Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Two-stage deep learning classification for diabetic retinopathy using gradient weighted class activation mapping

Abderaouf M. Moustari, Youcef Brik, Bilal Attallah & Rafik Bouaouina

To cite this article: Abderaouf M. Moustari, Youcef Brik, Bilal Attallah & Rafik Bouaouina (2024) Two-stage deep learning classification for diabetic retinopathy using gradient weighted class activation mapping, Automatika, 65:3, 1284-1299, DOI: 10.1080/00051144.2024.2363692

To link to this article: https://doi.org/10.1080/00051144.2024.2363692

AUTOMATIKA
2024, VOL. 65, NO. 3, 1284–1299
https://doi.org/10.1080/00051144.2024.2363692

OPEN ACCESS Check for updates

Two-stage deep learning classification for diabetic retinopathy using gradient weighted class activation mapping

Abderaouf M. Moustari[a], Youcef Brik [a], Bilal Attallah[a] and Rafik Bouaouina[b]

[a]LASS Laboratory, Faculty of Technology, University Mohamed Boudiaf of M'sila, M'sila, Algeria; [b]PIMIS Laboratory, Electronics and Telecommunications Department, University 08 Mai 1954 of Guelma, Guelma, Algeria

ABSTRACT

The fundus images of patients with Diabetic Retinopathy (DR) often display numerous lesions scattered across the retina. Current methods typically utilize the entire image for network learning, which has limitations since DR abnormalities are usually localized. Training Convolutional Neural Networks (CNNs) on global images can be challenging due to excessive noise. Therefore, it's crucial to enhance the visibility of important regions and focus the recognition system on them to improve accuracy. This study investigates the task of classifying the severity of diabetic retinopathy in eye fundus images by employing appropriate preprocessing techniques to enhance image quality. We propose a novel two-branch attention-guided convolutional neural network (AG-CNN) with initial image preprocessing to address these issues. The AG-CNN initially establishes overall attention to the entire image with the global branch and then incorporates a local branch to compensate for any lost discriminative cues. We conduct extensive experiments using the APTOS 2019 DR dataset. Our baseline model, DenseNet-121, achieves average accuracy/AUC values of 0.9746/0.995, respectively. Upon integrating the local branch, the AG-CNN improves the average accuracy/AUC to 0.9848/0.998, representing a significant advancement in state-of-the-art performance within the field.

ARTICLE HISTORY
Received 14 February 2024
Accepted 28 May 2024

KEYWORDS
Gradient weighted class activation mapping; deep learning; diabetic retinopathy classification; two-stage system; image preprocessing; region of interest extration

1. Introduction

Diabetes or diabetes mellitus (DM) is a worldwide issue of pandemic scale. As per the World Health Organization's projections, the number of individuals worldwide with diabetes is estimated to reach 360 million by 2030 [1]. DM is now more than just a problem in developed, prosperous countries. Due to nutritional changes, obesity, and a lack of physical activity, the disease has spread to developing countries. Everyone with DM runs the risk of developing diabetic retinopathy (DR).

A fundus examination is now the most accurate procedure for DR diagnosis. Microaneurysms, hemorrhages, and exudates are manually identified by trained ophthalmologists, which is labour-intensive and subject to inter-observer variation. An autonomous severity classification system could substantially impact the treatment of vision disorders connected to DR by improving the speed and reproducibility of DR diagnoses.

In the last ten years, machine and deep learning, in particular, have proven their ability to characterize DR severity properly. With the swift expansion of data and computing resource capacity, along with the accessibility of Datasets for classifying Diabetic Retinopathy in fundus images, like APTOS 2019 [2], solutions based on convolutional neural networks (CNNs) are growing in popularity. Using already existing large-scale CNN architectures, transfer learning is used to build the most effective CNN models for Retina diseases [3,4], but this requires a lot of training data and computing resources.

Our approach employs several preprocessing methods to enhance the detection of the presence of issues in the fundus images, as well as Grad-Cam, an algorithm that utilizes Class Activation Map (CAM) [5] and performs multiple evaluations using deep neural network models to ensure precise outcomes and effectively address the concerns above. This technique also makes it possible to create high-resolution graphics with enhanced class discrimination capabilities. Enhancing the comprehension of deep model representations and improving the clarity of classification systems pave the way for leveraging these capabilities to reinforce the model's predictions.

The remaining sections of this paper are structured as follows, Prior works in this field are summarized in Section 2. The data used for this research is presented in Section 3, and the proposed techniques are presented in Section 4. The experimental part of the proposed system is presented in Section 5. Finally, Section 5 draws a conclusion to wrap up the paper.

CONTACT Youcef Brik youcef.brik@univ-msila.dz B.P. 166, M'sila 28000, Algeria

2. Related works

The global image is commonly used for training in prior studies on DR classification. For instance, Khalifa et al. presented deep transfer learning using the models AlexNet, SqueezeNet, Res-Net18, GoogleNet, VGG16, and VGG19 [6]. Using a modified VGG16 model, Dekhil et al. developed a transfer learning strategy [7]. A custom Xception architecture was proposed by Kassani et al. [8]. Several unique pre-trained models were suggested by Tymchenko et al. [9]. Another automatic recognition of the DR was proposed by Qureshi et al. [10] proposed based on a new multi-layer architecture of active deep learning (ADL-CNN). The authors selected the most informative regions-of-interest within the retinograph image to grade five severity-levels of diabetic retinopathy using. The experiments were conducted on EyePACS dataset and the results outperformed the state-of-the-art methods.

In research conducted by Zeng et al. [11], DR classification was performed using a deep convolutional neural network based on a Siamese-like structure. The suggested architecture takes as inputs two individual fundus images that each represent one eye. Zeng's approach has an area under the ROC curve (AUC) of 95.10% and a sensitivity of 82.2%. Shanthi and Sabeenian [12] used a modified AlexNet architecture to identify DR fundus pictures on the MESSIDOR dataset [13] by adding the right pooling, softmax, and ReLU layers. This suggested model has a 96.25% average accuracy in the MESSIDOR dataset. Besides, a computer-aided diagnosis system for diabetic retinopathy (CAD-DR) was proposed to recognize the five stages of DR through retinal fundus images [14]. A preprocessing step was performed in a perceptual-oriented colour space to enhance the DR-related lesions and then an enhanced pre-trained CNN model was involved to get high classification results.

In a separate research, Jain et al. [15] assessed the performance of several pre-trained networks (VGG16, VGG19, and InceptionV3) for binary and 5-class DR classification. To rectify the significant class skewness, several data augmentation methods were applied. According to Jain's study, the number of layers directly affects how accurate a model is. With a score of 80.40%, VGG19 had the best accuracy. Hagos and Kant [16] used an Inception V3 network and a limited dataset transfer learning method to create another DR classification model. Hagos' method used an SGD optimizer and the cosine loss function to obtain 90.9% accuracy.

Quellec et al. [17] employed a backpropagation generalization and a weakly-supervised model to evaluate referable lesion locations in DR images. Utilizing Kaggle and E-Ophtha datasets, this approach has an Area Under Curve metric of 95.5% and 94.9%, respectively. Glaucoma was evaluated by Gargeya and Leng [18], who utilized automatic feature learning. Datasets from MESSIDOR 2 and E-Ophtha were used to test the model. The 5-fold cross-validation AUC of Gargeya's approach was 97.00%. Shakibania et al. [19] employed two pre-trained deep learning models for the detection and stage grading of diabetic retinopathy using a single fundus retinal image. The proposed model is trained on a large multi-centre dataset, including the APTOS 2019 dataset, obtained from publicly available sources.

Wang et al. [20] proposed Zoom-in-Net to classify DR. The Zoom-in-Net uses an attention mechanism to generate suspicious spots. A separate bilinear learning technique [21] with an attention mechanism was also utilized for DR classification. The writers used a strategy to highlight the most critical elements while downplaying the less important ones. Madarapu et al. [22] have proposed a novel deep integrative approach for DR classification, leveraging the strengths of residual blocks, channel-spatial attention mechanism (CSAM), and non-local blocks (NLB).

Narayanan et al. [23] and Majumder et al. [24] have proposed a two-stage system, calling it a hybrid system; the first stage is for just detection and the second one or grading the severeness of DR in APTOS 2019 DR dataset. While Shaban et al. [25] and Gangwar et al. [26] used a modified vgg-19 and inception ResNet v-2 models, respectively, to classify DR., Ramchandre et al. [27] used the AUGMIX, a data augmentation method to expand the APTOS 2019 dataset much.

Both Md. Nahiduzzaman et al. [28] and Md. Islam et al. [29] utilized CLAHE preprocessing technique to build their DR severity classification system based on APTOS 2019 datasets, Md. Nahiduzzaman et al. used a parallel CNN network for feature extraction followed by an Extreme Learning Machine (ELM) classifier, while Md. Robiul Islam et al. applied supervised contrastive learning to build his system. Jian et al. [30] used a "Triple-DRNet" which is a model that subdivides the classification of five types of DR.

3. Data description

The data used in this work is a collection of eye fundus images obtained using fundus imaging. This process uses reflected light to get a two-dimensional (2D) representation of semi-transparent retinal tissues in 3D that is projected onto the image plane. We use in this work the APTOS 2019 DR Kaggle dataset to develop and validate the proposed system. The APTOS 2019 DR Kaggle benchmark dataset is a part of the blindness detection challenge of the same name, is a sizable dataset of retinal photos obtained with a fundus camera under various imaging circumstances. To identify various DR severity levels, the images undergo manual grading on a scale from 0 to 4, where 0 represents no DR, 1 for mild, 2 for moderate, 3 for severe, and 4 for proliferative DR, as shown in Figure 1. The distribution of the classes in the APTOS 2019 DR dataset is shown in Table 1,

Table 1. Distribution of samples in the APTOS 2019 dataset.

Scale	Severity level	# Samples
0	No DR	1805
1	Mild DR	370
2	Moderate DR	999
3	Severe DR	193
4	Proliferative DR	295
	Total	3662

while Figure 1 illustrates some DR images extracted from different APTOS 2019 classes.

4. Methods

4.1. Densenet-121 model

Table 2. Densenet-121 summary.

