

**COVENTRY
UNIVERSITY**

**Faculty of Engineering, Environment and Computing School of
Computing, Electronics and Mathematics**

MSc. Data Science and Computational Intelligence
7151CEM
Computing Individual Research Project

Diabetic Retinopathy Detection Using Deep Learning

Author : **Aleena Alby**
SID : 11865340

1st Supervisor : **Prof. James Brusey**
2nd Supervisor : **Prof. Fei He**

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science and
Computational Intelligence

Academic Year: 2022/23

Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Statement of copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see www.coventry.ac.uk/ipr or contact ipr@coventry.ac.uk.

Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (<https://ethics.coventry.ac.uk/>) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed:



Date: 08-Dec-2022

First Name:	Aleena
Last Name:	Alby
Student ID number:	11865340
Ethics Application Number	P142645
1 st Supervisor Name	Prof. James Brusey
2 nd Supervisor Name	Dr Fei He

Abstract

Diabetic retinopathy is a complication of diabetes caused by high blood sugar levels damaging the back of the eye (retina) (“Diabetic Retinopathy - Illnesses & Conditions | NHS Inform,” n.d.). It can cause blindness if left undiagnosed and untreated. The more severe form of diabetic retinopathy, known as proliferative diabetic retinopathy (PDR), involves the growth of new blood vessels on the retinal surface. In order to prevent blindness, early DR identification is crucial. Deep learning is a long-term solution for screening, diagnosing, and keeping track of patients in primary healthcare. In this study, we compare various deep-learning models for DR detection. The models include CNN, EfficientNetV2L, ResNet50, VGG19 and DenseNet201. These models were developed using the Kaggle dataset, which has around 35,126 images with a 2416*1736 resolution. Methods including undersampling, contrast enhancement, and data preparation were used to improve the accuracy of the models. The unbalanced and bad dataset quality were major difficulties faced during the study. The stages of diabetic retinopathy are accurately identified using the residual network model.

Keywords— *Diabetic Retinopathy Detection; Residual Network; Transfer Learning; Deep Learning*

Contents

Abstract	3
Acknowledgements	9
Introduction	10
Background to the Project	11
Artificial Neural Networks	11
Convolutional neural networks (CNN)	11
Transfer Learning	13
Project Objectives	14
Overview of This Report	14
Literature Review	15
Summary	16
Methodology	17
Agile Methodology	17
Evaluation Metrics	18
Requirements	19
Dataset	19
Skill Requirements	20
System Requirements	20
Functionality Requirements	20
Design	21
Implementation	22
Data Loading	22
Data Preprocessing	23
Data Splitting	25
Model Creation	26
Implementation of Machine learning model for image classification	26
Implementation of Deep Learning for image classification	26
Model Evaluation	30

Results	31
ResNet50 Model	31
EfficientNetV2L Model	31
DenseNet201 Model	32
VGG19 Model	32
CNN	33
KNN	33
Plots for the model	34
ResNet50	34
EfficientNetV2L	34
DenseNet201	34
VGG19	35
CNN	35
Fine tuning the model	35
Project Management	37
Project Schedule	37
Risk Management	37
Quality Management	37
Social, Legal, Ethical and Professional Considerations	37
Conclusions	38
Achievements	38
Future Work	38
Student Reflections	39
References	40

List of Tables

2	DR detection methods	16
3	(“Diabetic Retinopathy - Stages” 2017)	19
4	System Configuration	20
5	Diabetic Retinopathy Dataset	25
6	Risk management	37
7	Transfer learning models comparison	38

List of Figures

1	Normal Retina Vs Diabetic Retinopathy Retina - Source: (Associates, n.d.)	10
2	A neural network with three hidden neurons in one hidden layer and four inputs- Source: (otexts.com, n.d.).	11
3	Typical CNN architecture- Source: (A. Kumar 2022)	12
4	Convolutional Layer - Source: (Reynolds, n.d.)	12
5	Pooling in CNN- Source: (IndoML.com 2018)	12
6	Dropout Layer - Source: (Budhiraja 2018)	13
7	Flatten Layer - Source: (Kongsilp 2019)	13
8	Transfer Learning Concept - Source (Turing Enterprises Inc, n.d.)	14
9	Agile Methodology - Source: (“Synopsys” 2017)	17
10	Dataset	19
11	High Level Design	21
12	Images Label CSV	22
13	The Orginal dataset	22
14	Sample Image	23
15	Resized image dimension	23
16	Unclear Image	23
17	Before Implementing CLAHE	24
18	After implementing CLAHE	24
19	Implementing CLAHE method	24
20	Data Distribution	25
21	Python code for under-sampling	25
22	Data Splitting	25
23	KNN diagram	26
24	Model Summary	27
25	Model construction	28
26	Early Stopping and checkpoints	28
27	Model compilation	29
28	Model Training	29
29	model Compilation	30
30	Classification Report	31
31	Classificatio report of EfficentNetV2L	31
32	Classification model of DeneseNet201	32
33	Classification report of VGG19	32
34	Classification report of CNN	33

35	Classification report of KNN	33
36	10 fold cross-validation	33
37	Fine tuning	35
38	Prediction using ResNet50	36

Acknowledgements

I would like to begin by expressing my sincere gratitude to Prof. James Brusey, who served as my first supervisor and offered continuous encouragement and constant guidance throughout this research. His feedback has enabled me to further improve my work. I'd also like to express my gratitude to Dr Fei He, who serves as my second supervisor, for his assistance with the project.

I want to express my gratitude to God and my parents, and my brother for always being there for me, even when I felt like finishing the project would be impossible and beyond my capabilities. Most importantly, I wish to thank my cousin, Vinojo Francis and his family, for their moral support.

I also want to express my appreciation and gratitude to every faculty member who has instructed me in this course and imparted their invaluable expertise, which has enabled me to finish this assignment.

Introduction

One of the major causes of eye vision loss is diabetes. While delayed examination would have a higher effect on the retinal area of the eye, early detection of diabetes is crucial. The key factors affecting the rise in the occurrence of this disease are people's lifestyles and other contributing factors, and it is anticipated that this trend will continue. According to Tien Y Wong et al, among the 285 million diabetics worldwide, 33 percent of those individuals exhibit DR symptoms(R, Ty, and C 2015). Nearly 90% of individuals can be diagnosed, and long-term effects can be reduced, with thorough screening and regular checkups. Statistics show that the percentage of diabetic patients over the age of 18 has increased from 4.7% to 8.5%. About 61.3 million Indians between the ages of 20 and 79 have been diagnosed as diabetic patients. By 2030, it is predicted that this number would likely rise to 101.2 million (Whiting et al. 2011).

The retinal blood vessels are harmed by a blood glucose spike. This rise in blood glucose levels primarily damages blood vessels, allowing blood to seep into the eyes and weakening the visual system. The brain activates the neighbouring cells to take control of the situation when it detects a blood leak. This behaviour causes aberrant blood vessel development. Newly created vessels lack strength. In time, they affect the person's vision. Therefore, it is essential that a diabetic patient constantly test routine eye exams. The significant issue here is that DR is primarily an asymptomatic eye condition that does not manifest distinctive symptoms until a late stage is reached. The manual examination of retinal image features is a challenging and taxing task, nevertheless. Many automated diagnostic technologies have been created recently to help ophthalmologists examine retinal abnormalities, which has helped to solve this problem.

In this paper, diabetic retinopathy is classified according to severity. To aid doctors in the diagnosis and treatment of diseases, we are building different neural networks through deep learning, comparing the structures and selecting the optimal network. A correct diagnosis of diabetic retinopathy is crucial to detecting the disease early.

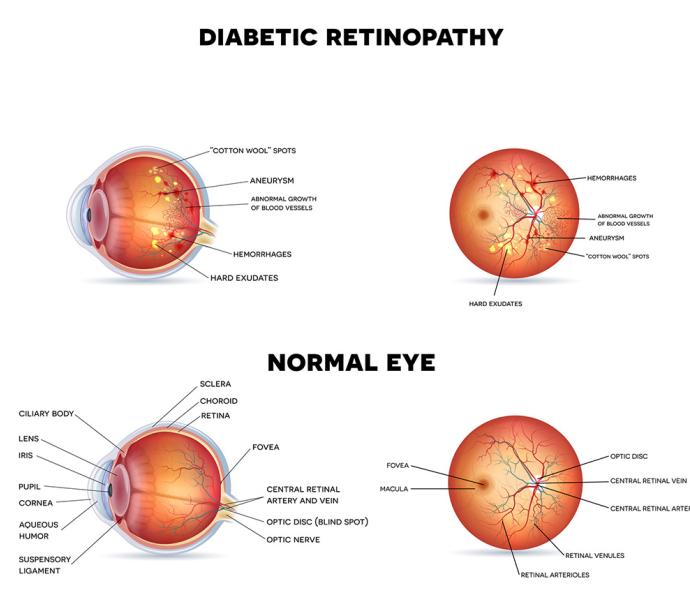


Figure 1: Normal Retina Vs Diabetic Retinopathy Retina - *Source:(Associates, n.d.)*

Background to the Project

Artificial intelligence (AI) will be used more often in the healthcare sector as a result of the complexity and growth of data within the field (Davenport and Kalakota 2019). Numerous studies have proven that AI is capable of doing important healthcare tasks including disease diagnosis as well as or better than humans. Deep learning, or neural network models with many levels of features or variables that predict outcomes, is one of the most difficult types of machine learning.

Artificial Neural Networks

Artificial neural networks have a structure that is comparable to biological neural networks. The history of neural networking probably began in the late 1800s with the pursuit for scientific understanding of how the human brain worked. In current artificial neural networks, the neuron model McCulloch and Pitts proposed in 1943 is still utilized. The first artificial neural network was developed by Narvin Minsky ("History of Artificial Neural Network - Javatpoint," n.d.).

Artificial neural networks use a range of mathematical processing layers to learn the data provided to them. Perceptrons are neural networks with only one layer. The input units (receptor), connection weights, summation function, computation, and output units make up an artificial neuron (effectors). The input layer receives information. The purpose of the neural network is to process or learn from this data. Data enters the input layer and then passes via one or more hidden levels. It is the job of the hidden layers to transform the input such that the output layer can use it. The weights typically represent the strength of the connections between the neurons. The activation function is used to obtain the desired result for the given problem. There are several activation functions, For instance, ReLU, linear regression, logistic regression, identity function, binary sigmoid, bipolar sigmoid, bipolar, hyperbolic tangent (V 2021). An enormous amount of data, known as a training set, must be provided to ANNs in order for them to learn.

