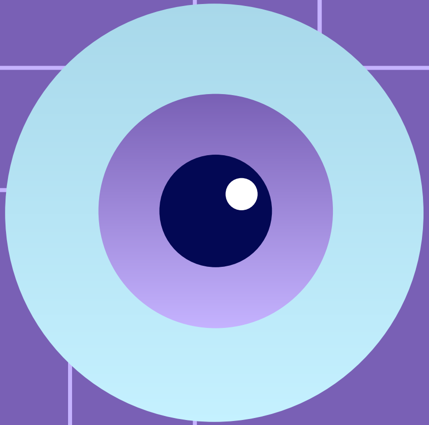
AUTOMATIC CLASSIFICATION OF RETINAL LESIONS



Using deep network features

TABLE OF CONTENTS

01

Introduction

Dataset, Preprocessing, Libraries

02

Transfer Learning

What are CNNS and Transfer Learning concept

03

Results analysis

You can describe the topic of the section here

04

Conclusions

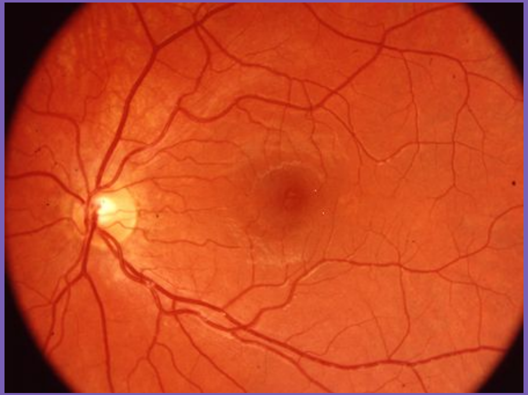
You can describe the topic of the section here

01

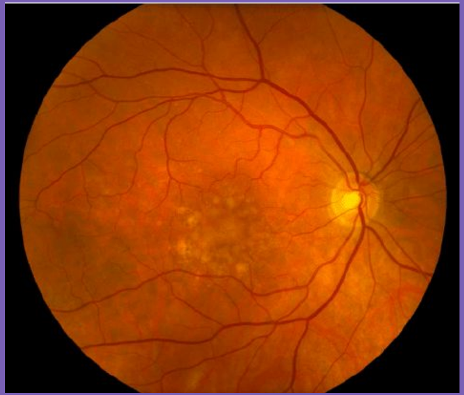
INTRODUCTION & PREPROCESSING

Project's Purpose

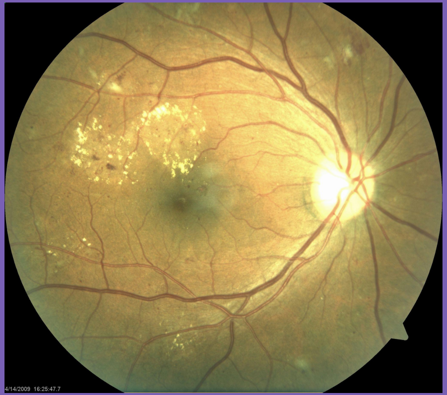
Using deep learning to classify normal, drusen, and exudate retinal fundus images.



?



?



?

Dataset

- Datasets are used to evaluate and measure the performance of deep learning models. By splitting the dataset into training, validation, and testing subsets, one can assess how well the model generalizes to unseen data and compare different models or techniques.

