Journal of Robotics and Control (JRC)
Volume 6, Issue 1, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i1.25331

Enhancing Diabetic Retinopathy Classification in Fundus Images using CNN Architectures and Oversampling Technique

Yuri Pamungkas [1*], Evi Triandini [2], Wawan Yunanto [3], Yamin Thwe [4]
[1] Department of Medical Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2] Department of Information System, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, Indonesia
[3] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[4] Department of Big Data Management & Analytics, Rajamangala University of Technology Thanyaburi, Bangkok, Thailand
Email: [1] yuri@its.ac.id, [2] evi@stikom-balu.ac.id, [3] d10615811@mail.ntust.edu.tw, [4] yamin_t@rmutt.ac.th
*Corresponding Author

Abstract—**Diabetic Retinopathy (DR) is a severe complication of diabetes mellitus that affects the retinal blood vessels and is a leading cause of blindness in productive-age individuals. The global increase in diabetes prevalence requires an effective DR classification system for early detection. This study aims to develop a DR classification system using several CNN architectures, such as EfficientNet-B4, ResNet-50, DenseNet-201, Xception, and Inception-ResNet-v2, with the application of the SMOTE oversampling technique to address data class imbalance. The dataset used is APTOS 2019, which has an unbalanced class distribution. Two scenarios were tested, the first without data balancing and the second with SMOTE implementation. The test results show that in the first scenario, Xception achieved the highest accuracy at 80.61%, but model performance was still limited due to majority class dominance. The application of SMOTE in the second scenario significantly improved model accuracy, with EfficientNet-B4 achieving the highest accuracy of 97.78%. Additionally, precision and recall increased dramatically in the second scenario, demonstrating SMOTE's effectiveness in enhancing the model's ability to detect minority classes and reduce prediction errors. DenseNet-201 achieved the highest precision at 99.28%, while Inception-ResNet-v2 recorded the highest recall at 98.57%. Overall, this study proves that the SMOTE method effectively addresses class imbalance in the fundus dataset and significantly improves CNN model performance. Although data balancing can help improve model quality by dealing with data imbalances, it comes at a higher computational cost. Using data balancing techniques with SMOTE significantly increased the iteration time per round on all tested CNN architectures.**

Keywords—Diabetic Retinopathy; CNN Architectures; SMOTE; Class Imbalance; Classification System.

I. INTRODUCTION

Diabetic retinopathy (DR) is a common microvascular complication of diabetes mellitus that can lead to visual impairment and blindness [1]. According to data from the International Diabetes Federation (IDF), in 2021 there were around 537 million adults (aged 20-79 years) living with diabetes worldwide [2]. Although specific data on the prevalence of diabetic retinopathy globally are not available, studies show that DR accounts for more than 50% of cases of visual impairment worldwide [3]. In the Asia-Pacific region, an estimated 51% of cases of visual impairment are caused by DR [4]. In Indonesia, diabetic retinopathy is the fifth leading cause of blindness and visual impairment. WHO estimates that 4.8% of the 37 million cases of blindness worldwide are caused by DR [5]. With the increasing prevalence of diabetes globally, the risk of developing diabetic retinopathy is also expected to increase. Therefore, it is important for people with diabetes to have regular eye screenings for early detection and appropriate treatment.

The rapid increase in the prevalence of diabetes, driven by unhealthy lifestyles, population aging, and urbanization, has significantly increased the number of DR cases [6]. To overcome these challenges, DR classification has become essential to support early diagnosis, severity stratification, and therapeutic decision-making [7]. Currently, DR classification is generally based on guidelines developed by international health organizations, such as the American Academy of Ophthalmology (AAO) and the International Clinical Diabetic Retinopathy Disease Severity Scale (ICDRS) [8]. This approach groups DR into stages, such as mild non-proliferative to severe and proliferative, based on the presence of microaneurysms, retinal hemorrhages, or neovascularization [9]. In recent decades, technological advances have enabled the development of automated classification systems for DR using artificial intelligence (AI) and machine learning (ML) [10]. These classifications aim to differentiate the severity of DR, ranging from mild non-proliferative to severe proliferative. This technology is based on the analysis of eye fundus images, allowing the detection of abnormal patterns such as microaneurysms, hard exudates, and neovascularization [11]. This system provides an opportunity for faster, more accurate and more cost-effective diagnosis compared to manual methods by specialist doctors.

The importance of proper DR classification is not only for diagnosis, but also for determining the appropriate treatment strategy [12]. Grouping patients based on DR severity allows for more targeted treatment, such as laser photocoagulation for proliferative DR or anti-VEGF therapy for macular edema [13]. In addition, AI-based classification also helps expand access to healthcare in remote areas, where ophthalmology specialists are limited. Thus, modern DR classification becomes an important foundation for improving early detection, patient management and prevention of more severe complications [14]. Suedumrong, et al [15] in their research used the CNN method to classify the severity of Diabetic Retinopathy (DR) based on retinal images (EyePACS dataset). From the dataset, 35155 images were categorized into 5 different classes (based on the severity of DR). The CNN models used include Inception V3, VGG, and ResNet with image pre-processing approaches including grayscale conversion, background removal, and data augmentation. Based on the research results, it was indicated that removing the image background was able to increase the performance of the CNN model in the classification process by 90.60%.

In addition, Zhang, et al [16] proposed a CNN-based Grading method to classify Diabetic Retinopathy in Fundus images. Data cleaning and enhancement were performed on fundus images to reduce noise and improve image quality. Then the SACGAN method was used to synthesize the number of Diabetic Retinopathy fundus images so that the unbalanced data between classes became balanced. The DRMC Net model which is a combination of ResNeXt-50 and residual convolution module is used to classify Diabetic Retinopathy which obtained a performance level of 92.3% (accuracy), 92.5% (specificity), 92.5% (sensitivity). Research conducted by Dutta, et al [17] successfully proposed an automatic Diabetic Retinopathy classification system using Ensemble Machine Learning such as Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), XGBoost (XGB), and LightGBM (LGM). In addition, there is a process of missing value imputation, feature selection, and K-fold cross validation in the proposed system. Based on the results of system testing and analysis of variance, the accuracy level reached 73.5% and AUC of 83.2%. Another study conducted by Fayyaz, et al [18] also succeeded in detecting and classifying DR based on fundus images using the AlexNet and ResNet101 models. In the proposed system, layer interconnection and Ant Colony method helps in the process of feature identification and selection of important features. The result obtained an optimal accuracy level of 93% in the system testing process.

Based on several previous studies, this study will also propose a classification system for Diabetic Retinopathy (DR) on fundus images using several CNN architectures and Oversampling techniques. The novelty that we carry in this study is the use of the SMOTE oversampling technique to overcome the imbalance of data classes before the DR classification process is carried out. Balancing the amount of data in each class is expected to optimize the performance level of the classification model used. The CNN architectures used in this study include EfficientNet-B4, ResNet-50, DenseNet-201, Xception, and Inception-ResNet-v2.

II. METHODOLOGY

In order to classify diabetic retinopathy, there are several stages taken in this study starting from fundus dataset exploration, pre-processing (resizing, intensity normalization, noise removal, data augmentation, and background subtraction), data scenario determination (using the SMOTE oversampling technique to balance data classes and without using SMOTE), data partitioning, data training and testing using CNN models (EfficientNet-B4, ResNet-50, DenseNet-201, Xception, and Inception-ResNet-v2), and evaluation of the classification model. The Fig. 1 is an overview of the methodology in this research.

A. Fundus Dataset

This study utilized fundus images obtained from the Asia Pacific Tele-Ophthalmology Society 2019 Blindness Detection (APTOS 2019 BD) dataset [19]. The dataset comprises 3,662 samples collected from participants residing in rural India. It was curated by Aravind Eye Hospital, India. The fundus images were gathered under diverse conditions and environments over an extended timeframe. Subsequently, a team of trained physicians analyzed and labeled the samples based on the International Clinical Diabetic Retinopathy Severity Scale (ICDRSS). According to this classification system, the APTOS 2019 BD dataset is categorized into five groups: no Diabetic Retinopathy (DR), mild DR, moderate DR, severe DR, and proliferative DR. The distribution of image samples for the condition of no Diabetic Retinopathy is 1805, mild DR is 370, moderate DR is 999, severe DR is 193, and proliferative DR is 295. Based on the data distribution, it can be seen that each data class has a different number of samples. The following is a fundus image from the APTOS 2019 BD dataset.

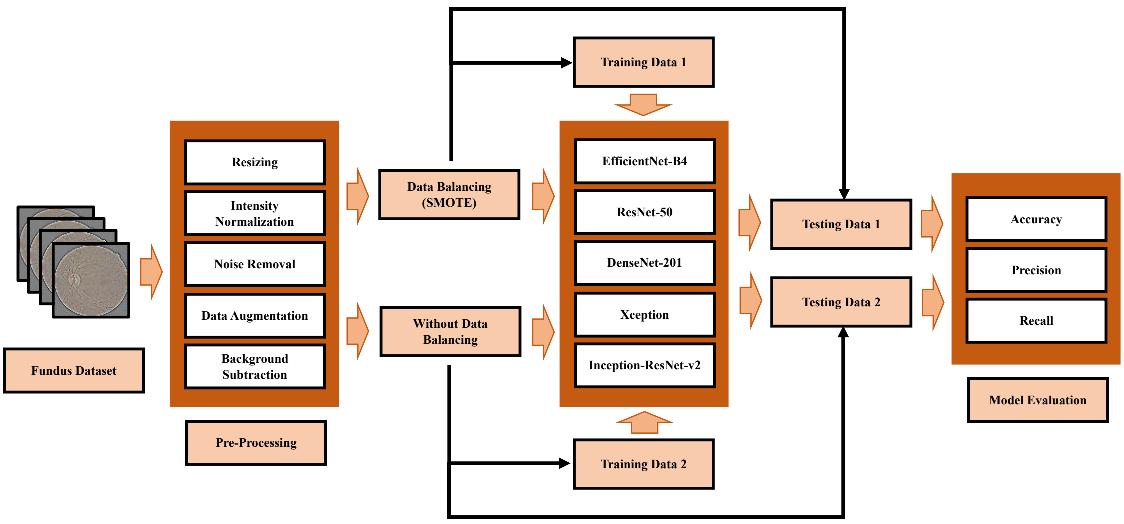

Fig. 1. Research methodology

The APTOS 2019 dataset (Asia Pacific Tele-Ophthalmology Society) presents several specific challenges for tasks such as diabetic retinopathy classification. One of the primary challenges is data imbalance, where certain classes, particularly those with higher severity levels, often have significantly fewer samples compared to more common classes [20]. This imbalance can lead to models being biased toward the majority class, reducing accuracy for minority classes [21]. Additionally, the retinal fundus images in the dataset often exhibit varying visual quality, such as inconsistent lighting, artifacts, noise, or blurriness, which can affect the model's ability to detect important features [22]. Another challenge is the similarity between classes, especially in borderline cases between two severity levels, making classification more difficult [23]. Therefore, preprocessing techniques, data augmentation, or balancing methods such as SMOTE are needed to enhance the model's performance when dealing with this dataset (Fig. 2).

