**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** |

**Faculty Advisor:**

**Dr. Mohammad Ashrafuzzaman Khan**

**Assistant Professor**

**ECE Department**

# Methodology

## System Design

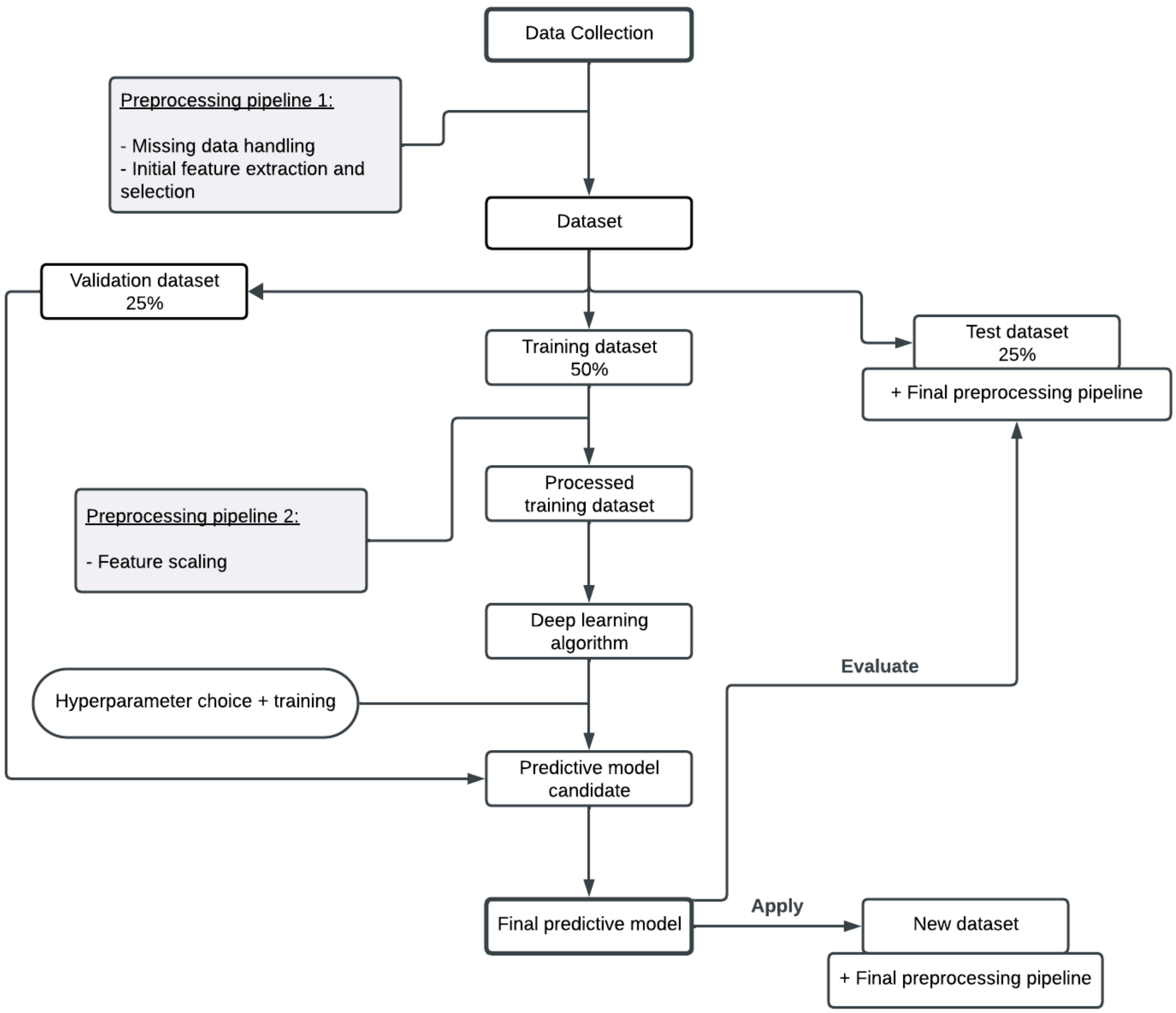


**Figure : System Design Diagram.**

## 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 : Image Distribution

## 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 : Dataset Distribution.

### 1.1.2 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.

### 1.1.3 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.

# Investigation/Experiment, Result, Analysis

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.