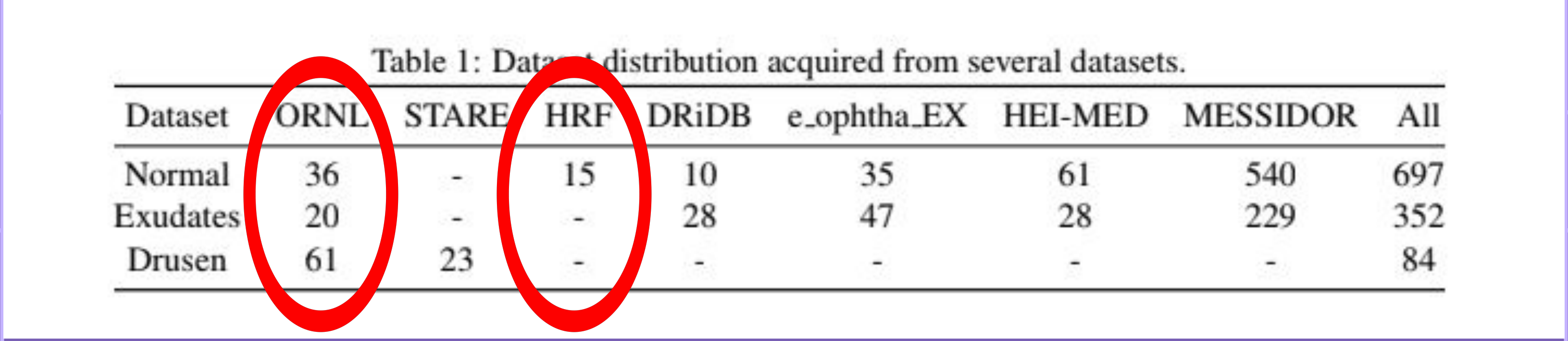


Table 1: Dataset distribution acquired from several datasets.

Dataset	ORNL	STARE	HRF	DRiDB	e_ophtha_EX	HEI-MED	MESSIDOR	All
Normal	36	-	15	10	35	61	540	697
Exudates	20	-	-	28	47	28	229	352
Drusen	61	23	-	-	-	-	-	84

Libraries Used

- **TensorFlow**: Utilized to implement CNN models with ease. It allows efficient computation on both CPUs and GPUs, enabling faster training and inference.
- **Keras**: Offers pre-built layers, optimizers, and loss functions, making it convenient.
- **Scikit-learn**: Provides a wide range of preprocessing, feature extraction, and evaluation techniques that can be beneficial in conjunction with CNN models.

Together, TensorFlow provides a robust deep learning platform, Keras simplifies model creation and prototyping, and scikit-learn offers additional tools for data preprocessing and evaluation.

Preprocessing

1. Resize the image
2. The three channels of the RGB color space are separated.
3. Apply intensity normalization to the green channel
4. Rescale image intensities to 0-255
5. Combine the channels back into an image

02

TRANSFER LEARNING & CNN

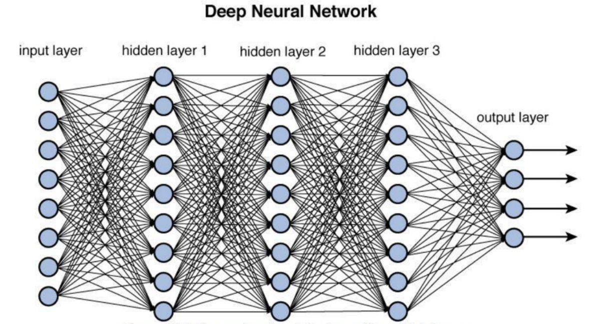


Figure 12.2 Deep network architecture with multiple layers.

