998	0.998	0.998
	MLP	0.011	0.989	0.989	0.989	0.989
	J48	**0.001**	**0.999**	**0.999**	**0.999**	**0.999**
KAGGLE [36]	SVM	0.008	0.992	0.995	0.995	0.995
	Random forest	0.001	0.999	0.999	0.999	0.999
	MLP	0.002	0.998	0.998	0.998	0.998
	J48	**0.00**	**1.00**	**1.00**	**1.00**	**0.999**

Table 12
Kappa statistic & validation accuracy of each classifier using IDRiD Database for binary classification.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	400	0.929	96.85
Random forest	408	0.972	98.78
MLP	408	0.972	98.78
J48	**412**	**0.995**	**99.76**

Table 13
Kappa statistic & validation accuracy of each classifier using MESSIDOR Database for binary classification.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	1175	0.958	97.92
Random forest	1198	0.996	99.8
MLP	1187	0.978	98.92
J48	**1199**	**0.998**	**99.92**

4.2. Analysis of system for binary classification

In binary classification, the system predicts whether the retinal fundus image belongs to DR or normal. The four classifiers used for analysis are SVM, Random Forest, MLP, and J48. The confusion matrices obtained by using IDRiD, MESSIDOR, and KAGGLE datasets are tabulated in Tables 5–7 respectively. From the confusion matrices, it is able to compute the performance measures that can be used to evaluate system performance. The detailed performance measures of normal and DR images using all the datasets individually is demonstrated in Tables 8–10. From the tabulation, the FPR of the J48 classifier is comparatively lower than other classifiers and almost producing high specificity, precision, recall, and F1-score values. Table 11 summarizes all the performance measures for binary classification by taking weighted average values indicate that the performance of the J48 classifier is better than others. The accuracy and K-score of each classifier using each dataset are listed in Tables 12–14. The interrater reliability indicated by K-score reflects the degree to which the data gathered in the experiment are accurate representations of the calculated variables. The K-score ranges from 0.81 to 1.00 can be considered as almost perfect agreement [37]. The proposed model with all classifiers thus gives K-score above 0.9 for binary classification, which suggests the system's reliability. The J48 classifier generates the highest K-score of the four classifiers. The average accuracy of the model using three datasets with the J48 classifier is 99.89%.

Table 14
Kappa statistic & validation accuracy of each classifier using KAGGLE Database for binary classification.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	34941	0.985	99.47
Random forest	35100	0.998	99.92
MLP	35054	0.995	99.79
J48	**35125**	**0.99**	**99.99**

Table 15
Confusion matrix for the evaluation of each classifier for multiclass classification using IDRiD Database.

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	130	0	1	3	0
Mild NPDR	0	0	20	0	0
Moderate NPDR	8	0	128	0	0
Severe NPDR	10	0	0	63	1
PDR	0	0	0	35	14

(a) SVM

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	125	0	1	8	0
Mild NPDR	0	5	15	0	0
Moderate NPDR	0	1	135	0	0
Severe NPDR	11	0	0	61	2
PDR	7	0	0	11	31

(b) Random forest

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	123	0	4	7	0
Mild NPDR	0	0	20	0	0
Moderate NPDR	5	0	31	0	0
Severe NPDR	3	0	0	68	3
PDR	0	0	0	3	46

(c) MLP

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	133	0	0	1	0
Mild NPDR	0	19	1	0	0
Moderate NPDR	1	0	135	0	0
Severe NPDR	0	0	0	73	1
PDR	0	0	0	0	49

(d) J48

4.3. Analysis of system for multiclass classification

For the analysis of multiclass classification, categorized data from IDRiD, MESSIDOR, and KAGGLE databases are used. The confusion matrices for multiclass classification using all the three datasets are

Table 16
Confusion matrix for the evaluation of each classifier for multiclass classification using MESSIDOR database.