Layer	DenseNet-121
Convolution	7 × 7 conv, stride 3
Pooling	3 × 3 max pool, stride 2
Dense Block (1)	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times 6$
Transition Layer (1)	1 × 1 conv 2 × 2 average pool
Dense Block (2)	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times 12$
Transition Layer (2)	1 × 1 conv 2 × 2 average pool
Dense Block (3)	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times 12$
Transition Layer (3)	1 × 1 conv 2 × 2 average pool
Dense Block (4)	$\begin{pmatrix}1\times1\ conv\\3\times3\ conv\end{pmatrix}\times 12$
Classification Layer	7 × 7 global average pool 14-D fully-connected, elementwise sigmoid

DenseNet [31] is a convolutional neural network that uses dense connections between layers. These connections are made through Dense Blocks, which connect all layers directly if their feature-map sizes are the same. To keep the feed-forward nature, each layer gets more information from all the layers before it and sends its feature maps to all the layers afterwards; this architecture is displayed in Figure 2, while its structure with all the layers is presented in Table 2.

4.2. Laplacian of Gaussian (LoG)

The second spatial derivative of an image's Laplacian is a 2-D isotropic measure. An image's Laplacian

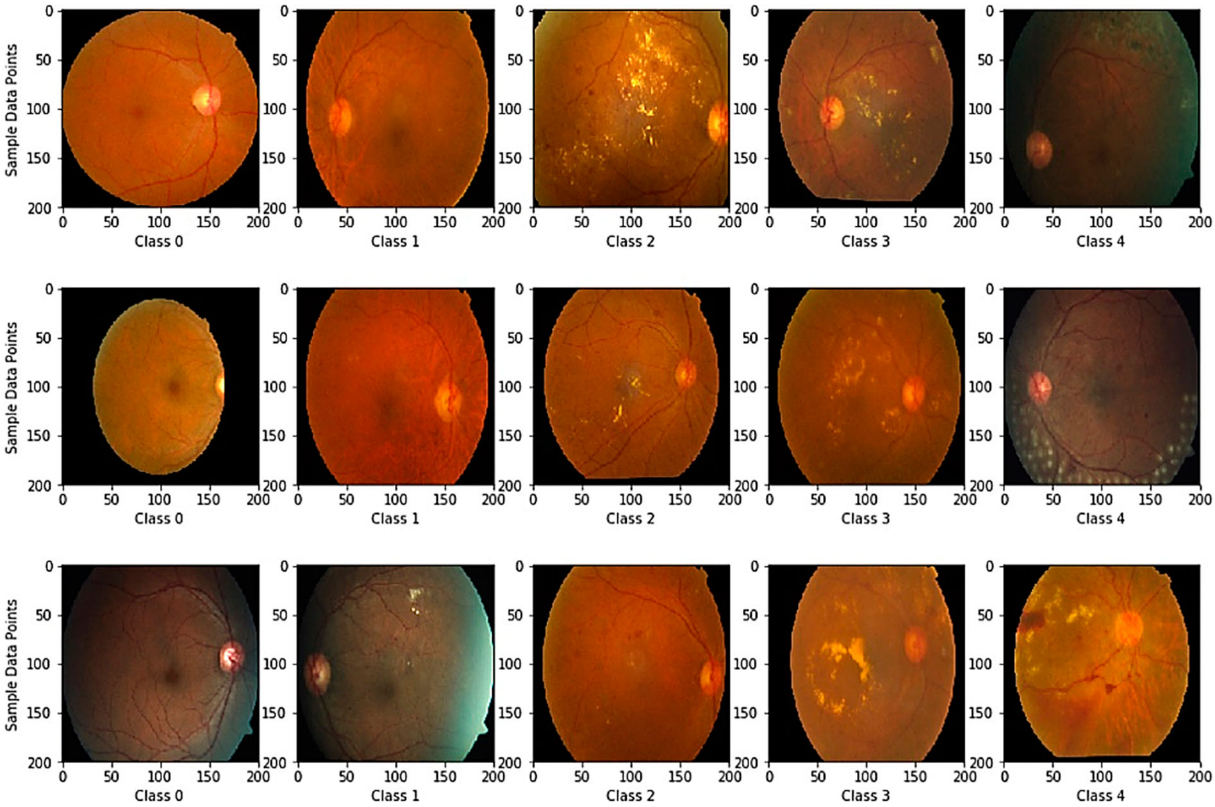


Figure 1. Samples from the APTOS 2019 DR dataset.

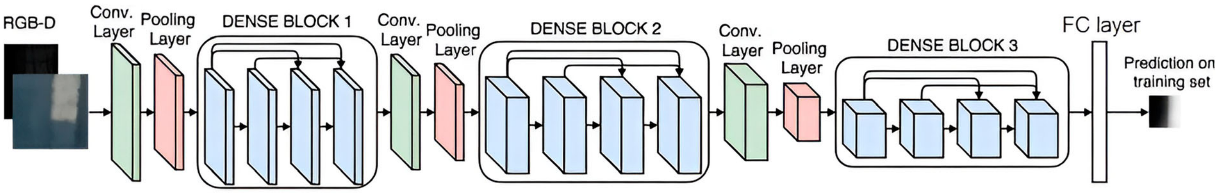


Figure 2. Illustration of the Densenet-121 architecture.

highlights the areas with sudden changes in intensity, making it a popular tool for edge detection. To lessen the sensitivity of an image to noise, the Laplacian is frequently applied after an image has been first smoothed with a method similar to a Gaussian Smoothing filter [32]. After using a Laplacian or Gaussian filter to $f(x, y)$, this detector searches for zero crossings to identify edges. This technique divides the image where the intensity varies to detect effectively the borders. It combines Gaussian filtering and Laplacian. It determines the boundaries and tests more extensive area before creating another image with a grey level as the output. The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is given by:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{1}$$

The typical small kernel is displayed in Figure 3, while Figure 4 shows the input and output of the LoG technique. The 2-D LoG function with a Gaussian standard deviation and a zero centre takes the following form:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2}$$

We note here that we used a sigma equal to 3 in the Laplacian of Gaussian filtering.

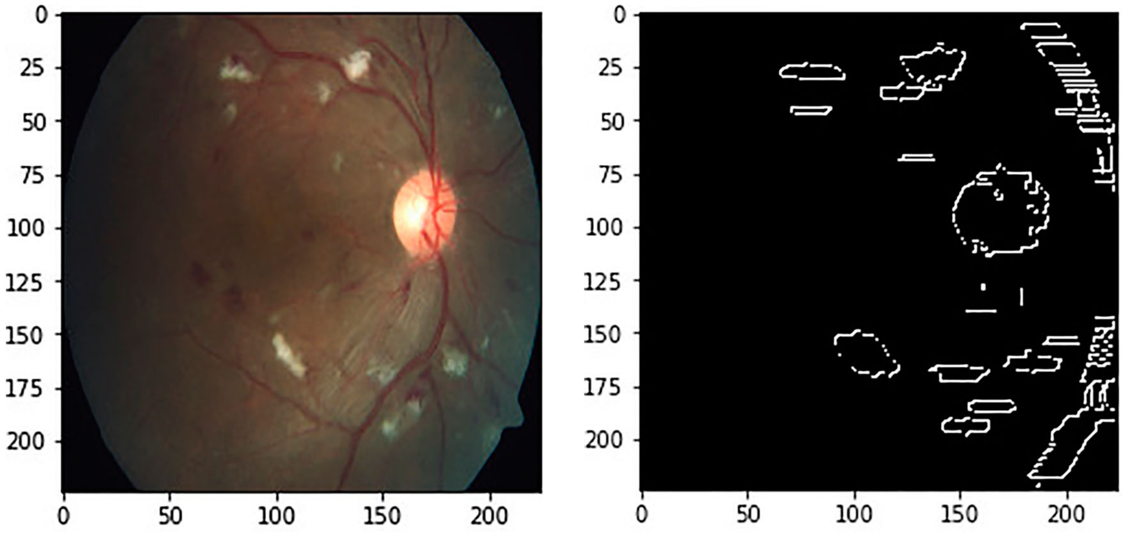
- 1	2	- 1
2	- 4	2
- 1	2	- 1

Figure 3. Standard LoG kernel.

4.3. Contrast limited adaptive histogram equalization (CLAHE)

An image histogram is a representation of the image intensity value. The primary purpose of the histogram is to provide an image's statistical data. This explains why we can adjust the histogram to perform image improvement. Histogram equalization is a widely used technique for image enhancement since it is straightforward and computationally light. We improve the colour fundus picture using Contrast Limited Adaptive Histogram Equalization (CLAHE). This preprocessing technique is commonly utilized in ophthalmology to enhance the visibility of the issues present. One essential aspect of a fundus image pertains to the contrast of blood vessels. Image contrast results from combining a range of intensity levels and distinguishing between the full spectrum of pixel values from the highest to the lowest. The application of histogram manipulation for image enhancement aims to achieve a uniform distribution of intensity. In low-contrast images, the effective intensity range is limited. However, histogram equalization expands the intensity distribution, effectively altering the original image's intensity. An example of the input and output of the CLAHE method is shown in Figure 5.

Figure 6 depicts the flowchart for the fundus image enhancement method suggested in this study. As an input image, we employ an RGB colour retinal image. The colour picture channel is split in the first step, leaving us with three images: R, G, and B image channels. Only the G channel is used in the picture enhancement procedure utilizing CLAHE because it contains more significant structural information about blood vessels than the other channels. The improved G channel image follows. The subsequent step combines the three image channels (R, enhanced G, and R). After



(a)

(b)

Figure 4. Applying LoG filter: (a) original image and (b) filtered image.

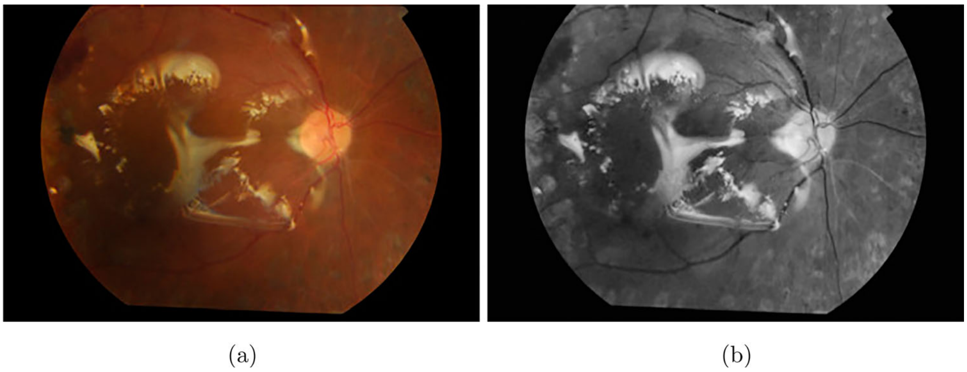


Figure 5. The effect of the CLAHE method: (a) input, (b) output.

