Johannes Kepler University Linz, Austria

And

University of South Bohemia Faculty of Science České Budějovice,
Czech Republic

Detection of Diabetic Retinopathy Using Deep Learning and Transfer Learning Techniques with Oversampling to Address Imbalanced Dataset

Bachelor's Thesis

To confer the academic degree of

Bachelor of Science in Bioinformatics

Teodora Ranđelović

Thesis Supervisors: Elisabeth Rumetshofer, MSc, Dipl.-Ing. Andreas Fürst
Institute for Machine Learning

April 2023

Teodora Ranđelović, (2023): Detection of Diabetic Retinopathy using Deep Learning and Transfer Learning Techniques with Oversampling to Address Imbalanced Dataset. Bc. Thesis, in English. – 35 p., Faculty of Engineering and Natural Sciences, Johannes Kepler University, Linz, Austria and Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic

Annotation

"The study aims to develop a system for detecting diabetic retinopathy using deep learning. In this study, I explored transfer learning with four distinct models and addressed the issue of an unbalanced dataset through oversampling. The final experiment achieved a significant improvement in accuracy and quadratic kappa score, highlighting the potential of deep learning and the importance of addressing dataset imbalances to obtain accurate results.".

Basic recommended literature:

- Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 316(22), 2402. https://doi.org/10.1001/jama.2016.17216

- Huang, K.-K., Ren, C.-X., Liu, H., Lai, Z.-R., Yu, Y.-F., & Dai, D.-Q. (2021). Hyperspectral image classification via discriminative convolutional neural network with an improved triplet loss. Pattern Recognition, 112, 107744. https://doi.org/10.1016/j.patcog.2020.107744

- Khalifa, N., Loey, M., Taha, M., & Mohamed, H. (2019). Deep Transfer Learning Models for Medical Diabetic Retinopathy Detection. Acta Informatica Medica, 27(5), 327. https://doi.org/10.5455/aim.2019.27.327-332

- Quellec, G., Charrière, K., Boudi, Y., Cochener, B., & Lamard, M. (2017). Deep image mining for diabetic retinopathy screening. Medical Image Analysis, 39, 178–193. https://doi.org/10.1016/j.media.2017.04.012

I declare that I am the author of this qualification thesis and that in writing it I have used the sources and literature listed in references.

Linz, **21.04.2023**
Teodora Ranđelović

Table of Contents

Abstract

Deep learning is a subfield of machine learning where algorithms can learn autonomously from provided examples, refining their performance over time. Using this technology in medical imaging could be helpful in the detection of disorders such as diabetic retinopathy (DR) and diabetic macular edema. This study is aimed to develop and explore an automated system using deep convolutional neural networks to detect DR from retinal fundus photographs so that the time and cost can be minimized while screening. I have explored the potential of transfer learning for training models with various preprocessing techniques. The model was trained using a publicly accessible dataset consisting of 35,126 retinal fundus images. The training phase faced an obstacle due to the imbalanced dataset. The dataset primarily consisted of images classified as normal, without any signs of disease. This presented a significant challenge that needed to be addressed in order to achieve accurate and reliable results. The dataset was subjected to training using both pre-trained and non-pre-trained models. Three experiments were performed utilizing the ResNet18, EfficientNet-B3, and Xception models, all initialized with pre-trained weights. Additionally, a comparative analysis was performed in the fourth experiment using the EfficientNet-B3 model, with and without pre-trained weights. To address the challenge of an imbalanced dataset, oversampling was implemented. The proposed solution achieved an accuracy of 0.92 and a quadratic kappa score of 0.96. These results indicate a significant improvement over other experiments conducted in this study. Overall, the findings highlight the potential of deep learning in automating the detection of DR, providing promising prospects for efficient and accurate screening.

Keywords: Diabetic Retinopathy, Deep learning, Transfer learning, Convolutional neural network, Image classification, medical imaging, diabetic macular edema, retinal fundus photographs, comparative analysis, oversampling, accuracy, quadratic kappa score

1. Introduction

Diabetic retinopathy (DR) is a leading cause of blindness among millions of individuals throughout the world. It is estimated that diabetes affects 382 million people, with the number anticipated to rise to 592 million by 2035 (Safi et al. 2018). DR is caused by high blood sugar levels and damages the blood vessels in the retina, resulting in vision loss or blindness if not addressed. Everyone with type 1 or type 2 diabetes can develop DR, and the longer a person has diabetes, the more likely they are to acquire DR. Additional risk factors for DR include high blood pressure, high cholesterol, pregnancy, and smoking. DR usually progresses through stages, with early stages often showing no symptoms. Diabetic retinopathy has four stages: mild, moderate, severe non-proliferative, and proliferative (Wang, W. et al. 2018). The condition progresses from minor swelling in the blood vessels to the growth of abnormal blood vessels, which can lead to severe vision loss and blindness (Figure 1).

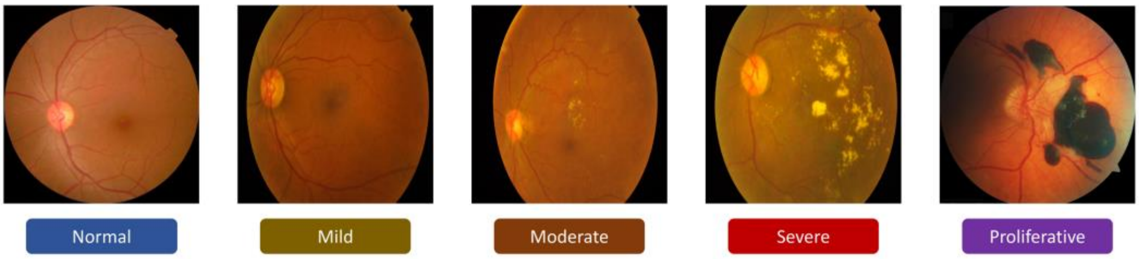

Figure. 1 Illustrates DR in various stages from normal to proliferative, indicating the severity class 4. In this final stage most, patients experience vision loss.

0 – **No diabetic retinopathy.**

1 – **Mild non-proliferative retinopathy:** Usually occurs when small areas of balloon-like swelling in the retina's blood vessel, called microaneurysms can be observed in the earliest stage of this disease. These areas may leak fluid or blood, results in distortion (Padhy, S. K. et al. 2019)

2 – **Moderate non-proliferative retinopathy:** If stage 1 is left untreated, a greater number of microaneurysms may form and block the blood arteries causing them to lose their ability to transport blood. This can result in growth of new blood vessels on the surface of retina. Some patients may experience difficulties seeing in dim light.

3 – **Severe non-proliferative retinopathy:** Significant amount of damage is done at this advanced stage where blood flow is less than the previous stage. This stage is characterized by hemorrhages or bleeding in the retina, and widespread swelling and fluid leakage. Patients might experience symptoms such as blind spot in their visual field.

4 – **Proliferative retinopathy:** When the new blood vessels leak blood or fluid, they cause scarring and damage to the retina (Khalifa et al., 2019). Proliferative retinopathy may lead to the formation of scar tissue which can cause the retina to detach from the back of the eye, a leading cause of blindness.

Medication such as injections into the eye may also be utilized in some circumstances to minimize swelling and inflammation. Controlling blood sugar levels, blood pressure, and cholesterol levels, on the other hand, can help lower the likelihood of developing DR and stop its progression. Machine learning-based automated DR detection systems have the potential to be a more efficient and cost-effective alternative to manual screening. The advancement of digital imaging technologies and machine learning algorithms have created new opportunities for automated identification and diagnosis of DR.

2. Aim

In the past, diabetic retinopathy was mostly detected after severe damage was already done to the eye had already affected the eye. This was primarily due to a lack of awareness regarding the disease or the symptoms being less noticeable. Prior to the availability of advanced technology, medical professionals relied on ophthalmoscopy as a method to examine the fundus and detect any potential abnormalities in the eye. This method was not very effective in detecting the early stages of retinopathy, as the disease often progresses without causing visible changes until it reaches an advanced stage. Currently more than 170 million people worldwide are affected, and it is estimated to increase by 366 million by year 2030 (Palavalasa & Sambaturu, 2018). Nowadays, the treatment of DR is often made with fundus images. Doctors use a specialized camera to take pictures of the back of the eye to see the abnormalities. Several studies have aimed to confirm the effectiveness of deep learning technologies in screening DR (Bhaskaranand et al., 2019; Gulshan et al., 2016a; Ruamviboonsuk et al., 2019; Ting et al., 2017a). There are three issues regarding DR treatment (Khalifa et al., 2019; Yu et al., 2018). Firstly, limited resources affect the screening uptake and accuracy of results. Secondly, delayed diagnosis due to the need for a second opinion and finally, poor tracking of patient follow-up and treatment referrals. Addressing the first issue, deep learning models can automate the process of screening which reduces the burden on the limited resources and manpower. Additionally, better accuracy and faster screening time can be ensured (Gulshan et al., 2016b). On the other hand, these machine learning models can reduce the need for a second opinion as they can accurately classify the images for signs of DR. As the whole process can be done within hours, overall management of patients with DR is easier and automatically generated reports can ensure timely treatment and follow-up care. One study by Ting et al. showed that a deep learning method achieved a high accuracy of 97.4% for DR detection Ting et al., (2017b). In several circumstances, the method outperformed human specialists, according to the study. In another study, Gulshan et al., (2016b) they examined the performance of a deep learning system on a large dataset of retinal pictures, attaining an area under the receiver operating characteristic (ROC) curve of 0.99. Most of these studies are binary classifications where the model can predict between 'DR' and 'No DR' (Pires et al., 2019; Quellec et al., 2017a; Xu et al., 2017). The objective of this thesis is to explore existing automated DR detection algorithms and investigate the potential of transfer learning in improving the accuracy of multi-class classification, which involves categorizing all five classes of DR. By conducting a comparative analysis between pre-trained and non-pre-trained convolutional neural network models, the aim of this study is to identify an effective approach that can achieve optimal accuracy for the screening of a large number of individuals for diabetic retinopathy, enabling early detection and treatment by ophthalmologists. Moreover, this research aims to demonstrate the potential of machine learning methods in the field of medical imaging analysis, while also highlighting the strengths and limitations of different pre-trained convolutional neural network models for the detection of diabetic retinopathy.