## 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 : 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 : accuracy comparison of training, testing, and validation sets of four models**

## EfficientNetV2S

**EfficientNetV2S training and validation accuracy:**

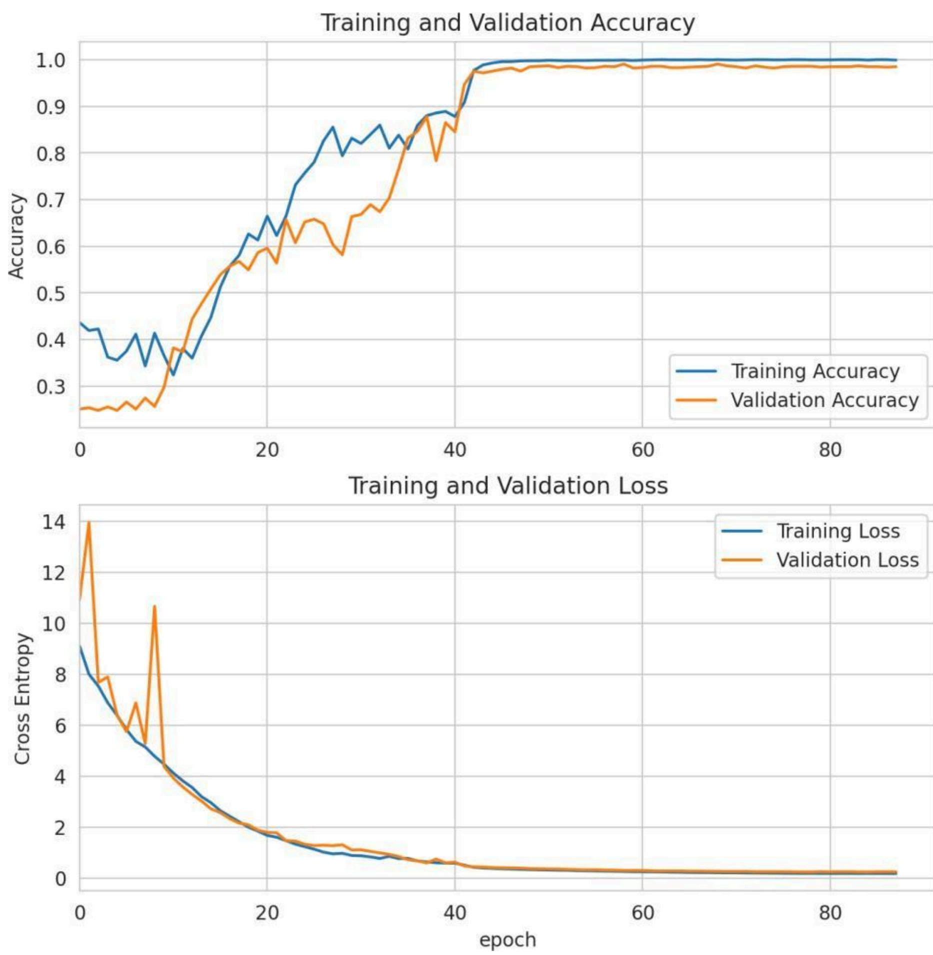