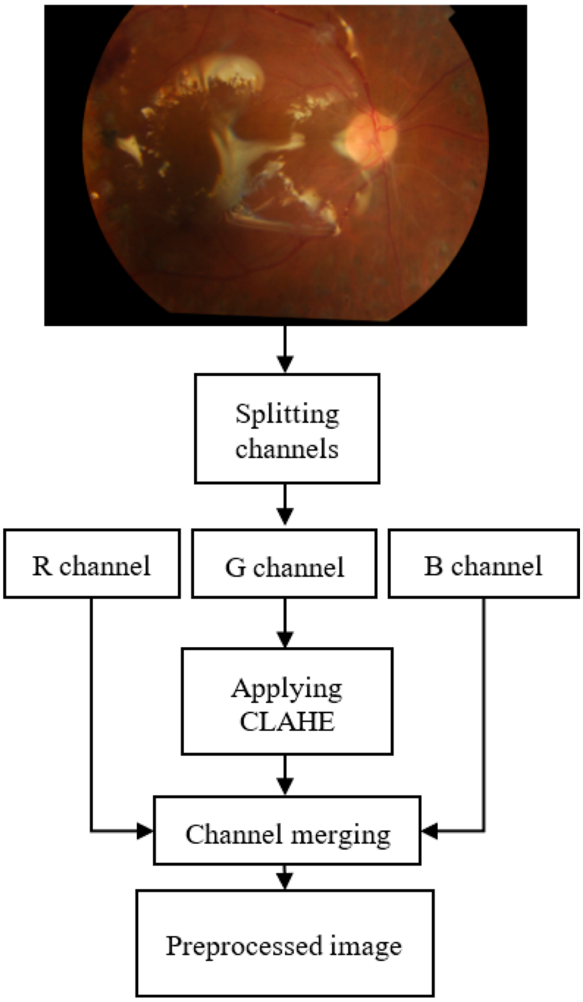


Figure 6. Flowchart of the CLAHE technique.

this procedure, we have an improved colour retinal image.

4.4. Grad-CAM

Selvaraju et al. [5] developed a method called Gradient Weighted Class Activation Mapping (Grad-CAM), which offers a clear understanding of deep learning models. Any highly linked neural network may be graphically described using Grad-CAM, further facilitating model understanding during the prediction phase. Illustrated in Figure 7, Grad-CAM utilizes an input image to make predictions using the proposed model. The model generates the output class based on its forecast, and any one of the Convolution layers is then subjected to the Grad-CAM algorithm. The last Conv layer is often utilized for Grad-CAM. Figure 8 show some examples of the results of the Grad-CAM algorithm, The steps of the Grad-CAM algorithm are:

- Acquire the final result of the network after completing the convolution phase.
- Calculate the gradient of the score for class c, denoted as Y^c, with respect to A^k, which represents the activation maps of a conv layer.

$$\frac{\partial Y^c}{\partial A^k} \tag{3}$$

- By utilizing the global average pooling technique, emphasize the importance of each activation map k for specific classes:

$$\alpha_c^k = 1/Z \sum_i \sum_j \frac{\partial Y^c}{\partial (A_i)_j^k} \tag{4}$$

  Where Z is the number of pixels in the activation map In this context, the “partial differentials” correspond to the gradients computed during the backpropagation process, while the “summation over *I* and *j*” represents the global average pooling operation.
- Generate a heatmap by merging activation maps with weights and applying the ReLU function to the resulting values.

$$L^c_{\text{Grad-CAM}} = \text{RELU} \sum_k \alpha_c^k A^k \tag{5}$$

  ReLU is the preferred activation function because it accentuates characteristics that are beneficial for the specific class being considered. ROIs are identified as pixels whose intensity are dependent on the gradient Yc. Without ReLU, localization maps might contain non-pos pixels associated with other image classes, negatively impacting localization accuracy.
- Normalize heatmap (optional but advised step since it enhances outcomes).

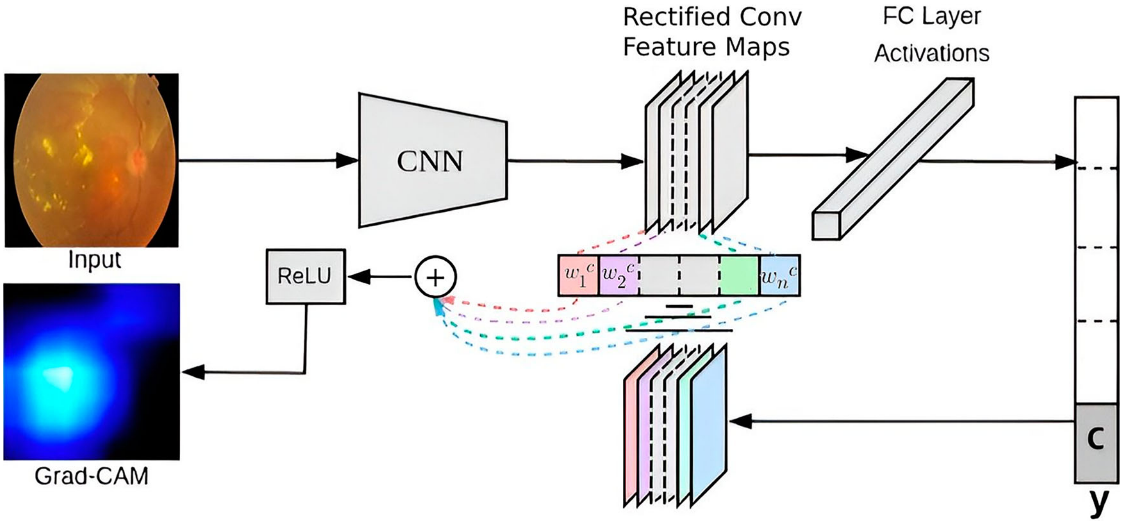


Figure 7. Explanatory diagram of the Gradient Weighted Class Activation Mapping (Grad-CAM).

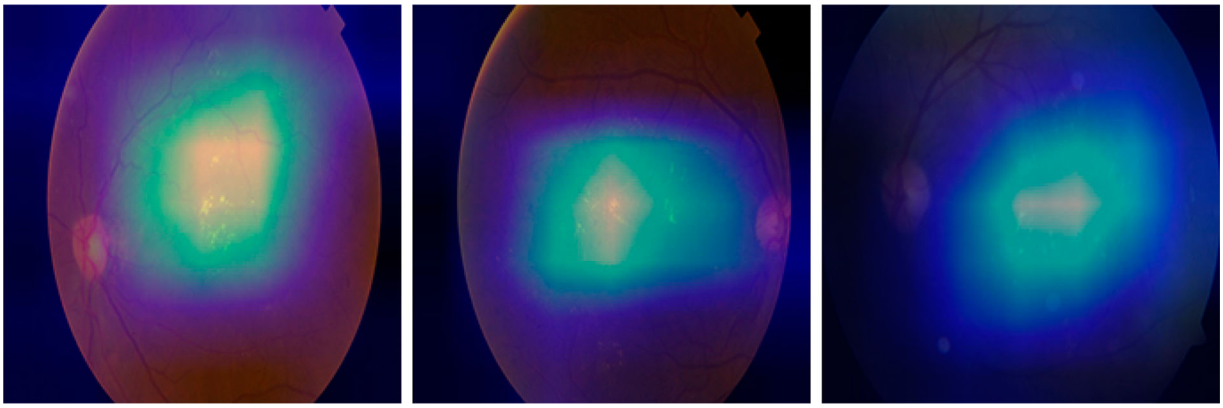


Figure 8. Extracted heatmaps superposed on their corresponding input images.

5. Proposed system

In this section, we describe the proposed Attention-guided system designed for classifying eye fundus images based on the DenseNet-121 model as a backbone. The overall attention-guided structure is explained in Section 5.2 and illustrated in Figure 9. The mask inference method for discriminative area detection is covered in Section 5.3. The approach involves two main stages: global and local. The input images are run through a preprocessing step and then are fed to dense convolutional blocks followed by a 1x1 convolution block and a pooling layer; these two are transition blocks. At the end, the blocks are connected to a max-pooling layer, an fully connected (FC) layer, and a sigmoid layer. Contrary to the global stage, the local stage's input is local images cropped by the mask generated from the global stage. The input image is overlaid over the heat map for visualization purposes.

5.1. *Multilabel setup*

In this classification model, we treat the problem as a multilabel scenario. Each image is assigned a 5-bit class vector representing different severity levels $L = [l1, l2, l3, l4, l5]$. Here, $l \in \{0, 1\}$, where 1 indicates that the severity level has been reached, and 0 suggests it hasn't. Therefore, if a sample belongs to a specific class (e.g. class 3), it is also considered to belong to all the classes below it (0, 1, and 2). This approach is grounded in the notion that once an eye attains severity levels 1 and 2, it will inevitably advance to level 3.