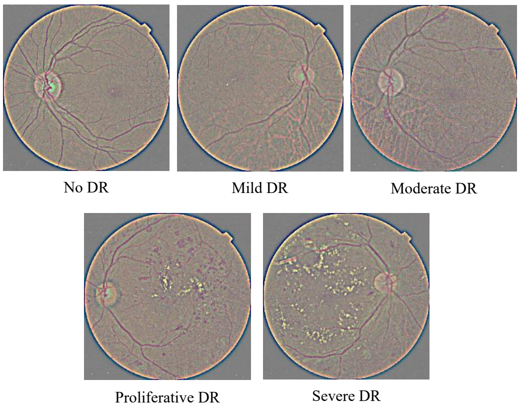

Fig. 2. Illustration of SMOTE

B. *Pre-Processing*

Image pre-processing in Convolutional Neural Network (CNN)-based Diabetic Retinopathy (DR) detection includes several important steps to ensure that the data is ready to be processed by the model with optimal results. One of the main steps is resizing, which aims to align the image dimensions according to the requirements of the CNN architecture [24]. This process ensures the consistency of the input dimensions, thus facilitating the model training process without losing important information from the original image. The next step is intensity normalization, which organizes the image pixel values into a certain range [25]. In this study, the selected pixel range is [0, 1]. This normalization helps the CNN model to be more stable and efficient during training, while reducing the influence of lighting variations between images. With a uniform distribution of pixel values, important features such as blood vessels, exudates, and hemorrhages become more visible, which is very important for improving the accuracy of DR detection.

Noise removal is also an important part of pre-processing. Retinal fundus images often contain noise from recording devices or artifacts. The Gaussian filter technique used in this study aims to reduce noise while retaining important edges and details [26]. With minimized noise, the CNN model can focus more on key features such as microaneurysms or retinal hemorrhages, which are important indicators of DR. Finally, data augmentation and background subtraction contribute significantly to improving the model performance. Data augmentation creates additional variations in the dataset through rotation, flipping, scaling, zooming, or changing color intensity, which helps reduce the risk of overfitting [27]. Meanwhile, background subtraction is used to remove irrelevant areas outside the retina, so that the model only processes important areas [28]. The combination of data augmentation and background subtraction ensures that the images used are of high quality, allowing the model to detect DR more accurately and reliably.

C. *Oversampling Technique*

Class imbalance is a common problem in Diabetic Retinopathy (DR) classification using fundus images [29]. Datasets such as APTOS 2019 BD often have an uneven distribution where classes such as “no DR” tend to have more samples than other classes such as “proliferative DR.” This imbalance can affect model performance, as the algorithm tends to focus more on the majority class, reducing the model’s ability to accurately recognize the minority class [30]. Therefore, a method is needed to balance the data so that the model can learn optimally across all classes. Synthetic Minority Over-sampling Technique (SMOTE) is one popular method to address data imbalance. SMOTE works by creating synthetic samples for the minority class through interpolation between existing samples [31]. This process is done by randomly selecting samples from the minority class, determining their nearest neighbors, and then creating new samples between those two points [32]. In this way, SMOTE adds variation to the minority class data without simply duplicating existing samples, reducing the risk of overfitting [33]. The Fig. 3 is an illustration of SMOTE in balancing data classes by synthesizing samples from the minority class.

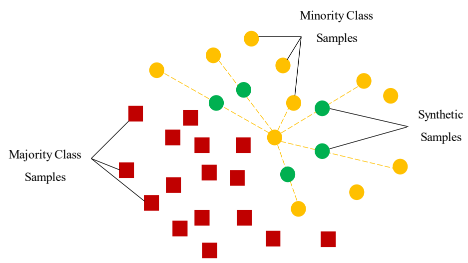

Fig. 3. Illustration of SMOTE

In the context of DR classification, SMOTE can be used to balance the distribution of classes such as “mild DR,” “moderate DR,” “severe DR,” and “proliferative DR.” By applying SMOTE, the number of samples in the minority classes can be increased to approach the number of samples in the majority class [34]. If the “proliferative DR” class only has 295 samples while “no DR” has 1,805 samples, SMOTE can generate additional samples for “proliferative DR” until they reach a balanced number. This ensures that the model has sufficient representation of each class. Using SMOTE before training a deep learning model, such as Convolutional Neural Networks (CNN), can improve the model’s performance in detecting all DR categories. With more balanced data, the model can learn relevant patterns for each class without bias towards the majority class [35]. As a result, evaluation metrics such as accuracy, precision, and recall on the minority class tend to increase. In addition, SMOTE helps ensure that the model not only performs well on the majority class but also provides reliable predictions for classes with low prevalence [36].

D. CNN Architectures

1) EfficientNet-B4

EfficientNet-B4 is a variant of the EfficientNet deep learning model architecture designed for image classification tasks [37]. EfficientNet uses a compound scaling approach to optimize model efficiency, namely by proportionally increasing the three main dimensions of the model, namely depth, width, and resolution of the input image [38]. EfficientNet-B4 offers a balance between high accuracy and computational efficiency, making it a popular choice for various applications that require a robust but resource-efficient model [39]. The main working principle of EfficientNet-B4 is compound scaling, where model enhancement is performed with a coordinated scale on the depth, width, and resolution of the input, rather than increasing only one aspect [40]. EfficientNet-B4 is built on the Mobile Inverted Bottleneck Convolution (MBConv) block, which is designed for efficiency by utilizing depthwise separable convolution and squeeze-and-excitation blocks to highlight important features [41]. With this approach, EfficientNet-B4 can achieve optimal performance on various dataset sizes without significantly increasing model complexity [42]. The following are the parameters of the EfficientNet-B4 architecture presented in Table I.

Table I describes the EfficientNet-B4 architectural parameters, which consist of several main stages, starting from the Stem stage to the Head stage. In the Stem stage, the model receives an input image measuring 380×380×3 and runs a Conv2D operation with a 3×3 kernel, producing 48 output channels with a stride of 2. Next, in stage 1, the model applies the MBConv1 block with 24 output channels. The MBConv6 block is applied in stages 2 and 3 with 32 and 48 output channels, using 3×3 and 5×5 kernels respectively and a stride of 2 to reduce the image resolution. In stages 4 and 5, the model uses the MBConv6 block again with 96 and 136 output channels, increasing the depth and number of features, and changing the kernel size to 5×5. Stages 6 and 7, which also use MBConv6, increase the number of output channels to 232 and 384, respectively, with more efficient and deeper convolution operations. Finally, in the Head stage, Conv2D operations followed by pooling and fully connected (FC) produce a final output with 1536 channels, which are used for classification. This architecture optimizes resource usage by proportionally increasing depth, width, and resolution using the compound scaling principle, allowing the model to achieve high performance with good computational efficiency.

2) ResNet-50

ResNet-50 is a variant of deep neural network architecture that uses the concept of residual learning to address the degradation problem in deeper networks [43]. ResNet-50 refers to the ResNet model with 50 layers, designed for image classification, often used for object recognition tasks on datasets such as ImageNet [44]. This model is known for being able to achieve high performance with a large number of layers, without experiencing overfitting or slow training problems [45]. The main working principle of ResNet-50 is the use of residual learning, which avoids the degradation problem that often occurs in deep neural networks [46]. With residual learning, the model learns to improve the results of previous layers by learning the difference between the output of the previous layer and the desired output [47]. This is done by adding skip connections or shortcuts that allow information to pass through one or more layers unchanged, so that gradients can flow more easily during the training process, reducing vanishing gradients [48]. Thus, ResNet-50 can be deeper and more efficient than more traditional models, without losing accuracy [49]. The following are the parameters of the ResNet-50 presented in Table II.