3. Methodology

3.1 Dataset

The Kaggle diabetic retinopathy dataset is a publicly available retina fundus dataset which was released in 2015 as a competition on Kaggle. This competition was hosted by EyePACS, a non-profit organization who provides cloud-based platforms for retinopathy screenings. The dataset contains 35,126 retinal fundus images. Each patient has two images of their left and right eyes.

The dataset used in this study includes a set of images for training, as well as a corresponding CSV file containing the names of the images and the corresponding severity levels for each patient, ranging from 0 to 4. It should be noted that this problem is a multi-class classification task that involves the classification of images into five distinct categories, namely 0, 1, 2, 3, and 4, as illustrated in Figure 1. Furthermore, the distribution of images across each class is presented in Figure 2.

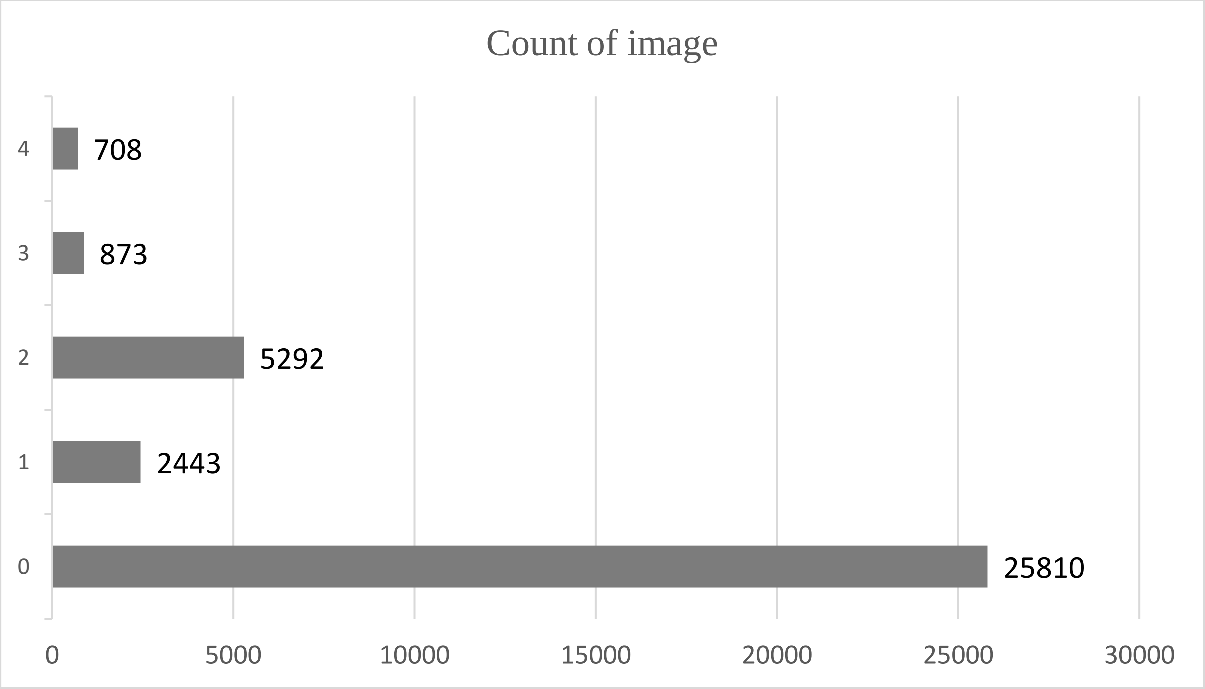

Figure 2. Distribution of severity class (bottom to top) in the dataset. x axis represents the number of images and y axis represents severity class (0-4.) 73.5% of the images are non-diabetic retinopathy patients. 7% of patients are diagnosed with class level 1 and 15% of the patients are diagnosed with moderate severity. Finally, the severity class 3 and 4 has 2.5% and 2% respectively.

In Figure 3 a visual representation of the fundus images for each of the five distinct classes can be observed. If class 0 is compared with class 4 then the cotton wool like structures can be observed without any image preprocessing.

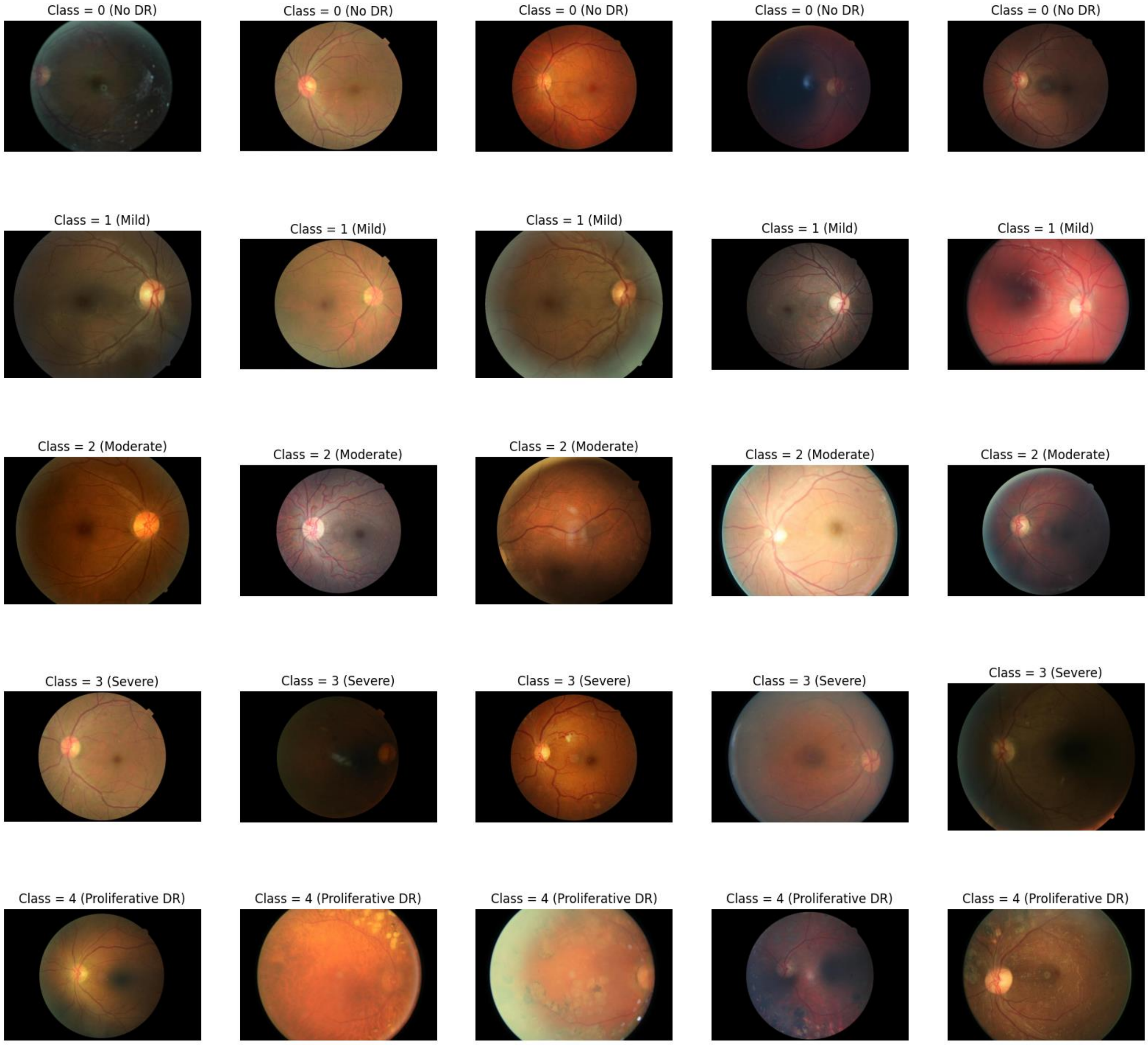

Figure 3. Sample images from the dataset for each class severity levels (from top to bottom): (a) No DR, (b) Mild DR, (c) Moderate (DR), (d) Severe DR, (e) Proliferative DR.

Due to the limitations of the dataset, which includes only the availability of the training CSV file, this study exclusively works with the training dataset by splitting it into training and testing sets. Despite the widespread use of this dataset for retinopathy detection, it is crucial to acknowledge its limitations, such as potential bias, class imbalance, and poor image quality. Therefore, it is important to interpret the results with these limitations in mind and to continue to refine and improve the dataset for more accurate and unbiased detection of diabetic retinopathy.

3.2 Data Pre-processing

To optimize the training process and improve the quality of results, two approaches will be implemented. The first approach involves training models using raw image data with minimal preprocessing to ensure compatibility with the input format of models, while another one is to use highly preprocessed data to facilitate more effective feature extraction. Considering diverse sizes of the images used for training, which can reach up to 4000x4000 pixels, it is essential to resize them before inputting them to the network. This resizing is not only to improve the training speed but also prevent memory allocation issues.

3.2.1 Splitting dataset into train and test sets

The dataset was split into two sets, namely "Train" and "Test". The data was partitioned in an 80:20 ratio, with 80% of the images being designated for training and 20% for testing. To facilitate a streamlined workflow, the raw data files were stored separately from the preprocessed images. Finally, the preprocessed images were stored in another directory.

3.2.2 Five characteristics

Important features to consider when detecting diabetic retinopathy in fundus images include microaneurysms, hemorrhages, exudates, neovascularization, and macular edema.