5.2. *Global and local stages*

The global input image gives the global stage access to pertinent information. The DenseNet-121 served as the primary global model in this study. It comprises three dense blocks and three transition layers, followed by a global average pooling layer and an FC layer for classification. Finally, the output vector $Z_g(I^c)$ is normalized by applying a sigmoid layer using:

$$\widetilde{p_g}(c|I) = \frac{1}{(1 + e^{-p(c|I)}} \tag{6}$$

Where I is the global image, $\widetilde{p_g}(c|I)$ indicates the probability score of I fitting to the c^th class, $c \in \{1, 2, \ldots C\}$ and C represents the total number of classes. Conversely, the local branch focuses on the relevant area and is expected to mitigate the disadvantages of only

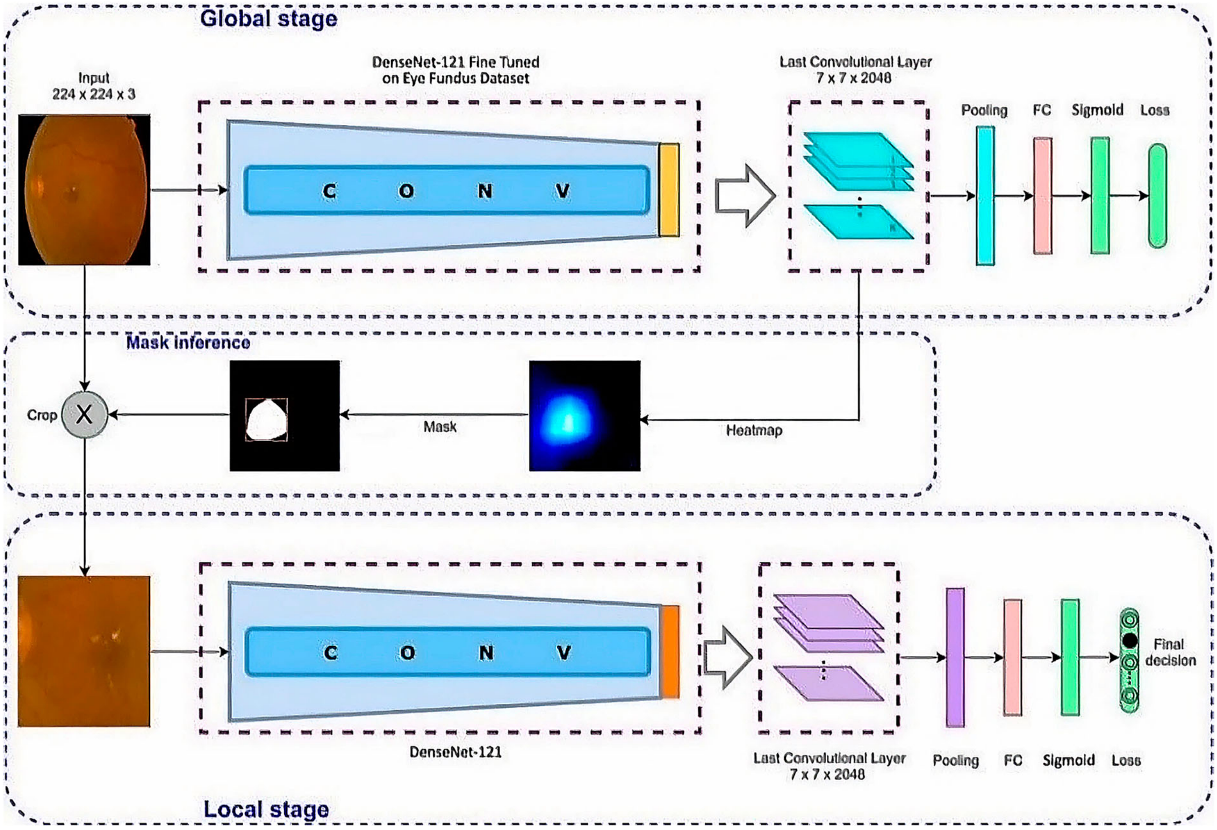


Figure 9. Proposed system structure.

using the global image. More specifically, the local branch shares the same convolutional network structure as the global branch. But do not share weights, as they serve different purposes. The probability score of a local branch is denoted by $\widetilde{p_l}(c|I_c)$ where I_c is the input image of the local branch. We perform the same normalization and optimization as the global branch.

5.3. Mask inference and ROIs union

In order to isolate the region of interest (ROI) in the global DR images, we build a binary mask by performing thresholding operations on feature maps (ROI heatmap). These heatmaps are extracted using the Grad-CAM technique.

After that, the heatmap is binarized according to a threshold called τ. We created a bounding box around it, then we cropped it out of the original image (see Figure 10). For some DR images where more than one region of interest are created, we apply a union of these ROIs and the resulted bounding box will be cropped from the original image. Figure 11 shows the principle of this process.

5.4. Training strategy of AG-CNN

This paper adopts a two-stage training scheme for AG-CNN:

Stage I: The global branch network pre-trained by ImageNet is finetuned using the global images. Equation (1) is used to normalize the resulting vector; after this stage, the weights of the global branch are fixed.

Stage II: Local images are obtained using mask inference with threshold τ, then we feed it to the local branch for finetuning and normalizing using Equation (1), and we get our final classification results at the output of this branch.

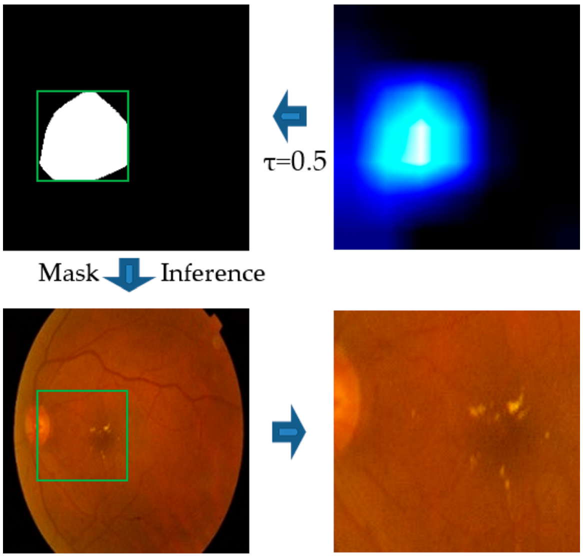


Figure 10. ROI cropping process using mask inference.

In each stage, we use the model with the highest accuracy/AUC on the validation set for testing. The overall AG-CNN training procedure is presented in Algorithm

Table 3. Evaluation of Multiple Models trained using the APTOS 2019 Dataset.

Model	Accuracy %	AUC %	F1 %	Val loss %	Precision %	Recall %
Densenet-121	96.18	98.99	95.32	0.112	96.90	93.85
Mobilenet v2	89.31	89.86	85.22	1.424	99.64	74.57
Resnet50	96.07	98.67	95.53	0.131	97.69	93.53
Inception V3	95.71	98.39	94.81	0.154	96.53	93.22
VGG16	95.85	99.13	95.02	0.110	94.68	95.42

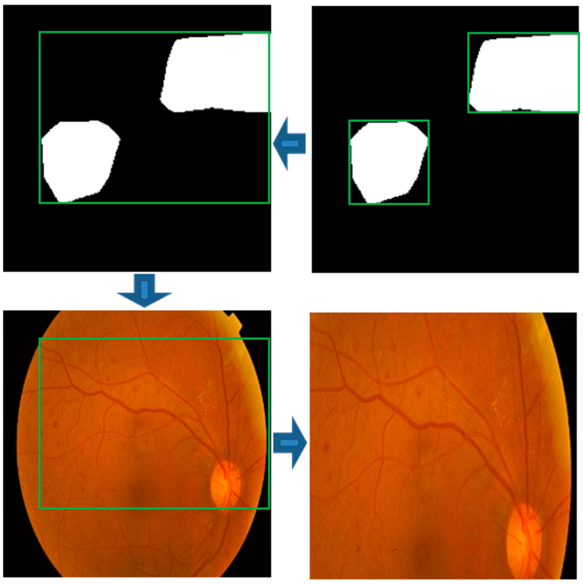


Figure 11. Bounding boxes union in case of multiple ROIs.

1. Variants of training strategy may influence the performance of AG-CNN. We discussed it in the next section.

6. Results and discussion

To investigate the applicability of the proposed approach, we evaluated its performance using the APTOS 2019 diabetic retinopathy dataset. Series of experiments were conducted to evaluate our approach with and without data augmentation. Besides, several preprocessing techniques were considered in order to enhance the DR image quality. For performance assessment, we used a set of measures namely the accuracy rate, the area under ROC curve (AUC), the recall rate, the precision rate and the f1-score (F1).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$\text{F1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

In order to justify the choice of the DenseNet architecture over others, we conducted an experiment with well-known deep learning models trained and validated on the APTOS 2019 DR dataset without preprocessing. The results of this experiment are displayed in Table 3. Upon analyzing these results, we notice that the DenseNet-121 model outperforms other models on this dataset. As a result, we select it as the backbone for the rest of our experiments.

6.1. Protocols and setups

During all experiments, the evaluation protocol of the APTOS 2019 database takes 85% of the dataset for training and 15% as the validation set. Experiments with and without data augmentation and different preprocessing strategies are established on the dataset classification. We have merged multiple transformation steps to change the original images and produce other variations in this work. The fundamental transformations applied to each training image in the gallery are:

- Resize the image to fit the model input (224 × 224 × 3).
- The random image zooms up to 15%.
- Flip the image vertically and horizontally at random.

6.2. Experiments

To investigate the applicability of the proposed approach, we evaluated its performance using the APTOS 2019 diabetic retinopathy dataset. We start by justifying the choice of the DenseNet architecture over others, where we conducted an experiment using the most used CNN models without preprocessing. The results of this experiment are displayed in Table 3. Upon analyzing these results, we determine that the DenseNet-121 model outperforms other models on this dataset. As a result, the DenseNet-121 was chosen in this work as the backbone CNN model to explore if Attention-guided CNN can improve the system performance.

After that, we perform a basic data augmentation for training by scaling the original images to 224 × 224, random flipping (horizontal and vertical) and zooming with a range of 15%. The batch size of 64 has been chosen for network optimization. Each branch receives 25 epochs of training. The learning rate is 0.0001 at first. We employ a weight decay of 50%. The same process is used during validation and test. We build AG-CNN with the TensorFlow Keras framework.

Table 4. Evaluation metrics for the scenario 1 experiment.