TABLE I. PARAMETERS OF THE EFFICIENTNET-B4

Stage	Input Resolution	Operator	Number of Blocks	Kernel Size	Expansion Ratio	Output Channels	Stride
Stem	380×380×3	Conv2D	1	3×3	-	48	2
1	190×190×48	MBConv1	1	3×3	1	24	1
2	190×190×24	MBConv6	2	3×3	6	32	2
3	95×95×32	MBConv6	2	5×5	6	48	2
4	48×48×48	MBConv6	3	3×3	6	96	2
5	24×24×96	MBConv6	5	5×5	6	136	1
6	24×24×136	MBConv6	5	5×5	6	232	2
7	12×12×232	MBConv6	3	3×3	6	384	1
Head	12×12×384	Conv2D + Pooling + FC	1	1×1	-	1536	-

TABLE II. PARAMETERS OF THE RESNET-50

Layer	Output Size	Operator	Number of Filters	Kernel Size	Stride	Activation Function
Initial Conv	112×112×64	Conv2D	64	7×7	2	ReLU
Max Pooling	56×56×64	MaxPool	-	3×3	2	-
Residual Block 1	56×56×256	Conv2D + BatchNorm + ReLU	64, 64, 256	1×1, 3×3, 1×1	1	ReLU
Residual Block 2	56×56×256	Conv2D + BatchNorm + ReLU	64, 64, 256	1×1, 3×3, 1×1	1	ReLU
Residual Block 3	28×28×512	Conv2D + BatchNorm + ReLU	128, 128, 512	1×1, 3×3, 1×1	2	ReLU
Residual Block 4	28×28×512	Conv2D + BatchNorm + ReLU	128, 128, 512	1×1, 3×3, 1×1	1	ReLU
Residual Block 5	14×14×1024	Conv2D + BatchNorm + ReLU	256, 256, 1024	1×1, 3×3, 1×1	2	ReLU
Residual Block 6	14×14×1024	Conv2D + BatchNorm + ReLU	256, 256, 1024	1×1, 3×3, 1×1	1	ReLU
Residual Block 7	7×7×2048	Conv2D + BatchNorm + ReLU	512, 512, 2048	1×1, 3×3, 1×1	2	ReLU
Residual Block 8	7×7×2048	Conv2D + BatchNorm + ReLU	512, 512, 2048	1×1, 3×3, 1×1	1	ReLU
Global Average Pooling	1×1×2048	GlobalAveragePooling	-	-	-	-
Fully Connected	1×1×1000	Dense	1000	-	-	Softmax

Table II illustrates the architectural parameters of ResNet-50 consisting of various layers and blocks that make up the network. Initially, the input image is processed through the first convolution layer with a 7×7 kernel, producing an output with 64 channels and a size of 112×112, followed by max pooling with a 3×3 kernel and stride 2 to reduce the image size to 56×56. Next, the model uses residual blocks, which consist of multiple convolution and batch normalization layers, with each block having two or three convolution layers, which are then followed by the ReLU activation function. The first and second residual blocks have 64 and 256 channels, using 1×1, 3×3, and 1×1 kernels, respectively. In the third and fourth residual blocks, the output increases to 512 channels, and the convolution kernels used are 1×1 and 3×3 with stride 2 to reduce the image dimension. In a further stage, ResNet-50 introduces a residual block with 1024 channels and a stride of 2 to further reduce the image dimensionality, followed by a block with 2048 channels in the final stage. After going through all the residual blocks, the model uses global average pooling to summarize the features, producing a 2048-channel output that is then processed by a fully connected layer with 1000 units, using the Softmax activation function for classification.

3) *DenseNet-201*

DenseNet-201 is one of the deep neural network architectures that uses the principle of dense connectivity where each layer is directly connected to all previous layers [50]. DenseNet-201 refers to the DenseNet model that has 201 layers, designed for image recognition and classification tasks [51]. Unlike conventional architectures, which connect each layer only to the next layer, DenseNet connects each layer to all previous layers, allowing for better information flow and more efficient feature reuse [52]. The main principle of DenseNet-201 is the concept of dense connectivity, which mitigates the vanishing gradient problem and enables very deep model training [53]. Each layer in DenseNet receives input from all previous layers, resulting in a more comprehensive feature map and improving the model's representation capabilities [54]. With this approach, DenseNet-201 not only improves information flow and gradients but also reduces the number of parameters because existing features are reused in subsequent layers [55]. This model also reduces memory requirements and redundant computation, making it efficient despite having many layers [56]. The following are the architectural parameters of DenseNet-201 presented in Table III.

Table III describes the architectural parameters of DenseNet-201 which consists of various layers that form this network. Starting with the first convolution layer using a 7×7 kernel with 64 channels, producing an output with a size of 112×112, followed by a max pooling layer to reduce the dimension to 56×56. Next, DenseNet-201 adopts several Dense blocks consisting of 3×3 convolutions with channels increasing gradually, such as 64, 128, 256, and finally 512 channels in the first block. Between each Dense block, there is a transition layer that reduces the number of channels by using 1×1 convolutions and pooling, which also reduces the spatial dimension of the image. This process is repeated for three additional Dense blocks with 512 and 1024 channels, reducing the image dimension from 56×56 to 28×28, 14×14, and finally 7×7 after each transition layer. After passing through all Dense blocks, the model uses global average pooling to summarize spatial information, producing an output of size 1×1×1024. Finally, a fully connected layer with 1000 units and a Softmax activation function is used for classification. This architecture optimizes the use of features through dense connections between layers, allowing DenseNet-201 to handle models with many layers efficiently.

4) *Xception*

Xception stands for Extreme Inception, which is an extension of the previous Inception architecture. The main difference between Xception and Inception is that Xception uses depthwise separable convolution instead of standard convolution [57]. This architecture is designed to improve computational efficiency and performance by reducing the number of parameters and speeding up the training process, making it suitable for image processing and object recognition applications [58]. The working principle of Xception focuses on the use of depthwise separable convolution, which divides the convolution operation into two separate steps [59]. First, depthwise convolution is applied to each input channel, and second, pointwise convolution (1×1) to combine information between channels [60]. This separable convolution replaces the usual convolution which usually uses a larger kernel and more parameters [61]. In this way, Xception reduces the number of parameters and speeds up computation, while maintaining the network's ability to extract complex features from images [62]. This allows Xception to handle large datasets and applications with high accuracy [63]. The following are the Xception architecture parameters presented in Table IV.

TABLE III. PARAMETERS OF THE DENSENET-201

Layer	Output Size	Operator	Number of Filters	Kernel Size	Stride	Activation Function
Initial Conv	112×112×64	Conv2D	64	7×7	2	ReLU
Max Pooling	56×56×64	MaxPool	-	3×3	2	-
Dense Block 1	56×56×256	Conv2D + BatchNorm + ReLU	64, 64, 128, 256	3×3	1	ReLU
Transition Layer 1	28×28×128	Conv2D + AvgPool	128	1×1, 2×2	2	-
Dense Block 2	28×28×512	Conv2D + BatchNorm + ReLU	128, 128, 256, 512	3×3	1	ReLU
Transition Layer 2	14×14×256	Conv2D + AvgPool	256	1×1, 2×2	2	-
Dense Block 3	14×14×1024	Conv2D + BatchNorm + ReLU	256, 256, 512, 1024	3×3	1	ReLU
Transition Layer 3	7×7×512	Conv2D + AvgPool	512	1×1, 2×2	2	-
Dense Block 4	7×7×1024	Conv2D + BatchNorm + ReLU	512, 512, 1024	3×3	1	ReLU
Global Average Pooling	1×1×1024	GlobalAvgPool	-	-	-	-
Fully Connected	1×1×1000	Dense	1000	-	-	Softmax

TABLE IV. PARAMETERS OF THE XCEPTION

Layer	Output Size	Operator	Number of Filters	Kernel Size	Stride	Activation Function
Initial Conv	299×299×32	Conv2D	32	3×3	2	ReLU
Depthwise Separable Conv 1	149×149×64	Depthwise Conv2D + Pointwise Conv2D	64	3×3, 1×1	1	ReLU
Depthwise Separable Conv 2	149×149×128	Depthwise Conv2D + Pointwise Conv2D	128	3×3, 1×1	2	ReLU
Depthwise Separable Conv 3	73×73×256	Depthwise Conv2D + Pointwise Conv2D	256	3×3, 1×1	2	ReLU
Depthwise Separable Conv 4	37×37×728	Depthwise Conv2D + Pointwise Conv2D	728	3×3, 1×1	2	ReLU
Depthwise Separable Conv 5	19×19×728	Depthwise Conv2D + Pointwise Conv2D	728	3×3, 1×1	2	ReLU
Depthwise Separable Conv 6	19×19×728	Depthwise Conv2D + Pointwise Conv2D	728	3×3, 1×1	1	ReLU
Depthwise Separable Conv 7	10×10×1024	Depthwise Conv2D + Pointwise Conv2D	1024	3×3, 1×1	2	ReLU
Depthwise Separable Conv 8	10×10×1024	Depthwise Conv2D + Pointwise Conv2D	1024	3×3, 1×1	1	ReLU
Global Average Pooling	1×1×1024	GlobalAvgPool	-	-	-	-
Fully Connected	1×1×1000	Dense	1000	-	-	Softmax

Table IV describes the Xception architecture parameters consisting of the various layers that make up the Xception network. Starting with an initial convolution layer, which uses a 3×3 kernel with 32 channels and produces an output of size 299×299. After that, the network continues with a series of depthwise separable convolutions consisting of two stages, the first stage using depthwise convolution for each channel separately and the second stage using pointwise convolution (1×1) to combine information across channels. Each depthwise separable convolution layer in Xception has two parts: a 3×3 kernel for depthwise convolution and a 1×1 kernel for pointwise convolution. In the first block, the output is 149×149 with 64 channels, and this process is repeated with increasing numbers of channels (128, 256, and 728) through the next few blocks. The transition layer reduces the spatial dimension of the image through stride 2 in some blocks, while other layers maintain the spatial size with stride 1. After several blocks of depthwise separable convolution, the architecture continues with a global average pooling layer that summarizes the spatial information of the extracted features and produces a 1×1 output with 1024 channels. At the end of the architecture, a fully connected layer with 1000 units and a Softmax activation function is used for classification, producing a final output representing the prediction for each class.

5) Inception-ResNet-v2

Inception-ResNet-v2 is a deep neural network architecture that combines the advantages of two major approaches in deep learning model development, namely Inception and ResNet (Residual Networks) [64]. This model is designed to improve the efficiency and accuracy of pattern recognition from images, especially for tasks such as classification and object detection [65]. Inception-ResNet-v2 combines the Inception module for multi-scale feature exploration with the residual learning technique from ResNet to accelerate training convergence and overcome the degradation problem that often occurs in very deep networks [66]. This architecture was developed by Google researchers and is a further development of the previous version, namely Inception-ResNet-v1 [67]. The working principle of Inception-ResNet-v2 lies in the combination of Inception blocks and residual connections [68]. The Inception block allows the network to extract features from various scales using filters with different kernel sizes (eg 1×1, 3×3, and 5×5) in one layer [69]. Residual connections, introduced in ResNet, add shortcut connections that skip several layers and directly connect inputs to outputs [70]. This helps mitigate the vanishing gradient problem in very deep networks and allows for more efficient model training [71]. In Inception-ResNet-v2, each Inception block has a residual connection that adds the original features to the features generated by the Inception module, thus accelerating learning and improving accuracy [72]. The following are the architectural parameters of Inception-ResNet-v2 presented in Table V.

