International Journal of Imaging Systems and Technology

WILEY

RESEARCH ARTICLE OPEN ACCESS

MEDCnet: A Memory Efficient Approach for Processing High-Resolution Fundus Images for Diabetic Retinopathy Classification Using CNN

Mohsin Butt[1,2] | D. N. F. NurFatimah[1] | Majid Ali Khan[3] | Ghazanfar Latif[3,4] | Abul Bashar[3]

[1]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Kota Samarahan, Malaysia | [2]College of General Studies, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia | [3]Department of Computer Science, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia | [4]Department of Computer Science, Thompson Rivers University, Kamloops, British Columbia, Canada

Correspondence: Ghazanfar Latif (glatif@pmu.edu.sa; glatif@tru.ca)

Received: 26 September 2024 | **Revised:** 7 February 2025 | **Accepted:** 2 March 2025

Funding: The authors received no specific funding for this work.

Keywords: convolutional neural networks (CNN) | deep learning | diabetic retinopathy | divide and conquer CNN | fundus images | medical diagnosis | memory efficient CNN | performance optimization

ABSTRACT

Modern medical imaging equipment can capture very high-resolution images with detailed features. These high-resolution images have been used in several domains. Diabetic retinopathy (DR) is a medical condition where increased blood sugar levels of diabetic patients affect the retinal vessels of the eye. The usage of high-resolution fundus images in DR classification is quite limited due to Graphics processing unit (GPU) memory constraints. The GPU memory problem becomes even worse with the increased complexity of the current state-of-the-art deep learning models. In this paper, we propose a memory-efficient divide-and-conquer-based approach for training deep learning models that can identify both high-level and detailed low-level features from high-resolution images within given GPU memory constraints. The proposed approach initially uses the traditional transfer learning technique to train the deep learning model with reduced-sized images. This trained model is used to extract detailed low-level features from fixed-size patches of higher-resolution fundus images. These detailed features are then utilized for classification based on standard machine learning algorithms. We have evaluated our proposed approach using the DDR and APTOS datasets. The results of our approach are compared with different approaches, and our model achieves a maximum classification accuracy of 95.92% and 97.39% on the DDR and APTOS datasets, respectively. In general, the proposed approach can be used to get better accuracy by using detailed features from high-resolution images within GPU memory constraints.

1 | Introduction

CNN (convolutional neural network) models are extensively used for image and video processing tasks, such as object detection, image classification, and segmentation. In certain application domains (such as medical imaging and remote sensing) CNN models can provide better accuracy with higher resolution input images [1–3]. Diabetic retinopathy (DR) is an eye complication that is caused by diabetes. The increased blood sugar levels of diabetic patients can affect the blood vessels inside the retina of the eye [4]. DR is a major cause of blindness in diabetes patients. It mostly remains undetected in the early stages until we reach the later stages of the disease. Fundoscopy is a widely used imaging technique that captures the back of the eye called the fundus [5]. In this imaging technique, the patient's eyes are dilated, and a fundus camera is used to capture a high-resolution color image of the fundus. The different abnormalities caused by DR in the eye include red lesions like microaneurysms (MA)

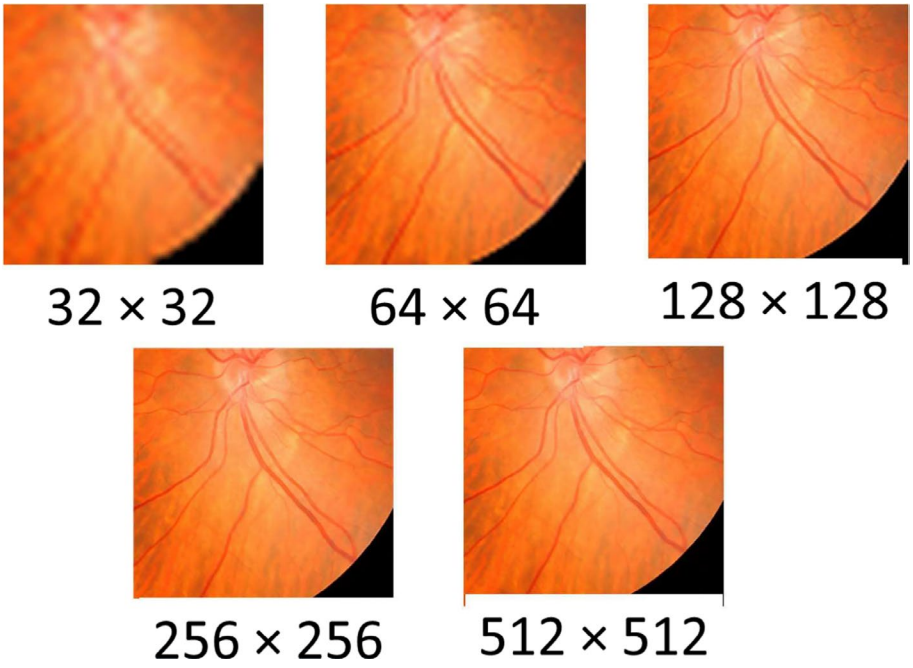

FIGURE 1 | Examples of level of details in fundus image at different resolutions.

and intra-retinal hemorrhages (HE) [6]. Besides these, white lesions that appear in the eye due to DR include exudates (EX) and cotton wool spots. Detecting DR from fundus images, for instance, relies heavily on the precise identification of subtle anomalies. Fundus images of varying quality and resolution are used by ophthalmologists to capture the internal structure of the eye and help diagnose DR in its early stages. CNNs trained on such high-resolution images can discern nuanced vessel textures, variations in density, and other critical features crucial for accurate diagnosis. This can be seen in Figure 1 that the level of detail perceptible in the image increases with higher resolution.

CNNs operate by leveraging hierarchical feature extraction through convolutional layers. While lower resolutions may capture general object shapes, higher resolutions offer a wealth of local details, often crucial for accurate classification. It is known that CNN models tend to rely mostly on local features rather than global features for classification purposes [7, 8], which is an inverse of how human perception works. Local features provide several advantages over global features. Firstly, local features capture more detailed and distinctive information about specific regions of an image. Global features, on the other hand, treat the entire image as a single entity and do not differentiate between different regions of the image. This can result in a loss of important information about the image that could be critical for accurate classification. Secondly, local features are more robust to variations in the image, such as changes in lighting, rotation, and scale. This is because local features are computed based on small patches of the image and are therefore less affected by changes in the overall appearance of the image. Global features, on the other hand, may not be as robust to such variations, as they consider the entire image as a whole. Lastly, local features can be combined in a spatially aware manner to capture the overall structure and layout of the image. Overall, local features are a powerful tool for image classification because they can capture fine-grained, distinctive information about specific regions of an image while also being more robust to variations and allowing for spatially aware feature combinations.

However, the increase in input image dimensions for CNN-based models substantially increases the number of trainable parameters and thus requires a significant amount of memory to store and process data. This results in a higher computational cost (increased time complexity) and requires larger memory (increased space complexity). There have been several attempts at mitigating the time complexity aspect of model training using data and model parallelism [9, 10].

A critical constraint in working with deep learning models using a large number of parameters with higher resolution images is the available graphics processing units (GPU) memory limitations. A popular approach for reducing space complexity is to reduce model complexity with a reduced number of parameters such as MobileNet, SqueezeNet, and EfficientNet. However, these memory-efficient models can still result in larger memory requirements for processing higher resolution images.

The major contribution of this paper is that it proposes a novel Memory Efficient Divide and Conquer based approach for training deep learning models (MEDCNet) with high-resolution input images. The main objective is to extract high-resolution image features from the fundus images of the eye that can help in providing better classification for DR images. The work proposes a divide and conquer approach by dividing the fundus image into various patches and extracts high-resolution features which help improve the performance of DR classification. The approach uses transfer learning (from a model trained on reduced sized images) to identify both high-level and detailed low-level features from input high-resolution fundus images. The approach results in improving overall test accuracy within given GPU memory constraints. In what follows, Section 2 discusses the related work in the

field of DR detection from high-resolution images and large images in general for image classification tasks. Further, the proposed methodology is presented in detail in Section 3. The details of the dataset used in the research work are provided in Section 4. Results and their related discussions are elaborated in Section 5. The paper concludes in Section 6 with directions toward future work.