		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet-121	96.18%	98.99%	95.32%	0.112	96.90%	93.85%
Local (Cropping)	$\tau = 0.1$	96.44%	99.38%	95.66%	0.097	96.35%	95.03%
	$\tau = 0.2$	96.25%	98.85%	95.47%	0.127	97.26%	93.78%
	$\tau = 0.3$	95.92%	98.92%	95.03%	0.131	96.77%	93.40%
	$\tau = 0.4$	96.18%	99.08%	95.37%	0.115	96.00%	94.80%
	$\tau = 0.5$	95.96%	98.77%	95.11%	0.133	96.79%	93.55%
	$\tau = 0.6$	96.14%	95.28%	97.29%	0.114	97.69%	93.05%
	$\tau = 0.7$	96.00%	98.51%	95.09%	0.140	97.40%	92.95%
	$\tau = 0.8$	95.67%	98.81%	94.71%	0.130	96.49%	93.08%
	$\tau = 0.9$	95.30%	98.71%	94.26%	0.137	96.74%	91.95%

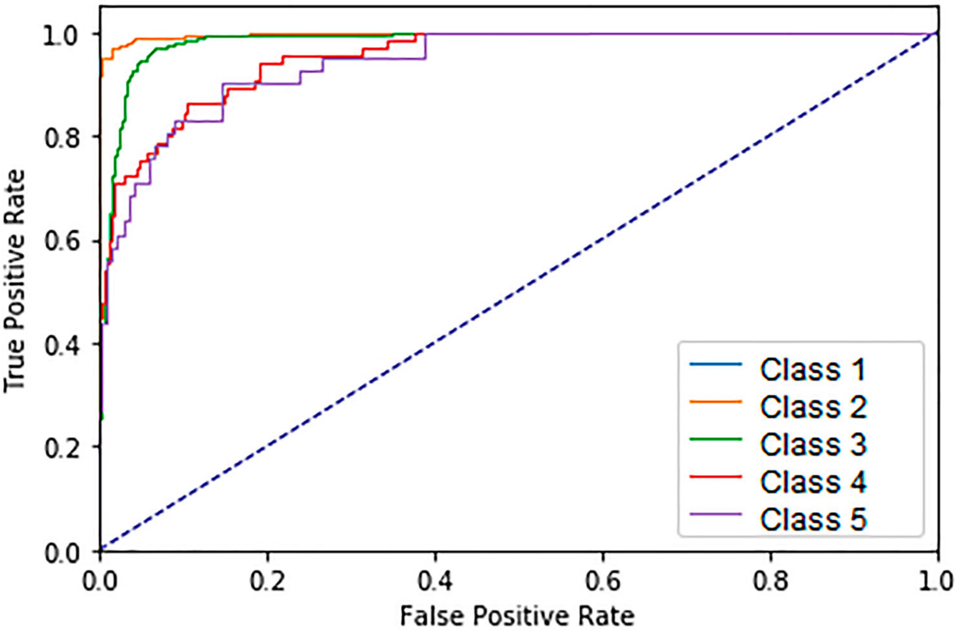


Figure 12. ROC curve of the best performing branch/threshold iteration of scenario 1.

By analyzing the potential of each thresholding value, we evaluated the performance of the proposed AG-CNN architecture. We also evaluate the performance of both global and local branches with and without preprocessing.

6.2.1. Scenario 1

The first scenario consists of training the system (global and local) with the original dataset without CLAHE or LoG. The only transformations applied are the basic ones mentioned earlier. Table 4 shows the resulting metrics of the global and local branches using this first scenario, while Figure 12 represents the ROC curve of the best-performing architecture.

6.2.2. Scenario 2

For the second scenario, three new datasets are generated using CLAHE and LoG (Figure 13). These new datasets are used for the training of this second scenario. Tables 5, 6 and 7 show the resulting validation metrics of the global/local branches corresponding to CLAHE, LoG and CLAHE+LoG preprocessed data, respectively. Figure 14 represents the ROC curves of the best-performing iteration for every preprocessing method used in this scenario.

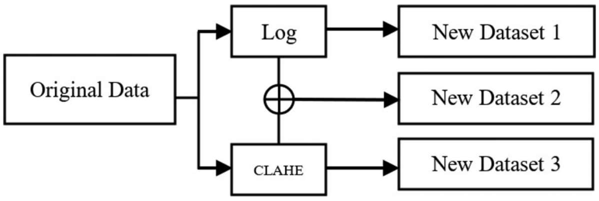


Figure 13. Diagram of the new datasets generation.

6.2.3. Scenario 3

Data class imbalance can lead to a loss in deep learning performance. For this reason, in this scenario, we implemented a data balance algorithm on the same datasets generated in the previous scenario by applying some transformations (rotation with $-/+45°$, horizontal flip, and brightness with range from 0.7 to 1.3). Consequently, the sample number for classes 'Mild DR', 'Moderate DR', 'Severe DR' and 'Proliferative DR' become 1730, 1740, 1710 and 1720, respectively. We

Table 5. Evaluation metrics for the CLAHE preprocessing process.

Branch	Threshold	Score					
		Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	96.51%	99.22%	95.77%	0.105	96.73%	94.87%
Local (Cropping)	\$\tau = 0.1$	96.18%	98.89%	95.33%	0.112	96.56%	94.22%
	$\tau = 0.2$	95.74%	98.55%	94.79%	0.144	96.17%	93.55%
	$\tau = 0.3$	95.89%	98.79%	95.03%	0.123	96.07%	94.05%
	$\tau = 0.4$	95.70%	98.53%	94.79%	0.146	96.17%	93.50%
	$\tau = 0.5$	95.67%	98.87%	94.66%	0.128	96.13%	93.33%
	$\tau = 0.6$	95.38%	98.82%	94.33%	0.134	95.79%	93.00%
	$\tau = 0.7$	95.27%	98.94%	94.24%	0.121	94.23%	94.35%
	$\tau = 0.8$	95.01%	98.74%	93.94%	0.148	93.60%	94.32%
	$\tau = 0.9$	94.50%	98.48%	93.32%	0.157	93.59%	93.13%

Table 6. Evaluation metrics for the LoG preprocessing approach.

Branch	Threshold	Score					
		Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	92.80%	97.20%	90.97	0.198	95.31%	87.09%
Local (Cropping)	$\tau = 0.1$	92.98%	97.29%	91.16%	0.203	96.04%	86.81%
	$\tau = 0.2$	93.27%	97.69%	91.60%	0.181	93.91%	89.51%
	$\tau = 0.3$	92.47%	97.07%	90.55%	0.208	94.52%	86.96%
	$\tau = 0.4$	92.65%	97.21%	90.63%	0.201	95.86%	86.01%
	$\tau = 0.5$	92.58%	96.94%	90.66%	0.211	94.52%	87.19%
	$\tau = 0.6$	92.54%	9691%	90.65%	0.209	95.21%	86.59%
	$\tau = 0.7$	92.18%	9629%	90.23%	0.238	93.90%	86.92%
	$\tau = 0.8$	91.89%	9617%	89.67%	0.242	93.93%	85.83%
	$\tau = 0.9$	89.56%	9512%	87.15%	0.275	88.43%	85.99%

recall here that we kept the number of samples in class ’No DR’ without change (1805 samples).

Tables 8, 9 and 10 show the results of this scenario with the three preprocessing methods used above. Meanwhile, Figures 15 and 16 illustrate the ROC curves of each preprocessing technique for the global branch and also for the local branch with best threshold that achieved the highest performance.

After examining the results of this scenario, we noticed that data balancing didn’t improve much in the performance of the model. In some cases, it was a bit worse due to the class imbalance that can be considered as a natural distribution of the diabetic retinopathy. As a result of the performance of this scenario, we continue the rest of the experiments without performing the data balancing.

6.2.4. *Scenario 4*

In this scenario, another boost for datasets is implemented to generate two new datasets by merging the original dataset with two modified ones (using CLAHE and LoG). Figure 17 shows the process of creating the two new datasets. Tables 11 and 12 show the resulting validation metrics of the system using this scenario, while Figure 18 shows the ROC curves of these results.

6.3. *Comparison*

After evaluating our different tests, the obtained results can be summarized in Table 13. We note that the results obtained when we used data augmentation are not mentioned in this table as it didn’t bring much improvement (scenario 3). We present an overview of the global branch’s performance, which serves as the benchmark for each preprocessing process. The accuracy and the AUC are low when using LoG preprocessing, with 92.62% and 97.25%, respectively. At the same time, the CLAHE+LoG preprocessing gave slightly better results with 93.20%/98.02% in term of accuracy and AUC, respectively. However, our system

Table 7. Evaluation metrics for the CLAHE+LoG preprocessing approach.

Branch	Threshold	Score					
		Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	93.49%	98.01%	91.91%	0.177	95.81%	88.42%
Local (Cropping)	$\tau = 0.1$	92.65%	97.82%	91.32%	0.194	90.11%	92.66%
	$\tau = 0.2$	92.95%	97.81%	91.29%	0.182	94.25%	88.57%
	$\tau = 0.3$	92.76%	97.61%	91.12%	0.187	93.78%	88.70%
	$\tau = 0.4$	92.62%	97.37%	90.92%	0.198	93.18%	88.85%
	$\tau = 0.5$	92.25%	97.19%	90.19%	0.219	94.87%	86.01%
	$\tau = 0.6$	92.87%	97.24%	91.22%	0.203	92.64%	89.96%
	$\tau = 0.7$	92.61%	97.48%	90.79%	0.205	94.24%	87.68%
	$\tau = 0.8$	92.15%	97.06%	90.42%	0.220	91.43%	89.50%
	$\tau = 0.9$	90.55%	96.02%	88.36%	0.253	90.55%	86.45%