**Figure : Training and validation accuracy Graph of EfficientNetV2S**

## EfficientNetV2S confusion matrix:

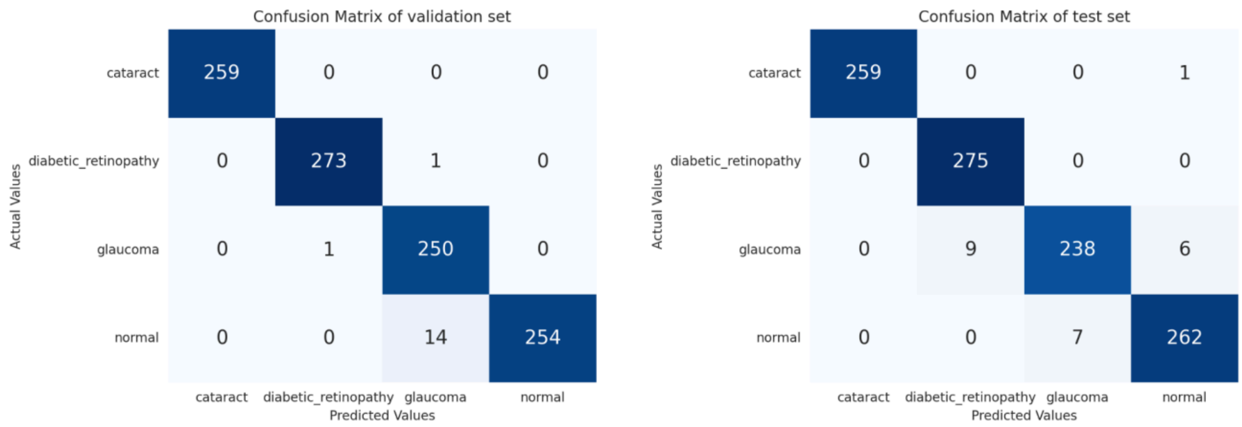


**Figure : confusion matrix of EfficientNetV2S**

## 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 : Classification report of EfficeintNetV2S**

## ResNet50

**ResNet50 training and validation