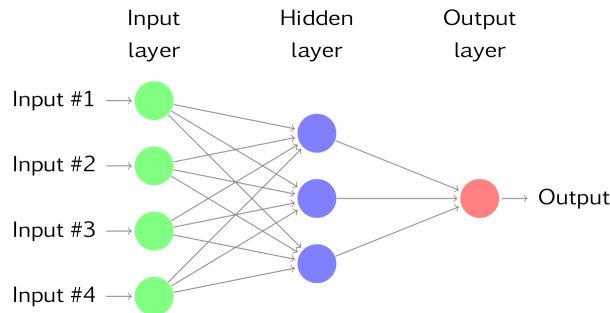


Figure 2: A neural network with three hidden neurons in one hidden layer and four inputs- **Source:** (otexts.com, n.d.).

According to a study of AI applications in health care, Artificial neural networks have been used in important illness areas like cancer or cardiology etc. Clinical diagnosis, cancer prediction, speech recognition, image analysis, and drug research are examples of uses of ANN in the healthcare industry (Shahid, Rappo, and Berta 2019).

Convolutional neural networks (CNN)

Convolutional neural networks, also known as convnets or CNNs. CNN is a powerful and effective image processing algorithm. It is a subset of the feed-forward artificial neural network models that are employed for diverse purposes. In the late 1990s, Bell Labs professor Yann LeCunn developed the first effective convolution networks. Convolutional Neural Networks specialized in image & video recognition applications. CNN is primarily utilised for image analysis applications such segmentation, object detection, and image recognition (Tripathi 2021). It has millions of parameters, a large number of hidden layers, and input and output layers.

Prior to applying an activation function, it subsamples the input using convolution and pooling techniques. All of the hidden layers are partially connected , with the completely connected layer at the end resulting in the output layer. All image processing applications, including face and pattern recognition, have shown CNN to be extremely effective. There are numerous architectures used by CNN, including LeNet, AlexNet, GoogleNet, ConvNet, and ResNet.

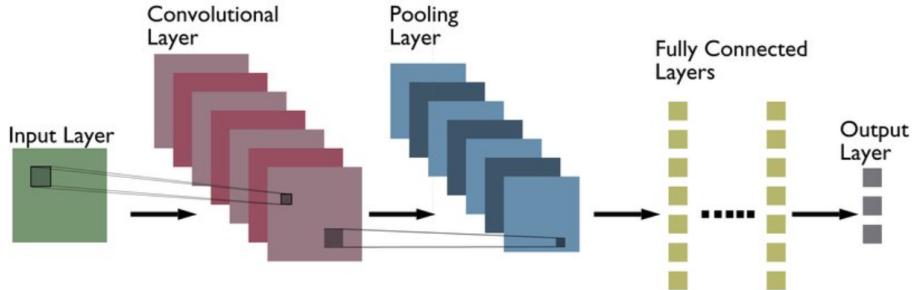


Figure 3: Typical CNN architecture- Source: (A. Kumar 2022)

Convolutional Layer

Filters are combined to produce convolutional layers, which are then applied to input images. A feature map also known as an activation map, a representation of the input image with the filters applied, is the outcome of the convolutional layer.

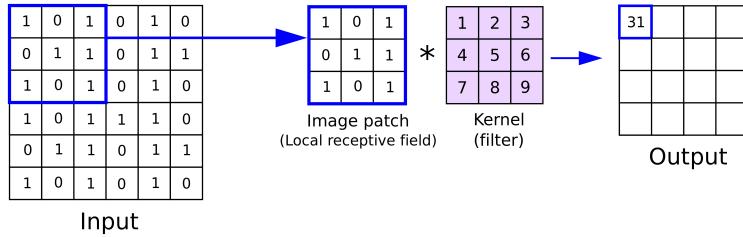


Figure 4: Convolutional Layer - Source: (Reynolds, n.d.)

Pooling layer

By pooling layers, the input can be processed more quickly and with less memory use due to its reduced spatial size. Every feature map is handled independently by the pooling layer. Pooling also facilitates a decrease in the number of parameters and accelerates training. Max pooling and average pooling are the two primary types of pooling. While average pooling uses the average value from each feature map, max pooling uses the maximum value.

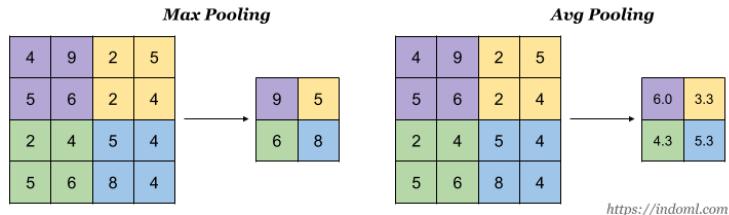


Figure 5: Pooling in CNN- Source: (IndoML.com 2018)

Dropout Layer

There are different deep neural network architectures, sometimes very deep and sometimes shallow. Nevertheless, while learning the data features, they sometimes learn the statistical noise in the dataset. This dramatically improves the model's performance on the training dataset but fails miserably when applied to the validation dataset. This is the problem of overfitting. To overcome this issue, we use the dropout layer.

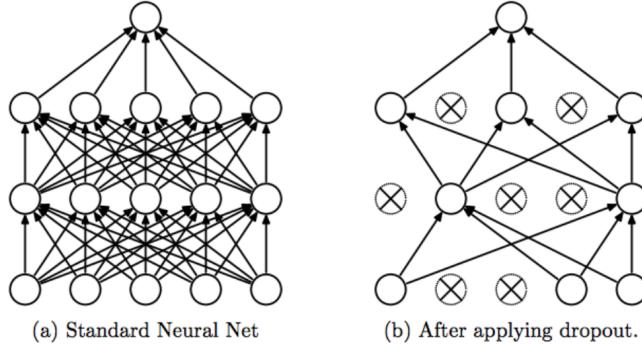


Figure 6: Dropout Layer - Source: (Budhiraja 2018)

Flatten Layer

Flattening transforms the data into a one-dimensional array before inputting it to the next layer. The convolutional layer output is flattened to create one long feature vector. Moreover, it is connected to the final classification model, a fully-connected layer.

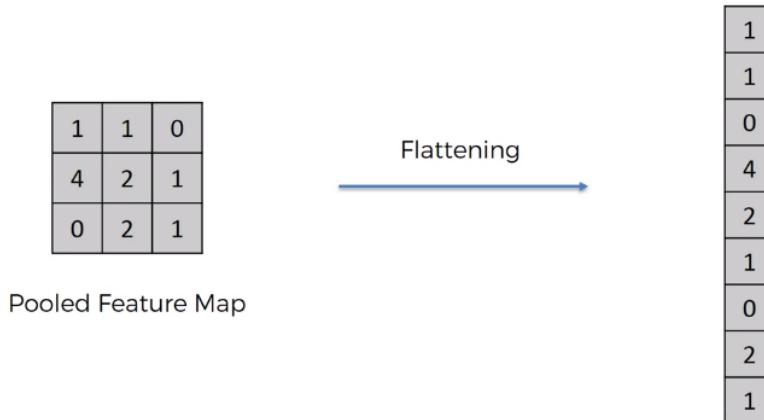


Figure 7: Flatten Layer - Source: (Kongsilp 2019)

Fully connected layer

The final layer of a convolutional neural network is fully connected layer. Each neuron in a fully connected layer is fully connected to every other neuron in the previous layer. The earlier layer's features map is flattened to a vector. The complicated correlations between high-level features are then captured by feeding this vector into a fully connected layer. The output of this layer is a feature vector with only one dimension (Pokhrel 2019).

Transfer Learning

Transfer learning is the process of applying a previously learned model to a new problem. Transfer learning is the process through which we attempt to apply the knowledge that have gained from one assignment to

another in order to better understand the concept. Transfer learning is useful in situations when it is not possible to get the large amounts of data required to train a neural network from scratch.

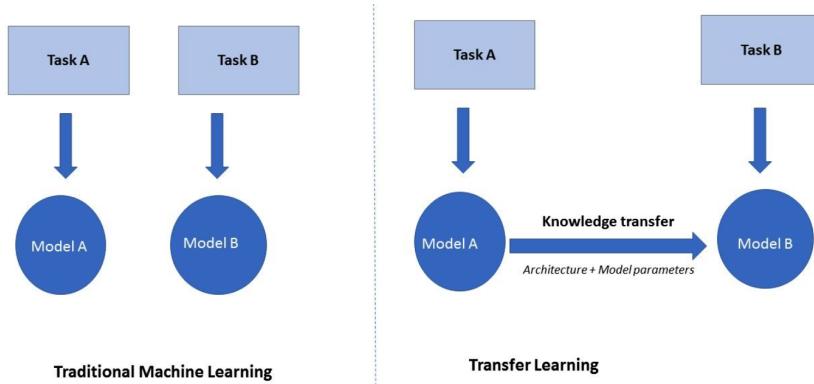


Figure 8: Transfer Learning Concept - **Source** (Turing Enterprises Inc, n.d.)

Transfer learning can provide an effective machine learning model with relatively minimal training data because the model has already been trained. Additionally, training time is reduced because it can take days to complete a complicated task after creating a deep neural network from scratch (Sharma 2021).

Project Objectives

Convolutional neural networks (CNNs) and a transfer learning model were both used in this study to build an image classification model for diagnosing diabetic retinopathy. The following objectives are considered at every stage of the modelling process:

1. Review existing classification methods and publications.
2. Obtain the required dataset.
3. Create a classification model using transfer learning and CNN.
4. Evaluate the performance of the models.

Overview of This Report

In the introduction, the report discusses the project's significance and objectives. A literature review in section 2 summarizes the major points outlined from researching relevant works by other researchers. Section 3 of the report, titled "Methodology," covers the usage of an agile methodology. Section 4 contains the dataset used for the model, the system and functional requirements, and the skills needed to implement the project. In Section 6, the implementation section is discussed, including how the models are trained using the data, how data preprocessing is performed, and how the model is trained and evaluated. In Section 7, we discuss the model outcomes, which allow us to compare them and decide which model is most efficient. Section 8 covers the project management aspect as well as various ethical, social, and legal issues. The project is concluded in Section 10, which gives a brief summary of all the implementation phases, outcomes, and predicted future scope of the project. After that comes student reflection, the report's bibliography and appendices.

Literature Review

In recent years, numerous deep learning-based automatic DR detection systems have emerged. In this section, some of the recent research projects have been addressed.

A technique for recognising diabetic retinopathy based on transfer learning is proposed by **Yuchen Wu et al.(Wu and Hu 2019)**. They used data from the official Kaggle website, and then they conducted data amplification, data flipping, data folding, and contrast adjusting. Then, employ pre-trained models such as VGG19, InceptionV3, Resnet50, etc. They oversample the data category with a small number due to the severely unbalanced data. The performance of VGG19 was 51 % accuracy, and Resnet50 was 49% InceptionV3 performed better than other models with a classification accuracy of 60%.