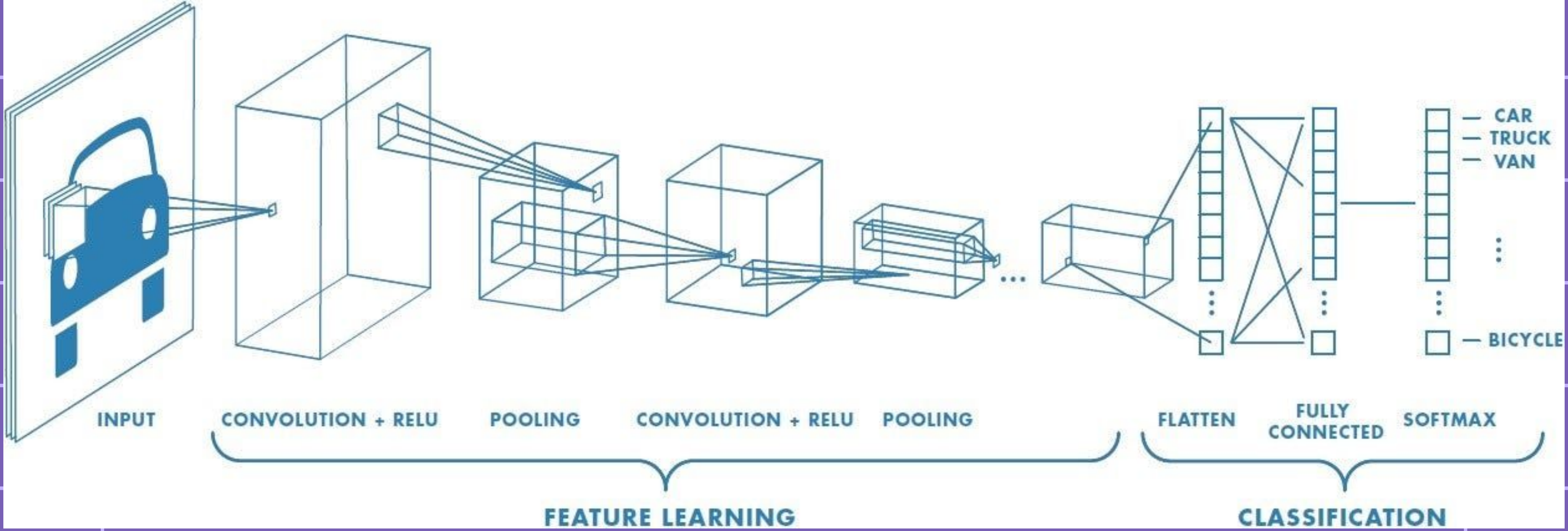
Transfer learning

- Leveraging Pre-trained Knowledge
- Improved Performance with Limited Data
- Time and Resource Efficiency

Convolutional Neural Network (CNN)

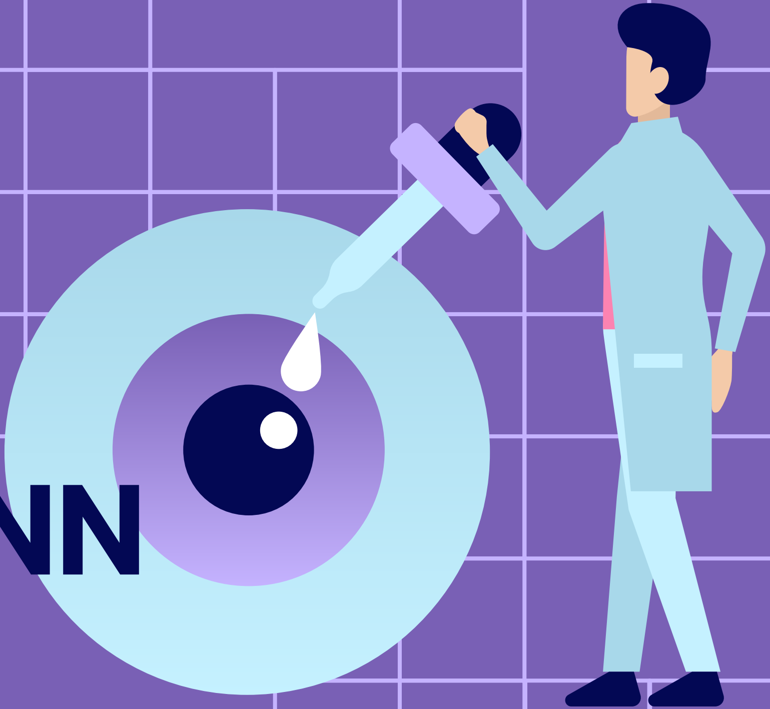
- Powerful Image Analysis Tool
- Robust to Image Variations
- Extensive Applications

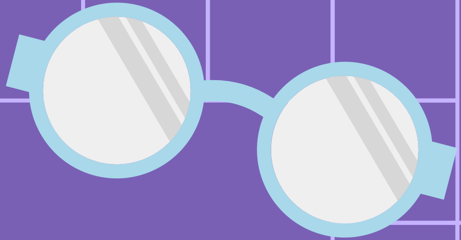
Convolutional Neural Network (CNN)