2 | Related Work

The issues associated with processing and analyzing images of high resolution, which have been identified earlier, have raised the interest of researchers in finding effective solutions to address them. There have been solutions that focus on image size variation, feature extraction methods, deep learning models, and various parameter optimization schemes.

In order to comprehend the knowledge in this domain, we have adopted a two-pronged approach, namely, research on high-resolution images (a generic approach) and research on Diabetic Retinopathy images (a specific approach). We first proceed with the study of research work performed in the domain of Deep Learning approaches utilized in the context of high-resolution images.

2.1 | DL Approaches for High-Resolution Images Classification

This study starts with a recent survey paper [11] which performs a comprehensive summary of where the high-resolution images exist and the approaches used to address their management and classification. The application areas include medical diagnosis, remote sensing, security and surveillance, agriculture, and material science. Some of the prominent approaches for working with high-resolution approaches include, uniform downsampling, high-resolution vision transformers, and lightweight scanner networks, which have been shown to be effective in reducing processing complexity and also improving performance in terms of image identification and classification. The paper has found that image resolutions in these applications can range from 360×360 to 200×200 K. Several prominently used datasets are also mentioned, which include, PANDA (for person detection), CAMELYON (for pathology), CAD-CAP (for endoscopy), INbreast (for breast cancer detection) and FAIR1M (for object detection). The related work on the recent research in the area of high resolution images in summarized in Table 1.

In order to define the scope of work, we choose the medical diagnosis field as a subset to focus on within the context of high resolution images. The authors in [1], have used the HyperKvasir dataset which had Endoscopy images and trained the CNN models like DenseNet-161 and ResNet-152 for identifying patients suffering from gastrointestinal tract infections. The study focused on DL model performance variation by the change in image resolutions from 32×32 to 512×512. They achieved a maximum MCC score of 0.9 with images having the highest resolution. In a similar work on chest radiographs data from NIH, CNNs (namely, ResNet-34 and DenseNet-121) were utilized on image sizes from a resolution of 32×32 to 600×600. Again, the experiments clearly suggested that the classifier performance improves with the increase in the image resolutions, with a maximum value of AUC as 86.7%. In a similar work by [17], the authors were varying the image resolutions from 32×32 to 600×600 for chest radiographs of the NIH dataset. This study utilized the pre-trained CNN models (ResNet34 and DenseNet121) and achieved a performance of 86.7% for the AUC metric.

In another interesting study [12], the authors have worked on the effect of DL performance by keeping the input image sizes fixed (with a resolution of 96×96), but varying the optimizer, batch size and learning rate of the DL algorithms. The experiments related to the VGG16 and ImageNet have shown that the higher batch size does not necessarily achieve higher accuracy. However they achieved a significantly higher accuracy with 96.77% as the AUC value. To deal with images of high resolution, the authors in [13], have used the cut into patches (CIP) approach on the whole slide images (WSI) relating to the pathological tests conducted for detecting eye diseases. The original WSI images in the ZJU-2 dataset were in the order of few gigapixels and formed as an input to the pre-trained VGG16 CNN model. It was observed through experiments that the CIP approach gave a lower classification accuracy of 94.9% with a lower computational complexity. However, with the WSI approach the accuracy increased to 98.2% and so did the computational complexity in terms of memory and CPU cycles.

An end-to-end part learning (EPL) approach was utilized on the BCC skin cancer dataset where the original images had a resolution in the order of a few gigapixels [14]. However, in this work, the ResNet34 CNN model was used, and the EPL approach observed tiles in an image and made classifications based on the discriminative features from these smaller tiles, as opposed to the features extracted from the whole images. This approach was successful in providing an enhanced AUC of 98.6%. In a similar work on image tiling, the authors utilized an RNN model on the multi-center WSI dataset for cervical cancer [15]. With image resolutions of 1 gigapixel, they achieved a sensitivity values of 95.1% with one WSI image taking about 1.5 min to process.

Finally, the authors have used the FCN with VGG16 on multi-gigapixel images from the Camelyon dataset for breast cancer detection [16]. From the computational complexity point of view, this approach took about 1 min to process a WSI image. With the image resolutions of 200×200 K an AUC score of 96.69% was obtained.

In summary, one of the popular approaches to deal with the computational complexity of high-resolution images is to use the CIP approach, which reduces the complexity; however, the performance in terms of classification accuracy also lowers. We now proceed to focus our study of the related work in the area of diabetic retinopathy detection using DL approaches.

2.2 | DL Approaches for Diabetic Retinopathy Detection

Now, we proceed with the study of research performed in the domain of Deep Learning approaches utilized in the context of

TABLE 1 | A comparative summary of recent related work in deep learning for high resolution images.

No.	Ref	Year	Approach	Dataset	Results	Comments
1	[1]	2021	CNN (DenseNet-161, ResNet-152)	HyperKvasir, Endoscopy images (512×512)	MCC of 0.9002	A study to see the performance variation with image resolution, Improved performance with higher image resolution
2	[2]	2020	CNN (ResNet-34, DenseNet-121)	NIH, chest radiographs (600×600)	AUC 86.7%	A study to see the performance variation with image resolution
3	[12]	2020	VGG16, ImageNet	Patch Camelyon, histo-pathologic scans (96×96)	AUC 96.77%	Fixed image size, variation of optimizer, variation of batch size, variation of learning rate, higher batch size does not necessarily achieve higher accuracy
4	[13]	2020	VGG16, Cut into Patches (CIP)	ZJU-2 pathological WSI eye diseases (few gigapixels)	WSI accuracy of 98.2% CIP accuracy of 94.9%	WSI performed better with more computational complexity
5	[14]	2020	ResNet34, End-to-end Part Learning (EPL)	BCC dataset for skin cancer (few gigapixels)	AUC of 98.6%	Observing the tiles in an image and making classification based on the discriminative features
6	[15]	2021	RNN with image tiles	Multi-center WSI dataset for cervical cancer (1 gigapixel)	Sensitivity of 95.1%	One WSI image takes about 1.5 min to process
7	[16]	2019	FCN using VGG16	Camelyon (2016) for breast cancer (200×100K)	AUC score of 96.69%	One WSI image takes about 1 min to process

diabetic retinopathy identification and classification. The related work on the recent research in the area of diabetic retinopathy images is summarized in Table 2.

A review paper by the authors in [23] has provided a comprehensive summary of the research domain where deep learning approaches have been specifically applied to the detection and classification of diabetic retinopathy (DR). They provide details of some of the popularly used datasets in this area, namely, Kaggle EyePACS, DDR, and Kaggle APTOS. Also, they present the usage of various DL models such as generic DL approaches, transfer learning, and ensemble learning approaches. One of the important techniques is about the model training with patches, which supports the researchers in dealing with computational complexities associated with high-resolution images.

The majority of the papers focus on pre-trained DL models. One such work [18], applied 16 pre-trained CNN models on three datasets (Kaggle DRD, IDRiD, and DDR) to perform multi-class classification of DR images. They achieved a maximum accuracy of 79.1% with the DenseNet121 model on image resolutions of 4288×2848 from the IDRiD dataset. The authors in [19], have used VGG16 and ImageNet models on 7 different datasets (DDR, IDRiD, Kaggle, Messidor, Messidor2, DIARETDB0, and DIARETDB1) by leveraging the patch-based method to achieve a ROC value of 91.2%. The original image sizes were 512×512 and the patch size was empirically chosen as $p = 65$ pixels. In a similar work by [20], eight different pre-trained models (namely, VGG16, ResNet18, GoogleNet, DenseNet-121, Inception, SSD, YOLO and RCNN) were used to work on the DDR dataset with image resolutions of 512×512. The maximum classification accuracy of 76.59% was reported for the ResNet18 model for a six-class grading experiments. However, the AP and IoU results were on the lower side. They also performed lesion detection by using SSD, YOLO and RCNN models which also did not perform well.