Using transfer learning, **Esra Kaya and Ismail Saritas** created CNN for the identification of diabetic retinopathy (Kaya and Saritas 2022). To find the best effective architecture, photographs from the DRIVE (Digital Retinal Images for Vessel Extraction) dataset of DR patients and healthy people were classified using Convolutional Neural Network (CNN) architectures as a transfer learning technique. They utilised contrast-limited adaptive histogram equalisation to improve the clarity of the image. They assessed the ResNet18, GoogleNet, and SqueezeNet CNN architectures' performances as feature extraction techniques and classifiers. For ResNet18 and squeezeNet, they employed adam optimiser, while for googleNet, they used sgdm. There are 71 layers in ResNet18, 144 layers in GoogleNet, and 68 layers in SqueezeNet. ResNet18 was discovered to be the most effective architecture as a classifier with 100% accuracy.

Fundus images from the Kaggle open-source dataset were used by **Nikhil Sathya Kumar and Dr B. Ramaswamy Karthikeyan** to identify DR using CNNs, Transformers, and MLPs (N. S. Kumar and Ramaswamy Karthikeyan 2021). The dataset includes more than 3600 photographs with a resolution of 2416*1736. ResNet and EfficientNet based on CNN, Vision-Transformer and SwinTransformer based on Transformer, and MLP-Mixer based on MLP architecture were the models chosen for this study. The findings show that, in comparison to CNN and MLP-based models, Transformer based models were more accurate. All models underwent 15 epochs of training. The most accurate Transformer-based model was Swin, with 92.49% accuracy.

ImageNet model was proposed by **Jayakumari.C et al.** (Jayakumari, Lavanya, and Sumesh 2020). Here the TensorFlow framework is used to build a convolutional neural network model in Python. The size of each image has reduced to 224 X 224 X 3. The dataset is under-sampled here because of the imbalance in the dataset. Additionally, to improve accuracy, the classes were reduced from 5 to 2. The model trained the network using the Adam optimiser and categorical cross-entropy as a loss function. The model has executed 100 epochs. The model accuracy in training was 98.8%, and the validation accuracy was 98.5%.

A CNN method was suggested by **Frans Coenen et al.** to diagnose DR with a sensitivity of 95 % and an accuracy of 75% (**Pratt et al. 2016**). They train the network using a high-end graphics processor unit (GPU) on the publicly available Kaggle dataset.

Pathak et al. classified early-stage DR using a deep learning method (Pathak et al. 2021). They have employed a variety of classifier-based techniques, including SVM (Support Vector Machine), CNN (Convolution Neural Network), DCNN (Deep Convolution Neural Network), ANN (Artificial Neural Network), NB (Naive Bayes), and threshold-based strategies. The model achieved 90% accuracy for the SVM, 91% for the ANN, 92.9% for the NB, 97% for the Thresholding-Based, and 96.5% for the DCNN. They concluded that the DCNN technique is highly accurate and productive.

Summary

Table 2: DR detection methods

Methods & Ref	Datasets Used	Techniques	Performance metrics
Transfer Learning models (Wu and Hu 2019)	Kaggle Dataset	VGG19, InceptionV3, Resnet50	VGG19 - 51%, InceptionV3 - 60%, Resnet50 - 49%
CNN (Kaya and Saritas 2022)	DRIVE dataset (40 images in the database were chosen randomly from 400 images)	They assessed the ResNet18, GoogleNet, and SqueezeNet	ResNet18 - 100 %, GoogleNet - 68.2 %, SqueezeNet - 67.4 %
CNN, MLP & Transfomer (N. S. Kumar and Ramaswamy Karthikeyan 2021)	Aptos dataset from Kaggle (6590 Images)	EfficientNet, ResNet, MLP-Mixer, ViT , ViT+MLP, Swin and Swin+ViT	EfficientNet -91.18 %, ResNet - 89.63%, MLP-Mixer - 94.47%, ViT - 91.13 %, ViT+MLP - 89.73%, Swin - 92.49%, Swin+ViT - 91.91 %
ImageNet (Jayakumari, Lavanya, and Sumesh 2020)	Kaggle Dataset	ImageNet	98.6%
CNN (Pratt et al. 2016)	Kaggle Dataset (80,000 Images)	CNN	75%
SVM classifier-Based technique, CNN classifier, DCNN classifier, ANN classifier, NB classifier, Thresholding Based techniques. (Pathak et al. 2021)	Indian Diabetic Retinopathy Image Dataset (IDRiD), High-Resolution Fundus (HRF) Image Database, Kaggle dataset	Using a classifier-based SVM (Support Vector Machine) method, Convolution neural network classifiers, DCNN (deep convolution neural network) classifiers, artificial neural network classifiers, naive bayes classifiers, and threshold-based methods are examples of classifiers.	SVM - 90% CNN - Best ANN - 91% NB - 92.6% Thresholding Based -97% DCNN - 96.5%

Methodology

The software development life cycle involves using common business procedures for creating software applications. After carefully examining a few well-known models, including Waterfall, Iterative, V-Model, and Agile approach, I have chosen to adopt the Agile model in this study.

Agile Methodology

Agile is a method of software development that aims to provide functional software consistently through rapid iterations (“What Is Agile Methodology?” 2022). An agile methodology emphasizes a cooperative, iterative, and incremental approach to project management. It comes in a variety of forms, including scrum, crystal, extreme programming (XP), and feature-driven development (FDD).

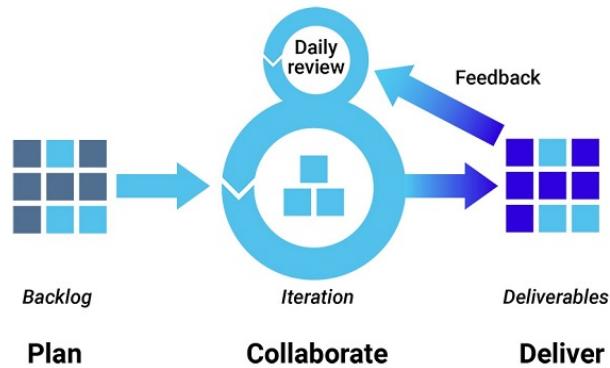


Figure 9: Agile Methodology - Source: (“Synopsys” 2017)

Pros:

1. Better quality
2. Speed and flexibility
3. cost management
4. Project's progress is fully visible and updated in real-time
5. stakeholders' participation

Agile software development's main advantage is that it enables iterative software releases. Iterative releases increase productivity by enabling teams to identify and correct flaws and set expectations early on. With regular incremental enhancements, they also enable consumers to enjoy the benefits of software earlier.

Cons:

Agile development methodologies focus on in-the-moment communication, thus new users frequently lack the necessary background information to get started. They are time- and labor-intensive because developers must finish each feature inside each iteration before asking users for approval.

Evaluation Metrics

1) **Metrics Evaluation:** We were able to determine metrics like as accuracy, precision, recall, and F1-Score Score, By making predictions on the test sets for all five models.

Accuracy: One of the popular evaluation metric is accuracy (see equation below). The total number of two right predictions ($TP + TN$) divided by the overall dataset size ($P + N$) is used to calculate accuracy. According to our evaluations, DenseNet201, VGG19, and EfficientNet V2L all have accuracy values of 61%, while ResNet50 has the greatest accuracy value of 63%.

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} = \frac{TP + TN}{P + N}$$

Precision and Recall: Other commonly used evaluation metrics include precision and recall. Recall tells us about what portion of actual positives is correctly identified, while precision shows us clearly what percentage of projected positives is indeed a true positive.

$$PREC = \frac{TP}{TP + FP}$$

$$RECALL = \frac{TP}{TP + FN}$$

F1-Score: The F1-score, which is the harmonic mean of Precision and Recall.

$$F1 - SCORE = \frac{(1 + \beta^2)(PREC \cdot REC)}{(\beta^2 \cdot PREC + REC)}$$

Typically, beta is 0.5, 1, or 2.

Requirements

In this section, I described the steps necessary to finish this research paper. I preferred the agile development methodology outlined in section 3. Since it allows us to change the requirements in accordance to how the project is evolving.

Dataset

Finding the appropriate dataset is essential for the creation of a best model. This study using dataset available at Kaggle("Diabetic Retinopathy Detection" 2015). This Retinal images were provided by EyePACS. The dataset containing large set of high-resolution retina images taken under a variety of imaging conditions. For each image, a left and right field is provided. Images are identified by a image id and either the left or right eye (for example, 1 left.jpeg represents the patient number 1's left eye).

Table 3: ("Diabetic Retinopathy - Stages" 2017)

DR classes	Level	Description
No DR	0	Healthy Retina (Normal)
Mild	1	Retina with tiny bulges (microaneurysms)
Moderate	2	Retina with microaneurysms, higher risk of developing vision problems in the future
Severe	3	Retina with severe and widespread microaneurysms, including bleeding into the retina
Proliferative	4	New blood vessels and scar tissue have formed on your retina, which can cause significant bleeding and lead to retinal detachment

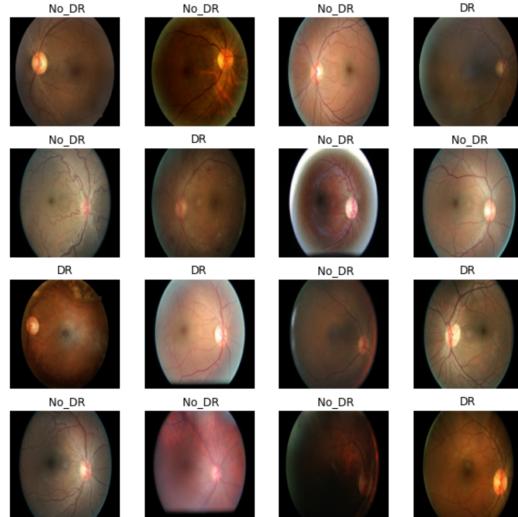


Figure 10: Dataset

Skill Requirements

We need to acquire a few specific deep learning related skills before we can begin our research project.

- Python
- Probability
- Keras
- Tensorflow
- Computer Science Fundamentals and Programming.
- Machine Learning Knowledge.
- Knowledge of Deep Learning Algorithms.
- Knowledge of Deep Learning Frameworks.

System Requirements

Table 4: System Configuration

System Type	64 bit
OS Name	Mac os mojave
Processor	Intel Core i5
Installed Physical Memory (RAM)	8 GB

Functionality Requirements

- Acquire the dataset of retinal images
- Employ preprocessing techniques
- Divide the dataset into training and testing in the proportion of 70-30%
- Build the CNN and Transfer Learning models for classification
- Plot the training and validation accuracy of the models
- Evaluate the models based on the classification report
- Fine Tune the model

Design

This programme is made up of various components, each of which carries out a distinct function. The blocks and their operation are shown in figure 18. Figure 19 illustrates how all the blocks are connected to create a high-level design. Below are brief descriptions of each block's function.