03

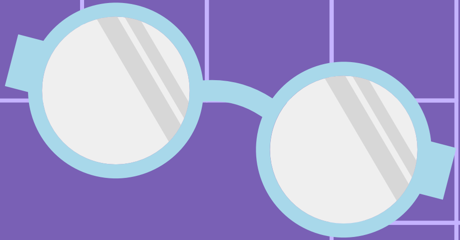
PRETRAINED CNN MODELS USED





ImageNet Dataset

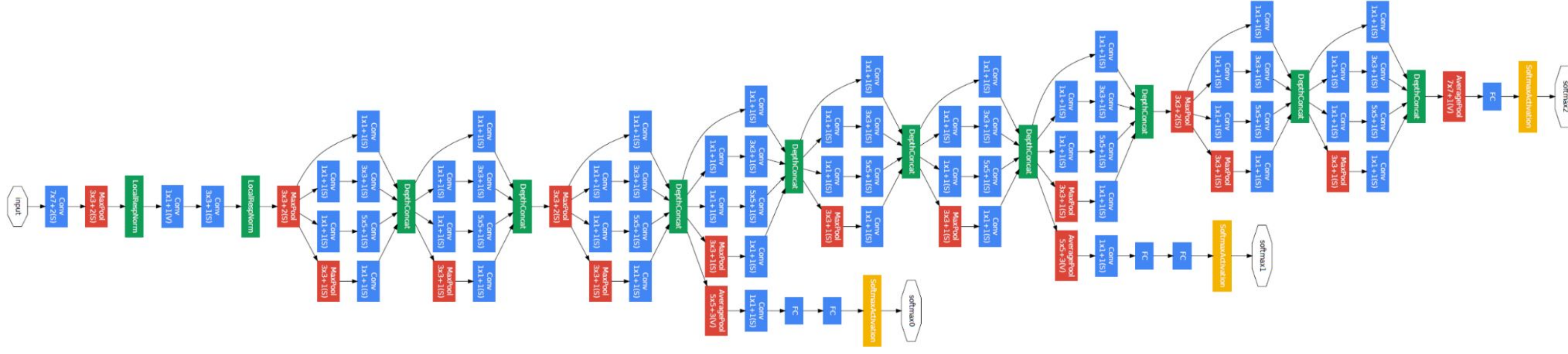
- ImageNet is a large dataset of labeled images used for training and evaluating computer vision algorithms.
- It contains over **14 million images** classified into more than **20,000 categories**.
- It is used as a benchmark in the annual ImageNet **Large Scale Visual Recognition Challenge**.
- It has **promoted the use of transfer learning** techniques, saving time and resources in other computer vision tasks.

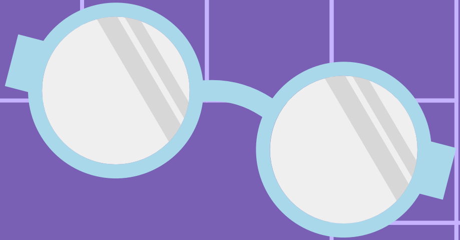


GoogLeNet

- InceptionV1, also known as GoogLeNet, was introduced in 2014.
- It was designed to address the limitations of deeper networks by introducing the concept of "inception modules."
- Inception modules utilize multiple filter sizes within the same layer to capture different scales of information.
- The model achieved state-of-the-art performance on the ImageNet dataset.
- InceptionV1 has approximately 6.6 million parameters.