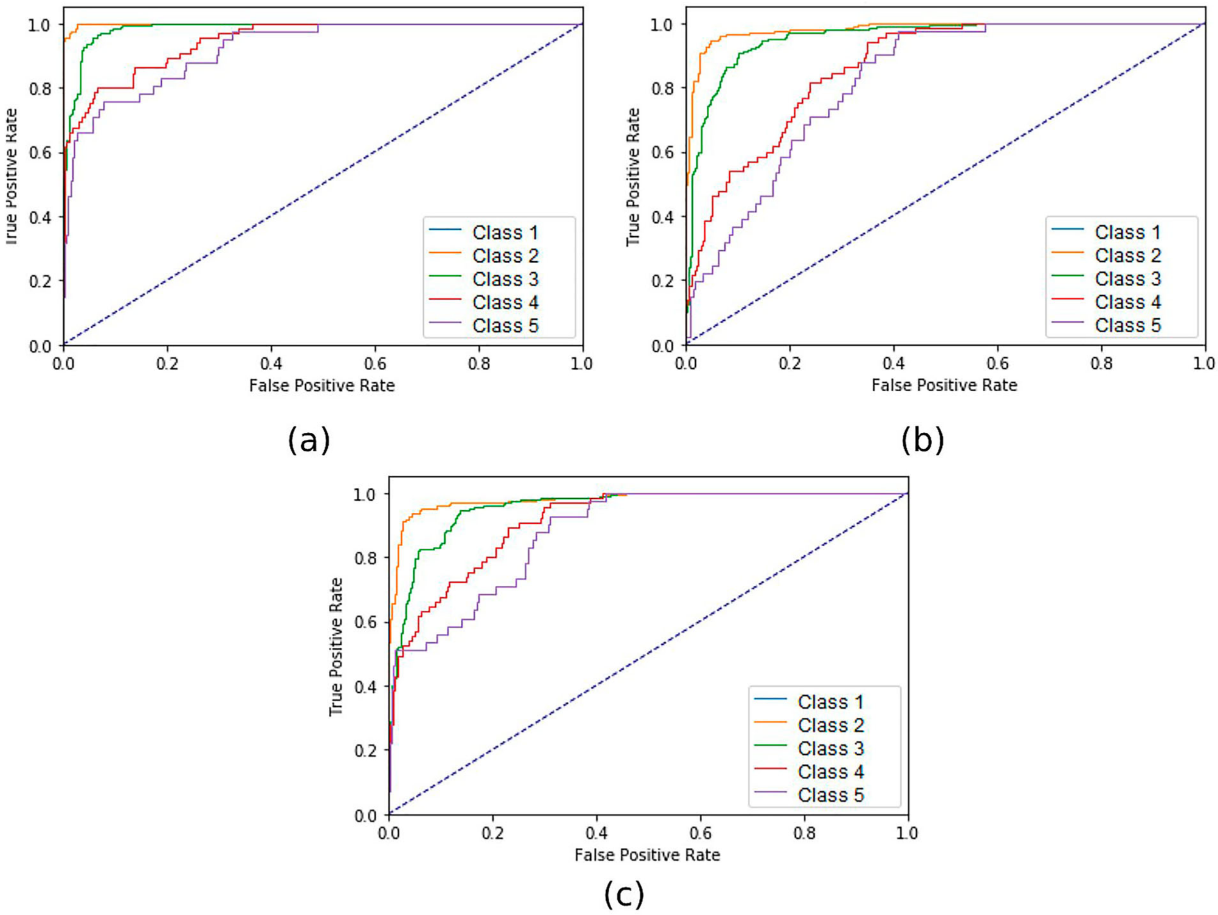


Figure 14. ROC curves of the best performing branch/threshold in scenario 2 with different preprocessing techniques: (a) CLAHE, (b) LoG, and (c) CLAHE+LoG.

Table 8. CLAHE preprocessing effect with data balancing.

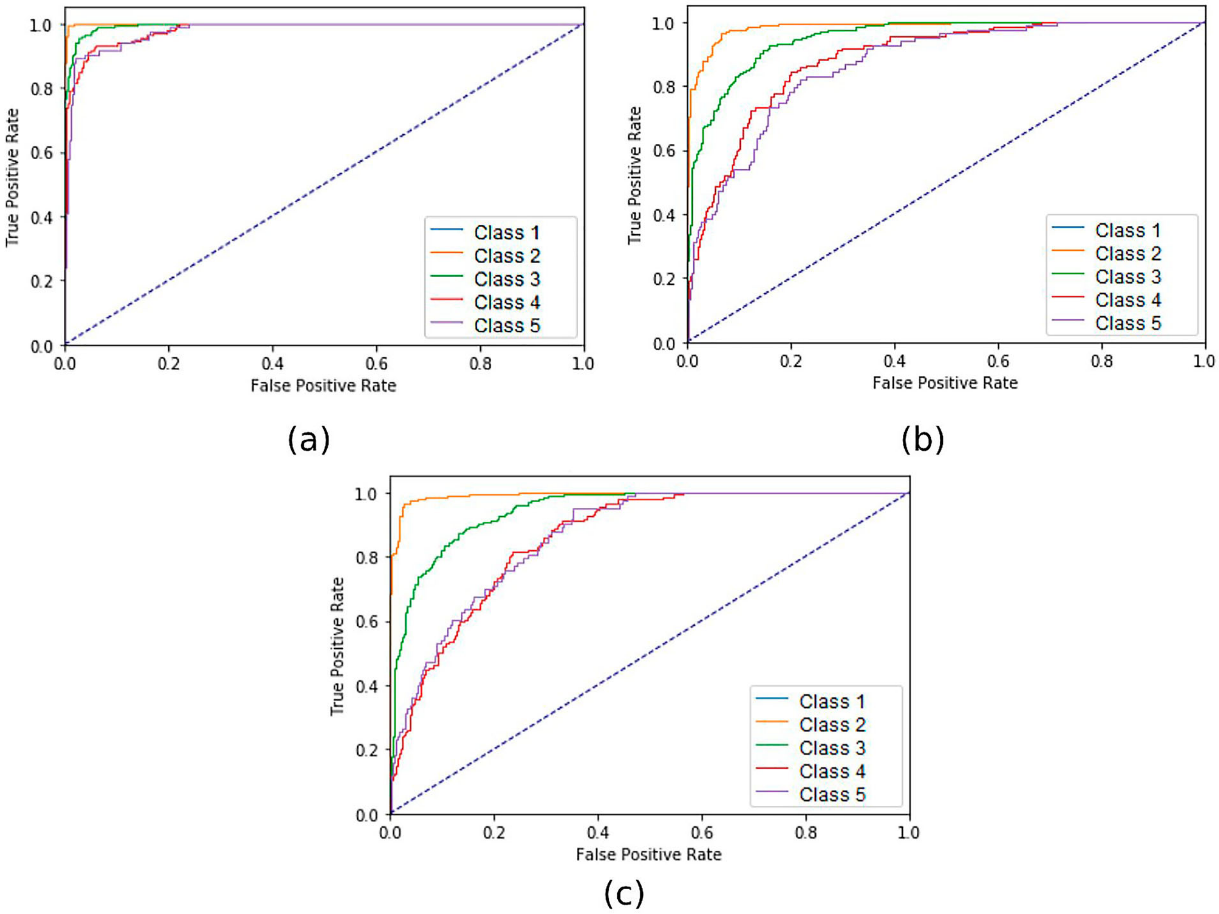
		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	96.51%	99.38%	96.47%	0.1	96.42%	96.56%
Local (Cropping)	$\tau = 0.1$	96.12%	98.54%	95.01%	0.102	96.33%	95.73%
	$\tau = 0.2$	95.64%	98.27%	94.84%	0.124	95.61%	94.12%
	$\tau = 0.3$	95.59%	98.42%	95.23%	0.118	95.49%	94.63%
	$\tau = 0.4$	94.67%	98.86%	94.44%	0.14	94.89%	94.03%
	$\tau = 0.5$	95.94%	98.21%	94.83%	0.108	94.57%	94.14%
	$\tau = 0.6$	95.27%	98.85%	94.08%	0.146	94.05%	94.16%
	$\tau = 0.7$	95.25%	98.27%	94.17%	0.105	94.06%	94.30%
	$\tau = 0.8$	95.84%	98.13%	93.64%	0.119	93.44%	93.87%
	$\tau = 0.9$	93.38%	98.68%	93.74%	0.16	93.6%	93.07%

Table 9. LoG preprocessing effect with data balancing.

		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	91.89%	97.49%	91.54%	0.2	92.57%	90.65%
Local (Cropping)	$\tau = 0.1$	91.21%	97.08%	90.76%	0.219	93.79%	88.02%
	$\tau = 0.2$	90.96%	97.03%	90.4%	0.219	93.26%	87.75%
	$\tau = 0.3$	90.66%	96.92%	89.97%	0.232	93.51%	86.73%
	$\tau = 0.4$	90.96%	96.68%	90.36%	0.233	92.73%	88.18%
	$\tau = 0.5$	90.51%	96.83%	89.83%	0.235	94.16%	85.93%
	$\tau = 0.6$	89.91%	96.31%	89.36%	0.243	91.02%	87.81%
	$\tau = 0.7$	88.69%	94.86%	87.34%	0.307	94.45%	81.29%
	$\tau = 0.8$	87.86%	94.42%	86.64%	0.294	92.29%	81.74%
	$\tau = 0.9$	85.96%	93.05%	84.32%	0.327	90.01%	79.46%

Table 10. CLAHE+LoG preprocessing effect with data balancing.

		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	91.61%	97.27%	91.00%	0.204	92.32%	90.25%
Local (Cropping)	$\tau = 0.1$	91.02%	97.40%	90.62%	0.218	93.49%	87.97%
	$\tau = 0.2$	91.15%	97.11%	90.90%	0.215	93.32%	87.68%
	$\tau = 0.3$	90.71%	97.01%	89.89%	0.224	93.63%	86.70%
	$\tau = 0.4$	91.03%	96.78%	90.23%	0.229	92.80%	88.16%
	$\tau = 0.5$	90.61%	96.91%	89.76%	0.231	94.13%	85.92%
	$\tau = 0.6$	89.88%	96.39%	89.33%	0.241	91.10%	87.77%
	$\tau = 0.7$	88.72%	94.90%	87.32%	0.304	94.39%	81.27%
	$\tau = 0.8$	87.87%	94.41%	86.61%	0.293	92.30%	81.76%
	$\tau = 0.9$	85.98%	93.03%	84.29%	0.325	90.02%	79.45%



(a)

(b)

(c)

Figure 15. ROC curves of the scenario 3 with data balancing and three different preprocessing: (a) CLAHE, (b) LoG, and (c) CLAHE+LoG.

without preprocessing gave better results than the previous ones with 96.26%/97.97%. This superiority can be justified by the failure of the LoG technique to properly highlight the features of the DR image. After that, we have CLAHE preprocessing with 96.20%/99.24%, while the augmented images with CLAHE and LoG yielded the best results, with 97.46%/99.5% and 97.26%/99.5%, respectively. In fact, the original images and the preprocessed images do not highlight the same features.