However, there are some approaches which do not resort to pre-trained CNN models, but rather use the standard ready-made CNN models or build their models from scratch. One such work by [17], uses the CNN512 and YOLOv3 deep learning models on the DDR and APTOS datasets and work on fixed sized images of resolution of 512×512. With this approach which also involves data augmentation, they achieve a classification accuracy of 89% for identifying five classes of DR cases. Some novel approaches have harnessed the optimization characteristics of bio-inspired optimization algorithms. One such paper by [21], uses Auto-regressive-Henry Gas Sailfish Optimization (Ar-HGSO) on the augmented DDR and IDRiD datasets, but re-sizing the images to a resolution of 256×256 for training their models. They achieve a five class classification accuracy of 91.4% for the IDRiD dataset. Yet another recent approach uses the Visual Transformer and Residual attention models for classifying five class of DR by achieving an AUC metric value of 90% [22]. They have used the original image resolution of 512×512 from the DDR and IDRiD datasets, but also perform a comparison with five other pre-trained CNN models.

Based on this literature survey of the deep learning based DR identification and classification, we come to the conclusion that images of high resolution need to be divided into segments or patches and then the DL model needs to be trained. This reduces the memory, processing resources, and time complexity of the models with a minimal loss of performance. However, we also propose a variation to this methodology by training the model with the image features that are generated by the DL model, but we perform the classification with the traditional ML classifiers to reduce the overall system complexity while minimizing the performance loss. The following section provides the details of the proposed methodology.

3 | Methodology

In this section, our proposed framework to deal with high-resolution fundus images that minimizes GPU memory usage is presented. Figure 2 shows the overall workflow of the proposed framework. The images in the DDR and APTOS datasets are passed through the pre-processing stage first where the retinal area of the eye is extracted from the fundus images. Contrast limited adaptive histogram equalization (CLAHE) is applied to the extracted images for enhancement. Next, the high-resolution features are extracted from the fundus images using our proposed framework, namely, the divide and conquer approach. These features are passed to different ML classifiers, that is, SVM, RF, and MLP. The results of the classification are evaluated using various performance metrics and compared with recent studies.

As stated earlier, higher resolution images that are given as input to the CNN models can improve the quality of features that are extracted from the input data. The inter-class similarities between different classes of DR can be better differentiated by the CNN model if the features are extracted from a high-resolution input image. However, increasing the size of the input image to a CNN causes increased training time and high GPU memory utilization. Most GPUs are unable to handle inputs with high pixel density due to memory limitations. Our framework proposes a divide and conquer-based approach for handling high-resolution images while keeping the GPU memory utilization within bounds. The main idea of the proposed algorithm is presented in Figure 3. The approach consists of three stages: (1) Model training on resized input images, (2) features extraction using transfer learning (from the trained model in previous stage) on patches of high-resolution fundus images, and (3) classification based on the detailed extracted features using standard machine learning algorithms. These steps are explained in detail in the following subsections.

3.1 | Model Training on Resized Input Images

The high-resolution input images are first resized to a smaller dimension. For example an image of dimension (w, h) would be resized to $\left(w/k_w, h/k_h\right)$ by a ratio of $k = k_w \times k_h$ where $k_w > 1$ and $k_h > 1$ are the resize ratios of width and height respectively and k defines the overall resize ratio. The resized images are then used to train an optimal CNN-based deep learning model using transfer learning. It should be noted that for using pre-trained models, the resizing should take into consideration the appropriate input size requirements.

In transfer learning, knowledge gained from one domain (source task) is used to improve learning in a different but related

TABLE 2 | A comparative summary of related work in deep learning for diabetic retinopathy detection.

No.	Ref	Year	Approach	Dataset	Results	Comments
1	[18]	2022	16 pre-trained CNN models	Kaggle DRD, IDRiD and DDR (4288×284) (IDRiD)	Accuracy of 79.1% for DenseNet121 (multiclass)	A study to see the performance of various pre-trained CNN Models, DenseNet121 performed the best on Kaggle dataset
2	[19]	2020	VGG16 and ImageNet with Patch-based method	7 datasets (DDR, IDRiD, Kaggle, Messidor, Messidor2, DIARETDB0 and DIARETDB1) (512×512)	ROC 91.2%	Patch size was chosen empirically ($p = 65$ pixels)
3	[20]	2019	VGG16, ResNet18, GoogleNet, DenseNet-121, Inception, SSD, YOLO, RCNN	DDR (512×512)	Accuracy ResNet-18 (76.59%)AP and IoU obtained are very low	Detecting six classes, DR grading experiments (VGG16, ResNet18, GoogleNet, DenseNet-121, Inception), Lesion detection (SSD, YOLO, RCNN) not performing well
4	[17]	2021	CNN512 and YOLOv3	DDR and APTOS (512×512)	Accuracy of 89%	Identification of five classes of DR with data augmentation
5	[21]	2022	Ar-HGSO (Autoregressive Henry Gas Sailfish Optimization)	DDR and IDRiD (256×256)	Accuracy 91.4% for IDRiD dataset	Data augmentation used to classify DR in five severity levels
6	[22]	2023	Visual Transformer and Residual attention	DDR and IDRiD (512×512)	AUC of 90%	Identification of five classes of DR, comparison with five other pre-trained CNN models

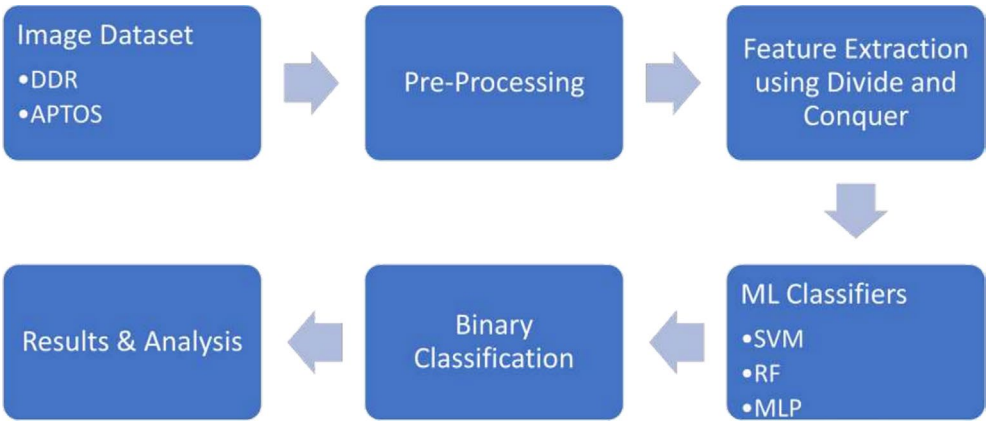

FIGURE 2 | Block diagram showing the overall proposed workflow.