GoogLeNet

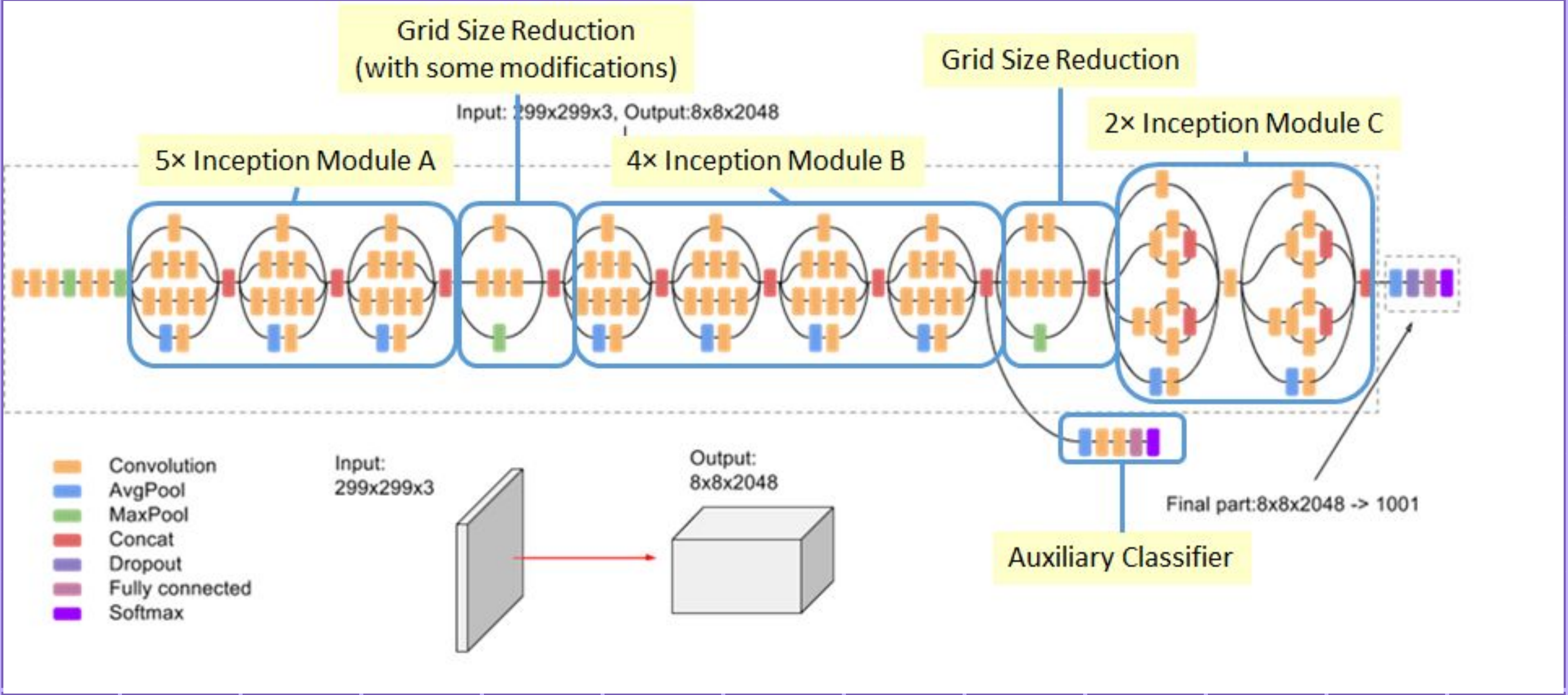

input
Conv 7x7+2(S)
MaxPool 3x3+2(S)
LocalRespNorm
Conv 1x1+1(V)
Conv 3x3+1(S)
LocalRespNorm
MaxPool 3x3+2(S)
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
MaxPool 3x3+1(S)
DepthConcat
AveragePool 5x5+3(V)
FC
SoftmaxActivation
softmax0
softmax1
AveragePool 7x7+1(V)
softmax2



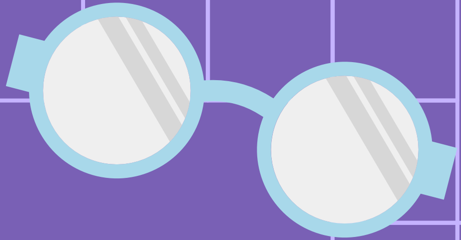
InceptionV3

- **InceptionV3** is an improved version of **InceptionV1**, released in 2015.
- It introduced several architectural enhancements, including **factorized convolutions** and aggressive use of **batch normalization.**
- The model is deeper and more complex than **InceptionV1**, with approximately **23 million parameters**.

