Department of Electrical and Computer Engineering
North South University

Senior Design Project

Enhancing Ocular Disease Diagnosis in Fundus Images with CNN Models and Deep Learning Approaches

Md. Marop Hossain	**ID# 2013982042**
Md. Zunayed Islam Pranto	**ID# 1921609642**
Tasnia Tasnim	**ID# 1912533642**

Faculty Advisor:

Dr. Mohammad Ashrafuzzaman Khan

Assistant Professor

ECE Department

SUMMER, 2023

LETTER OF TRANSMITTAL

June, 2023

To

Dr. Rajesh Palit
Chairman,
Department of Electrical and Computer Engineering
North South University, Dhaka

Subject: **Submission of Capstone Project Report on "Enhancing Ocular Disease Diagnosis in Fundus Images with CNN Models and Deep Learning Approaches"**

Dear Sir,

With due respect, we would like to submit our **Capstone Project Report** on **"Enhancing Ocular Disease Diagnosis in Fundus Images with CNN Models and Deep Learning approaches"** as a part of our BSc program. The project deals with developing a system using deep learning through CNN that will use the retinal fundus image to identify, extract, and evaluate disease-specific characteristics. The system will help to early detect the diseases, which allows patients to maintain a good quality of vision while avoiding serious vision loss and blindness.

We will be highly obliged if you kindly receive this report and provide your valuable judgment. It would be our immense pleasure if you find this report useful and informative to have an apparent perspective on the issue.

Sincerely Yours,

....

Md. Marop Hossain
ECE Department
North South University, Bangladesh

....

Md. Zunayed Islam Pranto
ECE Department
North South University, Bangladesh

....

Tasnia Tasnim
ECE Department
North South University, Bangladesh

APPROVAL

Md. Marop Hossain (ID # 2013982042), **Md. Zunayed Islam Pranto** (ID # 1921609642) and **Tasnia Tasnim** (ID # 1912533642) from Electrical and Computer Engineering Department of North South University, have worked on the Senior Design Project titled "**Enhancing Ocular Disease Diagnosis in Fundus Images with CNN Models and Deep Learning approaches**" under the supervision of **Dr. Mohammad Ashrafuzzaman Khan** partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

Supervisor's Signature

……………………………………….

Dr. Mohammad Ashrafuzzaman Khan

Assistant Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman's Signature

……………………………………….

Dr. Rajesh Palit

Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

This is to declare that this project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All project related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been properly acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students’ names & Signatures

1. Md. Marop Hossain

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

2. Zunayed Islam Pranto

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

3. Tasnia Tasnim

ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, **Dr. Mohammad Ashrafuzzaman Khan**, Assistant Professor, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance, and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

ABSTRACT

Enhancing Ocular Disease Diagnosis in Fundus Images with CNN Models and Deep Learning Approaches

Artificial intelligence based on deep learning is having an important effect on various sectors of healthcare, Ophthalmology is not an exception. Different deep-learning methods have been successfully applied in Ophthalmology to detect various kinds of eye diseases. In ophthalmology, deep learning methods have primarily been applied to eye fundus images and optical coherence tomography. These methods have achieved outstanding performance in the detection of ocular diseases such as diabetic retinopathy, glaucoma, diabetic macular degeneration, cataracts, and age-related macular degeneration. The impact and significance of these classifications lie in the potential to revolutionize ocular disease diagnosis, making it more accessible, accurate, and efficient. Also, it will potentially change the view of how ophthalmology is practiced in the future.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Background and Motivation

The eye is a vital organ for a human being that plays a critical role compared to other organs of the human body. Eye diseases can cause vision loss or blindness, which can have a profound impact on a person's quality of life. There are a lot of eye diseases, such as diabetes, glaucoma, cataracts, age-related macular degeneration, hypertension, pathological myopia, and abnormalities. Globally, 600 million people will have diabetes by 2040, with a third having DR [1]. It is projected that 288 million patients may have some forms of AMD by 2040 [2]. The global prevalence of glaucoma for people aged 40–80 is 3.4%, and by the year 2040 it is projected there will be approximately 112 million affected individuals worldwide[3]. To prevent such eye diseases, early detection and timely treatment can help reduce the risk of great loss of vision. Most eye care institutions are not well off in many underdeveloped nations. Furthermore, reliable medical treatment and ophthalmologists are scarce in rural areas. So, it becomes quite tough for the people of rural areas to carry the expenses of better treatment. As the population is increasing, the number of patients with eye diseases is also rapidly increasing. So, it's a community's or government's obligation to improve eye care facilities for its citizens. A system that can detect eye diseases from retinal fundus images can be developed using digital image processing and machine learning. The system will take the retinal fundus images as input. Then, from the fundus images, the system can extract and simplify the features of specific eye diseases.

1.2 Purpose and Goal of the Project

The purpose of this project is to develop a system using deep learning through CNN that will use the retinal fundus image to identify, extract, and evaluate disease-specific characteristics. This system will help to early detect eye diseases, which allows patients to maintain a good quality of vision while avoiding serious vision loss and blindness at an early age.

To achieve the goal required objectives taken:

- Determine a deep learning model that can accurately identify, extract, and evaluate disease-specific characteristics from retinal fundus images.
- Create a system that is easy to use and accessible to healthcare professionals, so that they can quickly and accurately diagnose eye diseases.
- Help to improve the early detection and treatment of eye diseases, which can lead to better patient outcomes. And early detection of disease.

1.3 Organization of the Report

Chapter 1 defines the project and its background, motivation, goal and purpose of the project.

Chapter 2 presents the literature reviews related to this project.

Chapter 3 presents the dataset, system designing, methodologies, as well as the hardware and software components implementations.

Chapter 4 presents the Investigation/Experiment, Result, Analysis and Discussion of various models.

Chapter 5 discusses the impact on societal, health, safety, legal and cultural issues as well as the Impact on environment and sustainability.

Chapter 6 projects the planning and budget.

Chapter 7 discusses a short summary, limitations and scope of future improvement to the project.

Chapter 2 Research Literature Review

2.1 Existing Research and Limitations

Deep learning (DL) techniques have the potential to revolutionize the field of ophthalmology, particularly in the area of early ocular disease detection. DL research in ophthalmology has progressed rapidly, and has the potential to become a part of daily clinical practice in the relatively near future [4]. Both papers reviewed, **"Deep learning in ophthalmology: a review"**[5] and **"Artificial intelligence and deep learning in ophthalmology"**[6], provide comprehensive overviews of the challenges in early ocular disease detection and the potential of DL techniques to address these challenges.

Both papers discuss the use of convolutional neural networks (CNNs) and transfer learning models for the development of DL systems for ophthalmology. CNNs are a type of deep learning model that are particularly well-suited for image analysis tasks. Transfer learning is a technique where a pre-trained CNN model is used as a starting point for a new model, which is then trained on a new dataset. This can be a very effective way to develop DL models for specialized tasks, such as ophthalmology, where the amount of available data may be limited.

Both papers also review the state-of-the-art DL systems for the detection of specific ocular diseases, such as diabetic retinopathy, age-related macular degeneration, glaucoma, and retinopathy of prematurity. These studies have shown that DL systems can achieve levels of accuracy that are comparable to human experts, and they can be used to screen large populations for eye diseases, monitor the progression of disease, and guide treatment decisions.

Overall, both papers discuss that developing general-purpose DL systems has the potential to make a significant impact on the early detection and treatment of eye diseases in economically disadvantaged regions. By using deep learning techniques, we can help improve access to quality eye care for everyone.