accuracy:**

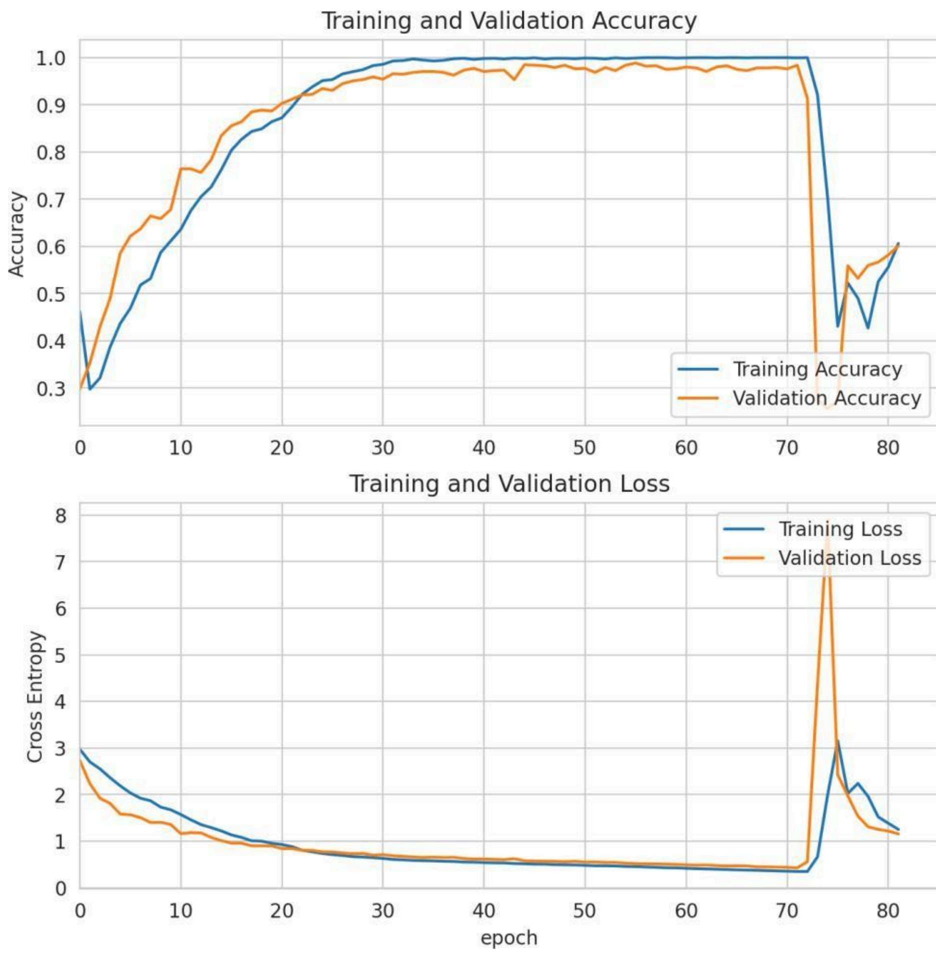


**Figure : Training and validation accuracy Graph of ResNet50**

**ResNet50 confusion matrix:**

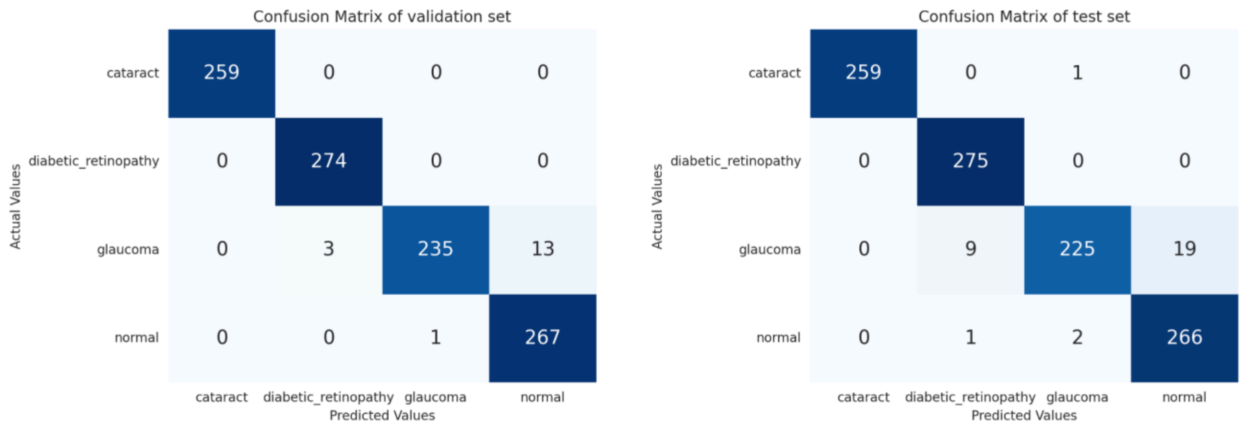


**Figure : confusion matrix of ResNet50**

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 : Classification report of ResNet50**