InceptionV3



ResNet50



- ResNet50 is a variant of the ResNet (Residual Network) architecture, introduced in 2015.
- ResNet incorporates **skip connections** or shortcuts to address the **vanishing gradient problem** in deep neural networks.
- ResNet50 specifically refers to a **50-layer** deep residual network.
- It achieved remarkable performance on the ImageNet dataset and won the **ImageNet challenge** in 2015.
- ResNet50 has approximately **25.6 million** parameters.

ResNet50

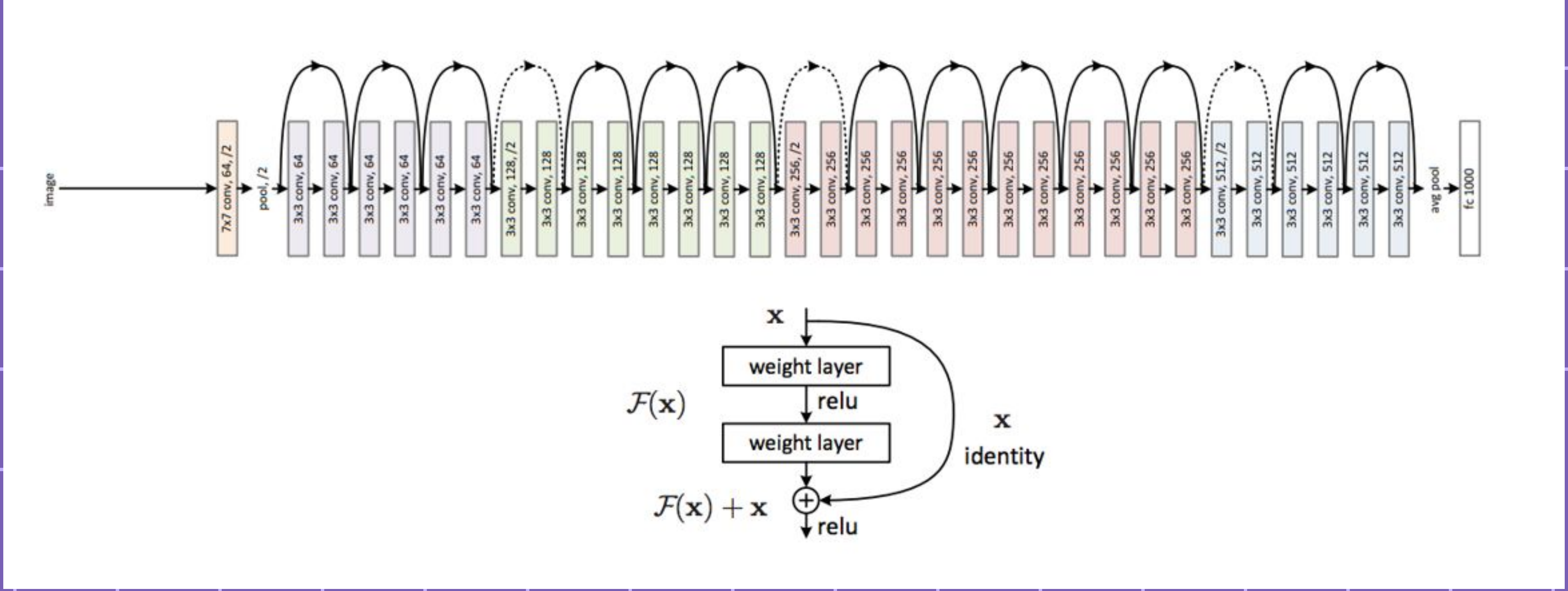
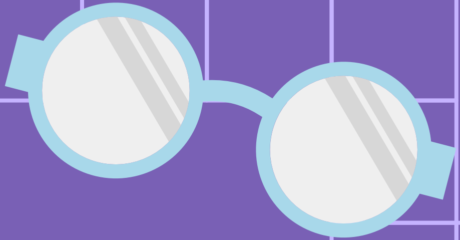