1. **Data Loading:** This data set was provided by EyePacs and was downloaded from Kaggle. The data was part of a Kaggle competition. The dataset contains high-resolution retinal images. The images are categorized into five categories based on the severity of the lesions. The TensorFlow input pipeline is used to load data. In tf.data.dataset, the data is loaded in chunks or batches, which makes it possible to manage large amounts of data.
2. **Data Preprocessing and Splitting:** In any machine learning project, preprocessing data is a critical step. It involves transforming or encoding data in order to make it easier for machines to parse. An algorithm must be able to easily interpret the features of data in order to be accurate and precise in its predictions. Using OpenCV, we reduced the image's resolution to 256×256 pixels. Furthermore, images that have more than 40% black and white pixels are eliminated. CLAHE also attempted to improve the contrast of images. The under-sampling method was also implemented due to the imbalance in the dataset.
Additionally, the dataset will also be divided into a training set and a testing set, which will enable us to train our model using the training set. The test set is analyzed to assess our model's performance and determine evaluation measures including recall, accuracy, and precision.
3. **Model Building:** We will create a CNN and transfer learning model for our classification in this block. To obtain the best accurate model, we will train the model using different hyperparameters.
4. **Model Evaluation and Fine tuning:** In order to identify the best model, we will evaluate each model's accuracy, recall, precision, and F1 score. We will later fine-tune the chosen model for better outcomes. Additionally, this block makes sure that data is recorded and stored on disc for future analysis.

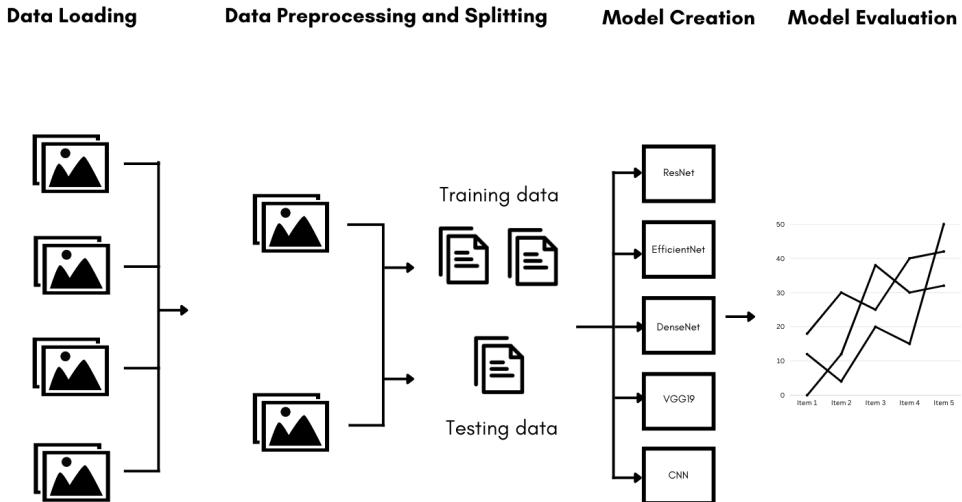


Figure 11: High Level Design

Implementation

In this section, we discuss how the project is implemented. The code for the project has been written in Python and is stored on GitHub. Data preprocessing was performed using Jupyter notebook, and model training was performed using Google Colab Pro. Google Collaboratory is a service that allows users to run their Python code directly through their web browser using high-end Collaboratory hardware. Since we are using pre-trained models, completing our research by training each model will take a long time. Because of its tremendous computational capacity, Google Colaboratory Pro makes it much simpler and quicker to train the models. Data was loaded and preprocessed using Tensorflow, Keras, and OpenCV, and the matplotlib package generated graphs. Additionally, OpenCV is used for removing blurred images and implementing the CLAHE method. The Keras API and TensorFlow library gave us all the tools we needed to complete our deep-learning experiment.

Data Loading

First, the dataset was downloaded from the Kaggle website. The data was in the form of a zip file with a 30 GB size. The downloaded data was unzipped using the keka extractor. The dataset's size became 58.5 GB after unzipping. It contains 35,126 high-resolution RGB images and a CSV file with the scores for the images.

In [11]: LabelsCsv.head(5)	
Out[11]:	
	image level
0	10_left 0
1	10_right 0
2	13_left 0
3	13_right 0
4	15_left 1

0 – Normal
1 – Mild NPDR
2 – Moderate NPDR
3 – Severe NPDR
4 – PDR

Figure 12: Images Label CSV

Tensorflow's input pipeline was utilised to load the dataset successfully. The result of this is a file called tf.data.Dataset. Using matplotlib, we first determined the image's original dimensions. The image's dimensions were 1944 x 2592. Figure x shows a sample image from the orginal dataset. For visualisation, we choose eight images from the dataset.

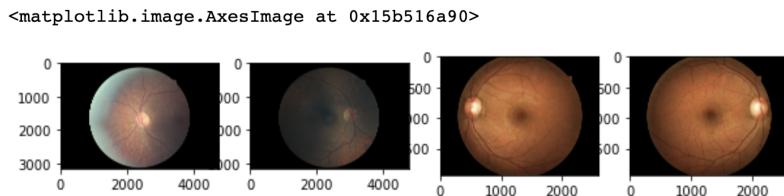


Figure 13: The Orginal dataset

```
#Show the image with matplotlib
%matplotlib inline
plt.imshow(img_size)
plt.show()

print('Original Dimensions : ',img_size.shape)
Original Dimensions :  (1944, 2592, 3)
```

Figure 14: Sample Image

Data Preprocessing

Image preprocessing is a significant phase in image classification. As we mentioned earlier, the image size was massive, so in order to speed up the process, we reduced the image size to 256 * 256 pixels using the cv2.resize() function of the OpenCV library cv2.

```
print('Resized Dimensions : ',resized.shape)
Resized Dimensions :  (256, 256, 3)
```

Figure 15: Resized image dimension

After examining the images, we apprehended that some images are blurred and dark. This kind of image influences the performance of the models. So utilizing the help of NumPy, we removed the images with 40% of black and white pixels. There were 15068 unclear images.

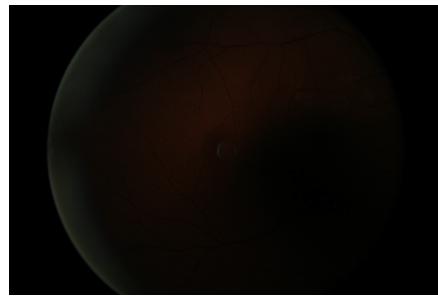


Figure 16: Unclear Image

Enhancement using Contrast Limited Adaptive Histogram Equalization (CLAHE)

After resizing and removing the blurred images, there were 20,059 images. Then We attempted to implement the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique employed by B. Ramasubramanian and S. Selvaperumal in their study to enhance the image's contrast.

Using contrast-limited adaptive histogram equalisation (CLAHE), the green channel of the retinal picture is divided into contextual regions, and histogram equalisation is then applied to each zone. CLAHE performs

contrast limiting and high precision histogram equalisation in small patches or tiles (Senaratne 2020). Instead of processing the entire image, CLAHE works with discrete sections of it called tiles. The false borders are then eliminated by combining the adjacent tiles using bilinear interpolation. To enhance the contrast of photographs, this method is used.

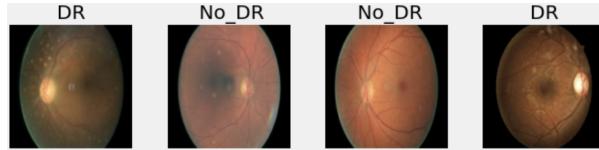


Figure 17: Before Implementing CLAHE

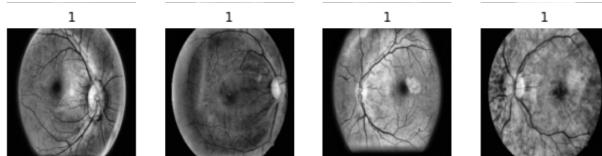


Figure 18: After implementing CLAHE

CLAHE was implemented using the cv2.createCLAHE() function of the OpenCV library with two clip limits and an eight-title grid size. The images are then saved in another new folder.

```
for img in glob.glob(inputFolder + "/*.jpeg"):
    #print(img)
    filename = img.rsplit('/', 1)[-1]
    #print(filename)
    image = cv2.imread(img)
    image_bw = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    #print(clahe)
    cl1 = clahe.apply(image_bw)
    cv2.imwrite(os.path.join(outputFolder, filename), cl1)
```

Figure 19: Implementing CLAHE method

After executing CLAHE technique, we separate the images into five folders based on the class labels 0 to 5. Below figure shows the number of images in each folder.

```
['/Volumes/Extreme SSD/DR/Classes/ No_DR', '/Volumes/Extreme SSD/DR/Classes/Mild', '/Volumes/Extreme SSD/DR/Classes/Moderate', '/Volumes/Extreme SSD/DR/Classes/Proliferative DR', '/Volumes/Extreme SSD/DR/Classes/Severe']
14906
1447
2866
404
435
```

Figure 20: Data Distribution

The dataset was imbalanced. The class with No DR have the majority of images. It will impact the experiment's outcome. To handle the imbalanced dataset, first, we reduced the number of classes to two, "With DR" and "Without DR." Then we implemented the under-sampling technique to balance the dataset.

```
path = r'/Volumes/Extreme SSD/DR/CLAH_E_new/0'
tot_no = len(os.listdir(path))
print("Total number of images in the folder:", tot_no)
n = tot_no - 10000 # Number of random images to be removed
print("Total number of images removed:", n)
img_names = os.listdir(path) # Get image names in folder
img_names = random.sample(img_names, n) # Pick 2500 random images
for image in img_names: # Go over each image name to be deleted
    f = os.path.join(path, image) # Create valid path to image
    #print(f)
    #break
    os.remove(f) # Remove the image
```

Figure 21: Python code for under-sampling

Table 5: Diabetic Retinopathy Dataset

Type of Diabetics Diagnosis	No. of Images
With DR	5152
Without DR	5152
Total	10304

Data Splitting

Using the `tf.keras.utils.image_dataset_from_directory()` function, we divided the dataset into training and validation sets. Before the split, we shuffled the dataset. The dataset has been divided into training and validation in a 7:3 ratio.