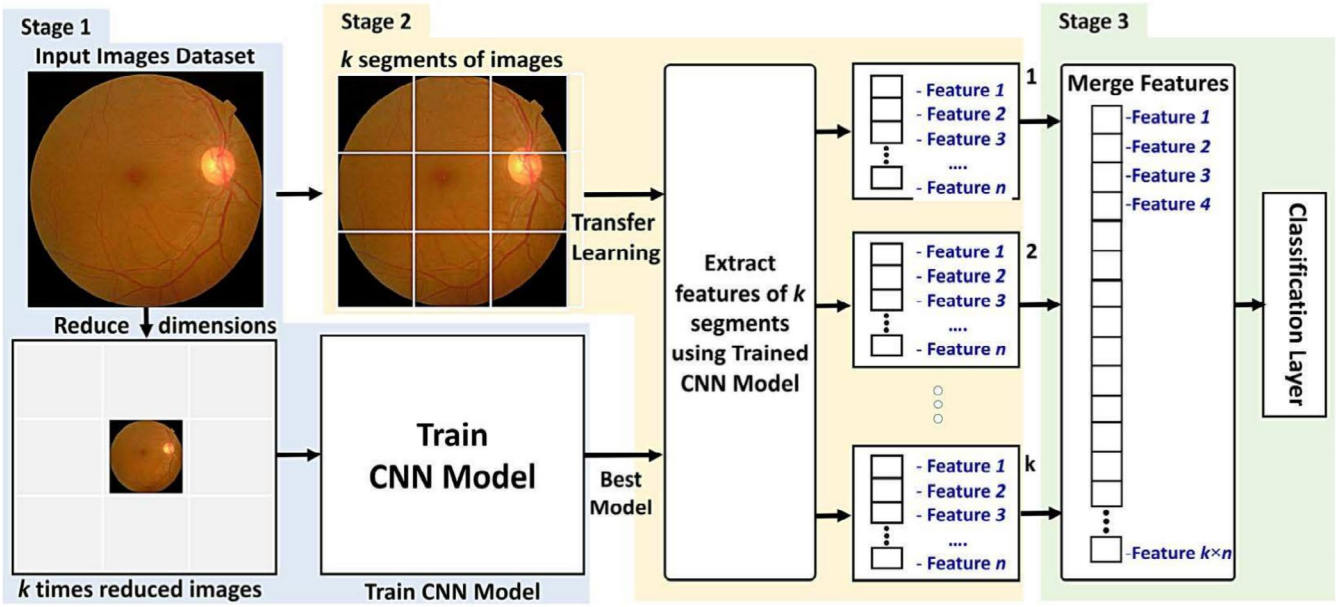

FIGURE 3 | Proposed approach for extracting features from high resolution images.

domain (target task). The transfer process involves taking the pre-trained model and adapting its feature extraction layers to the specific characteristics of the target task. This adaptation is typically done through fine-tuning, where the model is trained on the target task using a smaller dataset. There are different methods available for fine-tuning existing CNN models, including updating the architecture, retraining the model, or freezing partial layers of the model to utilize some of the pre-trained weights. Fine-tuning allows the model to adjust its learned features to better align with the nuances of the target task, leading to improved performance even when limited labeled data is available for the target task.

In this paper, we evaluated several pre-trained models for transfer learning, including GoogleNet, AlexNet, MobileNet, and Inceptionv3. Based on the initial comparative performance evaluation of these pre-trained models in our problem domain, we observed that GoogleNet provided better accuracy. Therefore, we chose to use GoogleNet for training our CNN models. GoogleNet [24] is a CNN-based architecture that has 22 layers and efficiently utilizes computational resources with repeated inception modules. The inception module employs multiple parallel convolutional filters of different sizes within the same layer. This allows the network to capture features at various spatial scales and helps prevent the vanishing gradient problem associated with very deep networks. The inception module for the GoogleNet model is shown in Figure 4. The 1×1 convolutional layers reduce the dimensions of the input and extract the local cross-channel features. The 3×3 and 5×5 convolutional layers help in capturing the spatial features of the input. The pooling layer is included in the inception module to reduce the dimensions of the input.

It is to be noted that training the above-mentioned models from scratch requires computation and data resources. On the other hand, because of the difference in the domain of the target task, transferring all learned weights as they are may not perform well in the new setting. Thus, it is generally better to freeze the initial layers and replace the latter layers with random initialization. This partially altered model is retrained on the current dataset to learn the new data classes. The number of layers that are frozen or fine-tuned depends on the available dataset

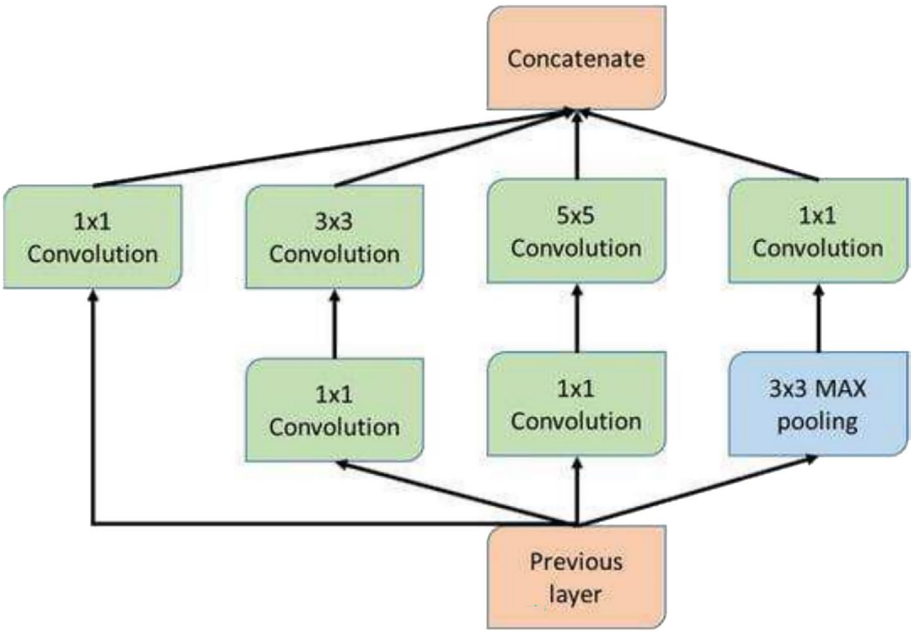

FIGURE 4 | Working of the inception module in the GoogleNet CNN model.

and computational power. If sufficient data and computational power are available, then we can unfreeze more layers and fine-tune them for the specific problem.

3.2 | Feature Extraction Using Transfer Learning From Optimal Trained Model

The pre-trained model trained in the previous stage can be used to extract features by using transfer learning. This feature extraction approach is a specific application of transfer learning that is focused on extracting relevant and generic features using the pre-trained models.

In this stage, the original high-resolution input images of dimension (w, h) are cut into k patches of dimensions $\left(w/k_w, h/k_h\right)$. The deep learning model trained in the previous stage is used to extract n features from each of the patch resulting in overall $k \times n$ features. These collective detailed merged features are then used for training a classification model. The specific steps of cutting images into patches and feature extractions are formally explained in an algorithmic format in Section 3.4.

3.3 | Classification

The classification using the merged features is performed using standard machine learning algorithms such as random forest (RF), support vector machines (SVM) and multi-layer perceptron (MLP). RF is an ensemble learning algorithm that combines the predictive power of multiple decision trees to achieve robust and accurate results. Given training data consisting of N data points, each with d features: $\left(x_1, y_1\right), \ \ldots, \left(x_N, y_N\right)$, where $x_i \in \mathfrak{R}^d$ and $y_i \in Y$. For each of the M trees in the forest, it randomly samples m features $(m < d)$ without replacement. Then for each internal node it randomly selects a subset of the m features. The best split among the selected features is then chosen based on an impurity measure (such as Gini index for classification). It then splits the data based on the chosen split point. The steps are repeated until a stopping criterion is met. There are several parameters that can be tuned for training RF based models such as (1) the number of individual decision trees within the forest (more trees improve robustness but with increased training time), (2) criteria to determine the quality of split (such as Gini impurity), and (3) maximum depth of the trees (deeper trees can capture more complex relationships but risk overfitting).