Chapter 3 Methodology

3.1 System Design

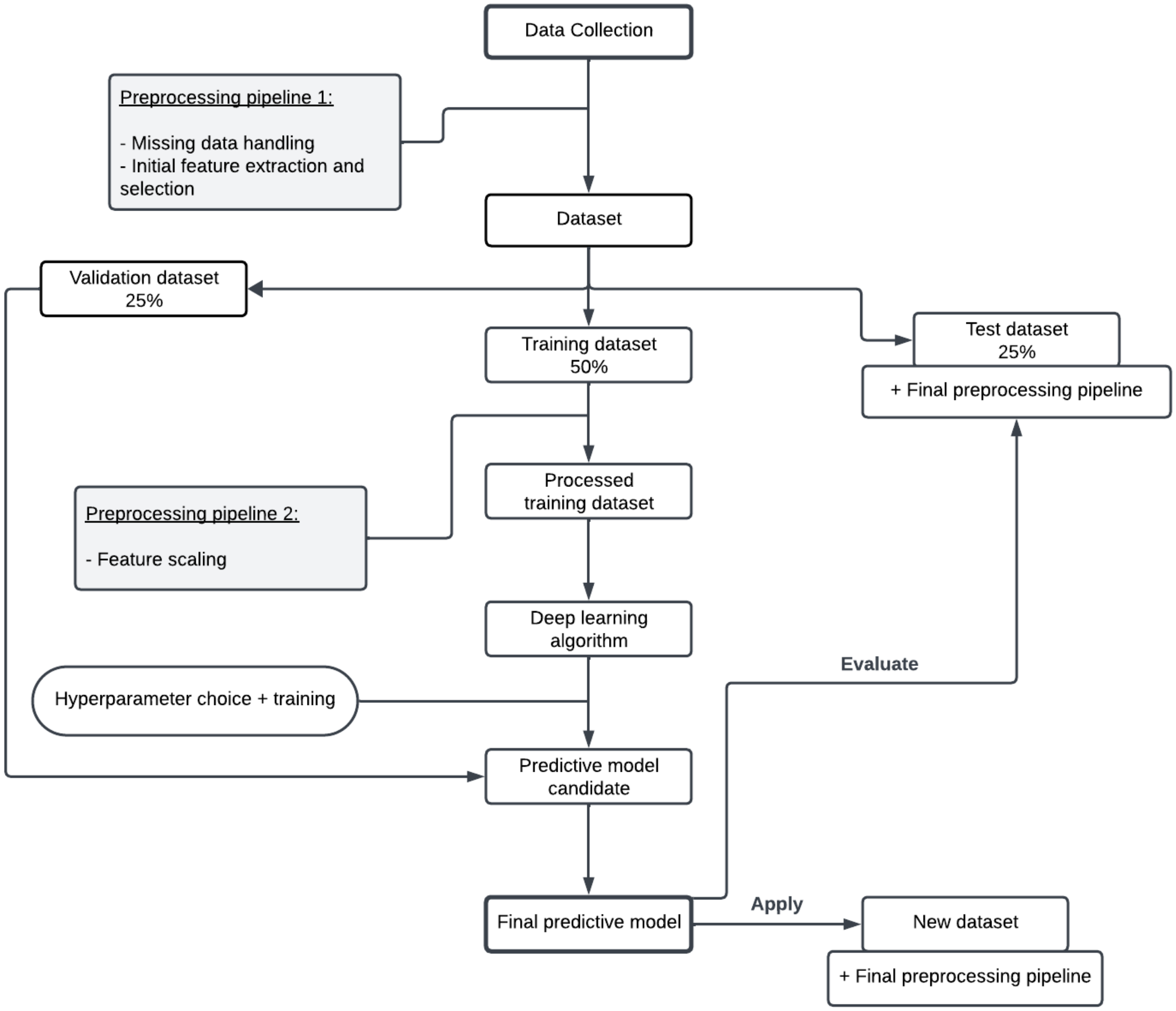

Data Collection
Preprocessing pipeline 1:
- Missing data handling
- Initial feature extraction and selection
Dataset
Validation dataset 25%
Training dataset 50%
Test dataset 25%
+ Final preprocessing pipeline
Processed training dataset
Preprocessing pipeline 2:
- Feature scaling
Deep learning algorithm
Hyperparameter choice + training
Evaluate
Predictive model candidate
Final predictive model
Apply
New dataset
+ Final preprocessing pipeline

Figure 3.1: System Design Diagram.

3.1.1 Data collection

We collected a large dataset of Ocular Disease Intelligent Recognition (ODIR) is a structured ophthalmic database of **5,000 patients** with age, color fundus photographs from left and right eyes and doctors' diagnostic keywords from doctors.

This dataset is meant to represent a ‘‘real-life’’ set of patient information collected by Shanggong Medical Technology Co., Ltd. from different hospitals/medical centers in China. In these institutions, fundus images are captured by various cameras in the market, such as Canon, Zeiss and Kowa, resulting in varied image resolutions. Annotations were labeled by trained human readers with quality control management. They classify patient into eight labels including:

- **Normal (N),**
- **Diabetes (D),**
- **Glaucoma (G),**
- **Cataract (C),**
- **Age related Macular Degeneration (A),**
- **Hypertension (H),**
- **Pathological Myopia (M),**
- **Other diseases/abnormalities (O)**

In this project, we have mainly conducted research on the three common types of disease, such as **diabetes retinopathy, glaucoma,** and **cataract.**

CLASSES	IMAGE NO
Cataract	**1038**
Diabetic retinopathy	**1098**
Glaucoma	**1007**
Normal	**1074**
Total image count	**4217**

Table 3.1.1: Image Distribution.

3.1.2 Preprocessing pipeline 1

The next step was to preprocess the data. This involved cleaning the data, removing any missing data, and performing initial feature extraction and selection. The goal of preprocessing was to prepare the data for training the neural network model, and we used image hashing to identify duplicate images. After finding duplicate images, we removed them from the dataset.

3.1.3 Dataset Splitting

I split the dataset into three subsets: training, validation, and test sets. The training set (50%) was used to train the model, a validation set (25%) was used to monitor the model's performance during training, and a test set (25%) was used to evaluate the model's performance on unseen data.

DIVIDED SET	IMAGE NO
Training set (50%)	**2108**
Validation set (25%)	**1052**
Testing set (25%)	**1057**
Total image count	**4217**

Table 3.1.3: Dataset Distribution.

3.1.4 Preprocessing pipeline 2

We preprocessed the training set once again using a second preprocessing pipeline. This pipeline involved feature scaling, which was a technique that normalized the features to have a common scale. This helped the neural network model learn more effectively.

3.1.5 Deep learning algorithm

The next step was to choose a deep learning algorithm to train the model. A variety of deep learning algorithms could be used for image classification, but convolutional neural networks (CNNs) were particularly well-suited for this task. CNNs are a type of neural network that are able to learn spatial features from images.

We used CNN and four deep learning-based models for targeted ocular disease diagnosis. For this project, we trained cutting-edge classification algorithms such as **EfficientNetV2S**, **DenseNet-121**, **ResNet-50**, and **ResNeXt-50** on the ODIR dataset consisting of **4217** fundus images that belonged to **4 different classes** such as **normal**, **cataract**, **glaucoma**, and **diabetic retinopathy**. Each of these classes represented a different ocular disease.

Our CNN model architecture involved convolutional layers, activation functions (ReLU and softmax), max-pooling layers, dropout layers, hidden fully connected layers, and the output layer. The model also used the Adam and RMSProp optimizers and a categorical cross-entropy loss function. Hyperparameters such as batch size, learning rate, and the number of epochs were also used.

3.1.6 Hyperparameter choice and training

After we had chosen a deep learning algorithm, the next step was to select the hyperparameters for the model. Hyperparameters were parameters that controlled the training process, such as the learning rate and the number of epochs. We chose the hyperparameters using a process called grid search. Once the hyperparameters had been chosen, we trained the model on the training set.