Table V illustrates the architectural parameters of Inception-ResNet-v2, which consists of the main structure of the model that combines Inception blocks with residual connections to improve efficiency and accuracy. In the early stage (Stem), the model uses multiple successive convolution layers with small filters (3×3 and 1×1) and pooling to efficiently reduce the spatial dimension of the input image while extracting initial features. After that, the network enters three main types of blocks: Inception-ResNet-A, Inception-ResNet-B, and Inception-ResNet-C, each designed to capture feature patterns with increasing complexity. Each Inception block combines multi-scale convolution operations with residual connections to accelerate training convergence and retain important information from the input. At each stage, reduction blocks (Reduction-A and Reduction-B) are used to significantly reduce the spatial dimension while increasing the feature depth (number of channels), allowing for more abstract and dense information processing. For example, the Reduction-A block reduces the size from 35×35 to 17×17 while increasing the number of channels to 1088. On the other hand, the Reduction-B block reduces the size from 17×17 to 8×8 with an increase in the number of channels to 2080. This stage is very important to simplify the data representation before moving on to the final stage. The Inception-ResNet-A, B, and C blocks each use filters with varying kernel sizes (e.g. 1×1, 3×3, 7×1) to extract features at different spatial scales, while the residual connections keep the initial information of the input from being lost during the process. The output of each residual block is added to the original input, improving the training stability and allowing very deep network training without experiencing the vanishing gradient problem. At the end of the network, a Global Average Pooling layer is used to summarize the spatial features into a single feature vector with dimensions of 1×1×2080. This vector then passes through a fully connected layer with Softmax activation function to generate probability predictions for 1000 classes.

E. Model Evaluation

Model evaluation matrices are metrics used to assess the performance of classification models, including Convolutional Neural Networks (CNN)-based models. Common evaluation matrices such as accuracy, precision, and recall are derived from a confusion matrix that maps model predictions to actual labels in four categories, such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [73]. Accuracy measures the percentage of correct model predictions from the entire data, while precision focuses on how accurate the model's positive predictions are [74]. Meanwhile, recall, also known as sensitivity, measures the extent to which the model can correctly detect positive data from all the positive data available [75]. The mathematical equations for these three metrics are as follows:

1) Accuracy

Accuracy shows the proportion of correct predictions (both positive and negative) compared to the entire data. However, accuracy can be misleading if the dataset is imbalanced, for example if one class is more dominant.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2) Precision

Precision measures the ratio of correct positive predictions to all positive predictions. This is important for cases where the consequences of false positives are high, such as in disease detection.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3) Recall

Recall measures the model’s ability to capture all true positive data. This is important in situations where missed detection has a major impact, such as in medical diagnosis.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

In CNN-based classification, accuracy, precision, and recall metrics are used to evaluate the model’s performance on previously unseen test data [76]. A good CNN should strike a good balance between these three metrics, especially on datasets with imbalanced classes [77]. For example, in the task of disease detection using medical images, high precision is important to reduce false positives, while high recall is important to ensure that no cases of a disease are missed [78]. By using these three metrics together, researchers can evaluate not only the model’s ability to make correct predictions overall, but also how well it handles positive and negative data specifically [79].

III. RESULTS AND DISCUSSIONS

As previously explained, this study uses the APTOS 2019 BD dataset for the Diabetic Retinopathy classification process based on CNN architecture. This fundus image dataset has a different amount of data in each class. The data distribution from each class is 1805 non-Diabetic Retinopathy data, 370 mild DR data, 999 moderate DR data, 193 severe DR data, and 295 proliferative DR data [19]. So we apply 2 types of scenarios in the Diabetic Retinopathy classification process.

TABLE V. PARAMETERS OF THE INCEPTION-RESNET-V2

Stage	Block/Layer	Output Size	Operator	Number of Filters	Kernel Size	Stride	Activation
Stem	Conv2D	149×149×32	Conv2D	32	3×3	2	ReLU
-	Conv2D	147×147×32	Conv2D	32	3×3	1	ReLU
-	Conv2D	147×147×64	Conv2D	64	3×3	1	ReLU
-	MaxPooling	73×73×64	MaxPooling	-	3×3	2	-
-	Conv2D	73×73×80	Conv2D	80	1×1	1	ReLU
-	Conv2D	71×71×192	Conv2D	192	3×3	1	ReLU
-	MaxPooling	35×35×192	MaxPooling	-	3×3	2	-
Inception-ResNet-A	10× Block A	35×35×320	Conv2D + Residual Connection	[32, 32, 64, 96, 96]	1×1, 3×3	1	ReLU
Reduction-A	Reduction Block A	17×17×1088	Conv2D + MaxPooling	[384, 256, 256]	3×3, 1×1	2	ReLU
Inception-ResNet-B	20× Block B	17×17×1088	Conv2D + Residual Connection	[128, 128, 256, 896]	1×1, 7×1, 1×7	1	ReLU
Reduction-B	Reduction Block B	8×8×2080	Conv2D + MaxPooling	[256, 384, 256, 256]	1×1, 3×3	2	ReLU
Inception-ResNet-C	10× Block C	8×8×2080	Conv2D + Residual Connection	[192, 192, 256, 1792]	1×1, 3×3	1	ReLU
Final Layers	Average Pooling	1×1×2080	Global Average Pooling	-	-	-	-
-	Fully Connected	1×1×1000	Dense	1000	-	-	Softmax

The first scenario, the fundus image dataset will be directly partitioned into training and testing data. The training data will be used as input for training CNN architectures such as EfficientNet-B4, ResNet-50, DenseNet-201, Xception, and Inception-ResNet-v2, and the results will be tested with testing data to see the performance level of each CNN architecture in the Diabetic Retinopathy classification process. While in the second scenario, the fundus image dataset will be balanced by data class using the oversampling technique (SMOTE) and then partitioned into training and testing data. The use of SMOTE aims to synthesize data samples from classes with a small amount of data and then the data samples will be increased until all classes have the same amount of data (balanced) [80]. Similar to scenario 1, the training data will be used as input for training CNN architectures and the results will be tested using testing data. The results of the accuracy, precision, and recall of the CNN architecture in scenarios 1 and 2 will be compared to observe the influence or effectiveness of using the oversampling technique (SMOTE) in the classification of Diabetic Retinopathy.

Based on the test results using several Convolutional Neural Networks (CNN) architectures, as shown in Fig. 4, a comparison of the classification performance of Diabetic Retinopathy (DR) in two different scenarios was obtained. In scenario 1 (without data balancing), the model performance showed quite varied accuracy. Xception recorded the highest accuracy of 80.61%, followed by Inception-ResNet-v2 with 77.55%, DenseNet-201 at 76.53%, ResNet-50 at 74.83%, and EfficientNet-B4 at 74.15%. These results show that architectures such as Xception are able to overcome data imbalance better than other models, but their performance is still limited by the dominance of the majority class. When data balancing was carried out using the SMOTE method in scenario 2, the model accuracy increased significantly. The highest accuracy rate was obtained by EfficientNet-B4 with 97.78%, followed by DenseNet-201 at 97.09%, ResNet-50 at 96.95%, Xception at 96.81%, and Inception-ResNet-v2 at 91.14%. This improvement shows that SMOTE successfully overcomes the problem of class imbalance in the dataset, allowing the model to learn better towards the minority class. The effectiveness of this method can be seen from the spike in performance across all CNN architectures, indicating that balanced data plays an important role in improving classification performance.

In addition to accuracy, precision is an important evaluation metric in assessing the performance of CNN models, especially in detecting DR (Fig. 5). In scenario 1, the highest precision was obtained by Xception at 73.76%, followed by EfficientNet-B4 at 73.33%, DenseNet-201 at 70.23%, Inception-ResNet-v2 at 67.86%, and ResNet-50 at 65.65%. However, the precision increased drastically after data balancing using SMOTE in scenario 2. DenseNet-201 recorded the highest precision of 99.28%, followed by EfficientNet-B4 with 98.75%, ResNet-50 with 98.56%, Xception with 97.68%, and Inception-ResNet-v2 with 90.77%. This increase shows that the SMOTE method is able to improve the model's ability to avoid false positives, so that predictions become more accurate. Meanwhile, recall, which measures the model's ability to detect positive classes, also showed a significant increase (Fig. 6). In scenario 1, the highest recall was obtained by Xception with 83.87%, followed by Inception-ResNet-v2 with 81.90%, DenseNet-201 with 75.41%, ResNet-50 with 74.78%, and EfficientNet-B4 with 71.22%. After data balancing using SMOTE in scenario 2, recall increased significantly. Inception-ResNet-v2 recorded the highest recall of 98.57%, followed by EfficientNet-B4 of 98.40%, Xception of 98.20%, ResNet-50 of 97.50%, and DenseNet-201 of 96.99%. This improvement indicates that the model can detect more positive cases more accurately after the data is balanced.