SVM based classification approach is based on finding a hyperplane (decision boundary) that maximizes the margin between the closest data points from different classes, known as support vectors. Suppose you have a dataset with input vectors X and corresponding labels y, where X is of size $m \times n$ (m examples, n features), and y is a vector of size m. The goal is to find a hyperplane defined by a weight vector w and a bias term b such that the hyperplane effectively separates the data into two classes and the margin between the two classes is maximized. The decision function for a linear SVM is given by: $f(x) = sign(\mathbf{w} \cdot \mathbf{x} + b)$ where, w is the weight vector, x is the input vector, and b is the bias term. The optimization problem to find w and b is typically formulated as follows: $\frac{1}{2}\|w\|^2$ subject to the constraints: $y_i\left(\mathbf{w} \cdot \mathbf{x}_i + b\right) > 1, \forall i = 1, 2, \ \ldots, m$. There are several SVM parameters that can be tuned for training such as (1) regularization parameter that can be used to determine a trade-off between maximizing the decision boundary margin and minimizing mis-classification errors, and (2) choice of several kernels which are used for mapping the input data points into a higher-dimensional space so that the separation between the classes becomes easier. The linear, RBF and polynomial kernels are the commonly used choices.

MLP represents a classical neural networks-based architecture. It is characterized by a layered structure that consists of an input

layer, one or more hidden layers, and an output layer. Each layer fully connects to the subsequent layer. MLPs leverage nonlinear activation functions in hidden layers that enable them to model complex relationships between inputs and outputs that are not linearly separable. Common activation functions include ReLU, sigmoid, and tanh. MLP models are trained using the backpropagation algorithm, which propagates the error signal backwards through the network, adjusting the weights and biases of connections to minimize the loss function. There are several parameters that can be tuned for training MLP-based models. Some of the common parameters include (1) number of hidden layers, (2) activation functions used by individual neurons to transform input (RELU is a common choice), (3) loss function to measure the discrepancy between actual and predicted output, (4) optimizing algorithm used to update weights based on error computed by the loss function, (5) learning rate to determine how quickly weights are updated, (6) different types of regularization and dropout layers to prevent overfitting, (7) number of epochs used for training, and (8) batch size to determine the number of training samples processed together.

3.4 | Proposed Algorithm

ALGORITHM 1 | Algorithm for Extracting High-Resolution Features From Input Images.

Require: Fundus images M with labels L and scaling factor k (number of image patches)

Algorithm:

read M, read L
P ← Pre-process (M)
R ← Resize the processed images P based on CNN model
T, V ← Divide R into training T and validation V
f_c ← Number of neurons in the fully connected layer of original CNN
n ← Number of features to extract for each patch that is, $\left\lfloor \frac{f_c}{k} \right\rfloor$
Net ← Modify CNN model with added f_n layer to extract
M ← Train modified model (*Net*) with (T, V)
$k_w, k_h \leftarrow \sqrt{k}$ (Dimension of patches based on scaling factor k)
P_w, P_h ← Height h and width w of pre-processed images P
for each image p in P **do**
  for x in range(k_w) **do**
    for y in range(k_h) **do**
      $S \leftarrow \frac{p_w}{k_w} * \frac{p_h}{k_h}$
      $I_{p,x,y}$ ← Extract patch S based on (x, y) from image p
end for
end for
end for
for each p in I **do**
  for x in range(k_w) **do**
    for y in range(k_h) **do**
      $f \leftarrow$ Transfer learning $\left(M, I_{p,x,y}\right)$ (Extract n features)
      $F_p \leftarrow F_p \frown f$
end for
end for
end for
ML_Model = Train (classifier, F, L)
DR_class = Classify (ML_Model, F_p)

The steps involved in implementing the proposed approach are formally presented in Algorithm1. The algorithm first reads and pre-processes the input images using thresholding and bounding box methods to extract retina structure from the input fundus images of varying sizes. In the next stage, the pre-processed images are resized by a scaling factor k that is defined in Equation (1).

$$k = k_w * k_h \tag{1}$$

where w and h are scaling ratios.

For example, if the scaling factor is nine, k_w is set to three and k_h is set to three. This means that pre-processed images will be reduced in width and height by a factor of three. The resized images are used to train a modified CNN model in stage 1 that extracts n features. The number of features to extract is calculated by dividing the number of neurons in the FC connected layer of the CNN divided by the scaling factor k and taking the floor of the resulting value, as given in Equation (2).

$$n = \left\lfloor \frac{f_c}{k} \right\rfloor \tag{2}$$

If the scaling factor is nine and there are 1000 neurons in the FC layer of the CNN, then the value of n will be $n = \text{floor}(1000/9)$, that is, 111. The CNN model in stage 1 is modified by adding a fully connected layer with n neurons before the classification layer of the CNN. This fully connected layer helps extract n features from the image patches in stage 2. Next, the high-resolution pre-processed images are divided into k patches according to Equation (1).

$$I \leftarrow \bigcup_{p=1}^{P} I_p \tag{3}$$

$$I_p \leftarrow \bigcup_{x,y=1}^{\sqrt{k}} I_{p,x,y} \tag{4}$$

where $I_{p,x,y}$ is the patch extracted from the high-resolution pre-processed image. The deep learning model trained in the previous step is then used to extract n features from each of the image patch. The resulting features are afterwards combined to create the feature vector F_p for each pre-processed image having $n \times k$ features. Equation (5) describes the features extracted from pre-processed images where f represents the features extracted from each patch.

$$F_p \leftarrow F_p \frown f \tag{5}$$

These collective features that contain high-resolution features from different patches of the image are used for classification by different ML classifiers. In this work, we have used RF, SVM, and MLP classifiers to check the performance of the proposed divide-and-conquer approach. The proposed approach thus allows the processing of high-resolution input images with minimal GPU memory. It should be noted that the approach is suitable for domains where features are evenly distributed across input images.

4 | Dataset Description

We have evaluated the proposed methodology using two different datasets: Diabetic Retinopathy dataset (DDR) and Asia Pacific Tele-Ophthalmology Society (APTOS) dataset.

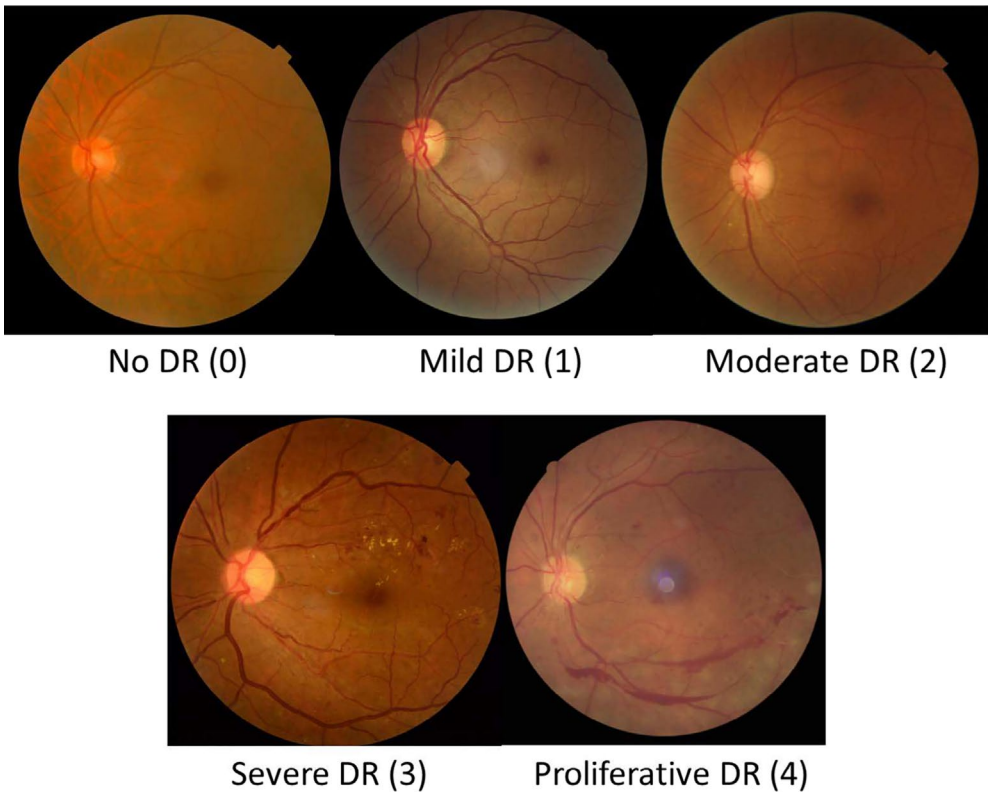

FIGURE 5 | Different severity scales of diabetic retinopathy in the DDR dataset.