## DenseNet121

**DenseNet121 training and validation accuracy:**

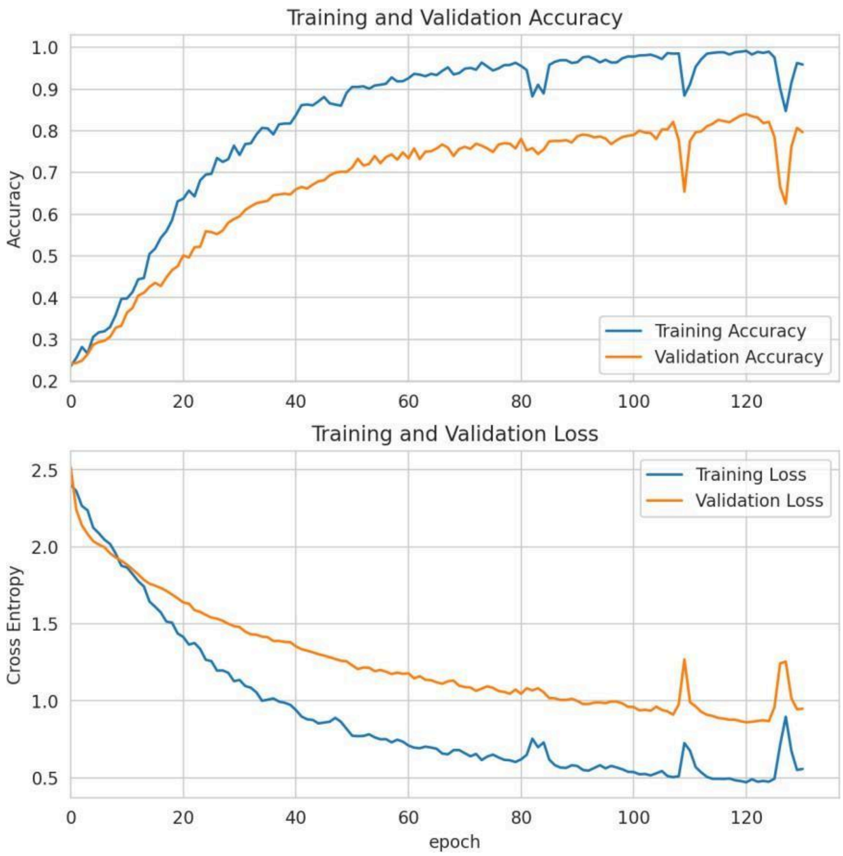


**Figure : Training and validation accuracy Graph of DenseNet121**

## DenseNet121 confusion matrix:

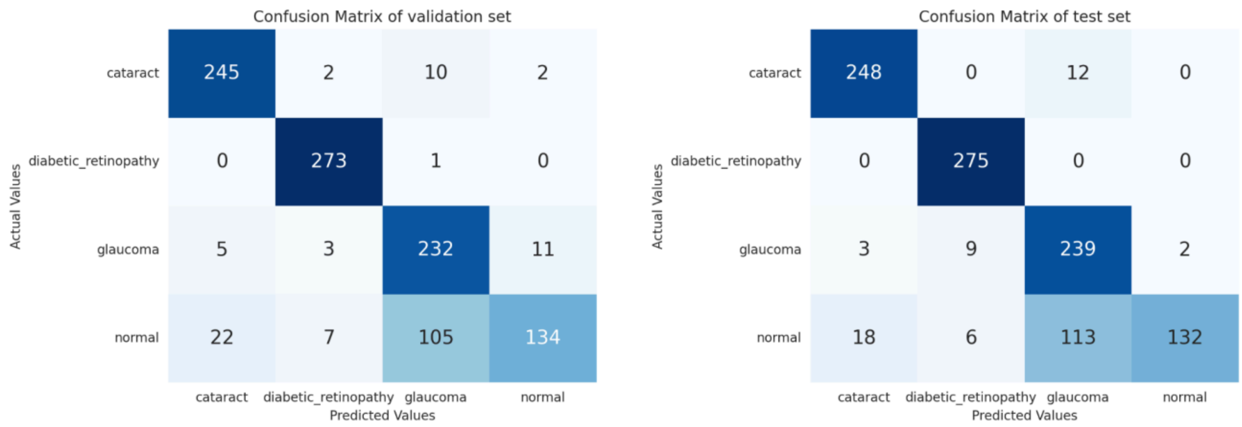


**Figure : confusion matrix of DenseNet121**

## 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 : Classification report of DenseNet121**

## ResNeXt50

**ResNeXt50 