Hyperparameter	EfficientNetV2S	ResNet50	DenseNet121	ResNeXt50
Data Augmentation	RandomFlip(horizontal), RandomRotation(0.1), RandomContrast(0.1)	RandomFlip(horizontal), RandomRotation(0.1), RandomContrast(0.1),	RandomFlip(horizontal), RandomRotation(0.1), RandomContrast(0.1), RandomCrop(height=160,width=160)	RandomFlip(horizontal), RandomRotation(0.1), RandomContrast(0.1), RandomCrop(height=160,width=160), RandomZoom(0.2)

Input shape	(160, 160, 3)	(160, 160, 3)	(160, 160, 3)	(160, 160, 3)
Base Model	EfficientNetV2S	ResNet50(pre-trained)	DenseNet121(pre-trained)	ResNeXt50(pre-trained)
Base Model Trainable	True	True	True	True
Re-scaling	1./255	1./255	1./255	1./255
Batch Normalization	Yes	Yes	Yes	Yes
Dense Layer Units	256	256	256	256
Dense Layer Activation	ReLU	ReLU	ReLU	ReLU
Kernel Regularizer	L2(0.016)	L2(0.001)	L2(0.001)	L2(0.001)
Bias Regularizer	L1(0.006)	L1(0.001)	L1(0.001)	L1(0.001)
Activity Regularizer	L1(0.006)	L1(0.001)	L1(0.001)	L1(0.001)
Dropout Rate	40%	50%	40%	50%
Optimizer	Adamax(LR=0.001)	Adam(LR=0.0001)	Adam(LR=0.00001)	Adam(LR=0.0001)
Loss Function	Categorical Cross-Entropy	Categorical Cross-Entropy	Categorical Cross-Entropy	Categorical Cross-Entropy
Number of Epochs	200	200	300	250
Early Stop Epochs (before overfitting)	88	82	131	41
Best Epoch (model return after complete execution)	78	72	121	31
Model Checkpoint	yes	yes	yes	yes

Early Stopping	monitor=’val_loss’,
	mode=’min’,
	verbose=1,
	patience=10,
	min_delta=0.001
		monitor=’val_loss’
		mode=’min’,
		verbose=1,
		patience=10,
		min_delta=0.001
			monitor=’val_loss’,
			mode=’min’,
			verbose=1,
			patience=10,
			min_delta=0.001
				monitor=’val_loss’,
				mode=’min’,
				verbose=1,
				patience=10,
				min_delta=0.001

Learning Rate	ReduceLRonPlatea	ReduceLRonPlate	ReduceLRonPl	ReduceLRonPlat

Scheduler	u(monitor=’val_loss,
	factor=0.2,
	patience=3,
	verbose=1,
	mode=’min’
	min_lr=0.00001)
		au(monitor=’val_loss,
		factor=0.2,
		patience=3,
		verbose=1,
		mode=’min’
		min_lr=0.00001)
			ateau(monitor=’val_loss,
			factor=0.2,
			patience=5,
			verbose=1,
			mode=’min’
			min_lr=0.00001)
				eau(monitor=’val_loss,
				factor=0.2,
				patience=3,
				verbose=5,
				mode=’min’
				min_lr=0.00001)

Learning Rate	initial : 0.001	initial : 0.0001	initial : 0.00001	initial : 0.0001
CSV Logger	yes	yes	yes	yes

Table 3.1.6: Different types of Hyperparameter of all models.

3.1.7 Evaluation

We evaluated the model on the validation set once it had been trained. This was done to assess the performance of the model and identify any areas where it could be improved. The model was then tuned by adjusting the hyperparameters and retraining the model. This process was repeated until the model was performing well on the validation set.

3.1.8 Predictive model candidate

In this case, we used different types of hyperparameters, a training set, and a validation set of all the models, as we will calculate each model’s best performance on the validation set and training set. We also calculated the validation and training losses, precision, recall, and f1-score. In these steps, we also knew that the current model is either face overfitting, underfitting, or balance fit or not.

3.1.9 Final Predictive Model

The model that performs best on the validation set and testing will be selected as the predictive model candidate.

In this step, we tested all the models on a dataset of 1057 images, as we could calculate the model performance and the number of misclassified test images in the testing set. If we get better accuracy and a lower number of misclassified images in the testing set, we will choose the best model for the tasks where it is important to correctly classify all of the images and minimize the number of errors. It is also important to note that the prediction accuracy of the test data is not the only factor to consider when choosing the best model. Other factors, such as the model's training time and inference speed, may also have been important.

3.2 Hardware and/or Software Components

3.2.1 Software Components:

- Google colab pro
- kaggle
- python

3.2.2 Hardware Components:

- **Hardware accelerator:** V100 GPU, A100 GPU, T4 GPU, CPU, TPU

System configuration	Hardware accelerator
51.0 GB RAM, 166.8 GB storage space	A100 GPU
32.0 GB RAM, 128.0 GB storage space	V100 GPU
16.0 GB RAM, 64.0 GB storage space	T4 GPU
8.0 GB RAM, 32.0 GB storage space	CPU

- **RAM:** The system had 51.0 GB of RAM.

- **Disk:** The system had 166.8 GB of storage space.

Other Software and Hardware Components:

- **Python:** Python was used as the programming language for the project.
- **TensorFlow:** TensorFlow was used as the deep learning framework.
- **Keras:** Keras was used as a high-level API for TensorFlow.
- **CUDA:** CUDA was used to accelerate deep learning computations.

Tool	Functions	Other similar Tools (if any)	Why selected this tool
Python	Programming language	TensorFlow, R, NumPy, Pandas	Python is a versatile programming language that is well-suited for deep learning. It is also relatively easy to learn and use.
TensorFlow	Deep learning framework	PyTorch, JAX, MXNet	TensorFlow is a popular deep learning framework that is known for its ease of use and flexibility.
Keras	High-level API for TensorFlow	PyTorch Lightning, Gluon	Keras is a high-level API for TensorFlow that makes it easier to build and train deep learning models.
GPU	Accelerates deep learning computations	CPU, TPU	GPUs are specialized processors that are well-suited for parallel computing tasks, such as deep learning.
Kaggle	Online data science and machine learning platform	GitHub, TopCoder, HackerRank	Kaggle provides access to a large dataset of labeled retinal fundus images, which is essential for training a deep learning model. It also provides a community of data scientists and machine learning experts who can help with the project.
Google Colab Pro	Cloud-based Jupyter Notebook environment	Google Cloud Platform, AWS SageMaker, Azure	Google Colab Pro provides access to powerful GPUs and TPUs, which are essential for training deep learning models. It also provides a convenient

		Machine Learning Studio	way to share and collaborate on projects.

Table 3.2: List of Hardware/Software tools

3.3 Hardware and Software Implementation

3.3.1 Hardware Implementation:

The hardware requirements for ocular disease recognition using deep learning vary depending on the size and complexity of the dataset, the model architecture, and the desired inference speed.

For small datasets and simple model architectures, a CPU may be sufficient. However, for larger datasets and more complex model architectures, a hardware accelerator such as a GPU or TPU is recommended.

Here are some specific examples of hardware that can be used for ocular disease recognition:

- **V100 GPU:** The NVIDIA V100 GPU is a high-performance GPU that is well-suited for deep learning applications. It has 5120 CUDA cores and 32GB of HBM2 memory.
- **A100 GPU:** The NVIDIA A100 GPU is the successor to the V100 GPU. It has 6912 CUDA cores and 40GB of HBM2e memory.
- **T4 GPU:** The NVIDIA T4 GPU is a more affordable GPU that is still well-suited for deep learning applications. It has 2560 CUDA cores and 16GB of GDDR6 memory.
- **TPU:** A TPU is a specialized machine learning accelerator that is designed for training and deploying deep learning models. TPUs can offer significant performance advantages over GPUs for deep learning applications.