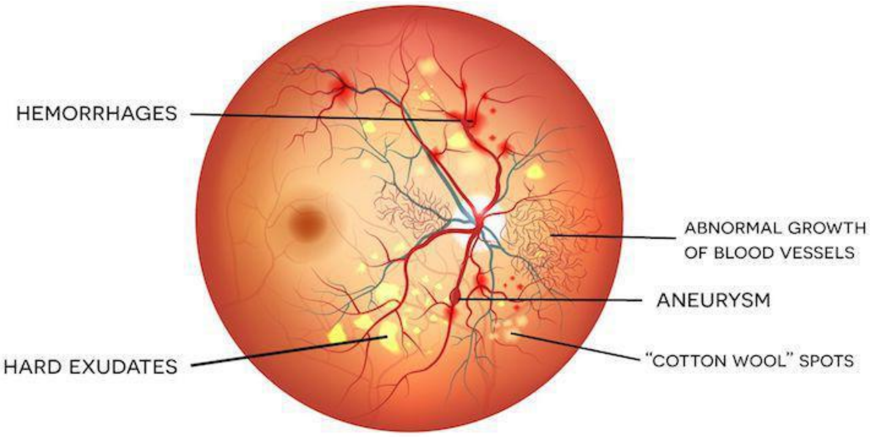

Figure 4: After conducting a brief analysis of the data in the pictures, it is observed that Hemorrhages, hard exudates, and cotton wool like spots are easily identifiable. However, the other two instances of Aneurysm or abnormal blood vessels growth are not very observable in the pictures. These two cases could be crucial for classification the disease.

Figure adapted from Nneji GU et. Al (2022) Identification of Diabetic Retinopathy Using Weighted Fusion Deep Learning Based on Dual-Channel Fundus Scans.

3.2.3 Cropping and drawing circle

Cropping the extra dark parts of retinal images is a useful technique for improving accuracy during training. In gray images, cropping the dark images might help to reduce noise and improve the signal to noise ratio, while cropping RGB images might help the network to improve feature extraction by focusing only on the relevant parts of the image. Finally, by drawing a circle around the edges we can select relevant parts of the images.

3.2.4 Applying Gaussian filter and blur

Adding Gaussian blur on the cropped images before feeding them to the neural network smooths out the image by reducing high-frequency components such as noise or sharp edges. This process can enhance feature extraction by allowing the network to focus on the important image features rather than noise or artifacts. The results of applying preprocessing functions are presented in Figure 5. The Gaussian blur function, which is an important preprocessing step, is configured using a parameter called sigmaX. In this study, a sigmaX value of 8 was found to be relatively more effective, as it allowed the blood vessels to be more visible compared to other values, such as 14, 40, or 50, as demonstrated in Figure 5.

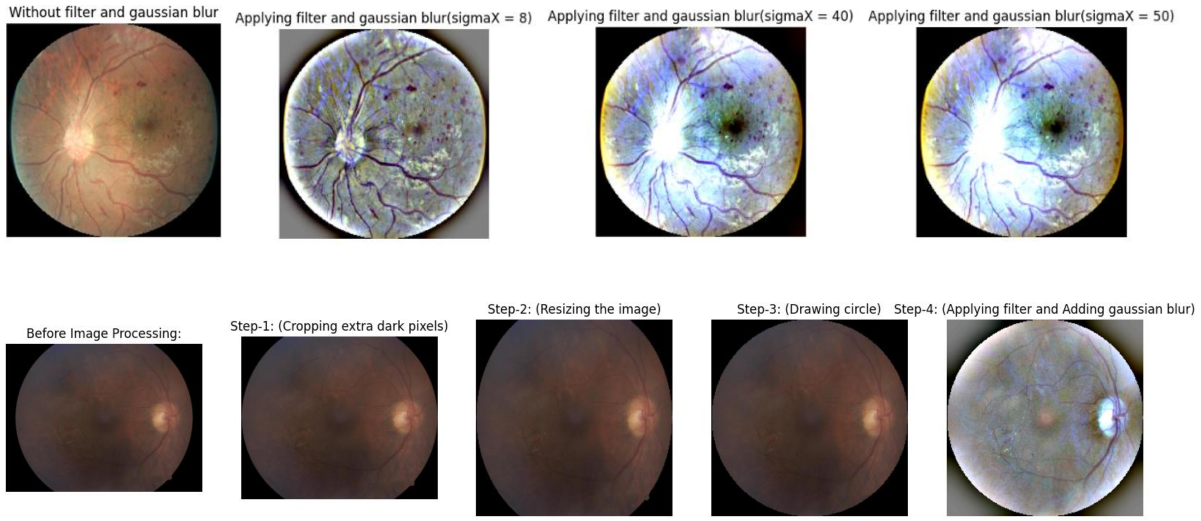
Without filter and gaussian blur
Applying filter and gaussian blur(sigmaX = 8)
Applying filter and gaussian blur(sigmaX = 40)
Applying filter and gaussian blur(sigmaX = 50)
Before Image Processing:
Step-1: (Cropping extra dark pixels)
Step-2: (Resizing the image)
Step-3: (Drawing circle)
Step-4: (Applying filter and Adding gaussian blur)

Figure 5. In the top row, Filter applied on the raw images and adding Gaussian blur to the images in different sigmaX values. Increasing the value of sigmaX leads to a degradation of image details. Therefore, the value 8 seems to be reasonable for this task. In the bottom row, Visual representation of preprocessing steps. For this study, two different datasets were employed: one containing data up to step 3 and another one including step 4. This allowed effective comparison of the training results.

3.3 Network architecture

After preprocessing was completed and the background noise was removed, the blood vessels were more visible. Utilizing a neural network helps with examination of each individual pixel and train system to find abnormalities. In the following section, I have explained in detail the functioning of deep neural networks and outlined their advantages.

3.3.1 CNN

A Convolutional Neural Network (CNN) is a type of neural network that is commonly used for image classification and image and video recognition. For instance, CNN can be used to classify images of different animals, detect faces in images, or identify handwritten digits in images. CNNs are designed to recognize patterns in visual data by using filters, or "kernels," to scan through the input image and identify relevant features. These features are then fed into a series of convolutional and pooling layers, where the network learns to extract increasingly complex features from the image. The first layer of a CNN applies a set of filters to the input image to identify relevant features, such as edges, corners, and curves. Each filter produces an output known as an activation map, which highlights areas of the image that are the most relevant to the task. The output from the convolutional layer is then passed through a pooling layer, which reduces the spatial dimension of the feature maps by down-sampling them. The convolutional and pooling layers are repeated multiple times in the network, with each subsequent layer learning more complex features by combining lower-level features from previous layers. The final layer of the network is typically a fully connected layer that takes the features learned from the previous layers and uses them to make a prediction about the input (Figure 6). The activation maps then are sampled down by pooling or stride convolutions. The final output consists of one or more fully connected layers which generate predictions of the input data. This innovation in the field of computer vision assists with tasks such as image classification, object detection, and image segmentation.

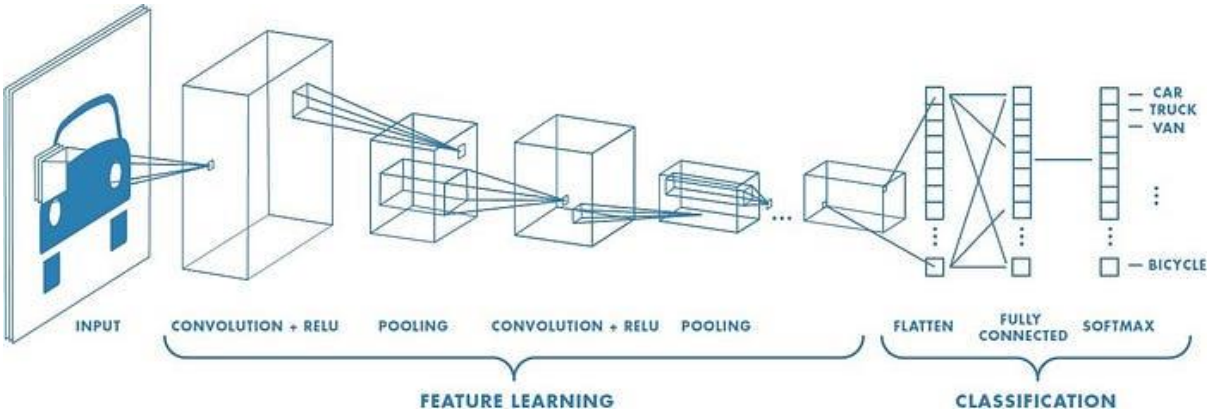

Figure 6. A convolutional neural network operates by taking an input and sliding a filter across each pixel block. The dot product is computed, and the result is then passed to the next layer. This process is repeated across multiple convolutional layers. As the network deepens, it becomes capable of detecting increasingly complex patterns.

Image: A comprehensive guide to convolutional neural networks by Sumit Saha (Sumit et al. 2018)

3.3.2 Transfer learning

Transfer learning enhances the learning process of a new task by using the knowledge gained from a related task that has already been successfully learned (Torrey & Shavlik, 2010). This approach enables systems to learn a new task by using pre-trained models. For instance, if someone can ride a bicycle then riding a motorcycle is easier because the knowledge is transferred here. Therefore, if a model is trained on a task, it can be used for another task which can reduce the amount of training data and computational power to achieve a better performance. A commonly employed strategy for utilizing a pre-trained model is to fine-tune to suit a desired task. For example, in image classification, a pre-trained model like ResNet (He et al., 2015) can be fine-tuned on a smaller dataset for different classification task such as identifying different types of animals or object. One of the main benefits of using transfer learning is that it can reduce the time and resources required to train a model from scratch. By starting with a pre-trained model that has already learned relevant features from a large dataset, transfer learning can significantly speed up the training process for a new task with similar features. However, it is important to note that the performance of transfer learning depends on the similarity between the source task and the target task, and the availability and quality of the pre-trained models. In this study, I have used several pre-trained models such as ResNet18, Xception, EfficientNet-B3, EfficientNet-B0. (Condori et al., 2021; Tan & Le, 2019a, 2019b).