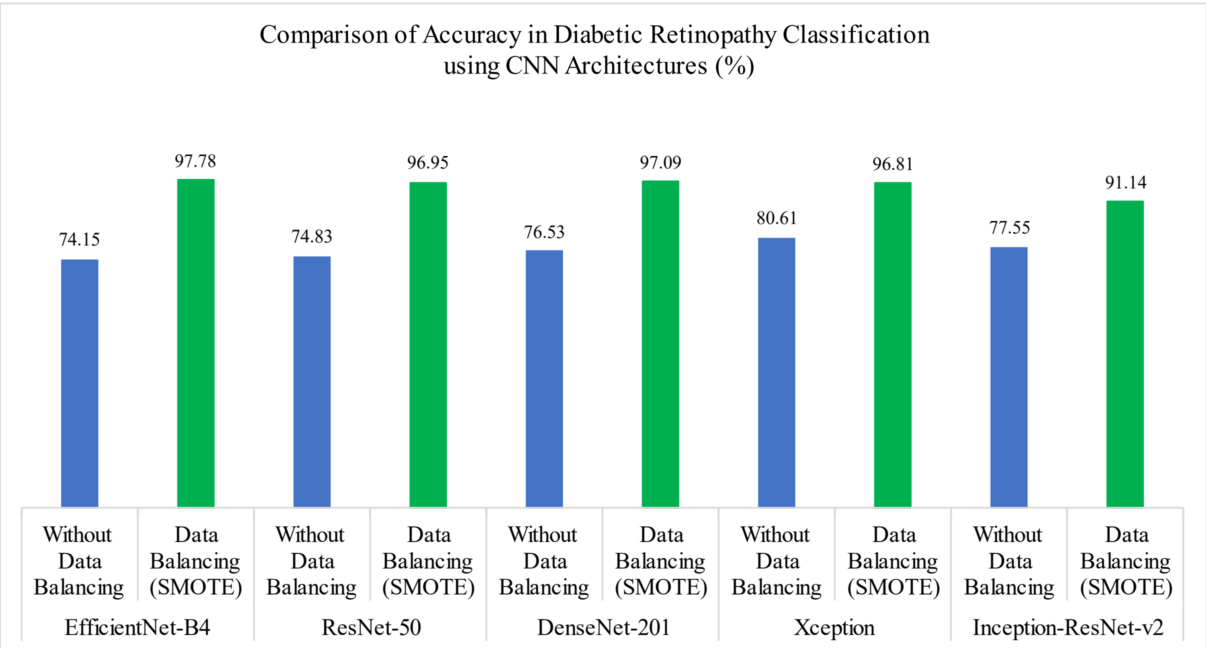

Fig. 4. Comparison of Diabetic Retinopathy Classification Accuracy using CNN Architecture with and without Data Balancing (SMOTE)

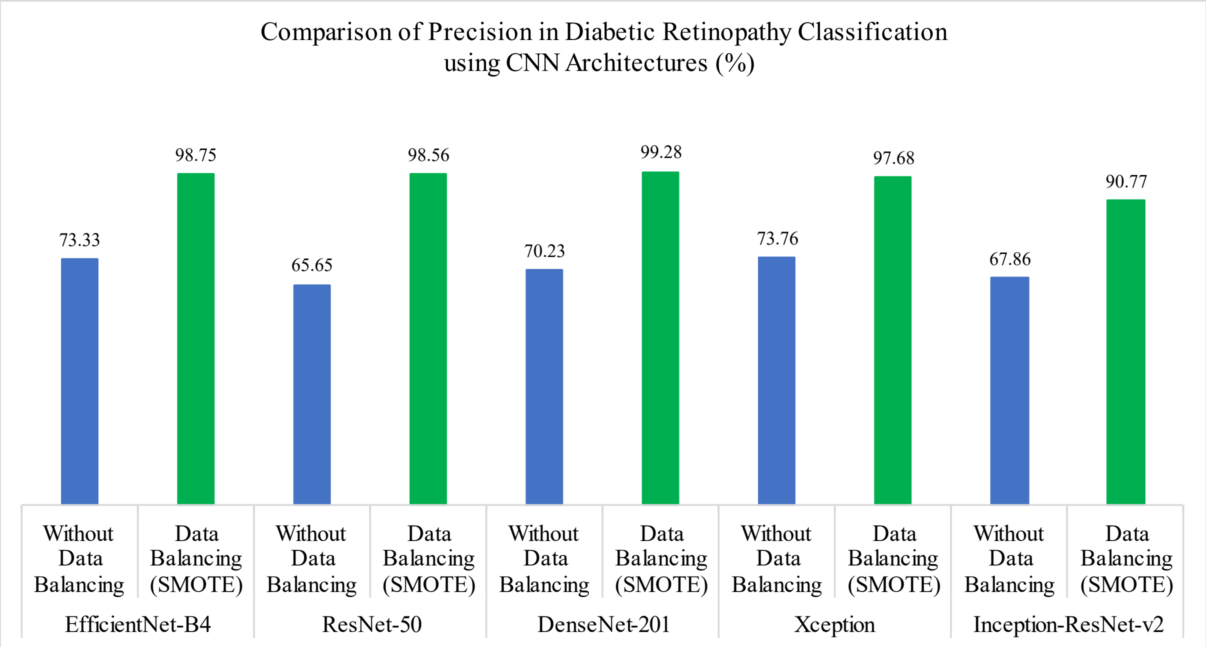

Fig. 5. Comparison of Diabetic Retinopathy Classification Precision using CNN Architecture with and without Data Balancing (SMOTE)

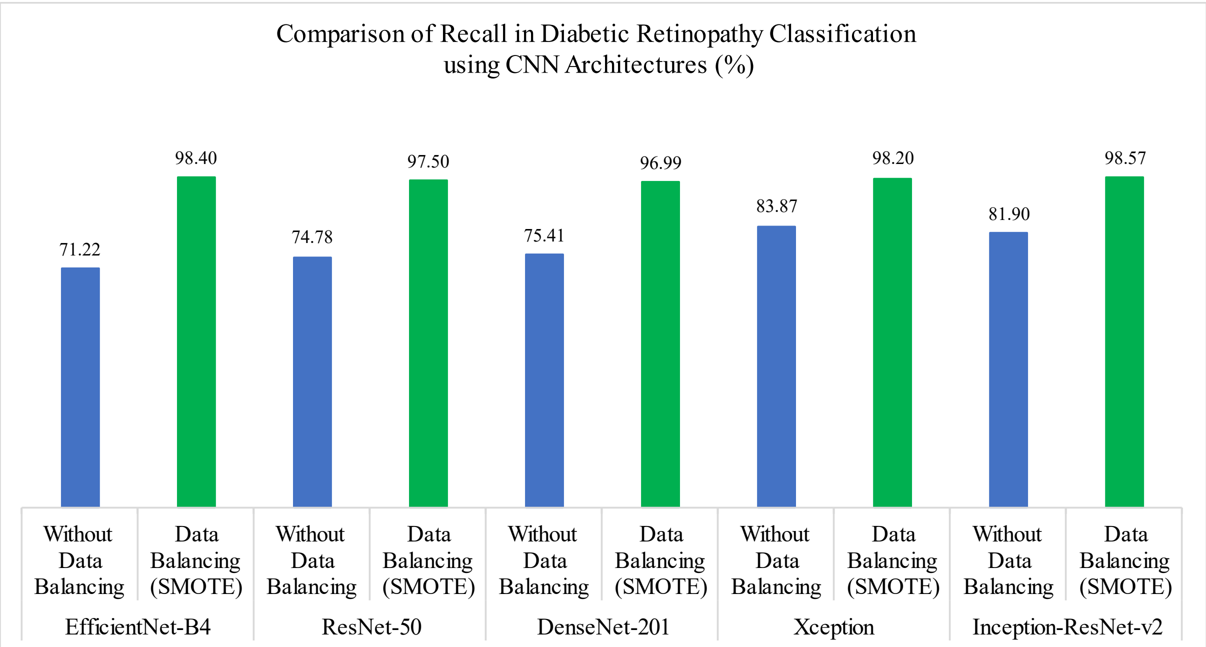

Fig. 6. Comparison of Diabetic Retinopathy Classification Recall using CNN Architecture with and without Data Balancing (SMOTE)

Overall, the test results emphasize the importance of implementing the SMOTE method in dealing with data imbalance in CNN-based Diabetic Retinopathy (DR) classification. Data imbalance is a common problem in medical datasets, where the number of samples for minority classes, such as severe DR and proliferative DR, tends to be much smaller than the majority class [81]. Without data balancing, the model tends to be biased towards the majority class, so that the performance in detecting rare disease conditions is not optimal [82]. The application of SMOTE successfully creates synthetic samples for the minority class, so that the data distribution becomes more balanced and the model can learn the patterns of each class more effectively [83]. Modern architectures such as EfficientNet-B4 and DenseNet-201 show significant performance improvements after applying SMOTE compared to scenarios without data balancing [84]. This can be seen from the spike in evaluation metrics, such as accuracy, precision, and recall, indicating that the model can better classify various levels of DR severity. The efficiency of architectures such as EfficientNet-B4, which optimizes the network scale in terms of depth, width, and resolution, contributes to more accurate detection capabilities [85]. Meanwhile, DenseNet-201 with its dense connectivity mechanism enables better feature propagation, which is very helpful in understanding the complex features of the retinal fundus images used in this study.

With the increasing ability of the model to detect positive classes accurately and consistently, the implementation of the SMOTE method can improve the reliability of the early detection system for Diabetic Retinopathy [86]. This system can help medical professionals make faster and more precise diagnoses, especially for cases with high severity conditions such as severe DR and proliferative DR that require immediate treatment [87]. Early detection is crucial because Diabetic Retinopathy is a progressive disease that can cause permanent blindness if not treated promptly [88]. Therefore, the combination of an optimal CNN architecture and an effective data balancing method opens up significant opportunities in the development of artificial intelligence-based systems in the field of ophthalmology [89]. Furthermore, this study shows that the SMOTE method not only improves model performance but also provides a fairer approach to disease classification. With a balanced data distribution, each disease category has an equal chance of being recognized by the model, so that the classification results are more reliable [90]. In addition, this implementation has the potential to be integrated into clinical decision support systems that can be used by doctors or medical personnel as a diagnostic tool [91].

Table VI compares the iteration time per round for various CNN architectures under two conditions, without data balancing and with data balancing using SMOTE (Synthetic Minority Oversampling Technique). It is evident that applying data balancing significantly increases the iteration time across all architectures. For instance, EfficientNet-B4 has an iteration time of 53.14 seconds without data balancing, which rises to 124.14 seconds with SMOTE. Similarly, ResNet-50 shows an increase from 31.84 seconds to 72.80 seconds. DenseNet-201 experiences the most significant rise, from 59.16 seconds to 142.16 seconds, while Inception-ResNet-v2 also exhibits a substantial increase, from 52.14 seconds to 132.15 seconds. The smallest relative increase is observed in ResNet-50, and the largest in DenseNet-201. This indicates that while SMOTE improves data balance, it imposes a computational cost, likely due to the additional synthetic data generation and processing required for balanced datasets. The results highlight a trade-off between computational efficiency and data balancing in CNN training. In the future, further optimization through a combination of the SMOTE method with other performance improvement techniques, such as ensemble learning or transfer learning, can further improve the accuracy and efficiency of the CNN-based Diabetic Retinopathy early detection system.