3.3.2 Software Implementation

The following software components can be used to implement ocular disease recognition using deep learning:

- **Google Colab Pro:** Google Colab Pro is a cloud-based Jupyter Notebook service that provides access to GPUs and TPUs. This makes it a convenient platform for developing and deploying deep learning models.
- **Kaggle:** Kaggle is a data science platform that hosts a variety of machine learning competitions, including competitions on ocular disease recognition. Kaggle can be a good resource for finding datasets and pre-trained models that can be used for ocular disease recognition.
- **Python:** Python is a popular programming language for data science and machine learning. Python libraries such as NumPy, Pandas, and scikit-learn can be used for data manipulation and preprocessing.
- **TensorFlow:** TensorFlow is a popular deep learning framework. It provides a variety of tools and libraries for building and training deep learning models.
- **Keras:** Keras is a high-level API for TensorFlow. It makes it easier to build and train deep learning models.
- **CUDA:** CUDA is a parallel computing platform and programming model developed by NVIDIA. CUDA can be used to accelerate deep learning computations on GPUs.

Why use Google Colab Pro and Kaggle?

Google Colab Pro and **Kaggl**e are both cloud-based platforms that provide access to powerful computing resources. Google Colab Pro provides access to GPUs and TPUs, while Kaggle provides access to a large dataset of labeled retinal fundus images. Both platforms are also free to use, which makes them a good option for students and researchers.

Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

4.1 Investigation/Experiment:

We investigated the performance of four deep learning models, **EfficientNetV2S, ResNet50, DenseNet121** and **ResNext50**, on an ocular disease project. We trained all models with data augmentation and evaluated their performance on a held-out validation set.

4.2 Result:

Model	**Training**						
	Best Epoch	**AUC**	**Precision**	**Recall**	**F1-score**	**categorical accuracy**	**Loss**
EfficientNetV2S	78/200	1.00	**1.00**	**0.99**	**0.99**	**1.00**	**0.18**
ResNet50	73/200	1.00	0.99	0.99	0.99	0.99	0.34
DenseNet121	121/300	0.99	0.99	0.99	0.99	0.99	0.47
ResNeXt50	31/250	0.99	0.95	0.92	0.93	0.94	0.78

Table 4.2 (a): Performance accuracy during training of four models

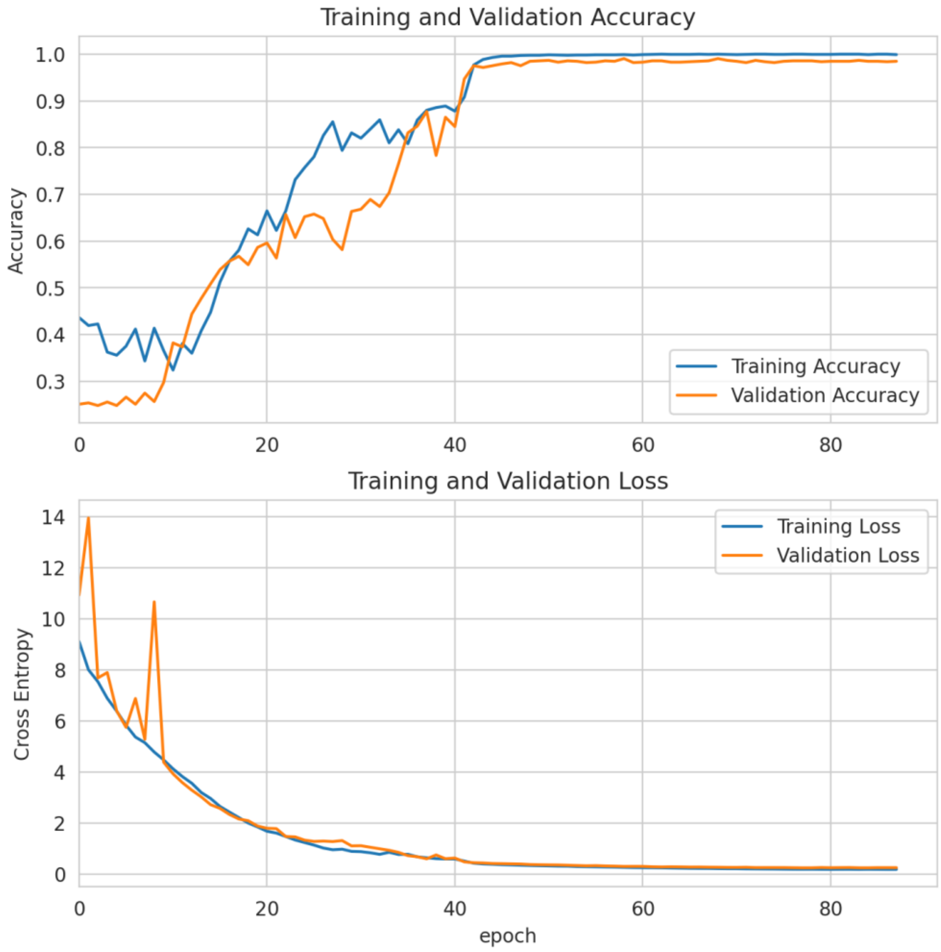
	EfficientNetV2S	**ResNet50**	**DenseNet121**	**ResNeXt50**
Categorical accuracy	**100%**	**99.96%**	**99.16%**	94.35%
validation categorical accuracy	**98.58%**	**98.39%**	**84.03%**	**64.92%**
AUC percentage difference	7.67%	3.98%	2.79%	11.11%
Accuracy percentage difference	1.42%	1.56%	15.24%	31.19%
Total testing images	1057	1057	1057	1057

Misclassified test images	23	32	163	377
Prediction accuracy on the test set	**97.82%**	**96.98 %**	**84.59%**	64.33%

Table 4.2 (b): accuracy comparison of training, testing, and validation sets of four models

4.2.1 EfficientNetV2S

EfficientNetV2S training and validation accuracy:



Training and Validation Accuracy

Training and Validation Loss

Figure 4.2.1(a): Training and validation accuracy Graph of EfficientNetV2S

EfficientNetV2S confusion matrix:

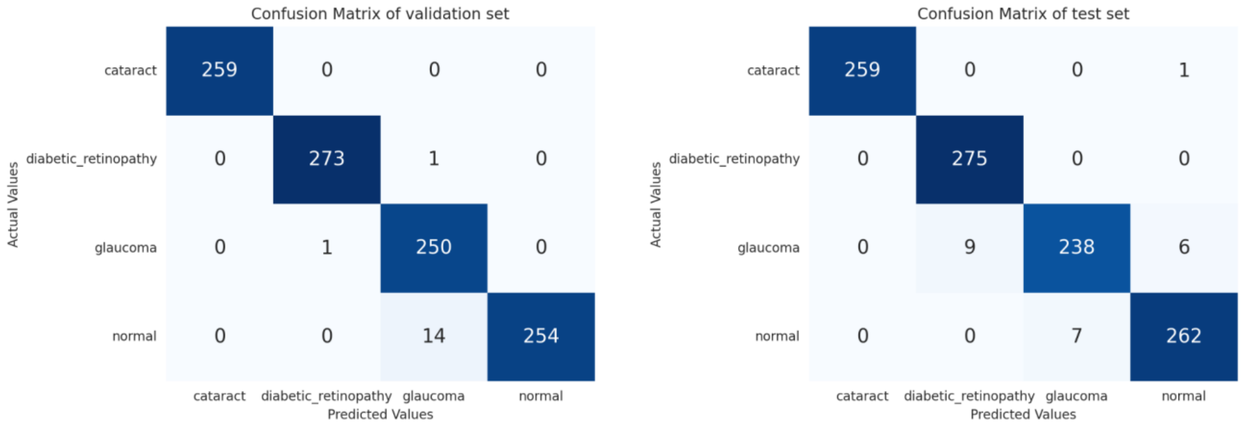


Figure 4.2.1(b): confusion matrix of EfficientNetV2S

EfficientNetV2S Classification report:

	Precision	Recall	F1-score	Support
Cataract	1.0000	0.9962	0.9981	260
Diabetic Retinopathy	0.9683	1.0000	0.9839	275
Glaucoma	0.9714	0.9407	0.9558	253
Normal	0.9740	0.9704	0.9704	269
Micro Avg	0.9782	0.9782	0.9782	1057
Macro Avg	0.9784	0.9777	0.9779	1057
Weighted Avg	0.9783	0.9782	0.9781	1057
Samples Avg	0.9782	0.9782	0.9782	1057