3.3.3 Basics of the pretrained models

There are a few ways to scale a convolutional network, the most common way is by increasing the depth of the network which is done by adding more layers. Another popular method to scale up the model is by image resolution (G. Huang et al., 2017; K.-K. Huang et al., 2021) Increasing the number of pixels can lead to better accuracy of model. Scaling up baseline models with different width, depth or resolution improves accuracy (Mingxing Tan, Quoc V. Le et al., 2019). However, as the size of the model increases, the accuracy gain starts to diminish after reaching around 80%. This indicates that while larger models may demonstrate enhanced performance, the extent to which accuracy improves in relation to the size of the model diminishes. As a result, the incremental gain in accuracy becomes less significant as the model size increases. I have used the EfficientNet-B3 pre-trained on ImageNet (Deng et al., 2009) which is a large-scale image recognition database used for training and evaluating computer vision models. I choose EfficientNet-B0 without prior pre-training on ImageNet to compare performance.

ResNet (He et al., 2015), short for “Residual Network” was introduced in 2015 and it is composed of several “Residual blocks”. ResNet typically has between 50 to over 100 layers. The key point of this architecture is that it uses residual connections to address the vanishing gradient problem that can occur in very deep networks. By incorporating skip connections between the layers, this approach enables efficient flow of gradients throughout the network, resulting in improved performance. During the training process, the weights of the model are updated proportionally to the partial derivative of the error function with respect to the current weight. When the gradient is very small, the weights in the neural network may no longer be effectively updated during training, resulting in a phenomenon known as vanishing gradients. This can occur when the gradient signal decreases as it spreads through the layers of the network, causing weight to update and become increasingly small and less effective. Therefore, the information in the data may "disappear" as it passes through the layers, leading to poor performance in the trained model. ResNet came up with a solution to this problem by breaking down a deep plain network into smaller chunks of network connected through skip or shortcut connections.

Xception (Extreme inception) is another example of deep convolutional neural network, a model based on the idea of separating the cross-channel correlations and spatial correlation in the convolutional layer. It uses depth wise separable convolutions, which perform a spatial convolution on each channel of the input tensor separately, followed by a point-wise convolution that mixes the resulting feature maps. This depth wise separable convolution reduces the number of parameters and computation resulting an efficient and faster network (Chollet, 2016).

3.4 Experiments

Four experiments were conducted in this study. The first Experiment involves using ResNet18 on raw datasets with pre-trained weights on ImageNet. The second Experiment is extensions of the first Experiment, using EfficientNet-B3 with pre-trained weights on ImageNet. The third Experiment involves using model Xception on down sampled datasets with pre-trained weights on ImageNet. The fourth experiment utilizes a balanced and oversampled dataset, employing the identical methodology as Experiments 2. Specifically, it employs the EfficientNet-B3 model with two distinct training runs: one with pre-training on ImageNet and the other without. For all Experiments, except Experiment 3, the raw images were normalized using a mean of [0.3199, 0.2240, 0.1609] and a standard deviation of [0.3020, 0.2183, 0.1741]. Images with filters applied were normalized with a mean of [0.502, 0.501, 0.501] and a deviation of [0.121, 0.116, 0.097]. This normalization step is crucial as images can have varying pixel values due to differences in lighting conditions, camera settings, or image resolution. Failure to normalize the pixels can prevent the network's ability to learn from the data. To standardize the pixel values across all images in the dataset, the normalization process includes subtracting the mean value of pixel intensities and dividing it by the standard deviation. This procedure ensures that the pixel values are consistent and comparable throughout the dataset. Each model was evaluated using different hyper-parameters and augmentation techniques, mentioned in detail in each experiment section. Furthermore, to keep the initializations constant I have tested different seed values including 42, 31415, 8854 before running the experiments. In order to effectively resolve the problem of class imbalance in classification task, I have conducted Experiment 4 as a comprehensive and decisive solution. This experiment aimed to compare the performance of models. The purpose of including these additional experiments with non-pre-trained models was to assess the impact of pre-trained weights.

3.4.1 Experiment 1

The first experiment involves the models ResNet18 where I have set the loss function to mean squared error which outputs a scaler value that represents the error of the current set of model parameters. The aim of MSE is to minimize this difference or error between the predicted and actual values during the training of a neural network. After I modified the final layers of the ResNet18 to have an input size of 512 and an output size of 1. Additional hyper-parameters are shown in Table 1. The prediction of output was a continuous float value which was then used by a function to classify each output value as belonging to one of the five possible classes by converting from float values to integer using a threshold of 0.5. For example, if the output is less than .50 then the integer prediction will be 0, meaning No DR. Similarly, if the prediction is between 0.5 and 1.5 then it will convert to 1, meaning mild DR and so on. This model is trained for 40 epochs and the pre-trained weights used for this network is ImageNet. The dataset used for this experiment is the raw dataset without gaussian filter and blur. However, the images were lightly preprocessed and augmented as shown in Table 2.

Hyper-parameters	**ResNet18**
Loss Function	MSE
Optimizer	Adam
Learning Rate	3e^{-5}
Batch Size	8
Epoch	40
Input size	728x728
Preprocessed	Resized
Pre-trained Weights	ImageNet

Table 1 Overview of the network hyperparameters that are used in ResNet18.

A comprehensive data augmentation technique was implemented for the ResNet18 model (Table 2). The transformation types include resizing the images to dimensions of 728 pixels in height and width, followed by random cropping to achieve a final size of 680 pixels in both dimensions. The augmentation also involves horizontal and vertical flipping with a probability of 50%, facilitating increased diversity in the training data. Furthermore, a random rotation of 90 degrees is applied with a probability of 50% to introduce additional variability. Additionally, a blurring effect is applied with a probability of 30% to enhance robustness against noise and improve the model's ability to generalize. These augmentation techniques collectively contribute to the training process of the ResNet18 model, enhancing its capacity to handle various image variations and improve overall performance.

Transformation Type	Description
Resize	Height = 728, Width = 728
Random Crop	Height = 680, Width = 680
Horizontal flip	Probability of 50%
Vertical Flip	Probability of 50%
Random Rotate 90 degrees	Probability of 50%
Blur	Probability of 30%

Table 2. Data augmentation for ResNet18 model.

3.4.2 Experiment 2

Experiments 2 was based on Experiment 1 with changes in hyperparameters such as batch size or number of epochs (Table 3). However, the methods are mostly the identical. Modifications to the model design, such as my selection of EfficientNet-B3 with a (1536,1) fully connected layer. The learning rate, loss function, optimizer, and other hyper-parameters as well as augmentation were set to identical as in Experiment 1 (Table 3).

Hyper-parameters	EfficientNet -B3
Loss Function	MSE
Optimizer	Adam
Learning Rate	3e^{-5}
Batch Size	4
Epoch	15
Input size	728x728
Preprocessed	Yes
Pre-trained Weights	ImageNet

Table 3. Overview of model hyperparameters of model EfficientNet-B3. The key difference between the two models is in training with pre-trained weights and without pre-trained weights.

3.4.3 Experiment 3

In contrast, Xception was trained on an under-sampled balanced dataset where I selected the minority class and took the same number of images for other classes to train in this architecture. The input size was kept to the minimum of 512x512 pixels to prevent memory allocation issues. In addition, while preprocessing the images I changed the sigmaX value to 10. Additional hyper-parameters are shown in Table 4.

Hyper-parameters	Xception
Loss Function	Categorical Cross Entropy
Optimizer	Adam
Learning Rate	1e^{-4}
Batch Size	4
Epoch	20
Input size	512x512
Preprocessed	Yes
Weight	ImageNet

Table 4. Overview of network hyper-parameter of model Xception.

For data augmentation I used ImageDataGenerator (Chollet, 2015). Before splitting data to train and test I changed the classes to categorical values and set the training and test size to 80% and 20% respectively. In addition, transformation such as horizontal, vertical flip, rotation range to 180 degrees, sample wise center, zoom range to 20% were set to introduce variability to data (Table 5).

Transformation Type	Description
Horizontal flip	TRUE
Vertical Flip	TRUE
Rotation Range	180 degrees
Zoom Range	20%
Sample-wise Center	TRUE
Sample-wise Standard Normalization	TRUE

Table 5. Data augmentation parameters for Model Xception.

A global average pooling layer was added to the model to reduce the number of parameters and improve the model's generalization capabilities. Additionally, a drop out layer was included to prevent overfitting and a dense layer with softmax activation function was added to produce probability distribution over 5 target classes. Since the model has multi class output, the loss function used is categorical cross entropy. It is one of the suitable loss-functions for multi-class classification problems. Categorical cross entropy calculates the dissimilarity between the predicted probability distribution and the true probability distribution across all classes. Finally, the model is trained using Adam optimizer and the learning rate was set to 1e^{-4} to minimize the loss.

3.4.4 Experiment 4

To address the issue of imbalanced data sets, where there are significant differences in the number of instances between the classes, a unique strategy was implemented. Oversampling was used to create random images for the minority class based on length of majority class, while down sampling was applied to the majority class, with each class then combined to create a single dataset out of original dataset. The resulting dataset is then finally split into train (80%) and test (20%) ratio. This balanced dataset was then used to train the EfficientNet-B3 model. The hyper-parameters as well as the augmentations used in this experiment were shown in Table 6 and Table 7.

Hyper-parameters	EfficientNet -B3	EfficientNet-B3 scratch
Loss Function	Cross-Entropy loss	Cross-Entropy loss
Optimizer	Adam	Adam
Learning Rate	1e^{-3}	1e^{-3}
Batch Size	16	16
Epoch	20	20
Input size	512x512	512x512
Preprocessed	Yes	Yes
Weight decay	1e^{-5}	1e^{-5}
Pre-trained Weights	ImageNet	No

Table 6. Overview of model hyperparameters of model EfficientNet-B3 on up sampled dataset.

Transformation Type	Description
Random Horizontal flip	Default
Rotation Range	-270, 270
Normalization	True

Table 7. Data augmentation parameters for the model EfficientNet-B3 on up sampled dataset.