TABLE VI. COMPARISON OF CNN ARCHITECTURES COMPUTATION TIMES

CNN Architectures	Iteration Time per Round (s)	
	Without Data Balancing	**Data Balancing (SMOTE)**
EfficientNet-B4	53.14	124.14
ResNet-50	31.84	72.80
DenseNet-201	59.16	142.16
Xception	33.91	81.90
Inception-ResNet-v2	52.14	132.15

IV. CONCLUSIONS

Based on the test results, the use of the SMOTE method has proven effective in handling data imbalance in CNN-based Diabetic Retinopathy (DR) classification. In scenarios without data balancing, model performance tends to be limited by the dominance of the majority class, with Xception showing the highest performance in accuracy, precision, and recall. However, when data balancing is carried out using SMOTE, there is a significant increase in all evaluation metrics across various CNN architectures. The highest accuracy increase was achieved by EfficientNet-B4 with 97.78%, followed by DenseNet-201, ResNet-50, Xception, and Inception-ResNet-v2. This shows that data balancing allows the model to learn more effectively from the minority class, so that classification performance becomes more optimal. Similar improvements are also seen in precision and recall, with DenseNet-201 recording the highest precision and Inception-ResNet-v2 recording the highest recall after applying SMOTE. These results confirm that SMOTE not only improves the overall accuracy of the model, but also improves the model's ability to detect positive cases (recall) and avoid false positives (precision). Architectures such as EfficientNet-B4 and DenseNet-201 have proven to be very reliable in utilizing balanced data, making them strong candidates for the implementation of early detection systems for DR. Thus, the use of the SMOTE method in the classification of Diabetic Retinopathy is highly recommended to improve the accuracy of the diagnosis. Although data balancing can help improve model quality by dealing with data imbalances, it comes at a higher computational cost. Using data balancing techniques with SMOTE significantly increased the iteration time per round on all tested CNN architectures. Therefore, it is important to consider the trade-off between improved model performance and increased computational time when applying techniques like SMOTE, especially for more complex architectures such as DenseNet-201 and Inception-ResNet-v2. This implementation can contribute significantly to the clinical decision support system, helping medical personnel to detect and treat Diabetic Retinopathy more quickly and accurately, thereby preventing the risk of blindness due to late diagnosis.

ACKNOWLEDGMENT

The authors would like to acknowledge the Department of Medical Technology, Institut Teknologi Sepuluh Nopember, for the facilities and support in this research. The authors also gratefully acknowledge financial support from the Institut Teknologi Sepuluh Nopember for this work, under project scheme of the Publication Writing and IPR Incentive Program (PPHKI) 2025.

REFERENCES

[1] W. L. Alyoubi, W. M. Shalash, and M. F. Abulkhair, “Diabetic retinopathy detection through deep learning techniques: A review,” *Informatics in Medicine Unlocked*, vol. 20, p. 100377, 2020, doi: 10.1016/j.imu.2020.100377.

[2] H. Sun *et al*., “IDF diabetes Atlas: Global, regional and country-level diabetes prevalence estimates for 2021 and projections for 2045,” *Diabetes Research and Clinical Practice*, vol. 183, no. 109119, Dec. 2021, doi: 10.1016/j.diabres.2021.109119.

[3] Z. Zlatarova *et al*., “Prevalence of Diabetic Retinopathy Among Diabetic Patients from Northeastern Bulgaria,” *Diagnostics*, vol. 14, no. 20, pp. 2340–2340, Oct. 2024, doi: 10.3390/diagnostics14202340.

[4] J. Chua, C. X. Y. Lim, T. Y. Wong, and C. Sabanayagam, “Diabetic retinopathy in the Asia-Pacific,” *Asia Pac. J. Ophthalmol. (Phila.)*, vol. 7, no. 1, pp. 3–16, Jan. 2018, doi: 10.22608/APO.2017511.

[5] P. Shrestha, R. Kaiti, and R. Shyangbo, “Blindness among patients with type II diabetes mellitus presenting to the Outpatient Department of Ophthalmology of a tertiary care centre: A descriptive cross-sectional

study," *JNMA J. Nepal Med. Assoc*., vol. 60, no. 254, pp. 877–880, Oct. 2022. doi: 10.31729/jnma.7702.

[6] T. H. Fung, B. Patel, E. G. Wilmot, and W. M. Amoaku, "Diabetic retinopathy for the non-ophthalmologist," *Clinical Medicine*, vol. 22, no. 2, pp. 112–116, Mar. 2022, doi: 10.7861/clinmed.2021-0792.

[7] T.-E. Tan and T. Y. Wong, "Diabetic retinopathy: Looking forward to 2030," *Frontiers in Endocrinology*, vol. 13, Jan. 2023, doi: 10.3389/fendo.2022.1077669.

[8] Z. Yang, T.-E. Tan, Y. Shao, T. Y. Wong, and X. Li, "Classification of diabetic retinopathy: Past, present and future," *Frontiers in Endocrinology*, vol. 13, Dec. 2022, doi: 10.3389/fendo.2022.1079217.

[9] E. Alizadeh, P. Mammadzada, and H. André, "The Different Facades of Retinal and Choroidal Endothelial Cells in Response to Hypoxia," *International Journal of Molecular Sciences*, vol. 19, no. 12, pp. 3846–3846, Dec. 2018, doi: 10.3390/ijms19123846.

[10] A. Sebastian, O. Elharrouss, S. Al-Maadeed, and N. Almaadeed, "A Survey on Deep-Learning-Based Diabetic Retinopathy Classification," *Diagnostics*, vol. 13, no. 3, p. 345, Jan. 2023, doi: 10.3390/diagnostics13030345.

[11] X. Huang *et al*., "Artificial intelligence promotes the diagnosis and screening of diabetic retinopathy," *Frontiers in Endocrinology*, vol. 13, Sep. 2022, doi: 10.3389/fendo.2022.946915.

[12] L. F. Nakayama *et al*., "Diabetic retinopathy classification for supervised machine learning algorithms," *International Journal of Retina and Vitreous*, vol. 8, no. 1, Jan. 2022, doi: 10.1186/s40942-021-00352-2.

[13] G. Alwakid, W. Gouda, M. Humayun, and N. Z. Jhanjhi, "Deep learning-enhanced diabetic retinopathy image classification," *Digital health*, vol. 9, Jan. 2023, doi: 10.1177/20552076231194942.

[14] P. Zang *et al*., "A Diabetic Retinopathy Classification Framework Based on Deep-Learning Analysis of OCT Angiography," *Translational vision science & technology*, vol. 11, no. 7, pp. 10–10, Jul. 2022, doi: 10.1167/tvst.11.7.10.

[15] C. Suedumrong, S. Phongmoo, T. Akarajaka, and K. Leksakul, "Diabetic Retinopathy Detection Using Convolutional Neural Networks with Background Removal, and Data Augmentation," *Applied Sciences*, vol. 14, no. 19, p. 8823, Sep. 2024, doi: 10.3390/app14198823.

[16] P. Zhang *et al*., "Fundus Image Generation and Classification of Diabetic Retinopathy Based on Convolutional Neural Network," *Electronics*, vol. 13, no. 18, p. 3603, Sep. 2024, doi: 10.3390/electronics13183603.

[17] A. Dutta *et al*., "Early Prediction of Diabetes Using an Ensemble of Machine Learning Models," *International Journal of Environmental Research and Public Health*, vol. 19, no. 19, p. 12378, Sep. 2022, doi: 10.3390/ijerph191912378.

[18] A. M. Fayyaz, M. I. Sharif, S. Azam, A. Karim, and J. El-Den, "Analysis of Diabetic Retinopathy (DR) Based on the Deep Learning," *Information*, vol. 14, no. 1, p. 30, Jan. 2023, doi: 10.3390/info14010030.

[19] Karthik, Maggie, and S. Dane, "APTOS 2019 Blindness Detection 2019," *Kaggle*, 2019, https://kaggle.com/competitions/aptos2019-blindness-detection.

[20] M. M. Hassan and H. R. Ismail, "Bayesian Deep Learning Applied to Diabetic Retinopathy with Uncertainty Quantification," *Heliyon*, pp. e41802–e41802, Jan. 2025, doi: 10.1016/j.heliyon.2025.e41802.

[21] T. Karkera, C. Adak, S. Chattopadhyay, and M. Saqib, "Detecting severity of Diabetic Retinopathy from fundus images: A transformer network-based review," *Neurocomputing*, vol. 597, p. 127991, Sep. 2024, doi: 10.1016/j.neucom.2024.127991.

[22] A. Rahman *et al*., "Diabetic Retinopathy Detection: A Hybrid Intelligent Approach," Computers, materials & continua/Computers, materials & continua (Print), vol. 80, no. 3, pp. 4561–4576, Jan. 2024, doi: 10.32604/cmc.2024.055106.

[23] M. Gargi, R. K. Eluri, O. P. Samantray, and K. Hajarathaiah, "Compact Pyramidal dense mixed attention network for Diabetic retinopathy severity prediction under deep learning," *Biomedical Signal Processing and Control*, vol. 100, pp. 106960–106960, Oct. 2024, doi: 10.1016/j.bspc.2024.106960.

[24] S. Abbasi *et al*., "Classification of diabetic retinopathy using unlabeled data and knowledge distillation," *Artificial Intelligence in Medicine*, vol. 121, p. 102176, Nov. 2021, doi: 10.1016/j.artmed.2021.102176.

[25] S. S. Mondal, N. Mandal, K. K. Singh, A. Singh, and I. Izonin, "EDLDR: An Ensemble Deep Learning Technique for Detection and Classification of Diabetic Retinopathy," *Diagnostics*, vol. 13, no. 1, p. 124, Dec. 2022, doi: 10.3390/diagnostics13010124.