The DDR dataset [20] consists of 13 673 fundus images. These images are collected from 9598 patients from 147 hospitals in China covering 23 provinces. The images are divided into five (5) categories as per DR severity scale that is, none, mild, moderate, severe, and proliferative DR as shown in Figure 5. The total number of images available for each category are shows in brackets. These fundus images are captured using 42 types of fundus cameras with a 45° FOV, using mainly Topcon D7000, Topcon TRC NW48, Nikon D5200, and Canon CR 2 cameras. The data suffers from imbalance where No DR (50%) and moderate DR (35%) classes cover 85% of the gradable images. The rest of the classes, that is, mild DR, severe DR, and proliferative DR cover 5.03%, 1.88%, and 7.29% of the total images, respectively.

In this paper, we have used this dataset for binary classification (positive DR (PDR) and negative DR (NDR)). This resulted in a balanced dataset for both DDR and APTOS datasets. The distribution of image samples per class (NDR and PDR) is shown in Table 3.

The resolution of the input images from the DDR dataset vary from1500×1100 pixels to3200×2400 pixels. The images are captured under varying conditions by different ophthalmologists and differ in size and quality. Different pre-processing methods have been proposed in recent studies for contrast and color enhancement [25, 26]. In the proposed pre-processing stage, bounding box is used to extract the eye patch area from the input data and CLAHE algorithm is applied to enhance the input fundus images. The images are then resized to a resolution of 672×672 which is three times the resolution of input images expected by GoogleNet, AlexNet, ResNet, and other CNN architectures.

TABLE 3 | Data distribution per class in DDR dataset.

Label (class)	DDR		APTOS	
NDR	No DR (6266)	6266	No DR (1805)	1805
PDR	Mild DR (630)	6256	Mild DR (370)	1857
	Moderate DR (4477)		Moderate DR (999)	
	Severe DR (236)		Severe DR (193)	
	Proliferative DR (913)		Proliferative DR (295)	

The APTOS dataset [27] (Kaggle dataset) consists of a total of 3662 retina images collected from multiple clinics under a variety of imaging conditions using fundus photography from Aravind Eye Hospital in India. Fundus images provided in this dataset are categorized into the same five (5) classes as mentioned earlier for DDR dataset. This dataset is also imbalanced. To use it for binary classification (PDR or NDR), the stages 1–4 of the severity scales are combined into PDR class whereas stage 0 categorized as NDR class. This makes the dataset have 1805 images in the NDR class and 1857 images in the PDR class. Like the DDR dataset, the resolution of fundus images in the APTOS dataset varies from 474×358 pixels to 3388×258 pixels.

5 | Results and Discussion

The hardware used in this work consists of an AMD Ryzen 2700× processor with 32 Gigabytes of memory. The Graphics

Processing Unit (GPU) installed in the system was an NVIDIA GeForce RTX 2080 with 8 GB graphics memory. MATLAB was used for running various image processing and ML algorithms. Different classifiers were used in MATLAB for binary and multi-class classification of input images.

In both datasets, classes 1–4, that is, Mild (1), Moderate (2), Severe (3), and Proliferative DR (4) are merged into a single class called "PDR." This creates a balanced dataset that was used for binary classification of the fundus images. The training data contains 80% of the dataset images and for testing purposes 20% of the dataset is used. In this work, we resize input fundus images after initial pre-processing to 672×672×3. The images are then resized by a scaling factor $k = 9$ where $k = k_w \times k_h$ and $k_w = k_h = 3$. This makes the input images reduced by a factor of nine. Experimental results are evaluated based on accuracy, precision, recall, F-1 measure and confusion matrix as explained in [28].

To select the CNN model that is used to test our divide and conquer approach, we first tested the performance of various CNN models by using the resized fundus images from the DDR dataset. For the CNN models, the learning rate is set to 0.0001 and Stochastic Gradient Descent with momentum is used as the loss function [29]. The training parameter values used for the experiments are shown in Table 4.

The results for binary classification of the DDR dataset are shown in Table 5. The DDR dataset is tested on different CNN models, that is, GoogleNet, AlexNet, MobileNet, and Inception v3. GoogleNet provides a maximum classification accuracy of 83.65%. Therefore, GoogleNet is selected as the CNN model of choice for initial training in this problem domain.

Table 6 shows the results of using Support Vector Machines for the classification of DDR image features. The features of DDR images are extracted with four different methods based on the GoogleNet model with 7.1 million parameters. In the first method, the GoogleNet CNN layers are re-trained using the training images of the DDR dataset. Once the model is trained, we extract features from the GoogleNet model using transfer learning by freezing the network layers and extracting weights of the modified classification layer of the network. The resized images are given as input to the CNN model. No patches are created for this method and scaling factor k is set to 1. These features are fed to the SVM classifier which gives an accuracy of 84.73%. The second results column in Table 6 shows the results that are observed when we use our divide and conquer approach. The scaling factor is set to 9, that is, the input of size 672×672×3 is divided into nine patches. The high-resolution features significantly improve the performance of the model by providing a classification accuracy of 95.92%. Other performance metrics of sensitivity, specificity, precision, and F1-score show a performance uplift to 96.02%, 96.02%, 95.92% and 96.02%, respectively. The base GoogleNet model used

TABLE 4 | CNN model training parameters.

Parameter	Value
Batch size	32
Epochs	32
Loss function	sgdm
Learning rate	0.0001
n (features for each patch)	111
k (number of patches)	9
$kw = kh$	3
Parameters for training	7.1 M

TABLE 6 | Experimental results of the SVM classifier for binary classification (testing done using DDR dataset).

Model: Support vector machines				
	Dataset used for training GoogleNet model			
Parameters	DDR		APTOS	
Image features	$K=1$	$K=9$	$K=1$	$K=9$
Accuracy	84.73%	95.92%	80.98%	94.60%
Recall	84.87%	96.02%	81.22%	94.88%
Specificity	84.63%	96.02%	80.82%	94.74%
Precision	84.68%	95.92%	80.87%	94.60%
F1-score	84.63%	96.02%	80.82%	94.74%
$K=1$ (no patch), $K=9$ (3×3 patch)				

TABLE 5 | Evaluation of different pre-trained CNN models for binary classification of the DDR dataset.

Binary classification of DDR dataset				
	CNN models			
Parameters	GoogleNet	AlexNet	MobileNet	Inception v3
Accuracy	83.65%	82.97%	82.93%	83.21%
Recall	85.56%	82.97%	82.93%	83.71%
Specificity	81.86%	80.00%	81.66%	80.79%
Precision	81.59%	80.00%	82.35%	83.73%
F1-score	83.53%	82.93%	82.80%	83.20%

to extract high resolution features is trained on DDR dataset and can show bias toward that dataset. The third column shows the results of using a GoogleNet CNN model that is retrained using the APTOS dataset. Transfer learning is used on this model to extract features from the DDR dataset which are used for classification using SVM classifier with simple resizing of the image without any patch. There is a decrease in performance compared to using DDR trained GoogleNet model with accuracy reducing to 80.98%. The last column in Table 6 represents the results of DDR image classification using our new approach on features extracted from a GoogleNet model that is trained on APTOS dataset. The accuracy of 94.6% compared with 84.73% and 80.98% indicates that our divide and conquer approach is not biased toward the DDR dataset.