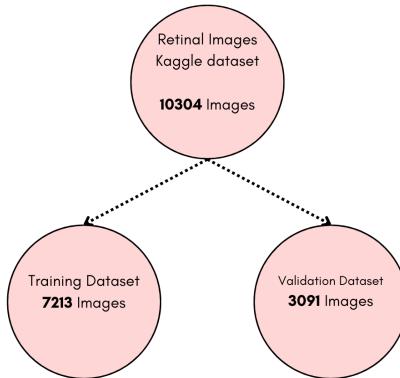


Figure 22: Data Splitting

Model Creation

Implementation of Machine learning model for image classification

This section will examine how we classified images using the KNN Classifier, the simplest supervised machine learning method. We will import our dataset into Sklearn and split it into training and testing sets. With three colour channels, each image is $256 * 256$ pixels in size. During preprocessing, the data are normalized by dividing x_{train} and x_{test} by 255. In order to use the image for categorization using machine learning techniques, we must first convert it into a 2d array because we are using sklearn.

K Nearest Neighbor

The k-nearest neighbours algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point (“What Is the k-Nearest Neighbors Algorithm? | IBM,” n.d.). By looking at which group the data points closest to a given data point belong to, it calculates the likelihood that a given data point will join that group or a different one.

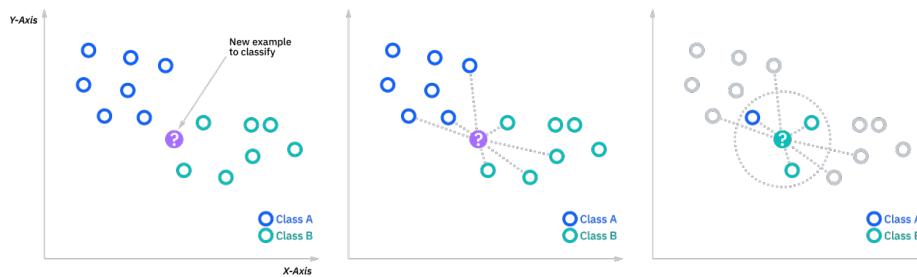


Figure 23: KNN diagram

Implementation of Deep Learning for image classification

Classification using CNN

Convolutional Neural Networks are a subset of artificial neural networks used in deep learning and are frequently employed for object and picture recognition and categorization. During the training phase, it includes feature extraction and weight calculation. The images are initially forwarded to a feature extraction network, and then the extracted features are subsequently sent to a classifier network.

We employed the keras library to construct the model. Hyperparameters like kernel size, stride, padding, and the number of feature mappings must also be considered while developing a model. Here we used batch size 16. We have added layers like batch normalization and dropout in addition to the max pooling layer. The model is then connected to a flatten layer, which creates a one dimension feature vector. A softmax activation function is used to predict the probabilities for each class using the vector before it is handed down to the dense or fully linked layer.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 32)	896
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0
batch_normalization (BatchN ormalization)	(None, 128, 128, 32)	128
dropout (Dropout)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 124, 124, 64)	51264
max_pooling2d_1 (MaxPooling 2D)	(None, 31, 31, 64)	0
batch_normalization_1 (Bathc hNormalization)	(None, 31, 31, 64)	256
dropout_1 (Dropout)	(None, 31, 31, 64)	0
flatten (Flatten)	(None, 61504)	0
dense (Dense)	(None, 512)	31490560
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_3 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_4 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 2)	66
<hr/>		
Total params: 31,717,730		
Trainable params: 31,717,538		
Non-trainable params: 192		

Figure 24: Model Summary

Model Construction

In the first convolution layer, 32 feature maps with kernel size 3 3, stride size one and padding equal to the same. We used max pool operation with pool size two and padding the same for the pooling layer. This layer's output will be (256, 256, 32). For the second convolution layer, 32 feature maps with kernel size 3 3, stride size one and same padding were used. Max pooling was used with pool size two and padding same. This layer's output will be (256, 256, 32).

Additionally, we have applied batch normalisation and dropout to both of our layers. The dropout layer can help correct overfitting and batch normalisation aid to normalise the layers to speed up and stabilise artificial neural network training. The model is then connected to a flatten layer, which creates a feature vector with just one dimension. The model is then connected to a fully connected layer, and the final layer will have two neurons which is the number of classes in our dataset.

```
model = models.Sequential([
    data_augmentation,
    layers.Conv2D(32, (3,3), input_shape=(256,256,3), strides=(1, 1), activation='relu',padding='same'),
    layers.MaxPool2D(pool_size=(2, 2), padding='same'),
    layers.BatchNormalization(),
    layers.Dropout(0.25),

    layers.Conv2D(32, (3,3), activation='relu', strides=(1, 1), padding='same'),
    layers.MaxPool2D(pool_size=(2, 2), padding='same'),
    layers.BatchNormalization(),
    layers.Dropout(0.25),

    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.25),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.25),
    layers.Dense(32, activation='relu'),
    layers.Dense(number_classes, activation='softmax')
])
```

Figure 25: Model construction

Model Training

In this step, we will compile and train our model. If specific requirements are met, the network will be stopped early in the training process using the early stopping callback method we have included. This function keeps track of the validation loss value after each epoch to stop the model from overfitting. If the validation accuracy has stopped improving after five epochs, the model stops training our network.

```
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping

path = 'best_model.hdf5'
checkpoint = ModelCheckpoint(path, monitor='val_accuracy', verbose = 1, save_best_only = True, mode='max')
es = EarlyStopping(monitor='val_accuracy', patience=5)

callback_list = [checkpoint,es]
```

Figure 26: Early Stopping and checkpoints

In the following section, we will train our model. We have set 50 training epochs for the model. We used categorical cross entropy as a loss function, a common loss function for multi-class problems. We employed an optimizer known as Adam for the optimization process with learning rate 0.001. During training, this method modifies the weights of a network. Figures 27 and 28 display the model compilation and training, respectively.

```
[16] from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
    path = 'best_model.hdf5'
    checkpoint = ModelCheckpoint(path, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
    es = EarlyStopping(monitor='val_accuracy', patience=5)
    callback_list = [checkpoint, es]

[17] #model compilation
    model.compile(optimizer='Adam',
                  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                  metrics=['accuracy'])
```

Figure 27: Model compilation

Now, we will start training our model. We have passed the training data, labels, number of epochs, batch size and callback functions.

```
> Epoch 1/50
451/451 [=====] - ETA: 0s - loss: 1.3143 - accuracy: 0.4879
Epoch 1: val_accuracy improved from -inf to 0.49951, saving model to best_model_den.hdf5
451/451 [=====] - 378 61ms/step - loss: 1.3143 - accuracy: 0.4879 - val_loss: 0.6942 - val_accuracy: 0.4995
Epoch 2/50
450/451 [=====] - ETA: 0s - loss: 0.7155 - accuracy: 0.5061
Epoch 2: val_accuracy improved from 0.49951 to 0.51213, saving model to best_model_den.hdf5
451/451 [=====] - 288 61ms/step - loss: 0.7154 - accuracy: 0.5066 - val_loss: 0.6914 - val_accuracy: 0.5121
Epoch 3/50
451/451 [=====] - ETA: 0s - loss: 0.7069 - accuracy: 0.5101
Epoch 3: val_accuracy did not improve from 0.51213
451/451 [=====] - 268 57ms/step - loss: 0.7069 - accuracy: 0.5101 - val_loss: 0.6972 - val_accuracy: 0.4930
Epoch 4/50
451/451 [=====] - ETA: 0s - loss: 0.7003 - accuracy: 0.4999
Epoch 4: val_accuracy did not improve from 0.51213
451/451 [=====] - 268 57ms/step - loss: 0.7003 - accuracy: 0.4999 - val_loss: 0.6960 - val_accuracy: 0.4930
Epoch 5/50
450/451 [=====] - ETA: 0s - loss: 0.6960 - accuracy: 0.5122
Epoch 5: val_accuracy did not improve from 0.51213
451/451 [=====] - 268 58ms/step - loss: 0.6960 - accuracy: 0.5120 - val_loss: 0.6940 - val_accuracy: 0.4930
Epoch 6/50
450/451 [=====] - ETA: 0s - loss: 0.6956 - accuracy: 0.5026
Epoch 6: val_accuracy did not improve from 0.51213
451/451 [=====] - 268 58ms/step - loss: 0.6956 - accuracy: 0.5031 - val_loss: 0.6935 - val_accuracy: 0.4930
Epoch 7/50
```

Figure 28: Model Training

Finally, the training stopped at the seventh epoch because the early stopping callback criteria met.

Classification using transfer learning model

ResNet50

The convolutional neural network (CNN) variant known as ResNet (Residual Network) was first introduced in the 2015. A 50-layer convolutional neural network is called ResNet-50 (48 convolutional layers, one MaxPool layer, and one average pool layer). Residual neural networks are a type of artificial neural network (ANN) that forms networks by stacking residual blocks (“ResNet-50: The Basics and a Quick Tutorial,” n.d.).

EfficientNetV2L

For the EfficientNet model, 50 training epochs have been set. We used categorical cross entropy as a loss function, a common loss function for multi-class problems. We employed an optimizer known as Adam for the optimization process.

DenseNet201

A DenseNet is a convolutional neural network that uses dense connections between layers by connecting all layers directly with one another using Dense Blocks. Each layer receives additional inputs from all previous layers and transmits its own feature-maps to all preceding layers to maintain the feed-forward structure (Huang et al. 2016). The model accepts an input image with dimensions of 224 * 224 * 3 and contains 201 deep layers.

VGG19 (Visual Geometry Group)

VGG19 is a state-of-the-art CNN with pre-trained layers. It is very deep and has been trained on millions of images with convoluted classification methods (Bardhi 2021). It utilizes 19 layers.

Model Compilation and Training

Pretrained Model on Keras

Keras has a pre-trained library. It makes using it quicker and more practical. Here's how to load the model in one line

```
inputs = keras.Input(shape=(img_height, img_width, 3))
x = base_eff(inputs, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.5)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
```

We modified all models, such as ResNet50, DeneseNet201, EfficientNetV2L and VGG19, by including Global Average Pooling and a dense output layer. Also, for all models, we have used the “Adam” optimizer. We have set 50 training epochs for all models. We used categorical cross entropy as a loss function, a common loss function for multi-class problems.

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Figure 29: model Compilation

Model Evaluation

After training, we will evaluate the model’s performance on the validation set. To evaluate the model, I have used a classification report from sklearn.metrics, where we pass the labels and attributes. We will be able to evaluate the model’s performance for the validation set after executing it. The findings will be covered in section 7.

Results

ResNet50 Model

The model is trained for 100 epochs, but it stopped performing at the 11th epoch because the validation accuracy did not improve. The model training took 26 minutes to complete, and The figure below shows the classification report of the model.

Accuracy: 61 % Precision: 66 % Recall: 63% F1-Score: 61%

Classification Report:					
	precision	recall	f1-score	support	
0	0.58	0.86	0.69	757	
1	0.75	0.40	0.52	779	
accuracy			0.63	1536	
macro avg	0.66	0.63	0.61	1536	
weighted avg	0.67	0.63	0.61	1536	

Figure 30: Classification Report

EfficientNetV2L Model

The model is trained for 100 epochs, but it stopped performing at the 17th epoch because the validation accuracy did not improve. The model training took 35 minutes to complete, and The figure below shows the classification report of the model.