To address the memory allocation issue associated with the dataset, the utilization of the Apex library (Nvidia Apex, n.d.) was employed, as the dataset had been oversampled. This library is specifically designed to optimize the performance of deep learning models using mixed-precision training techniques that combine single-precision and half-precision floating-point numbers for efficient training on modern hardware such as GPUs. It provides various tools, including automatic mixed-precision training, optimized memory management, and more efficient data loading, to enhance the scalability and performance of deep learning models. The newly created dataset was then resized, preprocessed, and compressed in hdf5 compression='gzip' (Delaunay et al., 2019). During the model definition process, the Model Freezer technique was used to freeze specific layers. This approach was adopted with the aim of expediting the training process, potentially mitigating overfitting, and stabilizing the model's performance. Additionally, an initial learning rate of 1e^{-3} was set, accompanied by the implementation of a learning rate scheduler. This scheduler dynamically adjusts the learning rate throughout the training procedure by gradually decreasing it over time. Finally, the model was trained twice: once with pre-trained weights and once without pre-trained weights, while maintaining the same augmentation techniques and hyperparameters throughout the training process.

4. Results

4.1 Performance measures

There are several metrics used to evaluate the performance of deep learning models in classification tasks. Accuracy measures the overall correctness of the model's predictions. Sensitivity, also known as recall or true positive rate, measures the model's ability to correctly identify positive cases. Specificity measures the model's ability to correctly identify negative cases. These metrics are important for assessing the performance of a deep learning model in a given task and can help to identify areas for improvement.

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) \quad (1)$$

$$\text{Sensitivity} = \text{TP}/ (\text{TP} + \text{FN}) \quad (2)$$

$$\text{Accuracy} = \text{TN} + \text{TP}/(\text{TN} + \underline{\text{TP}} + \underline{\text{FN}} + \text{FP}) \quad (3)$$

True positive (TP) refers to the number of images that are classified as positive (e.g., having a disease) and are actually positive according to ground truth labels. True negative (TN) refers to the number of images that are classified as negative (e.g., not having a disease) and are actually negative according to ground truth labels. False positive (FP) refers to the number of images that are classified as positive but are actually negative according to ground truth labels. False negative (FN) refers to the number of images that are classified as negative but are actually positive according to ground truth labels. Precision is calculated as the number of correct positive predictions divided by the total number of positive predictions. It is also called positive predictive value. The best precision is 1.0, whereas the worst is 0.0.

4.1.1 Cohen's Kappa

Cohen's kappa measures the degree of agreement between two classifiers beyond what would be expected by chance ranging from -1 to +1, where value of -1 is complete disagreement and +1 is complete agreement. It is used to measure the degree of agreement between raters for categorical data (Banerjee et al., 1999; Chicco et al., 2021; Fleiss & Cohen, 1973; Sim & Wright, 2005). Cohen's kappa is an extension of accuracy where it finds the simple percent agreement calculation (True prediction/ Total predictions). Kappa score makes it more robust by considering the agreement occurring by chance.

The Cohen's Kappa score is defined as,

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

where P_0 is observed agreement (same as accuracy) and P_e is expected agreement by chance. This expected agreement is calculated as the product of the marginal proportions of agreement for each label, assuming the two annotators assign labels independently of each other.

4.1.2 Quadratic Cohen's Kappa

An extension of Cohen's Kappa that considers the degree of disagreement between annotators or classifiers for ordinal or continuous variables where variables such as severity or intensity are measured on a scale rather than as discrete categories. The weighted kappa allows disagreement to be weighted differently and it is useful in ordinal data. It makes use of three matrices: Observed scores, expected scores and weight matrix.

The quadratic Cohen's Kappa is defined as,

$$\kappa = 1 - \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} x_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} m_{ij}}$$

where, W_{ij} is an element in weight matrix, X_{ij} is element in observed matrix, M_{ij} is element in expected matrix. In contrast, the prediction is penalized quadratically depending on the distance between the prediction and actual value. In that way, if the model is not predicting the true value of the class, at least it will try to be close to the true value. For the following reasons, it is useful to use quadratic kappa in life science.

For example, the penalty score will be higher if the true prediction is 3 and model predicted as 2 but the score will be lower if the model prediction is 4. In medical diagnosis, if the model predicts a higher severity level than expected, it may not necessarily be bad because there is room for correction. However, if the model predicts 2 but severity level is 3 or 4 such that a lower severity level than expected, it can be problematic because in that way patient's condition is not properly addressed.

4.1.3 Balanced Accuracy

Balanced accuracy is used when we have an imbalanced dataset with a large number of samples in one class. In this case we have the class 0 which is almost 73.5% of the whole dataset. Therefore, accuracy alone can be a misleading metric as it does not consider the class distribution of the data. Balance accuracy is calculated by taking the average of the recall for each class (Buitinck et al., 2013). The formula for balanced accuracy is defined as

Balanced accuracy = (Sensitivity + Specificity) / 2 (4)

4.2 Results of Experiment 1

After training for 40 epochs the best accuracy of quadratic kappa was found in 36th epoch. Based on the classification report in Table 8 ResNet18 model achieved an overall accuracy of 0.74 with balanced accuracy of 0.488 and a quadratic kappa score of 0.791. However, it showed low performance for the classification of class 1 and class 3 with F1 scores below 0.3. Class 0, 2 and 4 had F1-scores of 0.91, 0.41 and 0.44 respectively.

	ResNet18		
	Precision	Recall	F1-score
Class 0	0.92	0.9	0.91
Class 1	0.19	0.42	0.26
Class 2	0.74	0.29	0.41
Class 3	0.21	0.37	0.27
Class 4	0.41	0.46	0.44
Accuracy	0.74		
Balanced Accuracy		0.488	
Quadratic Kappa		0.791	
Parameters		11,177,025	

Table 8. The results of experiment 1, which were conducted on the raw dataset without any balancing of dataset. The F1 score is significantly higher for class 1 compared to the other classes, while the performance of the other classes is poor.

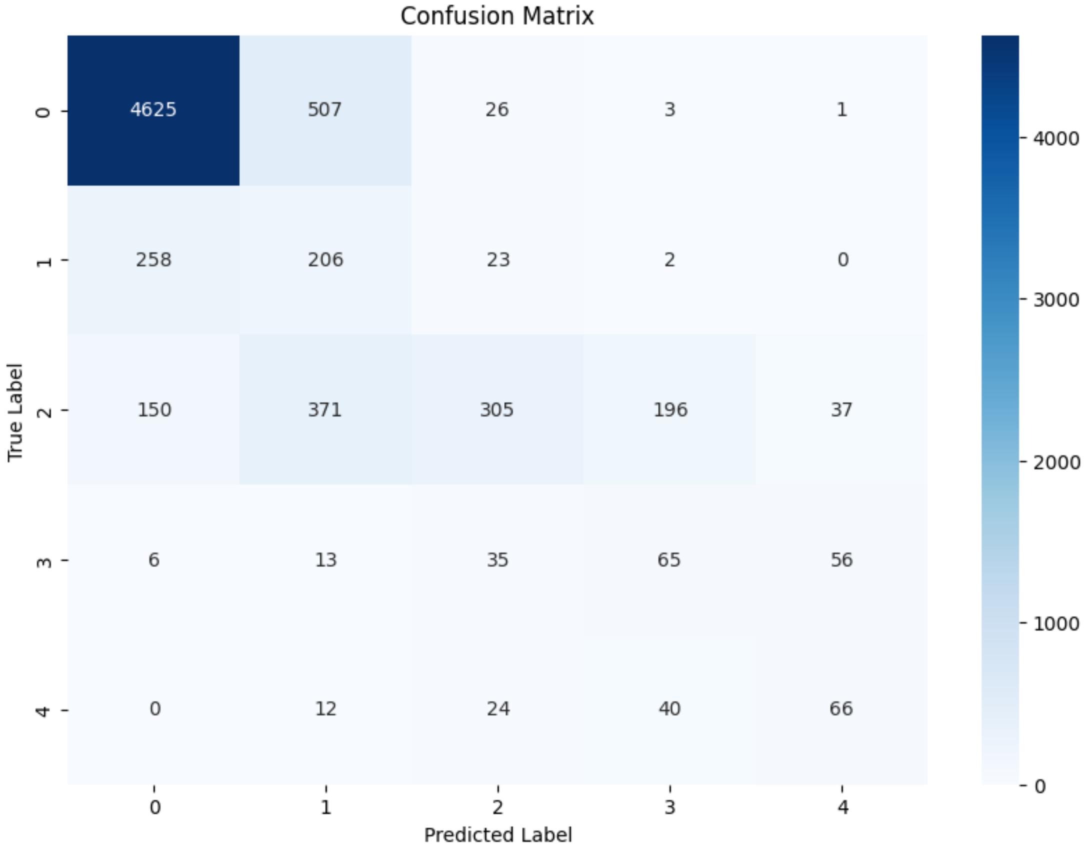

Figure 7. Confusion matrix generated by the best checkpoint of ResNet18. The matrix highlights the model's low performance of class 1,3 and 4 with misclassified samples while model performs well to classify the class 0.

4.3 Results of Experiment 2

EfficientNet-B3 performs the best among the other models therefore, I have run this model to train for different seed values (seeds = 42, 31415, 8854) to find optimal results. Although the difference in results did not differ much between different seed values, seed = 31415 had the quadratic kappa score 0.81. The mean quadratic kappa score of three different seeds runs was 0.79 ± 0.06. Furthermore, this model had limitation on classifying class and lowest F1-score 0.26 among other classes. Performance of Classes 0, 2, 3, and 4 were relatively better than other models with F1 scores 0.92, 0.51, 0.42 and 0.63 respectively. The confusion matrix also indicated that class 1 is poorly predicted (Figure 8). While training this model for 15 epochs, each iteration took approximately 30 minutes to complete.