	Normal	Mild	Moderate	Severe
Normal	534	12	0	0
Mild	5	140	8	0
Moderate	1	14	230	2
Severe	0	0	11	243

(a) SVM

	Normal	Mild	Moderate	Severe
Normal	545	1	0	0
Mild	5	142	6	0
Moderate	3	2	237	5
Severe	0	0	4	250

(b) Random forest

	Normal	Mild	Moderate	Severe
Normal	543	3	0	0
Mild	3	148	2	0
Moderate	0	2	237	8
Severe	0	0	9	245

(c) MLP

	Normal	Mild	Moderate	Severe
Normal	545	1	0	0
Mild	0	152	1	0
Moderate	0	0	246	1
Severe	0	0	0	254

(d) J48

Table 17
Confusion matrix for the evaluation of each classifier for multiclass classification using KAGGLE database.

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	25716	0	71	23	0
Mild NPDR	0	2365	78	0	0
Moderate NPDR	75	71	5146	0	0
Severe NPDR	421	0	0	447	5
PDR	52	0	0	634	22

(a) SVM

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	25810	0	0	0	0
Mild NPDR	0	2440	3	0	0
Moderate NPDR	1	0	5291	0	0
Severe NPDR	29	0	0	840	4
PDR	10	0	0	12	686

(b) Random forest

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	25772	0	38	0	0
Mild NPDR	0	2380	63	0	0
Moderate NPDR	47	51	5194	0	0
Severe NPDR	873	0	0	0	0
PDR	708	0	0	0	0

(c) MLP

	Normal	Mild NPDR	Moderate NPDR	Severe NPDR	PDR
Normal	25809	0	0	1	0
Mild NPDR	0	2442	1	0	0
Moderate NPDR	1	0	5291	0	0
Severe NPDR	0	0	0	872	1
PDR	0	0	0	0	708

(d) J48

demonstrated in Tables 15–17. The detailed performance metrics are computed and tabulated in Tables 18–20. The weighted average values of these performance metrics are summarized in Tables 21–23. It shows that SVM and MLP classifier's performance is not appropriate for the proposed feature extraction while using IDRiD and KAGGLE. In the case of SVM using the IDRiD database, the classifier misclassified the mild NPDR images as moderate NPDR. This might happen because of

Table 18
Detailed efficiency measures for multiclass classification using IDRiD Database.

Classifier	FPR	Specificity	Precision	Recall	F1-Score	Class
SVM	0.065	0.935	0.878	0.970	0.922	Normal
	0.00	1.00	–	0.00	–	Mild NPDR
	0.076	0.924	0.859	0.941	0.898	Moderate NPDR
	0.112	0.888	0.624	0.851	0.720	Severe NPDR
	0.003	0.997	0.933	0.286	0.437	PDR
Random forest	0.065	0.935	0.874	0.933	0.903	Normal
	0.003	0.997	0.833	0.250	0.385	Mild NPDR
	0.058	0.942	0.894	0.993	0.941	Moderate NPDR
	0.056	0.944	0.763	0.824	0.792	Severe NPDR
	0.005	0.995	0.939	0.633	0.756	PDR
MLP	0.029	0.971	0.939	0.918	0.928	Normal
	0.00	1.00	–	0.00	–	Mild NPDR
	0.087	0.913	0.845	0.963	0.90	Moderate NPDR
	0.029	0.971	0.872	0.919	0.895	Severe NPDR
	0.008	0.992	0.939	0.939	0.939	PDR
J48	**0.004**	**0.996**	**0.993**	**0.993**	**0.993**	**Normal**
	0.00	**1.00**	**1.00**	**0.950**	**0.974**	**Mild NPDR**
	0.004	**0.996**	**0.993**	**0.993**	**0.993**	**Moderate NPDR**
	0.003	**0.997**	**0.986**	**0.986**	**0.986**	**Severe NPDR**
	0.003	**0.997**	**0.980**	**1.00**	**0.990**	**PDR**

Table 19
Detailed efficiency measures for multiclass classification using MESSIDOR Database.