Accuracy: 61 % Precision: 61 % Recall: 61% F1-Score: 61%

Classification Report:					
	precision	recall	f1-score	support	
0	0.63	0.56	0.59	714	
1	0.59	0.67	0.63	686	
accuracy			0.61	1400	
macro avg	0.61	0.61	0.61	1400	
weighted avg	0.61	0.61	0.61	1400	

Figure 31: Classification report of EfficientNetV2L

DenseNet201 Model

The model is trained for 100 epochs, but it stopped performing at the 17th epoch because the validation accuracy did not improve. The model training took 35 minutes to complete, and The figure below shows the classification report of the model.

Accuracy: 60 % Precision: 62% Recall: 61% F1-Score: 60%

Classification Report:					
	precision	recall	f1-score	support	
0	0.58	0.76	0.66	1108	
1	0.65	0.45	0.53	1092	
accuracy			0.61	2200	
macro avg	0.62	0.61	0.60	2200	
weighted avg	0.62	0.61	0.60	2200	

Figure 32: Classification model of DeneseNet201

VGG19 Model

The model is trained for 100 epochs, but it stopped performing at the 13th epoch because the validation accuracy did not improve. The model training took 45 minutes to complete, and The figure below shows the classification report of the model.

Accuracy: 60 % Precision: 62% Recall: 61% F1-Score: 60%

Classification Report:					
	precision	recall	f1-score	support	
0	0.59	0.74	0.66	1557	
1	0.64	0.47	0.54	1515	
accuracy			0.61	3072	
macro avg	0.61	0.61	0.60	3072	
weighted avg	0.61	0.61	0.60	3072	

Figure 33: Classification report of VGG19

CNN

We used 50 epochs for training the model, and it took nearly 35 mins to complete the training process. The early stopping callback criteria were satisfied during the 13th epoch. Hence the training was terminated. The model scored 51% for accuracy, 51% for precision, 51% for recall, and 47% for f1-score on the validation set. The figure below shows the classification report of the model.

Classification Report:					
	precision	recall	f1-score	support	
0	0.53	0.25	0.34	1567	
1	0.50	0.77	0.61	1524	
accuracy			0.51	3091	
macro avg	0.51	0.51	0.47	3091	
weighted avg	0.52	0.51	0.47	3091	

Figure 34: Classification report of CNN

KNN

We attempted to execute the KNN algorithm to see the performance of the data in the machine-learning model. Unfortunately, it didn't turn out as expected. We used the default best value of 7 to K. To prevent conflicts while making decisions, K is typically taken as an odd number. KNN have the drawback that classification becomes slower as the dataset gets larger. KNN's performance is shown in the following figure.

	precision	recall	f1-score	support	
0	0.50	0.51	0.51	948	
1	0.52	0.51	0.51	972	
accuracy			0.51	1920	
macro avg	0.51	0.51	0.51	1920	
weighted avg	0.51	0.51	0.51	1920	

Figure 35: Classification report of KNN

To evaluate machine learning models on limited data, we applied a 10-fold cross-validation method. In machine learning, cross-validation is mostly used to assess how well a machine-learning model performs on unseen data. That is, to use a small sample to estimate how the model will generally perform to generate predictions on data.

```
from sklearn.model_selection import cross_val_score
score=cross_val_score(knn,x_train2,y_train,cv=10)

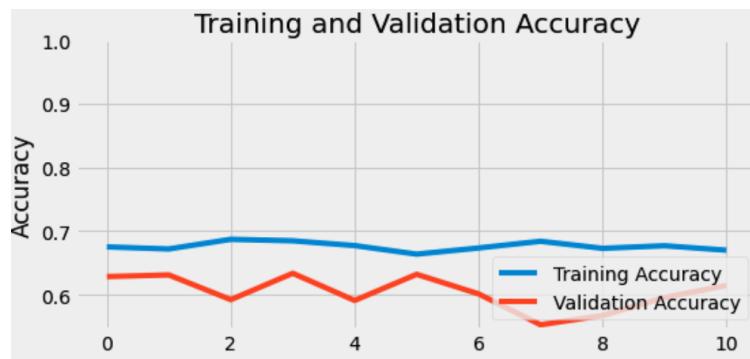
score
array([0.48883929, 0.546875 , 0.53125 , 0.56696429, 0.52901786,
       0.515625 , 0.49107143, 0.50669643, 0.47767857, 0.50892857])
```

Figure 36: 10 fold cross-validation

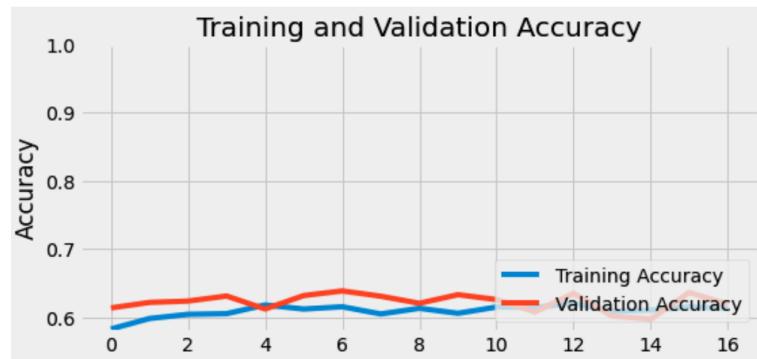
The above figure shows the accuracy in each fold. The highest score is 0.5669

Plots for the model

ResNet50

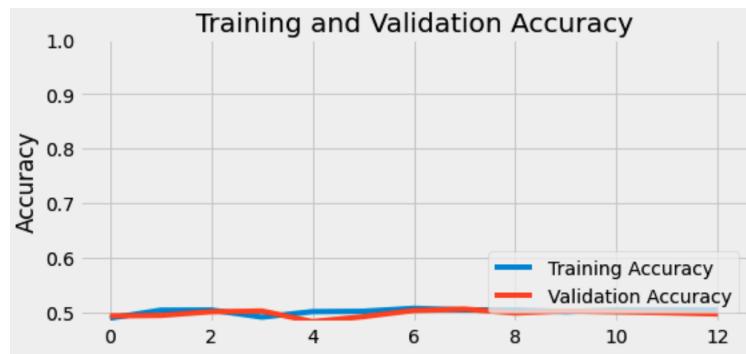


EfficientNetV2L



DenseNet201



VGG19**CNN****Fine tuning the model**

The method of fine-tuning involves taking a model that has already been trained to perform one specific task and then adjusting or modifying the model to have it execute a second comparable task. Setting base model.trainable = True will de-freeze the base model allowing for subsequent changes. We used the EfficientNet model to apply fine-tuning. The model's performance slightly increased to 62%.

Classification Report:				
	precision	recall	f1-score	support
0	0.61	0.69	0.65	1109
1	0.64	0.54	0.59	1091
accuracy			0.62	2200
macro avg	0.62	0.62	0.62	2200
weighted avg	0.62	0.62	0.62	2200

Figure 37: Fine tuning

The prediction generated by the model on random photos is depicted in the figure below.

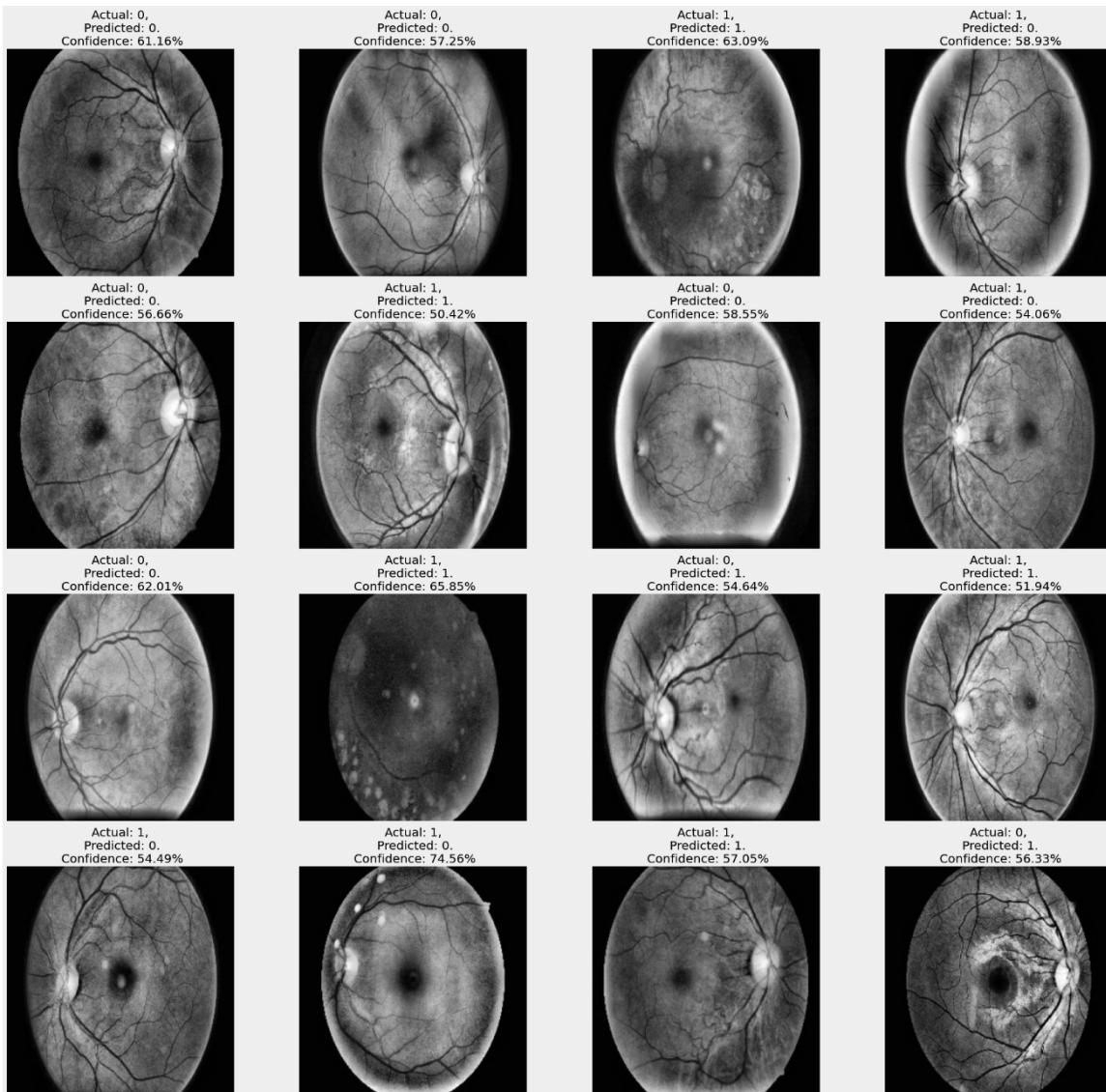


Figure 38: Prediction using ResNet50

Project Management

A project management comprises details like deadlines for various goals, whether all goals were achieved within the allotted time and budget with the desired quality, whether changes and substitute goals were made to the original goal, and how the project was organised and finished. To ensure a successful project, project management is crucial. Throughout this research project, three aspects of project management will be discussed, including Project schedule, Risk, and Quality management.