Table 4.2.1: Classification report of EfficeintNetV2S

4.2.2 ResNet50

ResNet50 training and validation accuracy:

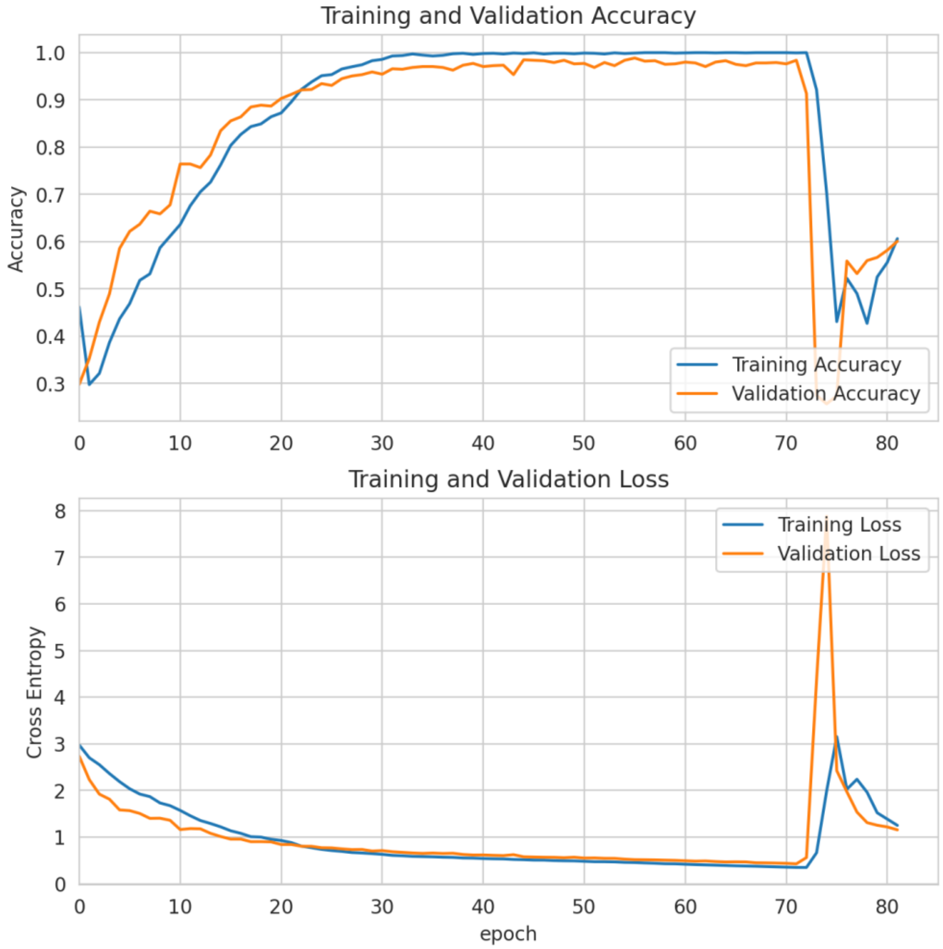


Figure 4.2.2(a): Training and validation accuracy Graph of ResNet50

ResNet50 confusion matrix:

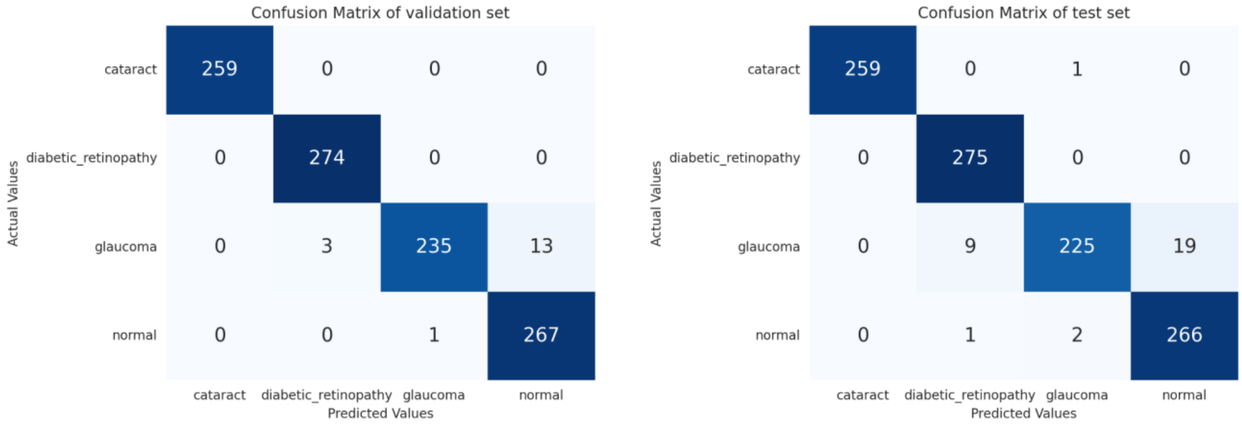


Figure 4.2.2(b): confusion matrix of ResNet50

ResNet50 Classification report:

	precision	recall	f1-score	support
cataract	1.0000	0.9962	0.9981	260
diabetic_retinopathy	0.9649	1.0000	0.9821	275
glaucoma	0.9868	0.8893	0.9356	253
normal	0.9333	0.9888	0.9603	269
micro avg	0.9697	0.9697	0.9697	1057
macro avg	0.9713	0.9686	0.9690	1057
weighted avg	0.9708	0.9697	0.9693	1057
samples avg	0.9697	0.9697	0.9697	1057

Table 4.2.2: Classification report of ResNet50

4.2.3 DenseNet121

DenseNet121 training and validation accuracy:

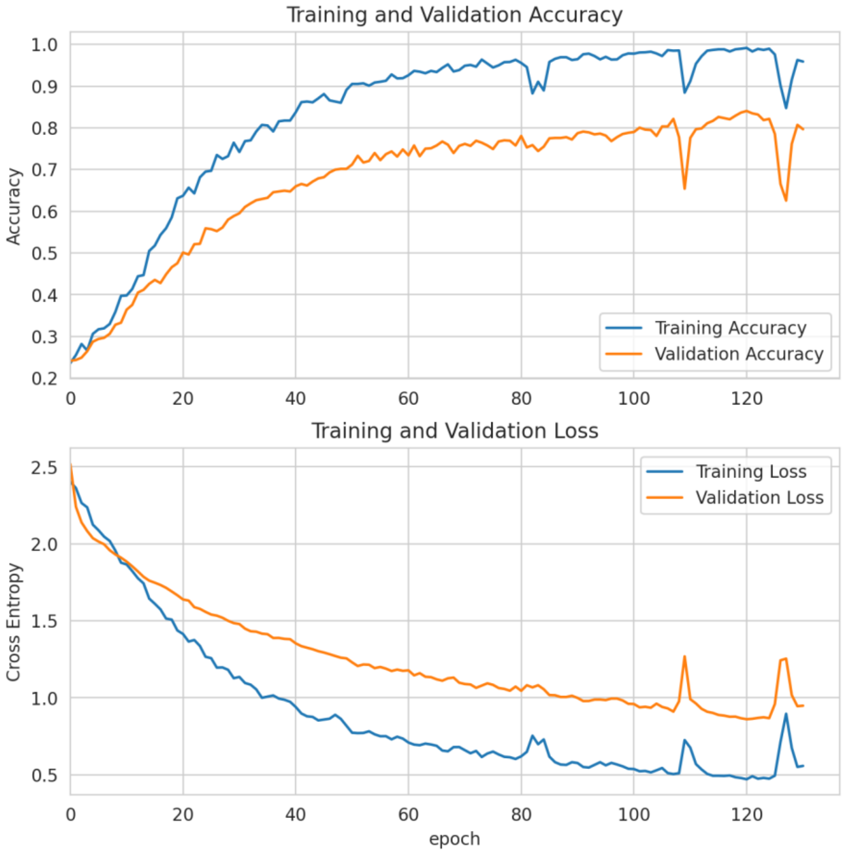


Figure 4.2.3(a): Training and validation accuracy Graph of DenseNet121

DenseNet121 confusion matrix:

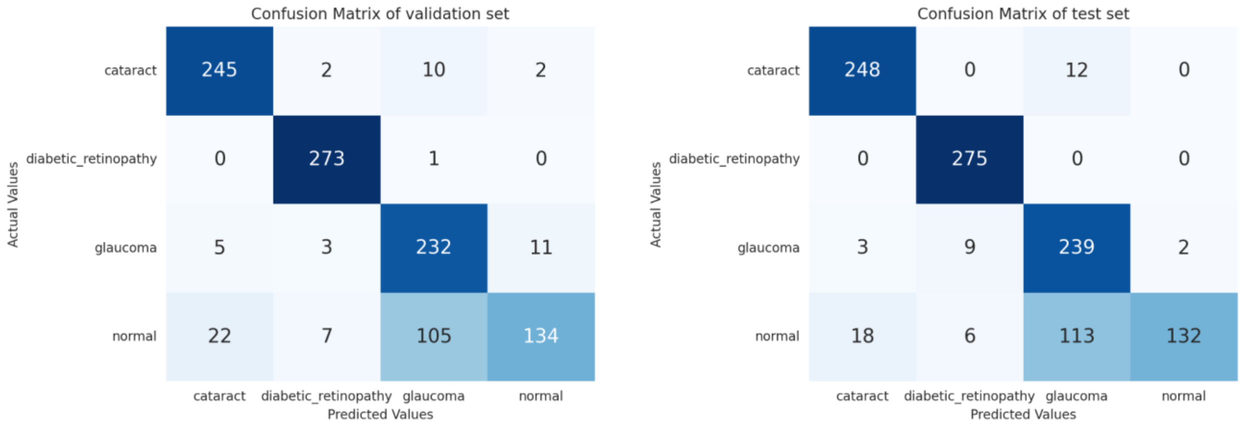


Figure 4.2.3(b): confusion matrix of DenseNet121

DenseNet121 Classification report:

	precision	recall	f1-score	support
cataract	0.9219	0.9538	0.9376	260
diabetic retinopathy	0.9483	1.0000	0.9735	275
glaucoma	0.6566	0.9447	0.7747	253
normal	0.9851	0.4907	0.6551	269
micro avg	0.8458	0.8458	0.8458	1057
macro avg	0.8780	0.8473	0.8352	1057
weighted avg	0.8813	0.8458	0.8360	1057
samples avg	0.8458	0.8458	0.8458	1057

Table 4.2.3: Classification report of DenseNet121

4.2.4 ResNeXt50

ResNeXt50 training and validation accuracy:

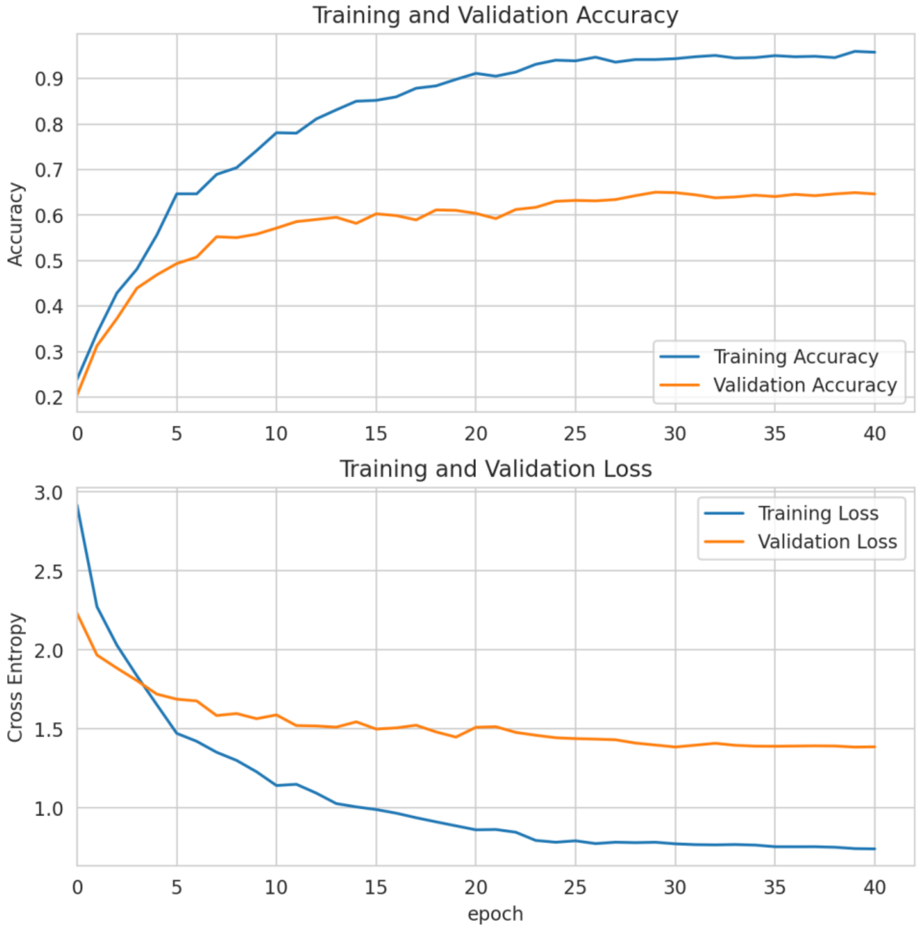


Figure 4.2.4(a): Training and validation accuracy Graph of ResNeXt50

ResNeXt50 confusion matrix:

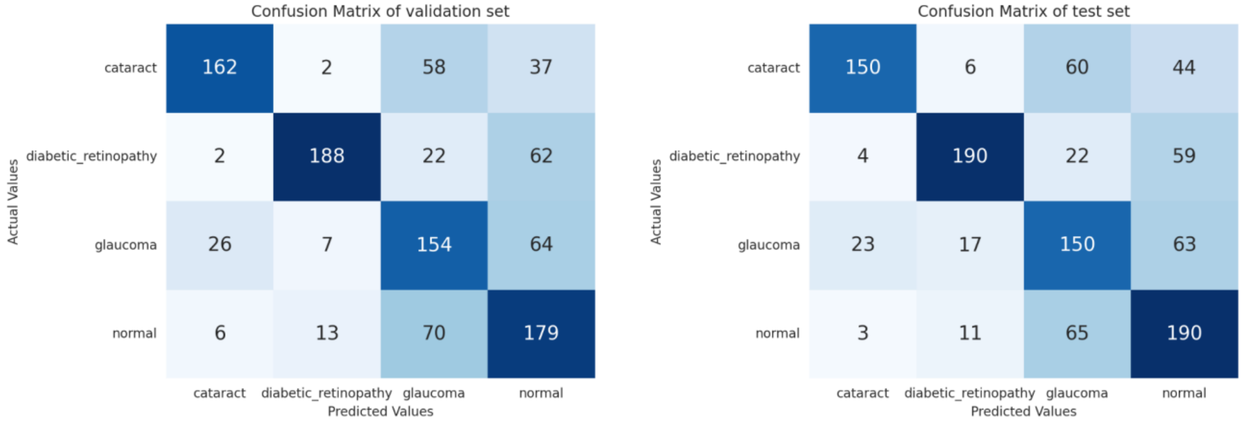


Figure 4.2.4(b): confusion matrix of ResNeXt50

ResNeXt50 Classification report:

	Precision	Recall	F1-score	Support
Cataract	0.8333	0.5769	0.6818	260
Diabetic retinopathy	0.8482	0.6909	0.7615	275
Glaucoma	0.5051	0.5929	0.5455	253
Normal	0.5337	0.7063	0.608	269
Micro avg	0.6433	0.6433	0.6433	1057
Macro avg	0.6801	0.6418	0.6492	1057
Weighted avg	0.6824	0.6433	0.6511	1057
Samples avg	0.6433	0.6433	0.6433	1057

Table 4.2.4: Classification report of ResNeXt50

4.3 Analysis

We evaluated the performance of four deep learning models (**ResNet50, EfficientNetV2S, DenseNet121,** and **ResNeXt50**) for the classification of ocular diseases diagnosis. We tested the models on a dataset of **1057 fundus images**, labeled with one of four classes: **cataract, diabetic retinopathy, glaucoma, and normal.**

As you saw, [section 4.2, Table 4.2 (b): accuracy comparison of training, testing, and validation sets of four models] EfficientNetV2S achieved the highest accuracy on the testing and validation sets, which were **97.82%** and **98.58%,** respectively**.** ResNet50 also performed well, with a testing accuracy of **96.98%** and a validation accuracy of **98.39%.** DenseNet121 and ResNeXt50, on the other hand, performed significantly worse, with testing accuracy of **84.58%** and **64.33%** and validation accuracy of **84.03%** and **64.92%**, respectively.

Also, we can see that ResNet50 and EfficientNetVS2 outperformed the other models in our project, with categorical accuracies of **99.95%** and **100%**, respectively. *This indicates that they were able to correctly classify almost all of the images in the dataset.* DenseNet121 performed slightly worse, with a categorical accuracy of 99.15%, while ResNeXt-50 had the lowest categorical accuracy at 94.35%.

- EfficientNetV2S misclassified **23 test images**, resulting in a prediction accuracy of 97.82%.
- ResNet50 misclassified **32 test images**, resulting in a prediction accuracy of 96.97%.
- DenseNet121 misclassified **163 test images**, resulting in a prediction accuracy of 84.58%.
- ResNeXt-50 misclassified **377 test images**, resulting in a prediction accuracy of 64.33%.
- **EfficientNetV2S** had the best precision and recall values [section 4.2, Table 4.2 (a): Performance accuracy during training of four models], followed by ResNet50 and DenseNet121. ResNeXt-50 had the lowest precision and recall values.

EfficientNetV2S vs. ResNet50 for Glaucoma and Normal Image Prediction [section 4.2, 4.2.1 EfficientNetV2S (EfficientNetV2S confusion matrix) and 4.2.2 ResNet50 (ResNet50 confusion matrix)]

Now, we compared the performance of our two best state-of-the-art deep learning models, EfficientNetV2S and ResNet50, on a four-class classification task of normal, cataract, diabetic retinopathy, and glaucoma images. Using the validation set, ResNet50 correctly predicted cataract, diabetic retinopathy, and normal class images but misclassified 16 glaucoma images. EfficientNetV2S, on the other hand, correctly predicted all cataract, diabetic retinopathy, and glaucoma images but misclassified 14 normal images. Based on these results, we concluded that ResNet50 is not well suited for predicting glaucoma images, while EfficientNetV2S is not well suited for predicting normal images.

4.4 Discussion

ResNet50 and EfficientNetV2S are our two best deep learning models that are well-suited for the task of classifying eye diseases. In our recent experiment, both models achieved high accuracy on a dataset of retinal images, with EfficientNetV2S outperforming ResNet50 on precision and recall metrics. However, DenseNet121, another deep learning model, achieved the highest overall accuracy on the validation set but struggled in the testing phase due to a higher number of misclassifications. ResNeXt-50, a variant of ResNet50, performed slightly worse than ResNet50.

From these findings, we concluded that both ResNet50 and EfficientNetV2S are strong candidates for ocular disease classification, with EfficientNetV2S being the better choice for applications where precision and recall are critical. DenseNet121 may be a better choice for applications where overall accuracy is the top priority, but further fine-tuning may be needed to improve its performance on the testing set. ResNeXt-50 is not as competitive as the other models, but it may still be a viable option for certain applications.

Also in our analysis part, we saw that ResNet50 is good at predicting cataracts, diabetic retinopathy, and normal class images, but not glaucoma images. EfficientNetV2S is good at predicting cataracts, diabetic retinopathy, and glaucoma images, but not normal images. In the future, we will need to fine-tune and ensemble these two models to improve their performance. Hopefully, the new model will be able to correctly predict all four classes of images.

Chapter 5 Impacts of the Project

5.1 Impact of this project on societal, health, safety, legal and cultural issues

- **Improved Healthcare Access**

One of this project's most significant social benefits is improved healthcare access. Automating the diagnosis process can make healthcare services more accessible to underserved populations, especially in remote or resource-constrained areas where access to specialists may be limited. In context to a country like Bangladesh where access to good healthcare is limited this project may come up with tremendous benefits.

- **Early Disease Detection**

Early detection of ocular diseases is critical for preventing vision loss. This project can contribute to the early detection of diseases such as diabetic retinopathy, glaucoma, and macular degeneration, leading to timely interventions and improved patient prognosis.

- **Reduced Healthcare Costs**

Efficient diagnosis and early intervention can lead to reduced healthcare costs in the long run. By identifying ocular diseases at an early stage, patients may require fewer intensive treatments, resulting in cost savings for both individuals and healthcare systems.

- **Skill Enhancement**

The project offers opportunities for medical professionals and researchers to enhance their skills in the field of ophthalmology and machine learning. This can lead to a more knowledgeable and skilled workforce in the healthcare industry.

5.2 Impact of this project on environment and sustainability

In today’s world, people are more concerned about Environmental sustainability as well as computer engineers. From that concern, a very important question arises: How Green is your software? It’s totally impossible to make software that does not have any environmental effects. The software makes hardware run, hardware uses electricity, and electricity production emits greenhouse gasses. But we can develop our software so that it has as little greenhouse emissions as possible or it can solve a problem that has greater greenhouse emissions in the Environment. This project will be developed in a way that it solves bigger greenhouse emission problems such as:

- **Reduction in Physical Records**

The digitization of medical records and the use of deep learning models can reduce the reliance on physical records and paperwork, resulting in reduced paper usage and storage needs in healthcare facilities. This has a positive environmental impact by reducing paper waste and resource consumption.

- **Energy Consumption**

The use of computational resources for deep learning models may increase energy consumption, particularly in data centers and facilities hosting the necessary infrastructure. To mitigate this, the project should consider energy-efficient computing solutions and renewable energy sources to minimize its environmental footprint.

- **Sustainability of code and project**

A sustainable project can limit invisible waste products as well as ensure environment-friendly ones. In the realm of environmentally conscious software development, the "Seven R's" principles find their application to minimize the ecological impact of the development process. By refusing unnecessary features, reducing resource-intensive coding practices, reusing code and components, repurposing existing solutions, and prioritizing software repair and maintenance, developers can create more energy-efficient and sustainable software products. Additionally, implementing efficient algorithms and recycling code and data when retiring further reduces waste and energy consumption. Embracing these principles in “Ocular Disease Diagnosis” not only makes it a greener software but also aligns with a broader commitment to environmental sustainability in the technology industry.

- **Data Privacy and Security**

While not a direct environmental effect, the proper handling of patient data is essential. Ensuring robust data privacy and security measures is crucial to prevent data breaches, which could lead to environmental consequences if data storage facilities are compromised.