Classifier	FPR	Specificity	Precision	Recall	F1-Score	Class
SVM	0.009	0.991	0.989	0.978	0.983	Normal
	0.025	0.975	0.843	0.915	0.878	Mild
	0.020	0.98	0.924	0.931	0.927	Moderate
	0.002	0.998	0.992	0.957	0.974	severe
Random forest	0.012	0.988	0.986	0.998	0.992	Normal
	0.003	0.997	0.979	0.928	0.953	Mild
	0.010	0.99	0.960	0.960	0.960	Moderate
	0.005	0.995	0.980	0.984	0.982	severe
MLP	0.005	0.995	0.995	0.995	0.995	Normal
	0.005	0.995	0.967	0.967	0.967	Mild
	0.012	0.988	0.956	0.960	0.958	Moderate
	0.008	0.992	0.968	0.965	0.966	Severe
J48	**0.00**	**1.00**	**1.00**	**0.998**	**0.999**	**Normal**
	0.001	**0.999**	**0.993**	**0.993**	**0.993**	**Mild**
	0.001	**0.999**	**0.996**	**1.00**	**0.998**	**Moderate**
	0.001	**0.999**	**0.996**	**1.00**	**0.998**	**severe**

input images. The same happens with the KAGGLE dataset too. Here, the MLP was not able to differentiate severe NPDR and PDR. But with the MESSIDOR dataset, the system performs very well. It is noticeable that the J48 classifier produces less misclassification for all the datasets. The average FPR is comparatively very less and the other performance measures (specificity, precision, recall, and F1-score) approximately equal 1 for the J48 classifier. The K-score and accuracy of the system are tabulated in Tables 24–26, which narrates the best performance of the J48 classifier with CNN feature extraction. The average K-score and accuracy of the J48 classifier for multiclass classification is 0.994 and 99.59% respectively.

The ResNet-50 architecture is also used as a feature extractor and the extracted features are then fed into the same classifiers for multiclass classification. The K-score and Accuracy for each classifier obtained from ResNet-50 extracted features are illustrated in Table 27.

Some works already existed for DR grading using CNN. The comparison of the existing approaches with the proposed method can be analyzed from Table 28. The method in [4] separates the lesions that are relevant for DR grading using a local network and then a global network is used for DR grading. They obtained both the non-weighted and weighted scores for evaluation metrics. The weighted network gives high values of evaluation metrics than the other. So, we compared the weighted values with the evaluation metrics of the proposed method. From the analysis, it is observed that the proposed

Table 20
Detailed efficiency measures for multiclass classification using KAGGLE Database.

Classifier	FPR	Specificity	Precision	Recall	F1-Score	Class
	0.059	0.941	0.979	0.996	0.988	Normal
	0.002	0.998	0.971	0.968	0.969	Mild NPDR
SVM	0.005	0.995	0.972	0.972	0.972	Moderate NPDR
	0.019	0.981	0.405	0.512	0.452	Severe NPDR
	0.00	1.00	0.815	0.031	0.060	PDR
	0.004	0.996	0.998	1.00	0.999	Normal
	0.00	1.00	1.00	0.999	0.999	Mild NPDR
Random forest	0.00	1.00	0.999	1.00	1.00	Moderate NPDR
	0.00	1.00	0.986	0.962	0.974	Severe NPDR
	0.00	1.00	0.994	0.969	0.981	PDR
	0.175	0.825	0.941	0.999	0.969	Normal
	0.002	0.998	0.979	0.974	0.977	Mild NPDR
MLP	0.003	0.997	0.981	0.981	0.981	Moderate NPDR
	0.00	1.00	–	0.00	–	Severe NPDR
	0.00	1.00	–	0.00	–	PDR
	0.00	**1.00**	**1.00**	**1.00**	**1.00**	**Normal**
	0.00	**1.00**	**1.00**	**1.00**	**1.00**	**Mild NPDR**
J48	**0.00**	**1.00**	**1.00**	**1.00**	**1.00**	**Moderate NPDR**
	0.00	**1.00**	**0.999**	**0.999**	**0.999**	**Severe NPDR**
	0.00	**1.00**	**0.999**	**1.00**	**0.999**	**PDR**

Table 21
Weighted average values calculated for each measures in Table 19 for multiclass classification using IDRiD database.