[26] D. Saproo, A. N. Mahajan, and S. Narwal, "Deep learning based binary classification of diabetic retinopathy images using transfer learning approach," *Journal of Diabetes & Metabolic Disorders*, Sep. 2024, doi: 10.1007/s40200-024-01497-1.

[27] J. Fan *et al*., "A Self-Supervised Equivariant Refinement Classification Network for Diabetic Retinopathy Classification," *Journal of Imaging Informatics in Medicine*, Sep. 2024, doi: 10.1007/s10278-024-01270-z.

[28] H. K. Vasireddi, S. D. K, and R. R. G N V, "Deep feed forward neural network–based screening system for diabetic retinopathy severity classification using the lion optimization algorithm," *Graefe's Archive for Clinical and Experimental Ophthalmology*, Sep. 2021, doi: 10.1007/s00417-021-05375-x.

[29] H. Shakibania, S. Raoufi, B. Pourafkham, H. Khotanlou, and M. Mansoorizadeh, "Dual branch deep learning network for detection and stage grading of diabetic retinopathy," *Biomedical Signal Processing and Control*, vol. 93, pp. 106168–106168, Jul. 2024, doi: 10.1016/j.bspc.2024.106168.

[30] S. Madarapu, S. Ari, and K. Mahapatra, "DFCAFNet: Dual-feature co-attentive fusion network for diabetic retinopathy grading," *Biomedical Signal Processing and Control*, vol. 96, pp. 106564–106564, Jun. 2024, doi: 10.1016/j.bspc.2024.106564.

[31] C. Huang, M. Sarabi, and A. E. Ragab, "MobileNet-V2 /IFHO Model for Accurate Detection of Early-Stage Diabetic Retinopathy," *Heliyon*, pp. e37293–e37293, Aug. 2024, doi: 10.1016/j.heliyon.2024.e37293.

[32] K. Ashwini and R. Dash, "Improving Diabetic Retinopathy grading using Feature Fusion for limited data samples," *Computers and Electrical Engineering*, vol. 120, p. 109782, Dec. 2024, doi: 10.1016/j.compeleceng.2024.109782.

[33] K. Ashwini and R. Dash, "Grading diabetic retinopathy using multiresolution based CNN," *Biomedical Signal Processing and Control*, vol. 86, p. 105210, Sep. 2023, doi: 10.1016/j.bspc.2023.105210.

[34] S. Piri, D. Delen, T. Liu, and H. M. Zolbanin, "A data analytics approach to building a clinical decision support system for diabetic retinopathy: Developing and deploying a model ensemble," *Decision Support Systems*, vol. 101, pp. 12–27, Sep. 2017, doi: 10.1016/j.dss.2017.05.012.

[35] J. P-Fontanilles, A. Valls, and P. R-Aroca, "Multivariate data binning and examples generation to build a Diabetic Retinopathy classifier based on temporal clinical and analytical risk factors," *Knowledge-Based Systems*, vol. 300, pp. 112154–112154, Jun. 2024, doi: 10.1016/j.knosys.2024.112154.

[36] S. Nayak *et al*., "Development of a machine learning-based model for the prediction and progression of diabetic kidney disease: A single centred retrospective study," *International Journal of Medical Informatics*, vol. 190, pp. 105546–105546, Jul. 2024, doi: 10.1016/j.ijmedinf.2024.105546.

[37] Y. Fu, Y. Ju, and D. Zhang, "MSEF-Net: A multi-scale EfficientNet Fusion for Diabetic Retinopathy grading," *Biomedical Signal Processing and Control*, vol. 98, pp. 106714–106714, Aug. 2024, doi: 10.1016/j.bspc.2024.106714.

[38] R. Raza *et al*., "Lung-EffNet: Lung cancer classification using EfficientNet from CT-scan images," *Engineering Applications of Artificial Intelligence*, vol. 126, pp. 106902–106902, Nov. 2023, doi: 10.1016/j.engappai.2023.106902.

[39] S. Tripathy, R. Singh, and M. Ray, "Automation of Brain Tumor Identification using EfficientNet on Magnetic Resonance Images," *Procedia Computer Science*, vol. 218, pp. 1551–1560, Jan. 2023, doi: 10.1016/j.procs.2023.01.133.

[40] B. Scholles *et al*., "Osteoporosis screening: Leveraging EfficientNet with complete and cropped facial panoramic radiography imaging," *Biomedical Signal Processing and Control*, vol. 100, pp. 107031–107031, Oct. 2024, doi: 10.1016/j.bspc.2024.107031.

[41] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, “Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer,” *Neuroscience Informatics*, vol. 2, no. 4, p. 100034, Dec. 2022, doi: 10.1016/j.neuri.2021.100034.

[42] K. Sun, M. He, Z. He, H. Liu, and X. Pi, “EfficientNet embedded with spatial attention for recognition of multi-label fundus disease from color fundus photographs,” *Biomedical Signal Processing and Control*, vol. 77, p. 103768, Aug. 2022, doi: 10.1016/j.bspc.2022.103768.

[43] C. Guo, Y. Chen, and J. Li, “Radiographic imaging and diagnosis of spinal bone tumors: AlexNet and ResNet for the classification of tumor malignancy,” *Journal of bone oncology*, vol. 48, pp. 100629–100629, Aug. 2024, doi: 10.1016/j.jbo.2024.100629.

[44] W. Xu, Y.-L. Fu, and D. Zhu, “ResNet and its application to medical image processing: Research progress and challenges,” *Computer Methods and Programs in Biomedicine*, vol. 240, p. 107660, Oct. 2023, doi: 10.1016/j.cmpb.2023.107660.

[45] C. J. Ejiyi *et al*., “ResfEANet: ResNet-fused External Attention Network for Tuberculosis Diagnosis using Chest X-ray Images,” *Computer Methods and Programs in Biomedicine Update*, vol. 5, pp. 100133–100133, Dec. 2023, doi: 10.1016/j.cmpbup.2023.100133.

[46] Y. Fan *et al*., “RMAP-ResNet: Segmentation of brain tumor OCT images using residual multicore attention pooling networks for intelligent minimally invasive theranostics,” *Biomedical Signal Processing and Control*, vol. 90, pp. 105805–105805, Dec. 2023, doi: 10.1016/j.bspc.2023.105805.

[47] N. G. Inan, O. Kocadağlı, D. Yıldırım, İ. Meşe, and Ö. Kovan, “Multi-class classification of thyroid nodules from automatic segmented ultrasound images: Hybrid ResNet based UNet convolutional neural network approach,” *Computer Methods and Programs in Biomedicine*, vol. 243, p. 107921, Jan. 2024, doi: 10.1016/j.cmpb.2023.107921.

[48] Q. Xiao *et al*., “A computer vision and residual neural network (ResNet) combined method for automated and accurate yeast replicative aging analysis of high-throughput microfluidic single-cell images,” *Biosensors and Bioelectronics*, vol. 244, p. 115807, Jan. 2024, doi: 10.1016/j.bios.2023.115807.

[49] Y. Zhou *et al*., “Optimization of automated garbage recognition model based on ResNet-50 and weakly supervised CNN for sustainable urban development,” *Alexandria Engineering Journal*, vol. 108, pp. 415–427, Aug. 2024, doi: 10.1016/j.aej.2024.07.066.

[50] G. Mohandass, G. Hari Krishnan, D. Selvaraj, and C. Sridhathan, “Lung Cancer Classification using Optimized Attention-based Convolutional Neural Network with DenseNet-201 Transfer Learning Model on CT image,” *Biomedical Signal Processing and Control*, vol. 95, p. 106330, Sep. 2024, doi: 10.1016/j.bspc.2024.106330.

[51] Z. Gu *et al*., “Assessing breast cancer volume alterations post-neoadjuvant chemotherapy through DenseNet-201 deep learning analysis on DCE-MRI,” *Journal of Radiation Research and Applied Sciences*, vol. 17, no. 3, pp. 100971–100971, Jun. 2024, doi: 10.1016/j.jrras.2024.100971.

[52] M. R. Khare and R. H. Havaldar, “Predicting the anterior slippage of vertebral lumbar spine using Densenet-201,” *Biomedical Signal Processing and Control*, vol. 86, pp. 105115–105115, Jun. 2023, doi: 10.1016/j.bspc.2023.105115.

[53] A. W. Saleh, G. Gupta, S. B. Khan, N. A. Alkhaldi, and A. Verma, “An Alzheimer’s disease classification model using transfer learning Densenet with embedded healthcare decision support system,” *Decision Analytics Journal*, vol. 9, p. 100348, Dec. 2023, doi: 10.1016/j.dajour.2023.100348.

[54] H. Zerouaoui and A. Idri, “Deep hybrid architectures for binary classification of medical breast cancer images,” *Biomedical Signal Processing and Control*, vol. 71, p. 103226, Jan. 2022, doi: 10.1016/j.bspc.2021.103226.

[55] F. B. Mofrad and G. Valizadeh, “DenseNet-based transfer learning for LV shape Classification: Introducing a novel information fusion and data augmentation using statistical Shape/Color modeling,” *Expert Systems with Applications*, vol. 213, p. 119261, Mar. 2023, doi: 10.1016/j.eswa.2022.119261.

[56] B. Cansiz, C. U. Kilinc, and G. Serbes, “Deep learning-driven feature engineering for lung disease classification through electrical impedance tomography imaging,” *Biomedical Signal Processing and Control*, vol. 100, pp. 107124–107124, Nov. 2024, doi: 10.1016/j.bspc.2024.107124.

[57] M. N. Akram, M. U. Yaseen, M. Waqar, M. Imran, and A. Hussain, “A Double-Branch Xception Architecture for Acute Hemorrhage Detection and Subtype Classification,” *Computers, materials & continua/Computers, materials & continua (Print)*, vol. 76, no. 3, pp. 3727–3744, Jan. 2023, doi: 10.32604/cmc.2023.041855.

[58] C. Upasana, A. S. Tewari, and J. P. Singh, “An Attention-based Pneumothorax Classification using Modified Xception Model,” *Procedia Computer Science*, vol. 218, pp. 74–82, 2023, doi: 10.1016/j.procs.2022.12.403.