To complement the attention processes of the global branch, the local branch is trained using clipped and downscaled lesion patches. Almost all of the methods gave lower results in the local branch than the global one, although the results are close to the baseline. The probable reason for this observation is that the lesion region estimation and cropping process may lead to information loss, which is critical for recognition. So, the local branch may need a more accurate estimation of the attention area. On the other hand, In the CLAHE augmented method, the local branch exhibited superior performance, achieving the best results when using a cropping threshold of 0.1. This configuration achieved an accuracy of 98.4% and an AUC score of 99.8%. The improved performance can be attributed to

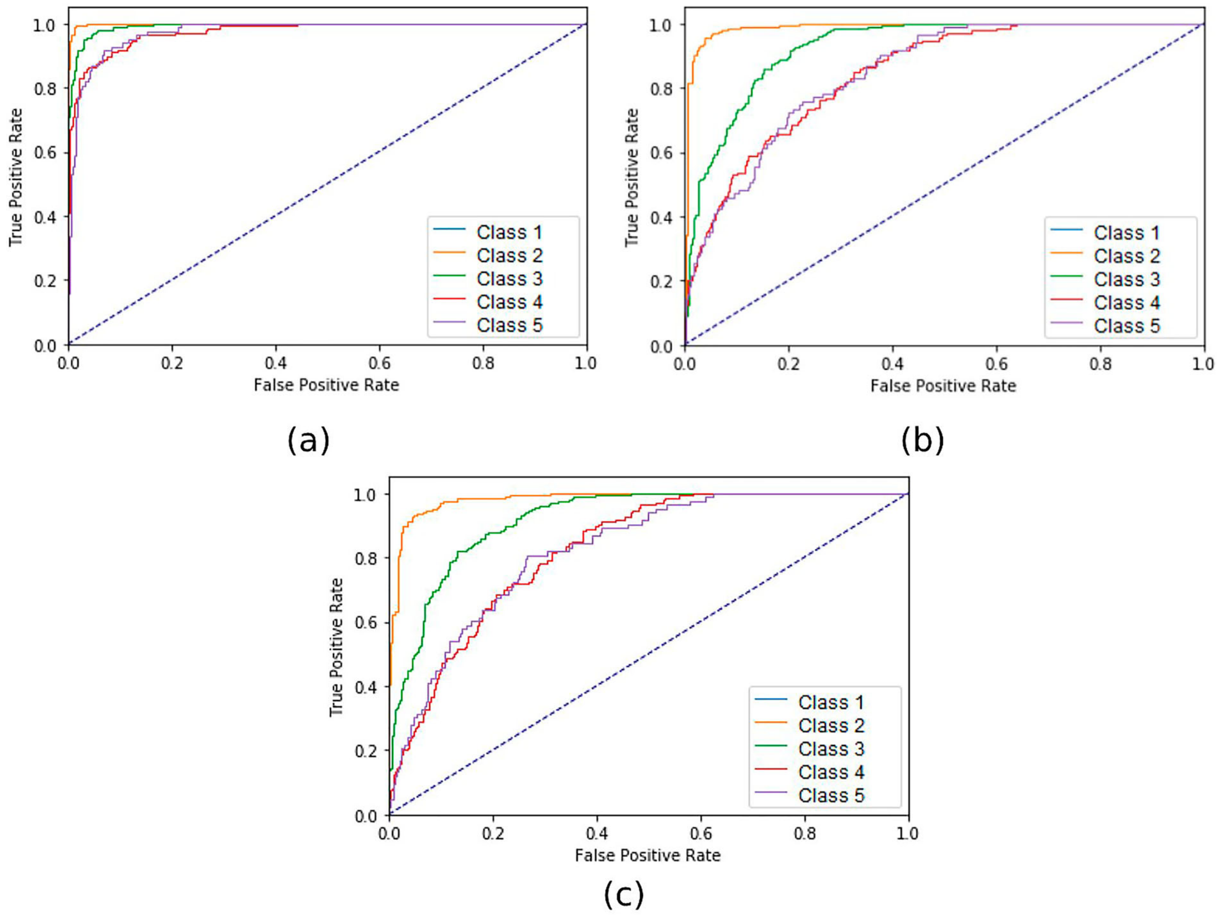


Figure 16. ROC curves of the best-performing threshold in scenario 3 for the local branch: (a) CLAHE with $\tau = 0.1$, (b) LoG with $\tau = 0.1$, and (c) CLAHE+LoG with $\tau = 0.2$.

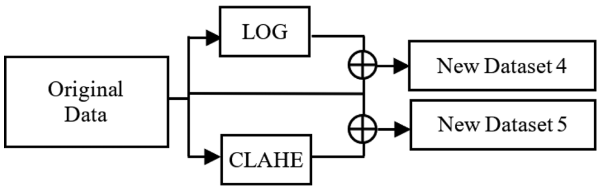


Figure 17. Diagram of the two augmented dataset generations.

the attention mechanism by eliminating irrelevant data facilitated by this threshold.

To demonstrate our method's advancements to deep learning-based Diabetic Retinopathy classification, we conducted a comparative analysis with state-of-the-art studies that used the same dataset (APTOS 2019). Table 14 reports the authors, used preprocessing operations, results and limitations of each method againt our one. Except the method presented by Narayanan et al. [23] where we achieved the same accuracy rate (98.4%) but our method achieved a superior AUC metric with 99.8%, our method outperforms the state-of-the-art studies in term of accuracy and AUC.

7. Conclusion

In this paper, we introduce a cutting-edge Convolutional Neural Network (CNN) architecture characterized by its unique dual-branch design and an integrated attention-guided mechanism tailored specifically for the early detection of Diabetic Retinopathy (DR). This novel architecture represents a significant departure from traditional approaches in the field. Unlike previous methods, which predominantly relied on global image features, our approach takes a more holistic perspective by considering both global and local details throughout the training process. One of the key innovations of our method lies in its ability to extract vital regions of interest within the global retinal images. We achieve this through the generation of attention heatmaps, which are an integral part of the training process for the local branch of our CNN. By focusing on these critical regions, our model gains a deeper understanding of the subtle variations and anomalies that can signify the presence and progression of DR. This approach effectively enhances the network's sensitivity and specificity, contributing to its remarkable performance. The extensive experiments conducted as part

Table 11. Results using augmented method with CLAHE preprocessing.

		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	97.29%	99.60%	96.41%	0.082	97.10%	95.77%
Local(Cropping)	$\tau = 0.1$	98.42%	99.81%	97.77%	0.052	99.06%	96.78%
	$\tau = 0.2$	98.28%	99.76%	97.96%	0.054	98.41%	97.26%
	$\tau = 0.3$	97.92%	99.83%	97.57%	0.059	97.41%	97.61%
	$\tau = 0.4$	97.79%	99.82%	97.30%	0.059	98.13%	96.53%
	$\tau = 0.5$	97.68%	99.69%	97.25%	0.059	96.98%	97.64%
	$\tau = 0.6$	97.82%	99.65%	97.42%	0.068	97.59%	97.01%
	$\tau = 0.7$	97.71%	99.59%	97.19%	0.065	98.08%	96.44%
	$\tau = 0.8$	97.53%	99.63%	96.97%	0.072	97.84%	96.20%
	$\tau = 0.9$	96.33%	99.34%	95.53%	0.096	94.86%	96.15%

Table 12. Results using augmented method with LoG preprocessing.

		Score					
Branch	Threshold	Accuracy	AUC	F1	Loss	Precision	Recall
Global	DenseNet121	95.12%	98.70%	94.05%	0.138	96.90%	91.41%
Local (Cropping)	$\tau = 0.1$	95.53%	99.10%	94.57%	0.116	95.64%	93.58%
	$\tau = 0.2$	94.86%	98.90%	93.78%	0.127	94.56%	93.10%
	$\tau = 0.3$	94.76%	98.69%	93.57%	0.138	95.97%	91.35%
	$\tau = 0.4$	94.34%	98.48%	93.11%	0.147	94.82%	91.53%
	$\tau = 0.5$	94.28%	98.42%	93.01%	0.149	94.52%	91.23%
	$\tau = 0.6$	94.50%	98.36%	92.88%	0.153	95.33%	91.17%
	$\tau = 0.7$	94.10%	98.38%	92.85%	0.155	94.15%	91.05%
	$\tau = 0.8$	94.09%	98.22%	92.73%	0.161	94.02%	91.01%
	$\tau = 0.9$	94.22%	98.11%	92.70%	0.168	94.22%	90.94%

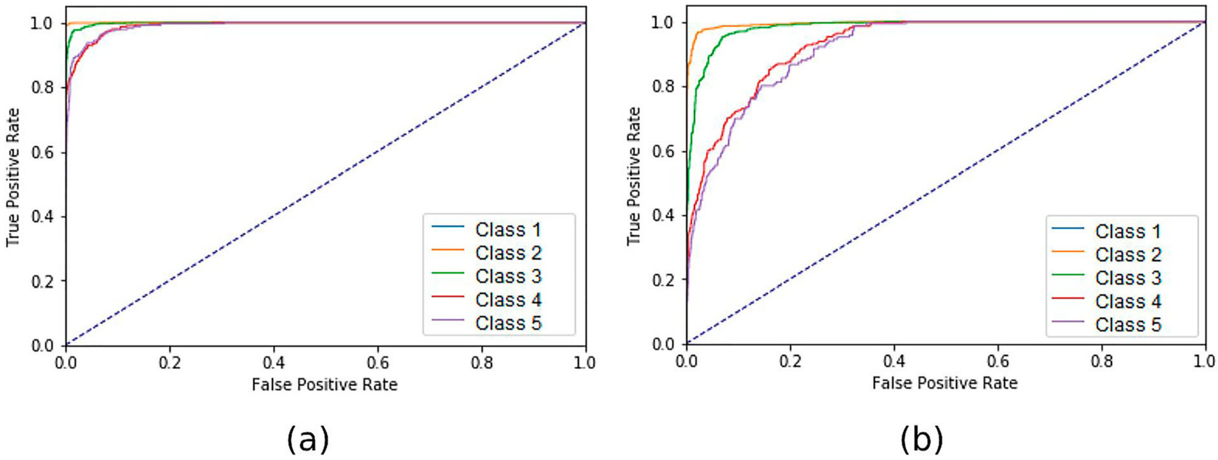


Figure 18. ROC curves obtained with augmented preprocessing scenario: (a) CLAHE and (b) LoG.

Table 13. Summary of the proposed method performances.

Scenario	Preprocessing technique	Validation accuracy (%)	Branch	Improvement (%)	Binarization ratio (τ) (%)
1	Basic	96.43	Local	0.3	0.1
2	• CLAHE	96.5	Global	/	/
	• LoG	93.27	Local	0.47	0.2
	• CLAHE + LoG	93.49	Global	/	/
3	• Data balancing + CLAHE	96.51	Global	/	
	• Data balancing + LoG	91.89	Global	/	
	• Data balancing + CLAHE&LoG	91.61	Global	/	
4	• *Original* + CLAHE	98.42	Local	1.03	0.1
	• *Original* + LoG	95.53	Global	/	/

of this study serve as compelling evidence of the efficacy of our proposed system. Leveraging the APTOS 2019 Diabetic Retinopathy dataset, we demonstrate that our dual-branch CNN, which harnesses the synergistic power of global and local cues, achieves peak performance in the critical task of DR identification. Our research drive with this suggested model involves multifaceted training across diverse scenarios. Through

Table 14. Comparison with the state-of-the-art methods.