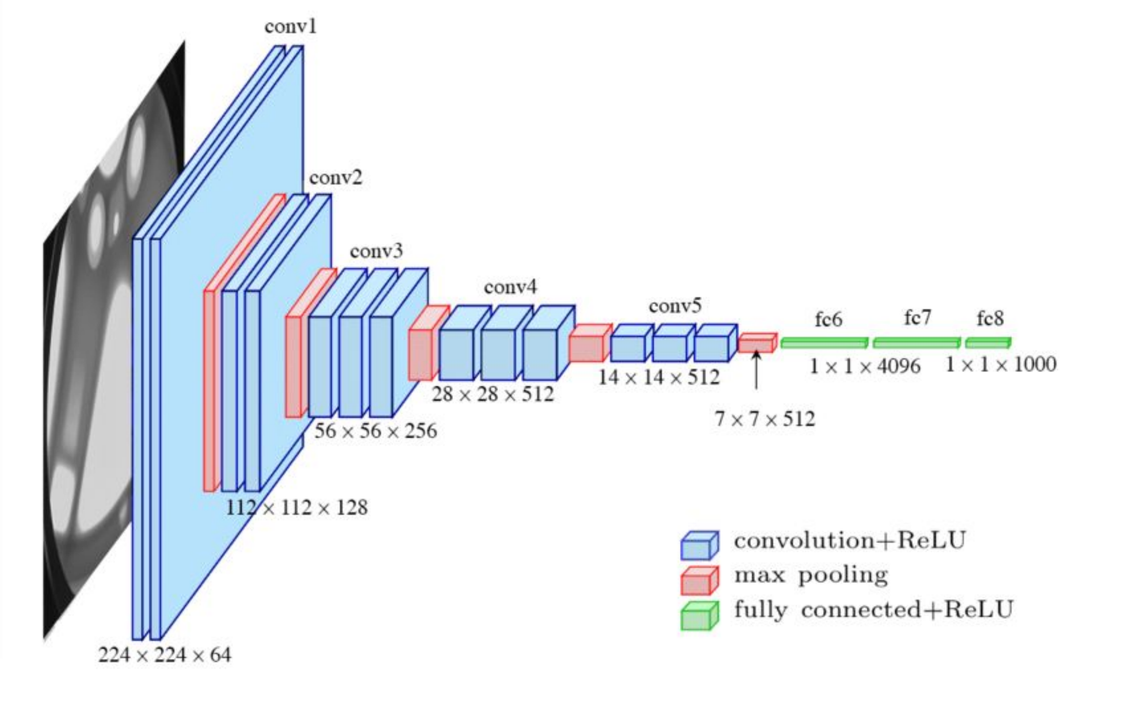
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000
$\mathbf{x}$
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
$\mathbf{x}$
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

VGG16



- **VGG16** is a deep convolutional neural network architecture introduced in **2014**.
- It stands for **Visual Geometry Group 16,** which represents its configuration with **16 layers**.
- Consists of a series of convolutional layers with small filter sizes (3x3) and max pooling layers.
- VGG16 has approximately **138 million** parameters.