training and validation accuracy:**

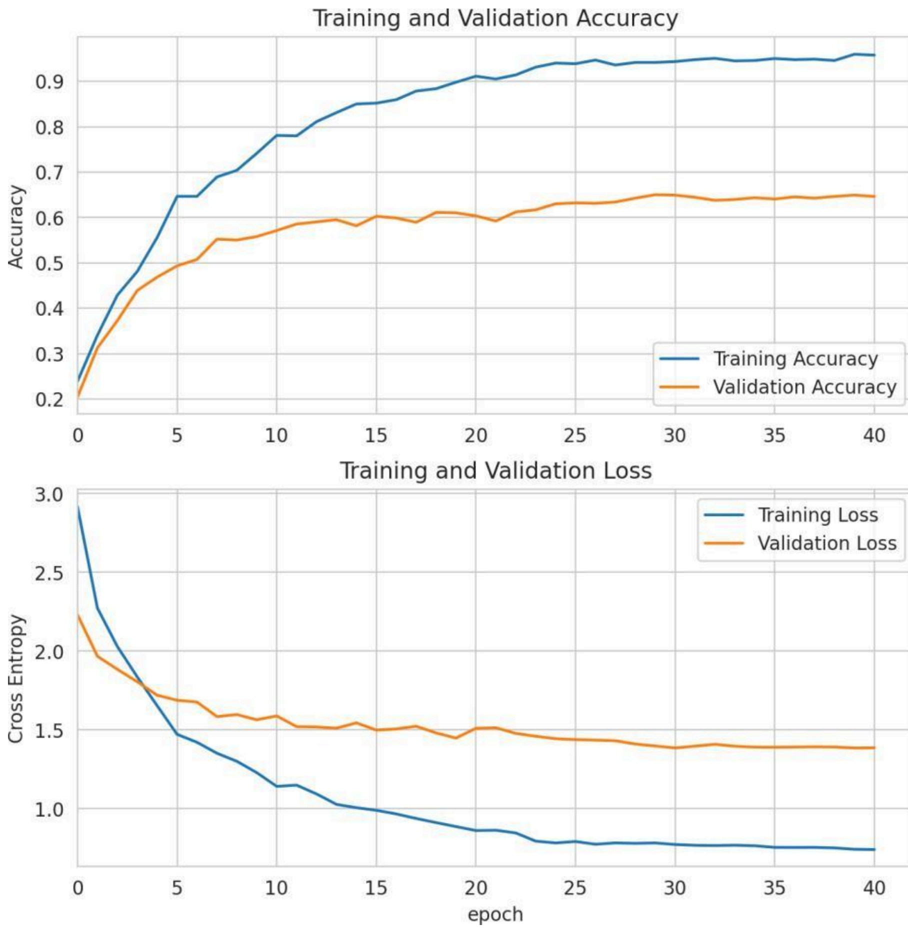


**Figure : Training and validation accuracy Graph of ResNeXt50**

**ResNeXt50 confusion matrix:**

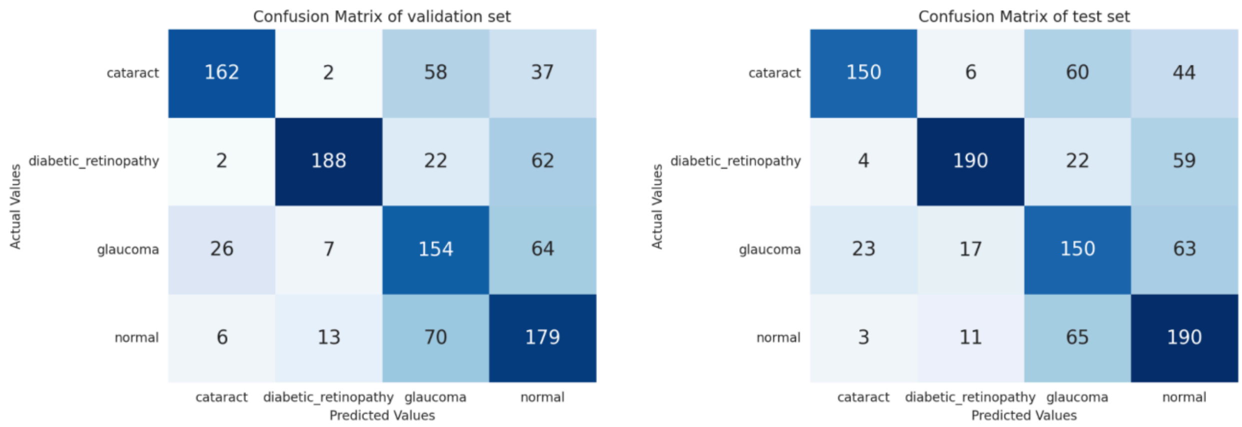


**Figure : confusion matrix of ResNeXt50**

**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 : Classification report of ResNeXt50**

# 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.