TABLE 7 | Experimental results of RF classifier for binary classification (testing done using DDR dataset).

Model: Random forests				
	Dataset used for training GoogleNet model			
Parameters	**DDR**		**APTOS**	
Image features	***K*=1**	***K*=9**	***K*=1**	***K*=9**
Accuracy	84.53%	94.44%	80.02%	93.33%
Recall	84.59%	94.83%	80.34%	93.89%
Specificity	84.46%	94.61%	79.84%	93.52%
Precision	84.49%	94.44%	79.88%	93.32%
F1-score	84.46%	94.61%	79.84%	93.52%
$K=1$ (no patch), $K=9$ (3×3 patch)				

TABLE 8 | Experimental results of MLP classifier for binary classification (testing done using DDR dataset).

Model: Multilayer perceptron				
	Dataset used for training GoogleNet model			
Parameters	**DDR**		**APTOS**	
Image features	***K*=1**	***K*=9**	***K*=1**	***K*=9**
Accuracy	83.65%	95.12%	79.18%	94.00%
Recall	83.69%	95.12%	79.38%	94.43%
Specificity	83.71%	95.16%	79.30%	94.11%
Precision	83.65%	95.12%	79.17%	94.00%
F1-score	83.71%	95.16%	79.30%	94.11%
$K=1$ (No patch), $K=9$ (3×3 patch)				

Tables 7 and 8 show the results of using the same methods of feature extractions on two different classifiers, that is, Random Forests and MLP. The divide and conquer approach is able to extract more meaningful features from the fundus images of the eye, which results in high performance gains. The tables show again that when the APTOS dataset is used to train the GoogleNet model, the performance of ML classifiers through the divide and conquer approach decreases by a small amount. The SVM model outperforms all the other ML models with the highest values in all performance metrics.

The confusion matrix of all the models is shown in Table 9. Another important observation from the confusion matrix shows that the RF classifier, although it does not provide the best overall performance, does have the best per-class accuracy for the PDR class when the divide and conquer approach is used.

The major disadvantage of the divide and conquer approach is the extra training time used for the CNN models with an added feature layer to extract 111 features from each patch of the input fundus image and use them for training an appropriate classifier. For example, it took 874 min and 16 s to train the GoogleNet CNN model without an additional feature layer. However, it took 903 min and 54 s to train the GoogleNet model with additional features layer. Both the models were trained with 20 epochs and batch size was set to 32.

Table 10 shows the best results comparison of our proposed divide and conquer approach with other recent works. To further reduce any bias of the DDR dataset during the validation phase, we tested the proposed divide and conquer approach by using the APTOS dataset to train the best performing CNN model. The GoogleNet CNN model was retrained using the APTOS dataset, but the feature extraction and validation were done on the DDR dataset. The results show that the performance

TABLE 9 | Confusion matrix for SVM, RF, and MLP models on the DDR dataset with and without the proposed framework.

	DDR		**DDR (3×3)**		**APTOS**		**APTOS (3×3)**		
	NDR	**PDR**	**NDR**	**PDR**	**NDR**	**PDR**	**NDR**	**PDR**	
SVM	1135	155	1200	90	1106	184	1164	126	NDR
	227	985	12	1200	292	920	9	1203	PDR
RF	1121	169	1153	137	1103	187	1126	164	NDR
	218	994	2	1210	313	899	3	1209	PDR
MLP	1056	234	1214	76	972	318	1134	142	NDR
	175	1037	46	1166	203	1009	8	1217	PDR

TABLE 10 | Comparison of binary classification results of the proposed framework with recent works.

Method	Dataset	Metrics			
		Accuracy	Precision	Recall	F-Measure
Best results of divide and conquer approach (training on DDR)	DDR	95.92%	95.92%	96.02%	96.02%
Best results of divide and conquer approach (training on APTOS)	DDR	94.60%	94.60%	94.88%	94.74%
Best results of divide and conquer approach (training on APTOS)	APTOS	97.39%	97.40%	97.40%	97.40%
Long et al. [30]	APTOS	92.5%	—	—	—
Mohanty et al. [31]	APTOS	97.30%	—	—	—
Elwin et al. [21]	DDR	90.25%	—	91.42%	—
Vives-Boix and Ruiz-Fernández [32]	APTOS	95.56%	96.07%	94.46%	94.24%
Zhang et al. [33]	APTOS	91.17%	90.42%	91.71%	90.88%
Farag et al. [34]	APTOS	97.00%	—	97.00%	94.55%

reduces by a small percentage but still outperforms all the recent works.

The third result in Table 10 is evaluated to compare the results of our proposed methodology with recent works that use the APTOS dataset for validation. In this experiment, the patches and the CNN are both trained and tested on the APTOS dataset. The experimental results using the APTOS dataset also indicate that the proposed divide-andand-conquer framework provides better performance over the recent classification approaches for DR.

6 | Conclusions

Diagnosing DR from high-resolution fundus images of the eye can greatly improve the performance of DR classification. However, GPU memory limits the excessive increase in the resolution of fundus images. In this paper, we have provided a divide-and-conquer framework for high-resolution fundus images that divides the image into various patches. A CNN model trained on resized fundus images is used to extract features from each patch of the fundus image. The extracted features from each patch are concatenated and passed on to ML classifiers. This approach resulted in classifying the DR images from the DDR dataset with maximum accuracy, recall, specificity, precision, and F1-score of 95.92%, 96.02%, 96.02%, 95.92%, and 96.02%, respectively using the SVM classifier on the extracted features from fundus image patches. The proposed approach allows the processing of high-resolution input images with limited GPU memory constraints. The approach works most effectively in problem domains where features are spread almost uniformly throughout the input images. Our proposed algorithms improve performance for DR classification of fundus images. However, the feature extraction methodologies and classifications are still limited by the amount of training data available for the fundus images of the eye. In the divide and conquer approach, the main limitation lies in the extra time taken to extract features from the image patches compared to regular full-size image features extraction. This extra time taken to extract features can ultimately affect the performance of the system.

It was observed in our work that extracting high-resolution local features from fundus images provides performance improvements in the DR classification. The global features of the image also play an important role in the classification of fundus images. In future, the global features of the fundus images can also be combined along with the high-resolution local features acquired using the divide and conquer approach to build DR detection systems that may provide an even better classification of the fundus images.

Author Contributions

Ghazanfar Latif and Mohsin Butt: conceptualization. **Majid Ali Khan, Ghazanfar Latif, and Mohsin Butt:** methodology. **Ghazanfar Latif and Mohsin Butt:** software. **D.N.F. NurFatimah, Abul Bashar, and Mohsin Butt:** validation. **Ghazanfar Latif and Majid Ali Khan:** formal analysis. **Mohsin Butt and D.N.F. NurFatimah:** investigation. **Ghazanfar Latif and Majid Ali Khan:** resources. **Mohsin Butt, Majid Ali Khan, and Abul Bashar:** writing – original draft preparation. **D.N.F. NurFatimah and Ghazanfar Latif:** writing – review and editing. **Mohsin Butt and Majid Ali Khan:** visualization. **D.N.F. NurFatimah and Ghazanfar Latif:** supervision. All authors have read and agreed to the published version of the manuscript.

Acknowledgements

The authors would like to acknowledge the support of KFUPM and PMU for providing facilities to perform the research work presented in this paper. Also, they would like to thank the expert reviewers for providing constructive comments in improving the paper's quality.

Ethics Statement

The authors have nothing to report.

Consent

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in PTOS 2019 Blindness Detection at https://www.kaggle.com/competitions/aptos2019-blindness-detection.

References

1. V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa, and M. A. Riegler, "Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images," *Diagnostics* 11, no. 12 (2021): 2183, https://doi.org/10.3390/DIAGNOSTICS11122183.