Project Schedule

For a clear understanding of how the project progresses, a work breakdown structure and timeline were developed. It is extremely important for the success of any project to have a clear timeline and a clear understanding of the work. As part of this work, I submitted an ethics application, created a project proposal, acquired knowledge of the various software programs we needed, conducted background research, analysed requirements, designed, gathered data, preprocessed the data, implemented the network, evaluated the network, and finally deployed it. All deadlines for submitting the ethical application, CW1 project proposal, CW2 project presentation, and CW3 final report submission have been met.

Risk Management

Project risk management is the process of locating, evaluating, and dealing with any risks that develop throughout the course of a project in order to keep it on track and achieve its objective. Being cautious in recognizing these risks and developing preventative actions is essential. The project's possible risks are shown in the table below, along with a preventive measure.

Table 6: Risk management

Risk	Mitigation	Risk Severity
Health Issue	Keep up a healthy sleep schedule and take medication if required.	High
Need of computing resources.	Reduce the size of the image because the dataset was so large, and try utilising Google Colab Pro and a high-performance computer at the university.	High
System failure	Regular data backups to Github and external hard drives	High

Quality Management

To produce an acceptable project outcome, quality management is another aspect of project management. Managing the quality of a project requires a careful consideration of how it will be executed in order to produce high-quality deliverables. In order to execute the project effectively, top-notch hardware and software tools were employed. As part of the project outcome, supervisors were asked to provide regular feedback to ensure the output is of good and acceptable quality. Moreover, the timelines and plan for developing the project were scheduled under the supervision of the supervisor.

Social, Legal, Ethical and Professional Considerations

Throughout the project, every social, legal, ethical, and professional concern has been taken into account, and every effort has been made to ensure that none of them has been violated. Deep learning studies

frequently raise social, legal, and ethical concerns because to the usage of datasets with sensitive data and the possibility for individual issues. The dataset used in this project was taken from the publicly accessible website and was utilised without raising any ethical or legal issues because it was intended for study and research purposes. If any information from other people's research is used in this report for any purpose, the appropriate sources are cited and given due credit.

Conclusions

Achievements

The diabetes retinopathy detection technology is studied in this work using image preprocessing technology and a deep learning algorithm and machine learning algorithm. In relation to the study's primary goal, "Can deep learning approach predict diabetic retinopathy," I think it's fair to say that this research used a rigorous approach to show that deep neural networks may predict diabetic retinopathy. This project tried to reproduce past work using the Kaggle dataset. We have referred to more than 15 research papers during the study of this project. We found only a few papers with the Kaggle Dataset. The majority of the papers use multiple datasets or different datasets. We tried to replicate the paper with the same dataset studied by Yuchen Wu et.al. They have achieved an accuracy of 60%. In data preprocessing, they applied data enhancement and contrast enhancement techniques. They employed the data enhancement method to deal with the unbalanced data. We managed the unbalanced data in my study using the data under-sampling technique.

In conclusion, To achieve the classification based on the severity of diabetic retinopathy, this paper mainly focus on CNN and transfer-learning methods and fine-tunes the dataset using Keras's built-in pre-training model. Compared to the performance of replicated work, my model performed better. The accuracy rate rises with epochs, as can be seen from the table below. Indicating that the proper number of training epochs must be increased. With more time and by utilizing a high-performance computer, we can train the model to more epochs. The performance of the ResNet 50 network is the best in this work. The outcomes demonstrate that the model performed well during training. Due to the low quality of images across categories with Severe and Proliferative DR, testing also good positive results with some misclassification. Our proposed method is quite efficient with this Kaggle dataset. Using multiple datasets or different balanced datasets, we can further improve the accuracy of the models.

Table 7: Transfer learning models comparison

Neural Networks	Epochs	Accuracy
ResNet50	27	63
EfficientNetV2L	17	61
DenseNet201	17	61
VGG19	13	61

Future Work

Several methods could not be executed due to time constraints, but this section nevertheless tackles them in the form of suggestions that can be used in future work, even if the project's results were as anticipated. First, it is suggested to use multiple datasets from a different sources that can give more successful and more stable results. Some of the images in this dataset were out of focus, and some algorithms did not successfully identify features in the DR. Therefore, in the future, the algorithm can be improved to solve the effect of the inconsistent image. Additionally, we can expand the scope of the project. We can deploy the models and build software with a good user interface to help doctors diagnose this disease.

Student Reflections

This research project has taught me a lot about machine learning, deep learning and image processing, and it has also assisted me in getting the most out of the concepts I've learned in class. It has also given me the chance to study diabetic retinopathy and its impact on this world.

This study project's secondary, yet still major, learning outcome has been Project management, especially time management. Because effective time management is a crucial skill in any corporate environment, the meetings I scheduled with the supervisor and the careful planning of the objectives through the timeline have given me project management expertise that may be very beneficial for my career.

The project's execution was the biggest challenge for me. But I was finally able to finish the project successfully and on time because of my efforts and the support I received, and I can say with certainty that I learnt a lot of hard and soft skills while working on this project.

I believe I applied good approaches in the data pre-processing and model building and am delighted with the outcome of those. If there had been more time and computing capacity to apply a few more methods, the error in final predictions across various models might have been lower. Ultimately, I am happy with how my research turned out because I was able to respond to the research questions to a certain extent and produce results.

References

- Associates, Advanced Retina. n.d. <https://www.advancedretinaassociates.com/patient-education/diabetic-retinopathy/>.
- Bardhi, Melisa. 2021. "# Image Detection Using Convolutional Neural Networks." <https://medium.com/mlearning-ai/image-detection-using-convolutional-neural-networks-89c9e21ffa3>.
- Budhiraja, Amar. 2018. "Learning Less to Learn Better — Dropout in (Deep) Machine Learning." <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bf5>.
- Davenport, Thomas, and Ravi Kalakota. 2019. "The Potential for Artificial Intelligence in Healthcare." *Future Healthcare Journal* 6 (2): 94. <https://doi.org/10.7861/futurehosp.6-2-94>.
- "Diabetic Retinopathy - Illnesses & Conditions | NHS Inform." n.d. <https://www.nhsinform.scot/illnesses-and-conditions/diabetes/diabetic-retinopathy>.
- "Diabetic Retinopathy - Stages." 2017. <https://www.nhs.uk/conditions/diabetic-retinopathy/stages/>.
- "Diabetic Retinopathy Detection." 2015. <https://kaggle.com/competitions/diabetic-retinopathy-detection>.
- "History of Artificial Neural Network - Javatpoint." n.d. <https://www.javatpoint.com/history-of-artificial-neural-network>.
- Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. "Densely Connected Convolutional Networks," August. <https://doi.org/10.48550/arXiv.1608.06993>.
- IndoML.com. 2018. <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/>.
- Jayakumari, C., Vidhya Lavanya, and E P Sumesh. 2020. "Automated Diabetic Retinopathy Detection and Classification Using ImageNet Convolution Neural Network Using Fundus Images." In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 577–82. <https://doi.org/10.1109/ICOSEC49089.2020.9215270>.
- Kaya, Esra, and Ismail Saritas. 2022. "Performances of CNN Architectures on Diabetic Retinopathy Detection Using Transfer Learning." In *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, 1–4. <https://doi.org/10.1109/ICEST55168.2022.9828576>.
- Kongsilp, Panadda. 2019. "CNN: Step 2 — Flattening." https://medium.com/@PK_KwanG/cnn-step-2-flattening-50ee0af42e3e.
- Kumar, Ajitesh. 2022. "Different Types of CNN Architectures Explained: Examples." <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>.
- Kumar, Nikhil Sathya, and B. Ramaswamy Karthikeyan. 2021. "Diabetic Retinopathy Detection Using CNN, Transformer and MLP Based Architectures." In *2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 1–2. <https://doi.org/10.1109/ISPACS51563.2021.9651024>.
- otexts.com. n.d. <https://otexts.com/fpp2/nnetar.html>.
- Pathak, Ketki C., Riddhi B. Shah, Reshma R. Tharakan, Bhavya N. Patel, and Dhruvi C. Jariwala. 2021. "Diabetic Retinopathy Diagnosis and Categorization Using Deep Learning - a Review." In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1063–69. <https://doi.org/10.1109/ICICCS51141.2021.9432312>.
- Pokhrel, Sabina. 2019. "Beginners Guide to Understanding Convolutional Neural Networks." <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d>.
- Pratt, Harry, Frans Coenen, Deborah M Broadbent, Simon P Harding, and Yalin Zheng. 2016. "Convolutional Neural Networks for Diabetic Retinopathy." *Procedia Computer Science* 90: 200–205.
- R, Lee, Wong Ty, and Sabanayagam C. 2015. "Epidemiology of Diabetic Retinopathy, Diabetic Macular Edema and Related Vision Loss." *Eye and Vision (London, England)* 2 (September). <https://doi.org/10.1186/s40662-015-0026-2>.
- "ResNet-50: The Basics and a Quick Tutorial." n.d. <https://datagen.tech/guides/computer-vision/resnet-50/>.
- Reynolds, Anh H. n.d. <https://anhreynolds.com/>.
- Senaratne, Ravindu. 2020. "CLAHE and Thresholding in Python." <https://towardsdatascience.com/clah-and-thresholding-in-python>.

- and-thresholding-in-python-3bf690303e40.
- Shahid, Nida, Tim Rappon, and Whitney Berta. 2019. “Applications of Artificial Neural Networks in Health Care Organizational Decision-Making: A Scoping Review.” *PLoS ONE* 14 (2). <https://doi.org/10.1371/journal.pone.0212356>.
- Sharma, Pranshu. 2021. “Transfer Learning | Understanding Transfer Learning for Deep Learning.” <https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/>.
- “Synopsys.” 2017. <https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/>.
- Tripathi, Mohit. 2021. “Image Processing Using CNN | Beginner’s Guide to Image Processing.” <https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/>.
- Turing Enterprises Inc. n.d. “Transfer Learning Using CNN (Vgg16).” <https://www.turing.com/kb/transfer-learning-using-cnn-vgg16>.
- V, Keerthana. 2021. “Artificial Neural Network | How Does Artificial Neural Network Work.” <https://www.analyticsvidhya.com/blog/2021/04/artificial-neural-network-its-inspiration-and-the-working-mechanism/>.
- “What Is Agile Methodology?” 2022. <https://www.redhat.com/en/topics/devops/what-is-agile-methodology>.
- “What Is the k-Nearest Neighbors Algorithm? | IBM.” n.d. <https://www.ibm.com/uk-en/topics/knn>.
- Whiting, David R, Leonor Guariguata, Clara Weil, and Jonathan Shaw. 2011. “IDF Diabetes Atlas: Global Estimates of the Prevalence of Diabetes for 2011 and 2030.” *Diabetes Research and Clinical Practice* 94 (3): 311–21.
- Wu, Yuchen, and Ze Hu. 2019. “Recognition of Diabetic Retinopathy Basedon Transfer Learning.” In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 398–401. <https://doi.org/10.1109/ICCCBDA.2019.8725801>.