Classifier	FP rate	Specificity	Precision	Recall	F1 Score
SVM	0.066	0.934	–	0.811	–
Random forest	0.051	0.949	0.866	0.864	0.853
MLP	0.044	0.956	–	0.891	–
J48	**0.003**	**0.997**	**0.990**	**0.990**	**0.990**

Table 22
Weighted average values calculated for each measures in Table 19 for multiclass classification using MESSIDOR database.

Classifier	FP rate	Specificity	Precision	Recall	F1 Score
SVM	0.012	0.988	0.958	0.956	0.956
Random forest	0.009	0.991	0.978	0.978	0.978
MLP	0.007	0.993	0.978	0.978	0.978
J48	**0.001**	**0.999**	**0.998**	**0.998**	**0.998**

Table 23
Weighted average values calculated for each measures in Table 20 for multiclass classification using KAGGLE database.

Classifier	FP rate	Specificity	Precision	Recall	F1 Score
SVM	0.045	0.955	0.960	0.959	0.952
Random forest	0.003	0.997	0.998	0.998	0.998
MLP	0.129	0.871	–	0.949	–
J48	**0.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 24
Kappa statistic & validation accuracy of each classifier for multiclass classification using IDRiD Database.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	335	0.736	81.11
Random forest	368	0.812	89.10
MLP	1032	0.849	86
J48	**409**	**0.986**	**99.03**

work shows higher performance than the two-stage CNN while using the KAGGLE database. They compared the two-stage network with Alexnet architecture also, which shows low performance than their global network. When we look at the binocular network which is introduced in [15] provided a K- score of 0.82, a specificity of 0.707, and a recall value of 0.82. When compared to this, the proposed method shows an improved classification performance with the highest values for specificity, recall (value of 1), and K-score (value of 0.99).

Table 25
Kappa statistic & validation accuracy of each classifier for multiclass classification using MESSIDOR database.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	1147	0.936	95.58
Random forest	1174	0.968	97.83
MLP	1173	0.967	97.75
J48	**1197**	**0.996**	**99.75**

Table 26
Kappa statistic & validation accuracy of each classifier for multiclass classification using KAGGLE database.

Classifier	Correctly classified instances	K-score	Accuracy (%)
SVM	33696	0.904	93.93
Random forest	35067	0.996	99.83
MLP	33346	0.873	94.93
J48	**35122**	**0.999**	**99.99**

Table 27
Kappa statistic & validation accuracy of each classifier for multiclass classification using RESNET-50 features.

Database	Classifier	K-score	Accuracy (%)
IDRiD [34]	SVM	0.595	70.46
	Random forest	0.434	60.29
	MLP	0.887	91.76
	J48	0.901	92.46
MESSIDOR [35]	SVM	0.76	83.75
	Random forest	0.251	56.25
	MLP	0.863	88.23
	J48	0.892	91.22
KAGGLE [36]	SVM	0.79	84.13
	Random forest	0.52	69.03
	MLP	0.883	91.72
	J48	0.92	93.46

Another advantage of the proposed method is that using a simple CNN as a feature extractor, the system achieves higher performance on DR classification while the existing methods are using pre-trained networks and transfer learning methods for the classification which may increase the system complexity. That means the proposed network along with the J48 classifier is a better option for DR grading. In [14], with the modified Alexnet architecture they evaluated the system using the MESSIDOR dataset and attained an average accuracy of 96.25%. The proposed system scores more than this for DR classification (Both binary and multi-class). Since the 32 relevant features from CNN are fed into the classifier reduces the computation time of classification.