	EfficientNet-B3		
	Precision	Recall	F1-score
Class 0	0.9	0.94	0.92
Class 1	0.23	0.3	0.26
Class 2	0.76	0.38	0.51
Class 3	0.29	0.74	0.42
Class 4	0.76	0.54	0.63
Accuracy	0.8		
Balanced Accuracy		0.578	
Quadratic Kappa		0.818	
Parameters		10,697,769	

Table 9. The model is performing well on the preprocessed, unbalanced dataset. The F1 score of class 0 is notable. However, while class 4 is also performing well, the scores for the other classes are comparatively low.

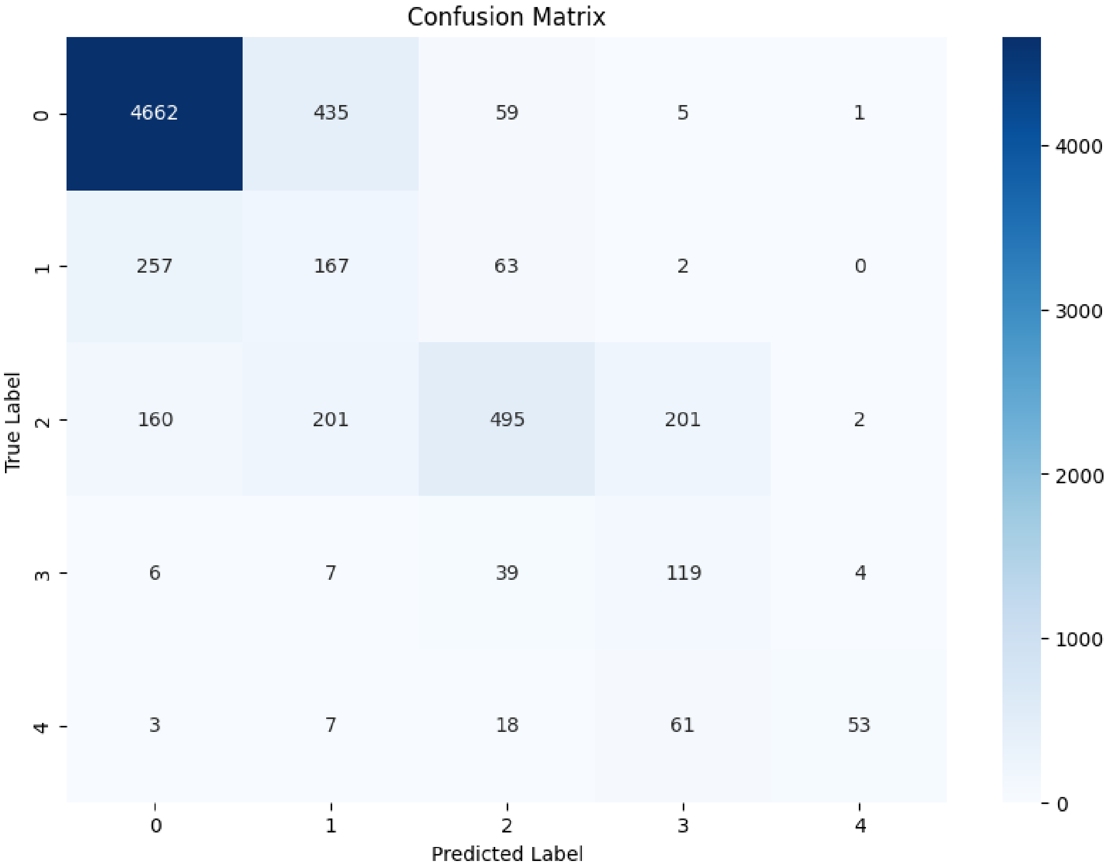

Figure 8. Confusion matrix generated by the best checkpoint of EfficientNet-B3. The matrix highlights the model's low performance of class 1 and 3 while model performs well to classify the class 0, 2 and 4.

4.4 Results of Experiment 3

Xception model was trained on a under sampled balanced dataset where all the classes had similar number of samples. This model achieved an accuracy of 0.54 on the validation set, with a weighted F1-score of 0.53. The model has the highest precision for class 4, which is 0.86 and the highest recall for class 0 is 0.82, while classes 1 and 2 had the lowest performance 0.30 and 0.45 respectively. The confusion matrix generated by the best checkpoint also indicates that the model is struggling to learn classification of class 1 and 2. The quadratic kappa score is relatively high at 0.77 indicating moderate agreement between the predicted and actual labels. The factors that may have contributed to the lower performance of these classes are the complexity of their features or the quality of the training data for corresponding classes. It may be worthwhile to investigate these factors further to improve the model's performance.

	Xception		
	Precision	Recall	F1-score
Class 0	0.51	0.82	0.63
Class 1	0.35	0.27	0.3
Class 2	0.45	0.45	0.45
Class 3	0.66	0.55	0.6
Class 4	0.86	0.52	0.65
Accuracy	0.54		
Balanced Accuracy		0.521	
Quadratic Kappa		0.765	
Parameters		20,871,725	

Table 10. The results of experiment 2, which was conducted on the preprocessed down sampled dataset. Each class consists of 700 images. Unlike experiment 1, the F1 score is considerable for class 0, 3 and 4 compared to the other classes, while the performance of the other classes are poor.

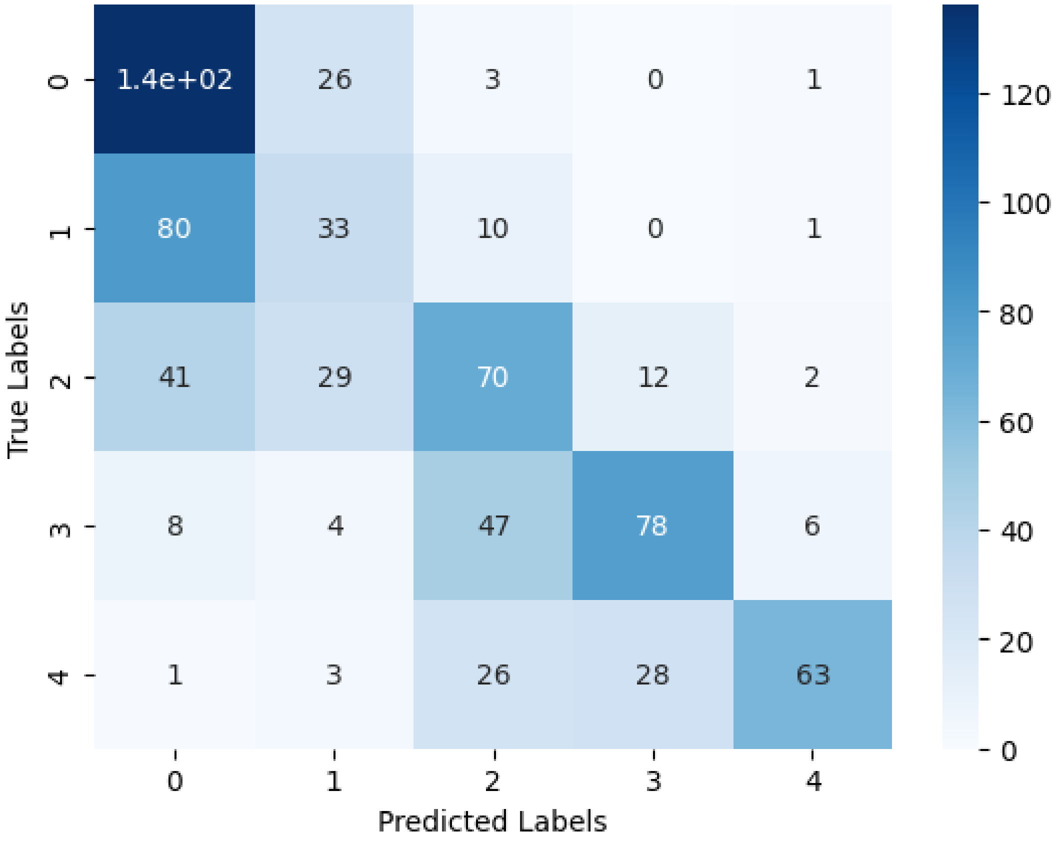

Figure 9. Confusion matrix generated by the best checkpoint of Xception. The matrix highlights the model's low performance of class 1 and 2 while model performs well to classify the class 0, 3 and 4.

It is important to highlight that the model was trained on an under-sampled dataset, where a significant number of samples were not considered during the training process. This approach can lead to various challenges and potential issues that need to be considered. Training a model on an under-sampled dataset can result in biased predictions. Since certain classes or instances have been underrepresented in the training data, the model may not adequately learn the patterns and characteristics of these underrepresented samples. As a result, when the model encounters such instances during inference or testing, it may struggle to make accurate predictions or provide reliable insights.

4.5 Results on Experiment 4

4.5.1 With pre-trained weights on ImageNet:

To mitigate with the class imbalance and the issue of under sampling I have conducted another sampling strategy by oversampling the minority class according to the majority class to balance the dataset. The model's performance was assessed on a test dataset of 20,648 instances and evaluated using a classification matrix and various performance metrics, including Balanced Accuracy, Quadratic Kappa Score, precision, recall, and F1-score. The confusion matrix presented in Figure 10 shows the true positives, false positives, true negatives, and false negatives for each class. The Balanced Accuracy of the model was 0.9392, indicating a high level of overall performance in correctly classifying instances across all classes. The Quadratic Kappa Score predicted a value of 0.9640. The average quadratic kappa score from three different runs with varying seed values was determined to be 0.93±0.03.

	EfficientNet-B3 on Up sampled dataset (Pre-trained)		
	Precision	Recall	F1-score
Class 0	0.89	0.84	0.86
Class 1	0.88	0.95	0.91
Class 2	0.94	0.91	0.93
Class 3	1.00	1.00	1.00
Class 4	1.00	1.00	1.00
Accuracy	0.94		
Balanced Accuracy		0.933	
Quadratic Kappa		0.963	
Parameters		10,697,769	

Table 11. The model achieved an accuracy of 0.94, which means that it is correctly classified 94% of the instances in the dataset. The precision, recall, and F1-score metrics are reported for each class separately, with class 0 achieving an F1-score of 0.86, class 1 achieving an F1-score of 0.91, class 2 achieving an F1-score of 0.93, and classes 3 and 4 achieving perfect scores of 1.00.