Appendix A – Interim Progress Report and Meeting Records

Meeting No	Discussed Topic	Date
1	<ul style="list-style-type: none"> First meeting with Dr Rochelle Sassman via MS Teams. She introduced the dissertation fundamentals and project timelines. 	15/08/2022
2	<ul style="list-style-type: none"> First meeting with supervisor Prof. James Brusey. Discussion about forming a Microsoft Teams group to hold regular meetings. Professor advised me to read various articles and find examples of dissertation-related documents in order to become familiar with and comprehend the project's topic. 	27/10/2022
3	<ul style="list-style-type: none"> Ethics submitted and approved 	07/10/2022
4	<ul style="list-style-type: none"> Physical meeting with the professor regarding clarification of the project's proposal. Professor advised me to create a GitHub repository and add him as a collaborator. Professor shares a GitHub repository that includes the necessary guidelines for the project. 	20/10/2022
5	<ul style="list-style-type: none"> Shared initial coding via GitHub Submitted initial report using Rmarkdown , and supervisor gave helpful feedback. 	08/10/2022
6	<ul style="list-style-type: none"> Discussion with supervisor 	15/11/2022
7	<ul style="list-style-type: none"> Submitted Interim report to supevisor Supervisor advised me to try cross-validation and take out data-augmentation 	23/11/2022
6	<ul style="list-style-type: none"> Conducted a meeting Via MS teams to discuss the project's presentation. 	5/12/2022

Appendix B – Project Presentation

COVENTRY UNIVERSITY

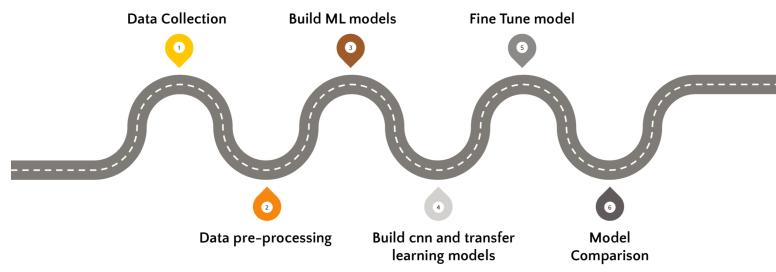
Diabetic Retinopathy Detection



Presented By: Aleena Alby
1st Supervisor : Prof. James Brusey
2nd Supervisor : Dr. Fei He
Date : 6 Dec 2022

COVENTRY UNIVERSITY

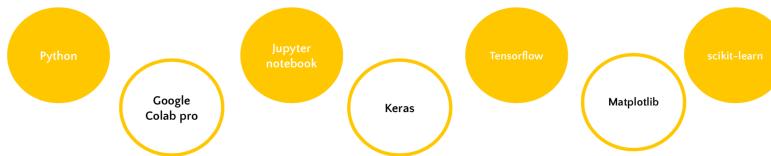
1 Project Overview



2

COVENTRY UNIVERSITY

2 Major Platforms & Libraries



3

4 Objective and scope

- To develop various deep learning models for the detection of diabetic retinopathy and then evaluate them using evaluation metrics such as accuracy, recall, f1-score, and precision.
- Also, I'm aiming to reproduce some earlier relevant work. Putting special emphasis on improving prediction performance.

Deliverables: A deep learning system that respects the privacy of the patient and can detect the diabetic retinopathy (DR)

4

3 Related work

2019

Recognition of Diabetic Retinopathy Based on Transfer Learning [1]

- They used the models VGG19, InceptionV3, Resnet50.
- Dataset : Kaggle dataset
- They performed data enhancement, include data amplification, flipping, folding, and contrast adjustment.
- Their classification accuracy of this method was 0.60.

2021

Diabetic Retinopathy Diagnosis and Categorization Using Deep Learning - a Review ([2] Pathak et al.)

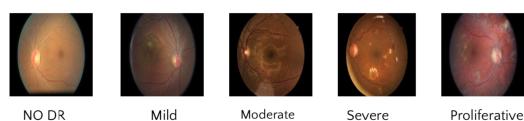
- Using a SVM method, CNN, DCNN, and ANN classifiers
- Dataset: IDRiD (Indian dataset), High-Resolution Fundus (HRF) Image Database and Kaggle dataset
- They summarised that DCNN technique is very productive and effective in terms of accuracy.

5

5 About Data

- Dataset:** Retinal images provided by EyePACS downloaded from Kaggle.
- DR images in the dataset are high-resolution RGB images.
- Images are divided according to the degree of lesions to 5 Categories.

IMAGE CLASSES	LEVEL
No DR	0
Mild	1
Moderate	2
Severe	3
Proliferative	4



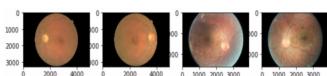
6

6 Methodology

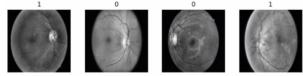
1. DATA COLLECTION : Dataset contain 35,126 set of images with 5 classes.



2. IMAGE PRE-PROCESSING: Images resized to 256 * 256 pixels.
Images with 40% black and white pixels removed.
To improve the contrast of the image CLAHE method is used.
The dataset was imbalanced, so a under-sampling strategy is implemented.



Before CLAHE

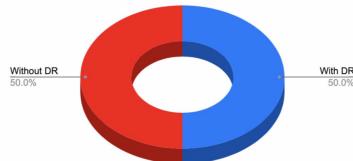


After CLAHE

7

6 Data distribution after image preprocessing

Type of Diabetics Diagnosis	No. of Images
With DR	5152
Without DR	5152
Total	10304



8

6 Methodology - continue..

3. BUILD ML MODEL: Models used here is KNN



4. BUILD DL MODELS: Models used are CNN, DenseNet201 VGG19, EfficientNetV2L, ResNet50



5. MODELS EVALUATION: Analyze models performances using evaluation metrics

9

K nearest neighbour

COVENTRY UNIVERSITY

7 Models Evaluation

- ❑ K nearest neighbour is one of the simplest supervised machine learning algorithm for classification and regression problems.
- ❑ KNN model performances shown in the figure.
- ❑ Model performed with only 51% accuracy.

```
: accuracy_score(y_pred_knn,y_test)
print(classification_report(y_pred_knn,y_test))

precision    recall   f1-score   support
          0       0.50      0.51      0.51     948
          1       0.52      0.51      0.51     972

   accuracy                           0.51
  macro avg                           0.51
weighted avg                          0.51
```

KNN model classification report

```
from sklearn.model_selection import cross_val_score
score=cross_val_score(knn,x_train2,y_train, cv=10)

score
array([0.48693929, 0.546675 , 0.53125 , 0.56696429, 0.52901786,
       0.515625 , 0.49107143, 0.50669643, 0.47767857, 0.50892857])
```

Cross validation score



10

CNN and Transfer Learning

COVENTRY UNIVERSITY

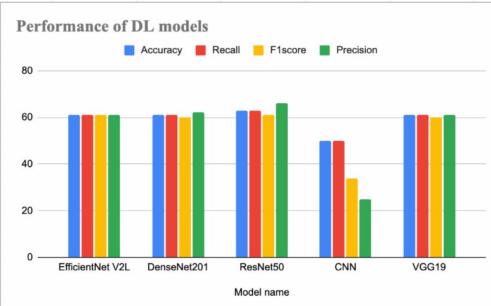
7 Models Evaluation continue..

- ❑ The performance of the CNN model and the Transfer Learning model is displayed in the figure.

```
44/44 [=====] - 16s 361ms/step
Classification Report:
precision    recall   f1-score   support
          0       0.63      0.56      0.59     714
          1       0.59      0.67      0.63     666

   accuracy                           0.61
  macro avg                           0.61
weighted avg                          0.61
```

Classification report of EfficientNet

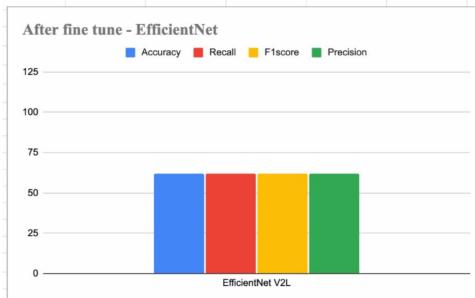


11

COVENTRY UNIVERSITY

7 Models Evaluation continue..

- ❑ EfficientNet model was chosen to Fine-tune to improve the accuracy.
- ❑ Here the model accuracy improved to 62%.



12

8 Limitation and future scope

Limitation

- Limitation of this work is that the dataset was an imbalanced dataset.

Future Scope

- This study has shown that boosting image quantity increases accuracy. It is recommended that more study be done to look into alternative image enhancement features.
- More data preprocessing steps could be implemented.
- Other model could also be explored.

13

9 Conclusion

- Experimental findings demonstrate that the model was overfitting on the training data we can diminish this problem by better dataset.
- After fine tune efficientNet the model accuracy didn't improve much.
- With a recall score of 0.63, the Resnet50 performed better.

14

10

References

[1] Wu, Yuchen, and Ze Hu. 2019. "Recognition of Diabetic Retinopathy Basedon Transfer Learning." In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 398–401. <https://doi.org/10.1109/ICCCBDA.2019.8725801>.

[2] Pathak, Ketki C., Riddhi B. Shah, Reshma R. Tharakan, Bhavya N. Patel, and Dhruvi C. Jariwala. 2021. "Diabetic Retinopathy Diagnosis and Categorization Using Deep Learning - a Review." In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1063–69. <Https://doi.org/10.1109/ICICCS51141.2021.9432312>.

15



Thanks!

16

Appendix F – Certificate of Ethics Approval

Diabetic Retinopathy prediction using deep learning

P142645



Certificate of Ethical Approval

Applicant: Aleena Alby

Project Title: Diabetic Retinopathy prediction using deep learning

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval: 07 Oct 2022

Project Reference Number: P142645

Page 1 of 1

Aleena Alby (7151CEM)

Page 1

07 Oct 2022