From the performance analysis, it is clear that the proposed system works well irrespective of the number of the training samples in each dataset, that is proposed CNN with J48 classifier seems to be the best one for DR classification. Thus by using CNN as a feature extractor, it reduces the computational time and complexity. The proposed system also compensates for the issues faced in deep learning classification by producing high precision and sensitivity scores. Our major contribution is reducing the CNN model parameters significantly to enable real-time deployment while improving the classification accuracy. So the novelty comes in using a CNN based feature extractor along with Machine Learning (ML) classifier (SVM, Random Forest, MLP, J48 Classifier) for the classification rather than using a softmax layer (multinomial logistic regression [22]), we found that features extracted from the CNN if further trained in ML classifier would give an outstanding result which can be a benchmark for DR classification. Also, the use of CNN enables the extraction of features directly from the fundus image without any segmentation and handcrafted feature extraction process. The proposed method also assures to consider the whole image and not leaving out any regions that may be affected due to DR. The issue in the proposed method is the variation in the evaluation metrics of each classifier while using different databases. One of the reasons might

Table 28
Comparison of Proposed work with existing methods.

Dataset	Methods	Specificity	Precision	Recall	F1-Score	K-score	Accuracy (%)
IDRiD [34]	RESNET-50 + J48 [38]	–	–	–	–	0.901	92.46
	Proposed Method(multiclass)	**0.997**	**0.990**	**0.990**	**0.990**	0.996	99.03
MESSIDOR [35]	Modified AlexNet [14]	0.97	92.07	0.923	–	–	96.25
	RESNET-50 + J48 [38]	–	–	–	–	0.892	91.22
	Proposed Method(multiclass)	**0.999**	**0.998**	**0.998**	**0.998**	0.996	99.75
KAGGLE [36]	Y. Yang et al [4]	–	–	–	–	0.75	56
	X. Zeng et al [15]	70.7	–	0.822	–	0.82	–
	Y. H. Li et al.[18]	–	–	–	–	–	86.17
	RESNET-50 + J48 [38]	–	–	–	–	0.92	93.46
	Proposed Method(multiclass)	**1.00**	**1.00**	**1.00**	**1.00**	0.99	99.9

be the resolution of the images in each database. So, in the future, a feature extraction method that can overcome this difficulty can provide another breakthrough in automated DR detection and grading.

5. Conclusion

Clinicians are currently diagnosing the DR by examining lesions associated with disease-caused vascular anomalies. This approach is quite effective but it is a cost-effective technique and also the minute lesions may not be visible at the primary stage of DR. In the proposed work, an effective CNN is introduced for feature extraction and the features are fed into simple machine learning classifier for binary and multi-class classification of DR. The FPR, specificity, precision, recall, F1-score, K-score and accuracy obtained for each classifier (SVM, Random Forest, MLP, J48) are evaluated and concluded that the proposed network with J48 classifier gives the best performance than others. The proposed model is capable of providing competitive performance in DR classification with an average accuracy of 99.89% for binary classification and 99.59% for multi-class classification. Thus an effective classifier is developed by combining a simple CNN with ML classifiers for both DR detection and grading. Our major contribution is reducing the CNN model parameters significantly to enable real-time deployment while improving the classification accuracy. In the future, the same network may be used for other retinal disease detection with some modifications.

CRediT authorship contribution statement

Gayathri S.: Conceptualization, Methodology, Software, Validation, Writing - original draft, Data curation, Visualization. **Varun P. Gopi:** Supervision, Writing - review & editing. **P. Palanisamy:** Investigation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.D. Abramoff, M.K. Garvin, M. Sonka, Retinal imaging and image analysis, IEEE Rev. Biomed. Eng. 3 (2010) 169–208, http://dx.doi.org/10.1109/RBME.2010.2084567.

[2] N. Cheung, J.J. Wang, R. Klein, D.J. Couper, A.R. Sharrett, T.Y. Wong, Diabetic retinopathy and the risk of coronary heart disease, Diabetes Care 30 (7) (2007) 1742–1746, http://dx.doi.org/10.2337/dc07-0264.

[3] J. James, E. Sharifahmadian, L. Shih, Automatic severity level classification of diabetic retinopathy, Int. J. Comput. Appl. 180 (2018) 30–35, http://dx.doi.org/10.5120/ijca2018916244.

[4] Y. Yang, T. Li, W. Li, H. Wu, W. Fan, W. Zhang, Lesion detection and grading of diabetic retinopathy via two-stages deep convolutional neural networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 533–540.