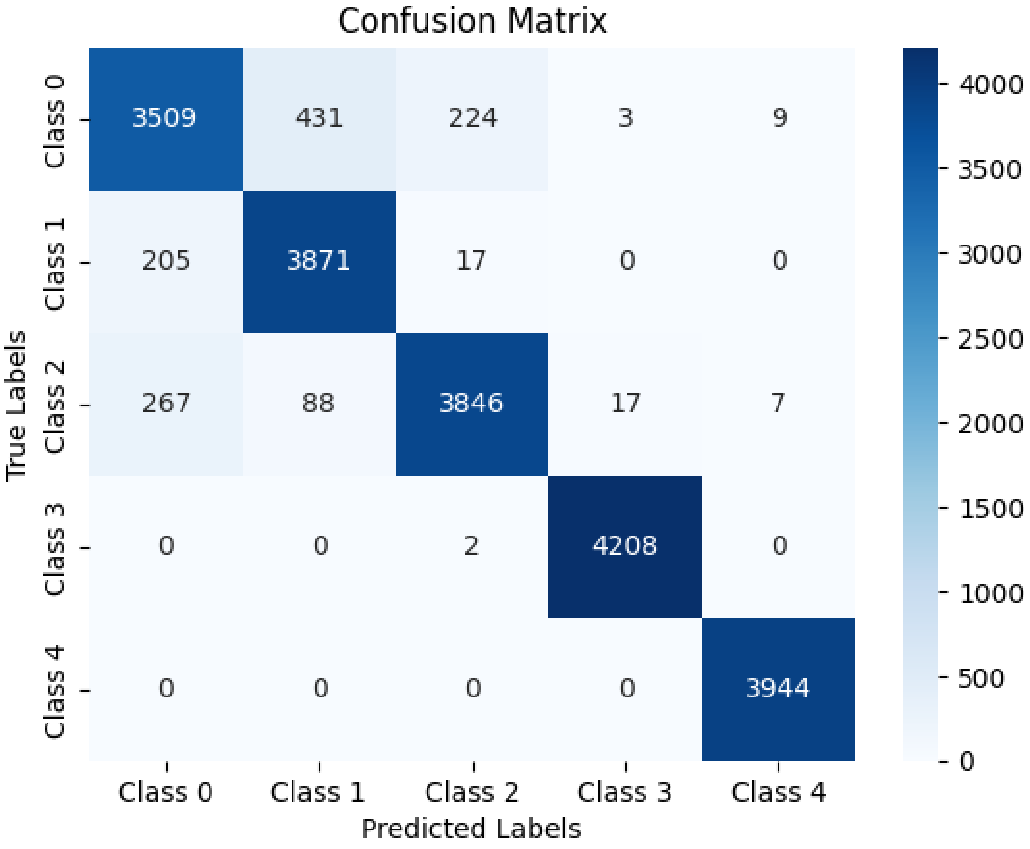

Figure 10. In this confusion matrix, the model predicted class 0 correctly 3509 times, but misclassified 431 instances as class 1, 224 instances as class 2, and 3 instances as class 3. Similarly, the model predicted class 1 correctly 3871 times, but misclassified 205 instances as class 0 and 17 instances as class 2. The same pattern is seen for classes 2, 3, and 4, with misclassifications occurring in various directions.

This score suggests that the model's predictions are in strong agreement with the true classes. The model demonstrated high precision, recall, and F1-scores across all five classes. Particularly strong performance is prominent in classes 3 and 4, where it achieved perfect scores. However, it should be noted that classes 3 and 4 had the lowest number of samples in the original dataset, and oversampling techniques were employed to balance the data. This could potentially lead to an overestimation of the model's performance in these classes, as it may be more prone to overfitting on the smaller sample sizes. The overall accuracy of the model was 0.94, which highlights its effectiveness in detecting diabetic retinopathy stages. Further research and validation with larger, balanced datasets are needed to confirm its effectiveness in these stages of diabetic retinopathy.

4.5.2 Without Pre-trained weights on ImageNet:

The confusion matrix reveals the distribution of predicted labels against the actual labels. The model achieved a balanced accuracy of approximately 71.42%, indicating reasonable overall performance (Table 12). In contrast to pre-trained on ImageNet, the performance decline is significant.

	EfficientNet-B3 on Up sampled dataset (Not Pre-trained)		
	Precision	Recall	F1-score
Class 0	0.58	0.75	0.66
Class 1	0.57	0.47	0.52
Class 2	0.64	0.51	0.57
Class 3	0.82	0.85	0.84
Class 4	0.94	0.98	0.96
Accuracy	0.71		
Balanced Accuracy		0.714	
Quadratic Kappa		0.878	
Parameters		10,697,769	

Table 12. The model attained an accuracy of 0.71, indicating that it accurately classified 71% of the instances within the dataset. Precision, recall, and F1-score metrics were calculated for individual classes, revealing class 3 and 4 to exhibit a respectable performance compared to the ImageNet pretrained model. However, classes 0, 1, and 2 demonstrated lower performance, with F1-scores ranging from 0.52 to 0.66.

The quadratic kappa score, measuring the agreement between predicted and actual labels beyond chance, yielded a high value of 0.88. This signifies a substantial agreement between the model's predictions and the true labels. The confusion matrix provides valuable insights into the model's misclassifications, aiding in understanding the specific areas where improvement may be needed. In Figure 11 the confusion matrix highlights the number of instances that are misclassified in each direction. For example, the model misclassified 899 instances of class 0 as class 1, 2078 instances of class 1 as class 0, and 853 instances of class 2 as class 1. Additionally, the matrix shows that class 4 achieved the highest accuracy, with only 92 instances misclassified, while class 0 had the highest number of misclassifications with 1292 instances misclassified.

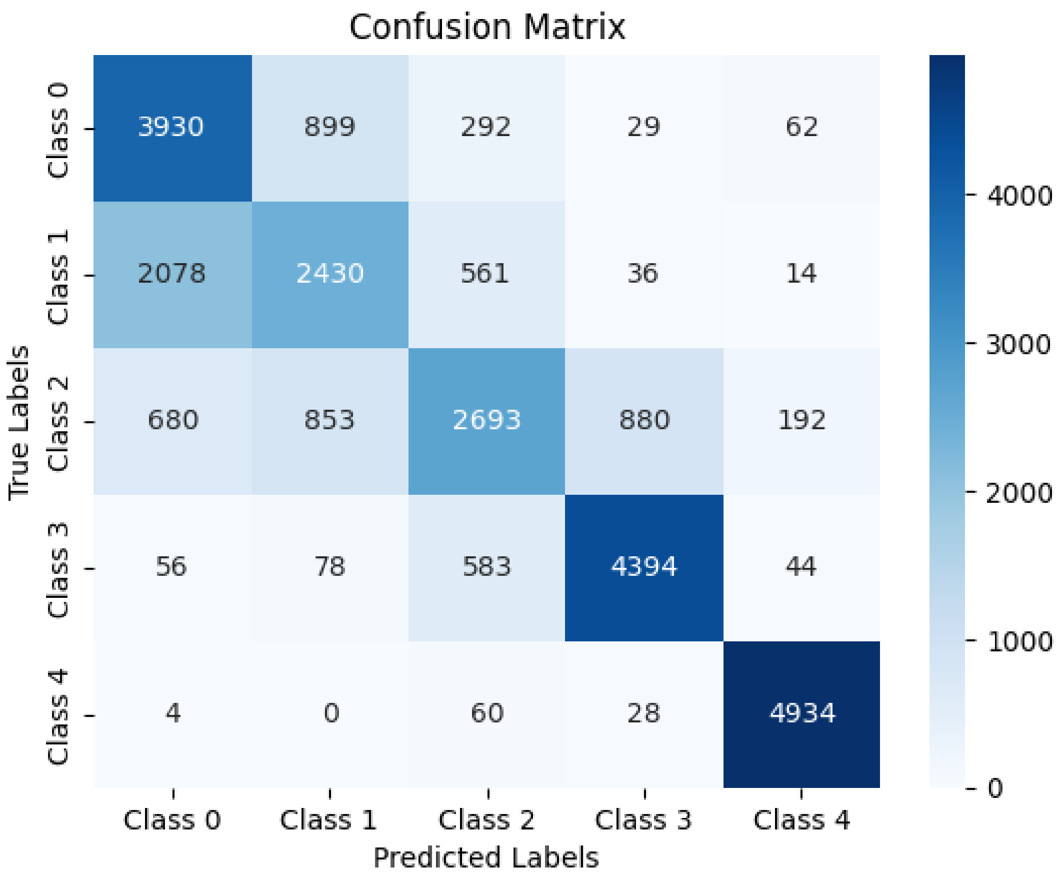

Figure 11. The confusion matrix illustrates the classification results obtained by the model, showcasing the distribution of predicted labels against the actual labels. The matrix provides a visual representation of the number of instances classified correctly and misclassified across different classes. In class 0, 899 instances were misclassified as class 1, 292 as class 2, 29 as class 3, and 62 as class 4. In class 1, 2078 instances were misclassified as class 0, 561 as class 2, 36 as class 3, and 14 as class 4 so on for other classes.

Analyzing the precision, recall, and F1-score for each class, it was observed that class 4 achieved the highest scores, indicating excellent performance. Classes 0, 1, 2, and 3 also achieved moderate scores, showing reasonable accuracy in their predictions. The overall weighted average of precision, recall, and F1-score is approximately 71%, consistent with the balanced accuracy.