Chapter 6 Project Planning

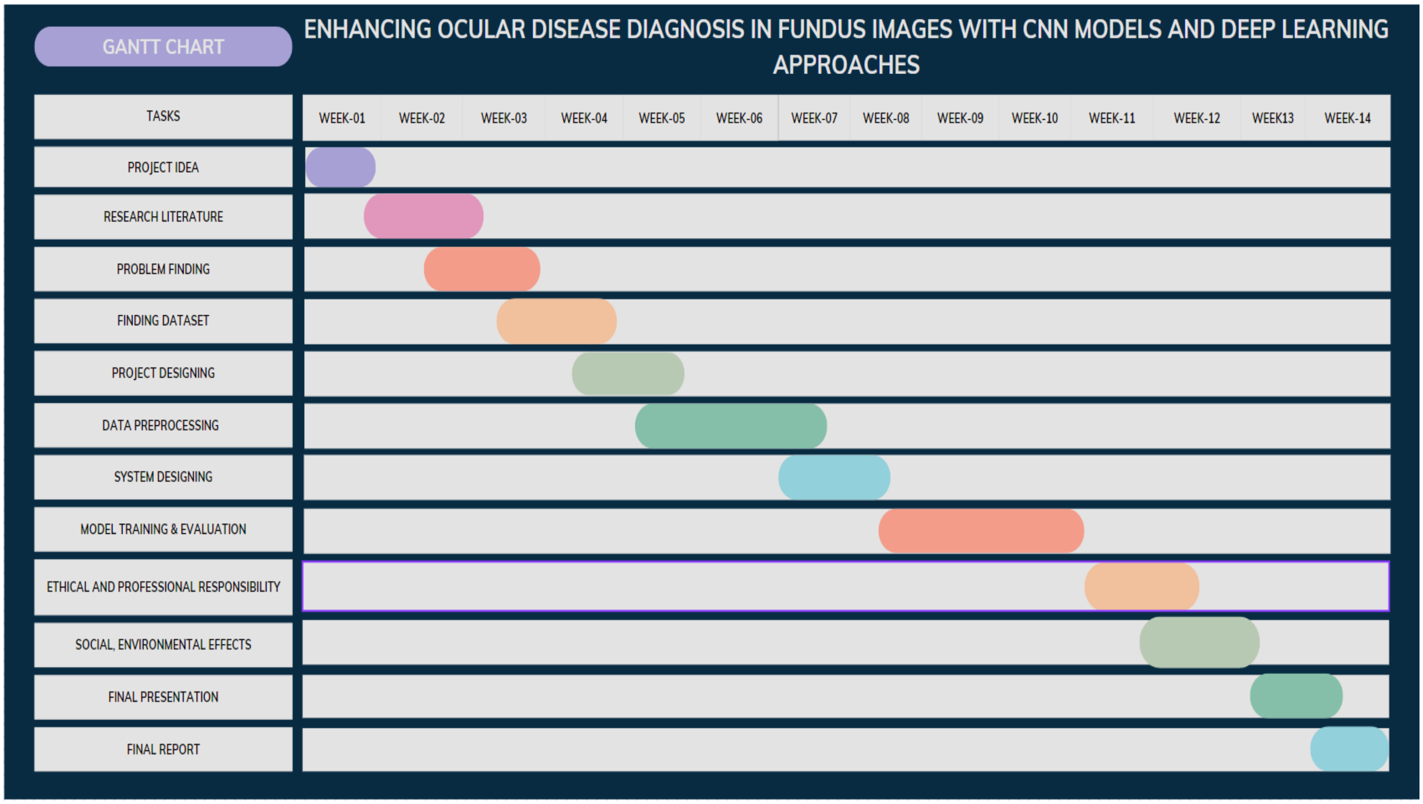


Figure 6. Gantt chart of project planning

Chapter 7 Conclusions

7.1 Summary

Our project focused on improving ocular disease diagnosis using advanced deep learning techniques. We employed the Ocular Disease Recognition (ODIR) dataset, containing 5,000 patient records with fundus images, reflecting real-world scenarios. We primarily focused on three common diseases: diabetes retinopathy, glaucoma, and cataract. We implemented Convolutional Neural Networks (CNNs), four pre-trained models including EfficientNetV2S, ResNet-50, DenseNet-121, and ResNeXt-50, to classify fundus images into specific disease categories. EfficientNetV2S was the best-performing model on the validation and testing sets after hyperparameter tuning, and the testing set confusion matrix shows that it has lower misclassified images than the other models. ResNet50 is close to performing well, with good testing and validation accuracy and the lowest misclassified images in the testing set. Also, we saw that EfficientNetV2S had the best precision and recall values, followed by ResNet50 and DenseNet121. ResNeXt50 had the lowest precision and recall values. However, DenseNet121, another deep learning model, achieved the highest overall accuracy on the validation set but struggled in the testing phase due to a higher number of misclassifications. ResNeXt-50, a variant of ResNet50, performed slightly worse than ResNet50. and our best two models, ResNet50 and EfficeintNetV2S's confusion matrix, also show that ResNet50 is not well suited for predicting glaucoma images, while EfficientNetV2S is not well suited for predicting normal images. In the future, we will need to fine-tune and ensemble these two models to improve their performance. Hopefully, the new model will be able to correctly predict all four classes of images.

Finally, we concluded that ResNet50 and EfficientNetV2s are our two best deep-learning models that are well-suited for the task of classifying eye diseases.

7.2 Limitations

- **Data Variability:** The availability of a variety of representative fundus image datasets impacts the effectiveness of CNN models and deep machine learning methods in the diagnosis of ocular diseases. The ability of these models to be applied to actual clinical situations may be hampered by a lack of diverse data.
- **Interpretability and Explain ability:** Clinicians may find it difficult to comprehend the reasoning behind a specific diagnosis due to the frequent lack of transparency and interpretability in deep learning models. This restriction may make it more difficult for physicians to accept and trust these models.
- **Hardware and Computational Resources:** High-performance GPUs are among the computational resources needed to implement deep learning models for the diagnosis of ocular diseases. Restricted availability of said resources may limit the project's potential for growth and feasibility.
- **Ethical and Regulatory Restrictions:** It can be difficult and time-consuming to ensure commitment to patient privacy laws, ethical standards, and to get the required approvals for clinical deployment. This can cause a delay in the project's adoption and execution in real healthcare settings.

7.3 Future Improvement

- Developing more accurate and robust deep learning models:
    - ▪ Researchers can continue to develop new deep learning model architectures and train them on larger and more diverse datasets. This will help to improve the accuracy and robustness of deep learning models for ocular disease diagnosis.
- Using multi-modal data:

    - Deep learning models can be trained on a variety of data modalities, such as fundus images, OCT images, and patient medical records. Using a multi-modal approach can improve the performance of the model.

- Using ensemble learning:
    - Ensemble learning is a machine learning technique that combines the predictions of multiple models to produce a more accurate prediction. Ensemble learning can be used to improve the performance of deep learning models for ocular disease diagnosis.
- Developing deep learning models for specific ocular diseases:
    - Researchers can develop deep learning models that are specifically designed to diagnose specific ocular diseases, such as diabetic retinopathy, glaucoma, and age-related macular degeneration. This could lead to the development of more accurate and efficient diagnostic tools for these diseases.
- Making deep learning models more accessible to clinicians:
    - Researchers can develop user-friendly interfaces for deep learning models that make them easy for clinicians to use. This could help to accelerate the adoption of deep learning in clinical practice.

References

1. Yau JW, Rogers SL, Kawasaki R, *et al* Global prevalence and major risk factors of diabetic retinopathy. Diabetes Care 2012;**35**:556–64.
2. Wong WL , Su X , Li X , *et al* Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis. Lancet Glob Health 2014;**2**:e106-16.
3. Tham YC, Li X, Wong TY, *et al* Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. Ophthalmology 2014;**121**:2081–90.
4. Lee A, Seattle M, Taylor P, Kingdom U. Machine learning has arrived ! Ophthalmology. 2017:1726-8.
5. Grewal, Parampal S., et al. "Deep learning in ophthalmology: a review." *Canadian Journal of Ophthalmology* 53.4 (2018): 309-313.
6. Ting, Daniel Shu Wei, et al. "Artificial intelligence and deep learning in ophthalmology." *British Journal of Ophthalmology* 103.2 (2019): 167-175.