[5] M.P. Paing, S. Choomchuay, M.D.R. Yodprom, Detection of lesions and classification of diabetic retinopathy using fundus images, in: 2016 9th Biomedical Engineering International Conference (BMEiCON), 2016, pp. 1–5.

[6] L. Seoud, J. Chelbi, F. Cheriet, Automatic grading of diabetic retinopathy on a public database, 2015.

[7] D.K. Prasad, L. Vibha, K.R. Venugopal, Early detection of diabetic retinopathy from digital retinal fundus images, in: 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2015, pp. 240–245, http://dx.doi.org/10.1109/RAICS.2015.7488421.

[8] S. Roychowdhury, D. Koozekanani, K. Parhi, DREAM: Diabetic retinopathy analysis using machine learning, IEEE J. Biomed. Health Inform. 18 (2014) 1717–1728, http://dx.doi.org/10.1109/JBHI.2013.2294635.

[9] S. Gayathri, A.K. Krishna, V.P. Gopi, P. Palanisamy, Automated binary and multiclass classification of diabetic retinopathy using haralick and multiresolution features, IEEE Access 8 (2020) 57497–57504.

[10] M.U. Akram, S. Khalid, A. Tariq, S.A. Khan, F. Azam, Detection and classification of retinal lesions for grading of diabetic retinopathy, Comput. Biol. Med. 45 (2014) 161–171, http://dx.doi.org/10.1016/j.compbiomed.2013.11.014.

[11] H. Pratt, F. Coenen, D.M. Broadbent, S.P. Harding, Y. Zheng, Convolutional neural networks for diabetic retinopathy, in: 20th Conference on Medical Image Understanding and Analysis (MIUA 2016), Procedia Comput. Sci. 90 (2016) 200–205, http://dx.doi.org/10.1016/j.procs.2016.07.014.

[12] M. Mookiah, U.R. Acharya, R.J. Martis, C.K. Chua, C. Lim, E. Ng, A. Laude, Evolutionary algorithm based classifier parameter tuning for automatic diabetic retinopathy grading: A hybrid feature extraction approach, Knowl.-Based Syst. 39 (2013) 9–22, http://dx.doi.org/10.1016/j.knosys.2012.09.008.

[13] E.M. Shahin, T.E. Taha, W. Al-Nuaimy, S. El Rabaie, O.F. Zahran, F.E.A. El-Samie, Automated detection of diabetic retinopathy in blurred digital fundus images, in: 2012 8th International Computer Engineering Conference (ICENCO), 2012, pp. 20–25, http://dx.doi.org/10.1109/ICENCO.2012.6487084.

[14] T. Shanthi, R. Sabeenian, Modified alexnet architecture for classification of diabetic retinopathy images, Comput. Electr. Eng. 76 (2019) 56–64, http://dx.doi.org/10.1016/j.compeleceng.2019.03.004.

[15] X. Zeng, H. Chen, Y. Luo, W. Ye, Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network, IEEE Access 7 (2019) 30744–30753.

[16] J. de la Calleja, L. Tecuapetla, M. Auxilio Medina, E. Bárcenas, A.B. Urbina Nájera, LBP and machine learning for diabetic retinopathy detection, in: E. Corchado, J.A. Lozano, H. Quintián, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2014, 8669 (2014) 110–117, http://dx.doi.org/10.1007/978-3-319-10840-7_14.

[17] D. Sidibé, I. Sadek, F. Mériaudeau, Discrimination of retinal images containing bright lesions using sparse coded features and SVM, Comput. Biol. Med. 62 (2015) 175–184, http://dx.doi.org/10.1016/j.compbiomed.2015.04.026.

[18] Y.-H. Li, N.-N. Yeh, S.-J. Chen, Y.-C. Chung, Computer-assisted diagnosis for diabetic retinopathy based on fundus images using deep convolutional neural network, Mob. Inf. Syst. 2019 (2019).

[19] J. Sahlsten, J. Jaskari, J. Kivinen, L. Turunen, E. Jaanio, K. Hietala, K. Kaski, Deep learning fundus image analysis for diabetic retinopathy and macular edema grading, Sci. Rep. 9 (1) (2019) 1–11.