[59] M. Aparna and B. Srinivasa Rao, “Xception-Fractalnet: Hybrid Deep Learning Based Multi-Class Classification of Alzheimer’s Disease,” *Computers, Materials & Continua*, vol. 74, no. 3, pp. 6909–6932, 2023, doi: 10.32604/cmc.2023.034796.

[60] S. Sharma and S. Kumar, “The Xception model: A potential feature extractor in breast cancer histology images classification,” *ICT Express*, Nov. 2021, doi: 10.1016/j.icte.2021.11.010.

[61] J. Banumathi *et al*., “An Intelligent Deep Learning Based Xception Model for Hyperspectral Image Analysis and Classification,” *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2393–2407, 2021, doi: 10.32604/cmc.2021.015605.

[62] A. Panthakkan, S. M. Anzar, S. Jamal, and W. Mansoor, “Concatenated Xception-ResNet50 — A novel hybrid approach for accurate skin cancer prediction,” *Computers in Biology and Medicine*, vol. 150, pp. 106170–106170, Oct. 2022, doi: 10.1016/j.compbiomed.2022.106170.

[63] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” *Informatics in Medicine Unlocked*, vol. 19, p. 100360, 2020, doi: 10.1016/j.imu.2020.100360.

[64] M. Neshat, M. Ahmed, H. Askari, M. Thilakaratne, and S. Mirjalili, “Hybrid Inception Architecture with Residual Connection: Fine-tuned Inception-ResNet Deep Learning Model for Lung Inflammation Diagnosis from Chest Radiographs,” *Procedia Computer Science*, vol. 235, pp. 1841–1850, 2024, doi: 10.1016/j.procs.2024.04.175.

[65] Y. Chen *et al*., “Classification of Lungs Infected COVID-19 Images based on Inception-ResNet,” *Computer Methods and Programs in Biomedicine*, p. 107053, Jul. 2022, doi: 10.1016/j.cmpb.2022.107053.

[66] H. Wang, X. Shen, K. Fang, Z. Dai, G. Wei, and L.-F. Chen, “Contrast-enhanced magnetic resonance image segmentation based on improved U-Net and Inception-ResNet in the diagnosis of spinal metastases,” *Journal of Bone Oncology*, vol. 42, pp. 100498–100498, Oct. 2023, doi: 10.1016/j.jbo.2023.100498.

[67] G. S. Sunsuhi and S. A. Jose, “An Adaptive Eroded Deep Convolutional neural network for brain image segmentation and classification using Inception ResnetV2,” *Biomedical Signal Processing and Control*, vol. 78, p. 103863, Sep. 2022, doi: 10.1016/j.bspc.2022.103863.

[68] S. Peng, H. Huang, W. Chen, L. Zhang, and W. Fang, “More Trainable Inception-ResNet for Face Recognition,” *Neurocomputing*, vol. 411, pp. 9-19, May 2020, doi: 10.1016/j.neucom.2020.05.022.

[69] M. Nawaz, A. Javed, and A. Irtaza, “A deep learning model for FaceSwap and face-reenactment deepfakes detection,” *Applied Soft Computing*, vol. 162, p. 111854, Sep. 2024, doi: 10.1016/j.asoc.2024.111854.

[70] X. Yu, J. Tian, Z. Chen, Y. Meng, and J. Zhang, “Predictive breast cancer diagnosis using ensemble fuzzy model,” *Image and Vision Computing*, vol. 148, p. 105146, Aug. 2024, doi: 10.1016/j.imavis.2024.105146.

[71] R. Khattab, I. R. Abdelmaksoud, and S. Abdelrazek, “Automated detection of COVID-19 and pneumonia diseases using data mining and transfer learning algorithms with focal loss from chest X-ray images,” *Applied Soft Computing*, vol. 162, pp. 111806–111806, May 2024, doi: 10.1016/j.asoc.2024.111806.

[72] S. Patnaik, S. Ghosh, R. Ghosh, and S. Sahay, “Identifying Skeletal Maturity from X-rays using Deep Neural Networks,” *The Open Biomedical Engineering Journal*, vol. 15, no. 1, pp. 141–148, Dec. 2021, doi: 10.2174/1874120702115010141.

[73] G. Nirmala, P. P. Nayudu, A. R. Kumar, and R. Sagar, “Automatic cervical cancer classification using adaptive vision transformer encoder with CNN for medical application,” *Pattern Recognition*, pp. 111201–111201, Nov. 2024, doi: 10.1016/j.patcog.2024.111201.

[74] X.-L. Pan *et al*., “EL-CNN: An enhanced lightweight classification method for colorectal cancer histopathological images,” *Biomedical Signal Processing and Control*, vol. 100, p. 106933, Feb. 2025, doi: 10.1016/j.bspc.2024.106933.

[75] J. Rabbah, M. Ridouani, and L. Hassouni, “Improving pneumonia diagnosis with high-accuracy CNN-Based chest X-ray image classification and integrated gradient,” *Biomedical Signal Processing and Control*, vol. 101, p. 107239, Mar. 2025, doi: 10.1016/j.bspc.2024.107239.

[76] S. Rajeashwari and K. Arunesh, “Enhancing pneumonia diagnosis with ensemble-modified classifier and transfer learning in deep-CNN based classification of chest radiographs,” *Biomedical Signal Processing and Control*, vol. 93, pp. 106130–106130, Feb. 2024, doi: 10.1016/j.bspc.2024.106130.

[77] H. M. El-Hoseny, H. F. Elsepae, W. A. Mohamed, and A. S. Selmy, “Optimized Deep Learning Approach for Efficient Diabetic Retinopathy Classification Combining VGG16-CNN,” *Computers, materials & continua*, vol. 77, no. 2, pp. 1855–1872, Jan. 2023, doi: 10.32604/cmc.2023.042107.

[78] S. Liu, W. Wang, L. Deng, and H. Xu, “Cnn-trans model: A parallel dual-branch network for fundus image classification,” *Biomedical Signal Processing and Control*, vol. 96, pp. 106621–106621, Jul. 2024, doi: 10.1016/j.bspc.2024.106621.

[79] J. Barbero-Gómez, R. P. M. Cruz, J. S. Cardoso, P. A. Gutiérrez, and C. Hervás-Martínez, “CNN explanation methods for ordinal regression tasks,” *Neurocomputing*, pp. 128878–128878, Nov. 2024, doi: 10.1016/j.neucom.2024.128878.

[80] Y. Pamungkas, M. R. N. Ramadani, and E. N. Njoto, “Effectiveness of CNN Architectures and SMOTE to Overcome Imbalanced X-Ray Data in Childhood Pneumonia Detection,” *Journal of Robotics and Control (JRC)*, vol. 5, no. 3, pp. 775–785, 2024, doi: 10.18196/jrc.v5i3.21494.

[81] M. Javed et al., “An advanced deep neural network for fundus image analysis and enhancing diabetic retinopathy detection,” *Healthcare analytics*, vol. 5, pp. 100303–100303, Jun. 2024, doi: 10.1016/j.health.2024.100303.

[82] A. Samanta, A. Saha, S. C. Satapathy, S. L. Fernandes, and Y.-D. Zhang, “Automated detection of diabetic retinopathy using convolutional neural networks on a small dataset,” *Pattern Recognition Letters*, vol. 135, pp. 293–298, Jul. 2020, doi: 10.1016/j.patrec.2020.04.026.

[83] Z. Lu, J. Miao, J. Dong, S. Zhu, X. Wang, and J. Feng, “Automatic classification of retinal diseases with transfer learning-based lightweight convolutional neural network,” *Biomedical Signal Processing and Control*, vol. 81, p. 104365, Mar. 2023, doi: 10.1016/j.bspc.2022.104365.

[84] S. Zhou, J. Wang, and B. Li, “A multi-class fundus disease classification system based on an adaptive scale discriminator and hybrid loss,” *Computational Biology and Chemistry*, vol. 113, pp. 108241–108241, Oct. 2024, doi: 10.1016/j.compbiolchem.2024.108241.

[85] A. AbuKaraki *et al*., “Pulmonary Edema and Pleural Effusion Detection Using EfficientNet-V1-B4 Architecture and AdamW Optimizer from Chest X-Rays Images,” *Computers, materials & continua*, vol. 80, no. 1, pp. 1055–1073, Jan. 2024, doi: 10.32604/cmc.2024.051420.

[86] M. Sahoo, S. Ghorai, M. Mitra, and S. Pal, “Improved detection accuracy of red lesions in retinal fundus images with superlearning approach,” *Photodiagnosis and Photodynamic Therapy*, vol. 42, pp. 103351–103351, Feb. 2023, doi: 10.1016/j.pdpdt.2023.103351.

[87] Y. Kim, Y. Kwon, and M. C. Paik, “Valid oversampling schemes to handle imbalance,” *Pattern Recognition Letters*, vol. 125, pp. 661–667, Jul. 2019, doi: 10.1016/j.patrec.2019.07.006.

[88] S. Min *et al*., “Deep Imbalanced Regression Model for Predicting Refractive Error from Retinal Photos,” *Ophthalmology Science*, pp. 100659–100659, Nov. 2024, doi: 10.1016/j.xops.2024.100659.

[89] T. Felfeli *et al*., “Assessment of predictive value of artificial intelligence for ophthalmic diseases using electronic health records: A systematic review and meta-analysis,” *JFO Open Ophthalmology*, vol. 7, p. 100124, Jul. 2024, doi: 10.1016/j.jfop.2024.100124.

[90] Y. Pamungkas and M. R. N. Ramadani, “Leveraging of recurrent neural networks architectures and SMOTE for dyslexia prediction optimization in children,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 5, p. 1178, Jul. 2024, doi: 10.12928/telkomnika.v22i5.26092.

[91] R. Vij and S. Arora, “Modified deep inductive transfer learning diagnostic systems for diabetic retinopathy severity levels classification,” *Biomedical Signal Processing and Control*, vol. 99, p. 106885, Jan. 2025, doi: 10.1016/j.bspc.2024.106885.