The primary observation from these two training results is that the utilization of a pre-trained model yields more precise feature capturing compared to training the model from scratch. When using pre-trained weights, the model benefits from the knowledge gained from a large dataset, such as ImageNet, which helps in learning generalized and discriminative features. This transfer of knowledge enhances the model's ability to recognize patterns and distinguish between different classes. On the other hand, training a model from scratch requires learning features solely from the provided dataset. This process may be more challenging as the model has to start from random weights and iterate through numerous training iterations to develop effective features.

5. Discussion

A comparison of Experiment 1 and Experiment 2 reveals that Experiment 2 achieved a noteworthy accuracy improvement of over 0.06. This improvement can be attributed to the utilization of a preprocessed dataset specifically designed for EfficientNet-B3. However, the quadratic kappa score improved only by 0.01 for these two experiments. Nonetheless, the F1-Score improved significantly within these two experiments among class 2, 3, and 4. Overall, in experiments 1, 2 and 3 the models have varying performance across different classes, with some models performing better in certain classes than others. However, the performance for class 1 is generally the lowest among experiments mentioned, indicating this class may be more difficult to classify accurately. Comparative analysis of the models trained from scratch and with pre-trained weights reveals that the pre-trained models exhibit faster feature learning on this dataset than the models trained from scratch. Nonetheless, by solving the imbalance data in Experiment 5 I have achieved the highest accuracy of 0.94 and kappa score of 0.964. The reference standard used for the study was the majority decision of ophthalmologist graders (Quellec et al., 2017b). The dataset is manually labeled by professional ophthalmologists. However, there may be small details in the images that even the doctors didn't notice, so training based on that dataset, the model might struggle to tell the difference between two classes. Another possible limitation is the nature of the neural networks because the network was provided with only the images and associated level of disease without explicit definitions of features. Therefore, the network learned those features that were most predictive for referability implicitly, which could include features unknown or be ignored by ophthalmologists. In future, retinopathy detection could involve several areas of research. One possible area of focus is to use multimodal data sources, such as combining retinal images with other clinical data, for instance, blood sugar level or patient medical history to improve accuracy and reliability. Finally, research efforts could also focus on deploying and testing the effectiveness of diabetic retinopathy detection models in real-world clinical settings, such as in telemedicine or remote patient care scenarios, to ensure that these tools are practical and useful for healthcare providers.

6. Conclusion

Automated screening systems have been proving as an effective tool in reducing the time and costs associated with diagnosing patients, especially in the field of ophthalmology. By leveraging deep learning systems, Ophthalmologists can discover potential diagnoses and start therapy on time. Extensive research is essential in the field of detecting DR at an early stage, as these systems significantly contribute to the process. In this thesis I have explored issues related to screening of this disease and proposed possible solutions using deep learning techniques. Furthermore, I investigated the effects of image preprocessing techniques, as well as data oversampling and down-sampling methods, on the accuracy of the models. Additionally, I compared the impact of using pre-trained versus non-pretrained weights on the models' performance. Through this process, one transfer learning solution emerged that performs the best when combined with image preprocessing technique. Where, the best accuracy 0.94 and quadratic kappa score 0.964 was reported on multi-label classification method with EfficientNet-B3 pretrained model. Furthermore, I have included potential explanations for a failure to detect certain classes accurately and highlighted areas that would benefit from future research and development.

7. References:

Banerjee, M., Capozzoli, M., McSweeney, L., & Sinha, D. (1999). Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, *27*(1), 3–23. https://doi.org/10.2307/3315487

Bhaskaranand, M., Ramachandra, C., Bhat, S., Cuadros, J., Nittala, M. G., Sadda, S. R., & Solanki, K. (2019). The Value of Automated Diabetic Retinopathy Screening with the EyeArt System: A Study of More Than 100,000 Consecutive Encounters from People with Diabetes. *Diabetes Technology & Therapeutics*, *21*(11), 635–643. https://doi.org/10.1089/dia.2019.0164

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). *API design for machine learning software: experiences from the scikit-learn project*.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access*, *9*, 78368–78381. https://doi.org/10.1109/ACCESS.2021.3084050

Chollet, F. (2016). *Xception: Deep Learning with Depthwise Separable Convolutions*.

Condori, H. C., de la Cruz, J. C., & Machaca, W. M. (2021). ResNet neural network hyperparameter tuning for Rigid Pavement Failure Assessment. *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 317–322. https://doi.org/10.1109/SACI51354.2021.9465547

Delaunay, X., Courtois, A., & Gouillon, F. (2019). Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netCDF-4 or HDF5 files. *Geoscientific Model Development*, *12*(9), 4099–4113. https://doi.org/10.5194/gmd-12-4099-2019

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *CVPR09*.

Fleiss, J. L., & Cohen, J. (1973). The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. *Educational and Psychological Measurement*, *33*(3), 613–619. https://doi.org/10.1177/001316447303300309

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016a). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, *316*(22), 2402. https://doi.org/10.1001/jama.2016.17216

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. Q. (2017). *Multi-Scale Dense Networks for Resource Efficient Image Classification*.

Huang, K.-K., Ren, C.-X., Liu, H., Lai, Z.-R., Yu, Y.-F., & Dai, D.-Q. (2021). Hyperspectral image classification via discriminative convolutional neural network with an improved triplet loss. *Pattern Recognition*, *112*, 107744. https://doi.org/10.1016/j.patcog.2020.107744

Khalifa, N., Loey, M., Taha, M., & Mohamed, H. (2019). Deep Transfer Learning Models for Medical Diabetic Retinopathy Detection. *Acta Informatica Medica*, *27*(5), 327. https://doi.org/10.5455/aim.2019.27.327-332

Nvidia Apex. (n.d.).

Palavalasa, K. K., & Sambaturu, B. (2018). Automatic Diabetic Retinopathy Detection Using Digital Image Processing. *2018 International Conference on Communication and Signal Processing (ICCSP)*, 0072–0076. https://doi.org/10.1109/ICCSP.2018.8524234

Pires, R., Avila, S., Wainer, J., Valle, E., Abramoff, M. D., & Rocha, A. (2019). A data-driven approach to referable diabetic retinopathy detection. *Artificial Intelligence in Medicine*, *96*, 93–106. https://doi.org/10.1016/j.artmed.2019.03.009

Quellec, G., Charrière, K., Boudi, Y., Cochener, B., & Lamard, M. (2017a). Deep image mining for diabetic retinopathy screening. *Medical Image Analysis*, *39*, 178–193. https://doi.org/10.1016/j.media.2017.04.012

Ruamviboonsuk, P., Krause, J., Chotcomwongse, P., Sayres, R., Raman, R., Widner, K., Campana, B. J. L., Phene, S., Hemarat, K., Tadarati, M., Silpa-Archa, S., Limwattanayingyong, J., Rao, C., Kuruvilla, O., Jung, J., Tan, J., Orprayoon, S., Kangwanwongpaisan, C., Sukumalpaiboon, R., … Webster, D. R. (2019). Deep learning versus human graders for classifying diabetic retinopathy severity in a nationwide screening program. *Npj Digital Medicine*, *2*(1), 25. https://doi.org/10.1038/s41746-019-0099-8

Safi, H., Safi, S., Hafezi-Moghadam, A., & Ahmadieh, H. (2018). Early detection of diabetic retinopathy. *Survey of Ophthalmology*, *63*(5), 601–608. https://doi.org/10.1016/j.survophthal.2018.04.003

Sim, J., & Wright, C. C. (2005). The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, *85*(3), 257–268. https://doi.org/10.1093/ptj/85.3.257

Tan, M., & Le, Q. V. (2019a). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*.

Ting, D. S. W., Cheung, C. Y.-L., Lim, G., Tan, G. S. W., Quang, N. D., Gan, A., Hamzah, H., Garcia-Franco, R., San Yeo, I. Y., Lee, S. Y., Wong, E. Y. M., Sabanayagam, C., Baskaran, M., Ibrahim, F., Tan, N. C., Finkelstein, E. A., Lamoureux, E. L., Wong, I. Y., Bressler, N. M., … Wong, T. Y. (2017a). Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images From Multiethnic Populations With Diabetes. *JAMA*, *318*(22), 2211. https://doi.org/10.1001/jama.2017.18152

Torrey, L., & Shavlik, J. (2010). Transfer Learning. In *Handbook of Research on Machine Learning Applications and Trends* (pp. 242–264). IGI Global. https://doi.org/10.4018/978-1-60566-766-9.ch011

Xu, K., Feng, D., & Mi, H. (2017). Deep Convolutional Neural Network-Based Early Automated Detection of Diabetic Retinopathy Using Fundus Image. *Molecules*, *22*(12), 2054. https://doi.org/10.3390/molecules22122054

Yu, K.-H., Beam, A. L., & Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nature Biomedical Engineering*, *2*(10), 719–731. https://doi.org/10.1038/s41551-018-0305-z

Zhang, W., Zhong, J., Yang, S., Gao, Z., Hu, J., Chen, Y., & Yi, Z. (2019). Automated identification and grading system of diabetic retinopathy using deep neural networks. *Knowledge-Based Systems*, *175*, 12–25. https://doi.org/10.1016/j.knosys.2019.03.016

Wang, W., & Lo, A. C. Y. (2018). Diabetic Retinopathy: Pathophysiology and Treatments. International journal of molecular sciences, 19(6), 1816. https://doi.org/10.3390/ijms19061816

Padhy, S. K., Takkar, B., Chawla, R., & Kumar, A. (2019). Artificial intelligence in diabetic retinopathy: A natural step to the future. Indian journal of ophthalmology, 67(7), 1004–1009. https://doi.org/10.4103/ijo.IJO_1989_18

Nneji GU, Cai J, Deng J, Monday HN, Hossin MA, Nahar S. Identification of Diabetic Retinopathy Using Weighted Fusion Deep Learning Based on Dual-Channel Fundus Scans. Diagnostics. 2022; 12(2):540. https://doi.org/10.3390/diagnostics12020540

Sumit Saha. A comprehensive guide to convolutional neural networks- the ELI5 way. 2018; https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53