[20] Z. Gao, J. Li, J. Guo, Y. Chen, Z. Yi, J. Zhong, Diagnosis of diabetic retinopathy using deep neural networks, IEEE Access 7 (2018) 3360–3370.

[21] M.M. Butt, G. Latif, D.A. Iskandar, J. Alghazo, A.H. Khan, Multi-channel convolutions neural network based diabetic retinopathy detection from fundus images, Procedia Comput. Sci. 163 (2019) 283–291.

[22] J. Wu, Convolutional neural networks, 2017, Published online at https://cs.nju.edu.cn/wujx/teaching/15CNN.pdf.

[23] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, arXiv preprint arXiv:1502.03167.

[24] N. Bjorck, C.P. Gomes, B. Selman, K.Q. Weinberger, Understanding batch normalization, in: Advances in Neural Information Processing Systems, 2018, pp. 7694–7705.

[25] J.P. Kandhasamy, S. Balamurali Kadry, L.K. Ramasamy, Diagnosis of diabetic retinopathy using multi level set segmentation algorithm with feature extraction using SVM with selective features, Multimedia Tools Appl. (2019) 1573–7721., http://dx.doi.org/10.1007/s11042-019-7485-8.

[26] G. Daqi, Z. Tao, Support vector machine classifiers using RBF kernels with clustering-based centers and widths, in: 2007 International Joint Conference on Neural Networks, 2007, pp. 2971–2976, http://dx.doi.org/10.1109/IJCNN.2007.4371433.

[27] A. Roychowdhury, S. Banerjee, Random forests in the classification of diabetic retinopathy retinal images, in: S. Bhattacharyya, T. Gandhi, K. Sharma, P. Dutta (Eds.), Advanced Computational and Communication Paradigms, vol. 475, Springer Singapore, 2018, pp. 168–176, http://dx.doi.org/10.1007/978-981-10-8240-5_19.

[28] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[29] G.-B. Huang, Y.-Q. Chen, H.A. Babri, Classification ability of single hidden layer feed forward neural networks, IEEE Trans. Neural Netw. 11 (3) (2000) 799–801, http://dx.doi.org/10.1109/72.846750.

[30] H. Saifuddin, H. Vijayalakshmi, Prediction of diabetic retinopathy using multi layer perceptron, Int. J. Adv. Res. 4 (2016) 658–664, http://dx.doi.org/10.21474/IJAR01/714.

[31] S. Sharma, J. Agrawal, S. Sharma, Classification through machine learning technique: C4. 5 algorithm based on various entropies, Int. J. Comput. Appl. 82 (2013) 28–32, http://dx.doi.org/10.5120/14249-2444.

[32] S. Yadav, S. Shukla, Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification, in: 2016 IEEE 6th International Conference on Advanced Computing (IACC), 2016, pp. 78–83, http://dx.doi.org/10.1109/IACC.2016.25.

[33] S. Visa, B. Ramsay, A. Ralescu, E. Knaap, Confusion matrix-based feature selection, in: CEUR Workshop Proceedings, vol. 710, 2011, pp. 120–127.

[34] P. Porwal, S. Pachade, R. Kamble, M. Kokare, G. Deshmukh, V. Sahasrabuddhe, F. Meriaudeau, Indian diabetic retinopathy image dataset (IDRiD): A database for diabetic retinopathy screening research, Data 3 (2018) 1–8.

[35] E. Decencière, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay, B. Charton, J.-C. Klein, Feedback on a publicly distributed database: The Messidor database, Image Anal. Stereol. 33 (3) (2014) 231–234, http://dx.doi.org/10.5566/ias.1155.

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

[37] M. McHugh, Interrater reliability: The kappa statistic, in: Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB, vol. 22, 2012, pp. 276–282, http://dx.doi.org/10.11613/BM.2012.031.

[38] S. Targ, D. Almeida, K. Lyman, Resnet in resnet: Generalizing residual architectures, 2016, arXiv preprint arXiv:1603.08029.