2. C. F. Sabottke and B. M. Spieler, "The Effect of Image Resolution on Deep Learning in Radiography," *Radiology: Artificial Intelligence* 2, no. 1 (2020): e190015, https://doi.org/10.1148/ryai.2019190015.

3. W. Li, Z. Wang, Y. Wang, et al., "Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020): 1986–1995, https://doi.org/10.1109/JSTARS.2020.2988477.

4. A. Ikram, A. Imran, J. Li, et al., "A Systematic Review on Fundus Image-Based Diabetic Retinopathy Detection and Grading: Current Status and Future Directions," *IEEE Access* 12 (2024): 96273–96303, https://doi.org/10.1109/ACCESS.2024.3427394.

5. D. Bhulakshmi and D. S. Rajput, "A Systematic Review on Diabetic Retinopathy Detection and Classification Based on Deep Learning Techniques Using Fundus Images," *PeerJ Computer Science* 10 (2024): 1947.

6. P. Saranya and K. Umamaheswari, "Detection of Exudates From Retinal Images for Non-Proliferative Diabetic Retinopathy Detection Using Deep Learning Model," *Multimedia Tools and Applications* 83, no. 17 (2024): 52253–52273.

7. N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, "Local Features and Global Shape Information in Object Classification by Deep Convolutional Neural Networks," *Vision Research* 172 (2020): 46–61, https://doi.org/10.1016/J.VISRES.2020.04.003.

8. Y. Zheng, J. Huang, T. Chen, Y. Ou, and W. Zhou, "CNN Classification Based on Global and Local Features," *Real-Time Image Processing and Deep Learning* 10996 (2019): 96–108.

9. A. Gholami, A. Azad, P. Jin, K. Keutzer, and A. Buluç, "Integrated Model, Batch and Domain Parallelism in Training Neural Networks," *Annual ACM Symposium on Parallelism in Algorithms and Architectures* (2017): 77–86, https://doi.org/10.1145/3210377.3210394.

10. T. Ben-Nun and T. Hoefler, "Demystifying Parallel and Distributed Deep Learning: An in-Depth Concurrency Analysis," *ACM Computing Surveys* 52, no. 4 (2019): 1–43, https://doi.org/10.1145/3320060.

11. A. Bakhtiarnia, Q. Zhang, and A. Iosifidis, "Efficient High-Resolution Deep Learning: A Survey," 2022, arXiv preprint arXiv:2207.13050.

12. I. Kandel and M. Castelli, "The Effect of Batch Size on the Generalizability of the Convolutional Neural Networks on a Histopathology Dataset," *ICT Express* 6, no. 4 (2020): 312–315.

13. L. Wang, L. Ding, Z. Liu, et al., "Automated Identification of Malignancy in Whole-Slide Pathological Images: Identification of Eyelid Malignant Melanoma in Gigapixel Pathological Slides Using Deep Learning," *British Journal of Ophthalmology* 104, no. 3 (2020): 318–323.

14. C. Xie, H. Muhammad, C. M. Vanderbilt, et al., "Beyond Classification: Whole Slide Tissue Histopathology Analysis by End-To-End Part Learning," in *Medical Imaging With Deep Learning* (PMLR, 2020), 843–856.

15. S. Cheng, S. Liu, J. Yu, et al., "Robust Whole Slide Image Analysis for Cervical Cancer Screening Using Deep Learning," *Nature Communications* 12, no. 1 (2021): 5639.

16. H. Lin, H. Chen, S. Graham, Q. Dou, N. Rajpoot, and P.-A. Heng, "Fast Scannet: Fast and Dense Analysis of Multi-Gigapixel Whole-Slide Images for Cancer Metastasis Detection," *IEEE Transactions on Medical Imaging* 38, no. 8 (2019): 1948–1958.

17. W. L. Alyoubi, M. F. Abulkhair, and W. M. Shalash, "Diabetic Retinopathy Fundus Image Classification and Lesions Localization System Using Deep Learning," *Sensors* 21, no. 11 (2021): 3704.

18. M. Saini and S. Susan, "Diabetic Retinopathy Screening Using Deep Learning for Multi-Class Imbalanced Datasets," *Computers in Biology and Medicine* 149 (2022): 105989.

19. G. T. Zago, R. V. Andreão, B. Dorizzi, and E. O. T. Salles, "Diabetic Retinopathy Detection Using Red Lesion Localization and Convolutional Neural Networks," *Computers in Biology and Medicine* 116 (2020): 103537.

20. T. Li, Y. Gao, K. Wang, S. Guo, H. Liu, and H. Kang, "Diagnostic Assessment of Deep Learning Algorithms for Diabetic Retinopathy Screening," *Information Sciences* 501 (2019): 511–522.

21. J. G. R. Elwin, J. Mandala, B. Maram, and R. R. Kumar, "Ar-Hgso: Autoregressive-Henry Gas Sailfish Optimization Enabled Deep Learning Model for Diabetic Retinopathy Detection and Severity Level Classification," *Biomedical Signal Processing and Control* 77 (2022): 103712.

22. Z. Gu, Y. Li, Z. Wang, et al., "Classification of Diabetic Retinopathy Severity in Fundus Images Using the Vision Transformer and Residual Attention," *Computational Intelligence and Neuroscience* 2023 (2023): 1–12.

23. N. Tsiknakis, D. Theodoropoulos, G. Manikis, et al., "Deep Learning for Diabetic Retinopathy Detection and Classification Based on Fundus Images: A Review," *Computers in Biology and Medicine* 135 (2021): 104599.

24. C. Szegedy, W. Liu, Y. Jia, et al., "Going Deeper with Convolutions," 2014.

25. D. Vijayalakshmi and M. K. Nath, "A Systematic Approach for Enhancement of Homogeneous Background Images Using Structural Information," *Graphical Models* 130 (2023): 101206, https://doi.org/10.1016/j.gmod.2023.101206.

26. D. Vijayalakshmi and M. K. Nath, "A Strategic Approach Towards Contrast Enhancement by Two-Dimensional Histogram Equalization Based on Total Variational Decomposition," *Multimedia Tools and Applications* 82, no. 13 (2023): 19247–19274.

27. S. D. Karthik, "Maggie: APTOS 2019 Blindness Detection. Kaggle," 2019, https://kaggle.com/competitions/aptos2019-blindness-detection.

28. T. Anbalagan, M. K. Nath, D. Vijayalakshmi, and A. Anbalagan, "Analysis of Various Techniques for Ecg Signal in Healthcare, Past, Present, and Future," *Biomedical Engineering Advances* 6 (2023): 100089, https://doi.org/10.1016/j.bea.2023.100089.

29. M. Yaqub, J. Feng, M. S. Zia, et al., "State-of-The-Art Cnn Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images," *Brain Sciences* 10, no. 7 (2020): 427.

30. F. Long, H. Xiong, and J. Sang, "A Classification Method for Diabetic Retinopathy Based on Self-Supervised Learning," in *Advanced Intelligent Computing in Bioinformatics*, ed. D.-S. Huang, Q. Zhang, and J. Guo (Springer, 2024), 347–357.

31. C. Mohanty, S. Mahapatra, B. Acharya, et al., "Using Deep Learning Architectures for Detection and Classification of Diabetic Retinopathy," *Sensors* 23, no. 12 (2023): 5726.

32. V. Vives-Boix and D. Ruiz-Fernández, "Diabetic Retinopathy Detection Through Convolutional Neural Networks With Synaptic

Metaplasticity," *Computer Methods and Programs in Biomedicine* 206 (2021): 106094.

33. C. Zhang, T. Lei, and P. Chen, "Diabetic Retinopathy Grading by a Source-Free Transfer Learning Approach," *Biomedical Signal Processing and Control* 73 (2022): 103423.

34. M. M. Farag, M. Fouad, and A. T. Abdel-Hamid, "Automatic Severity Classification of Diabetic Retinopathy Based on Densenet and Convolutional Block Attention Module," *IEEE Access* 10 (2022): 38299–38308.