Author, Year	Preprocessing	Method	Accuracy	AUC	Limitations
Majumder et al. [24]	Yes	CNN	88.5%	/	• Lack of Generalization with other models
Shaban et al. [25]	Yes	DCNN	88%	0.95	• 3 DR classes instead of 5 • Feature study and selection are not available
Gangwar and Ravi [26]	Yes	Hybrid model (Inception, ResNet v2)	82.18%	/	• Computationally expensive
Ramchandre et al. [27]	Yes (Image augmentation with AUGMIX)	EfficientNetb3 SEResNeXt32x4d	91.4% 85.2%	/	• Lack of image enhancement
Nahiduzzaman et al. [28]	Yes	Parallel CNN/ELM	97.27%	0.988	• Lack of data augmentation
Islam et al. [29]	Yes	SCL/Xception	84.36%	/	• Low batch size
Jian et al. [30]	Yes	Triple-DRNets	92.08%	/	• Limitation of feature fusion in the subnets
Narayanan et al. [23]	Yes	Hybrid model (AlexNet, ResNet, VGG16, Inception v3)	98.4%	0.985	• Lack of data augmentation techniques
Madarapu et al. [22]	Yes	Custom deep CNN (Residual blocs + CSAM + NLB blocs)	89.38%	0.97	• Complex architecture
Shakibania et al. [19]	Yes	Dual Branch network (ResNet50 and EfficientNetB0)	89.60%	/	• Misclassifications for severe and proliferative DR
Selvakumar and Akila [33]	Yes	U-Net-KNN model	80.78%	/	• Computational complexity
Proposed method	Yes	DenseNet-121	98.4%	0.998	• Computationally expensive in the training part

rigorous experimental evaluations, we have substantiated the inherent strength of our approach, which combines global image features with localized regions of interest within DR images. What's particularly noteworthy is that our model consistently delivers promising results, even when confronted with the challenge of limited training data. This resilience underscores its potential for real-world clinical applications, where obtaining extensive labelled datasets can often be a formidable obstacle. As we cast our gaze into the future of research in the realm of diabetic retinopathy detection, our commitment remains unwavering. We are poised to delve deeper into the exploration of diverse preprocessing techniques. These techniques hold the promise of further refining the accuracy and robustness of our model, potentially unlocking new dimensions in DR diagnosis. Additionally, we are eager to investigate opportunities for synergizing our dual-branch architecture with other state-of-the-art attention-guided methods. This avenue of research holds the potential to push the boundaries of DR detection, ultimately leading to more timely and effective interventions for individuals at risk of this debilitating condition. Our work stands as a testament to the ever-evolving landscape of medical AI, with the potential to make a profound impact on patient care and public health.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability

Dataset is available on Kaggle website.

ORCID

Youcef Brik http://orcid.org/0000-0003-1565-0594

References

[1] Wild S, Roglic G, Green A, et al. Global prevalence of diabetes. Diab Care. 2004;27(5):1047–1053. doi: 10.2337/diacare.27.5.1047

[2] Karthik M, Sohier D. APTOS 2019 Blindness Detection. Kaggle. 2019. https://kaggle.com/competitions/aptos2019-blindness-detection

[3] Abbas Q, Qureshi I, Ibrahim ME. An automatic detection and classification system of five stages for hypertensive retinopathy using semantic and instance segmentation in DenseNet architecture. Sensors. 2021;21(20):6936. doi: 10.3390/s21206936

[4] Asiri N, Hussain M, Al Adel F, et al. Deep learning based computer-aided diagnosis systems for diabetic retinopathy: a survey. Artif Intell Med. 2019;99:101701. doi: 10.1016/j.artmed.2019.07.009

[5] Selvaraju RR, Cogswell M, Das A, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Venice. 2017 IEEE International Conference on Computer Vision (ICCV). 2017. p. 618-626.

[6] Khalifa NEM, Loey M, Taha MHN, et al. Deep transfer learning models for medical diabetic retinopathy detection. Acta Inf Med. 2019;27(5):327. doi: 10.5455/aim.

[7] Dekhil O, Naglah A, Shaban M, et al. Deep learning based method for computer aided diagnosis of diabetic retinopathy. In: 2019 IEEE International Conference on Imaging Systems and Techniques (IST), Abu Dhabi: 2019. p. 1–4.

[8] Kassani SH, Kassani PH, Khazaeinezhad R, et al. Diabetic retinopathy classification using a modified xception architecture. In: 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT); 2019. p. 1–6.

[9] Tymchenko B, Marchenko P, Spodarets D. Deep learning approach to diabetic retinopathy detection; 2020. arXiv preprint arXiv:2003.02261.

[10] Qureshi I, Ma J, Abbas Q. Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. Multimed Tools Appl. 2021;80(8):11691–11721. doi: 10.1007/s11042-020-10238-4

[11] Zeng X, Chen H, Luo Y, et al. Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. IEEE Access. 2019;7:30744–30753. doi: 10.1109/ACCESS.2019.2903171

[12] Shanthi T, Sabeenian R. Modified Alexnet architecture for classification of diabetic retinopathy images. Comput Electr Eng. 2019;76:56–64. doi: 10.1016/j.compeleceng.2019.03.004

[13] Decencière E, Zhang X, Cazuguel G, et al. Feedback on a publicly distributed image database: the Messidor database. Image Anal Stereol. 2014;33(3):231–234. doi: 10.5566/ias.1155

[14] Abbas Q, Ibrahim ME, Baig AR. Transfer learning-based computer-aided diagnosis system for predicting grades of diabetic retinopathy. Comput Mater Contin. 2022;71(3):4573–4590.

[15] Jain A, Jalui A, Jasani J, et al. Deep learning for detection and severity classification of diabetic retinopathy. In: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT). Chennai: IEEE; 2019. p. 1–6.

[16] Hagos MT, Kant S. Transfer learning based detection of diabetic retinopathy from small dataset; 2019. arXiv preprint arXiv:1905.07203.

[17] Quellec G, Charriere K, Boudi Y, et al. Deep image mining for diabetic retinopathy screening. Med Image Anal. 2017;39:178–193. doi: 10.1016/j.media.2017.04.012

[18] Gargeya R, Leng T. Automated identification of diabetic retinopathy using deep learning. Ophthalmology. 2017;124(7):962–969. doi: 10.1016/j.ophtha.2017.02.008

[19] Shakibania H, Raoufi S, Pourafkham B, et al. Dual branch deep learning network for detection and stage grading of diabetic retinopathy. Biomed Signal Process Control. 2024;93:106168. doi: 10.1016/j.bspc.2024.106168

[20] Wang Z, Yin Y, Shi J, et al. Zoom-in-net: deep mining lesions for diabetic retinopathy detection. In: Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11–13, 2017, Proceedings, Part III 20. Springer; 2017. p. 267–275.

[21] Zhao Z, Zhang K, Hao X, et al. Bira-net: bilinear attention net for diabetic retinopathy grading. In: 2019 IEEE International Conference on Image Processing (ICIP). Taipei: IEEE; 2019. p. 1385–1389.

[22] Madarapu S, Ari S, Mahapatra K. A deep integrative approach for diabetic retinopathy classification with synergistic channel-spatial and self-attention mechanism. Expert Syst Appl. 2024;249:123523. doi: 10.1016/j.eswa.2024.123523

[23] Narayanan BN, Hardie RC, De Silva MS, et al. Hybrid machine learning architecture for automated detection and grading of retinal images for diabetic retinopathy. J Med Imaging. 2020;7(3):034501–034501. doi: 10.1117/1.JMI.7.3.034501

[24] Majumder S, Elloumi Y, Akil M, et al. A deep learning-based smartphone app for real-time detection of five stages of diabetic retinopathy. In: Real-time image processing and deep learning 2020. Vol. 11401, SPIE; 2020. p. 13–22. doi: 10.1117/12.2557554

[25] Shaban M, Ogur Z, Mahmoud A, et al. A convolutional neural network for the screening and staging of diabetic retinopathy. PLoS ONE. 2020;15(6):e0233514. doi: 10.1371/journal.pone.0233514

[26] Gangwar AK, Ravi V. Diabetic retinopathy detection using transfer learning and deep learning. In: Evolution in computational intelligence: frontiers in intelligent computing: theory and applications (FICTA 2020). Vol. 1, Surathkal: Springer Singapore; 2021. p. 679–689. doi: 10.1007/978-981-15-5788-0_64

[27] Ramchandre S, Patil B, Pharande S, et al. A deep learning approach for diabetic retinopathy detection using transfer learning. In: 2020 IEEE International Conference for Innovation in Technology (INOCON). Bangalore: IEEE; 2020. p. 1–5.

[28] Nahiduzzaman M, Islam MR, Goni MOF, et al. Diabetic retinopathy identification using parallel convolutional neural network based feature extractor and elm classifier. Expert Syst Appl. 2023;217:119557. doi: 10.1016/j.eswa.2023.119557

[29] Islam MR, Abdulrazak LF, Nahiduzzaman M, et al. Applying supervised contrastive learning for the detection of diabetic retinopathy and its severity levels from fundus images. Comput Biol Med. 2022;146:105602. doi: 10.1016/j.compbiomed.2022.105602

[30] Jian M, Chen H, Tao C, et al. Triple-DRNet: a triple-cascade convolution neural network for diabetic retinopathy grading using fundus images. Comput Biol Med. 2023;155:106631. doi: 10.1016/j.compbiomed.2023.106631

[31] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Hawaii: 2017. p. 4700–4708.

[32] Garg B, Sharma G. A quality-aware energy-scalable gaussian smoothing filter for image processing applications. Microprocess Microsyst. 2016;45:1–9. doi: 10.1016/j.micpro.2016.02.012

[33] Selvakumar V, Akila C. Efficient diabetic retinopathy diagnosis through U-Net – KNN integration in retinal fundus images. Automatika. 2023;64(4):1148–1157. doi: 10.1080/00051144.2023.2251231