VGG16

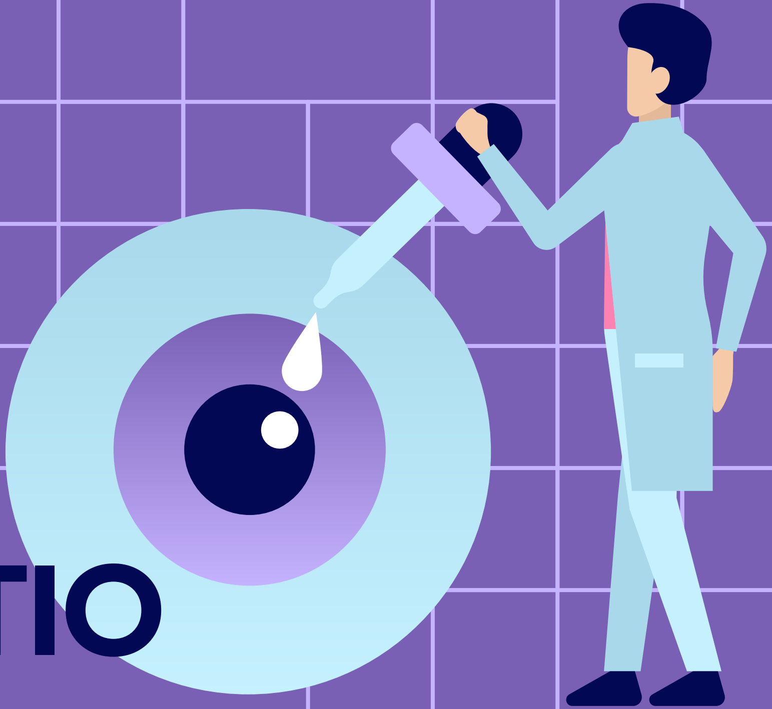


conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

04

SVM IMPLEMENTATIO N

And dealing with a skewed dataset



Model Implementation

01

Train-test split

Stratified splitting of input data

02

SMOTE

Synthetic Minority Oversampling Technique on training set

03

Train SVM

With a RBF kernel

04

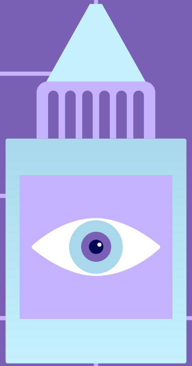
Feed test split

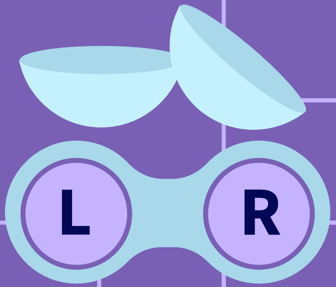
And let the model predict each image’s label

05

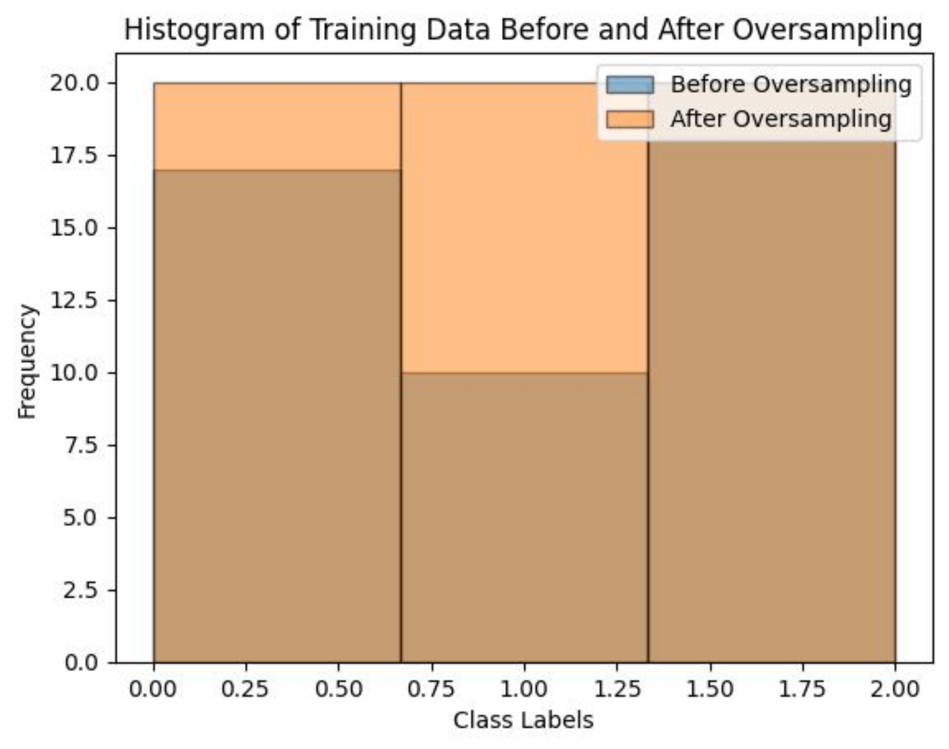
Evaluation

Evaluate the model





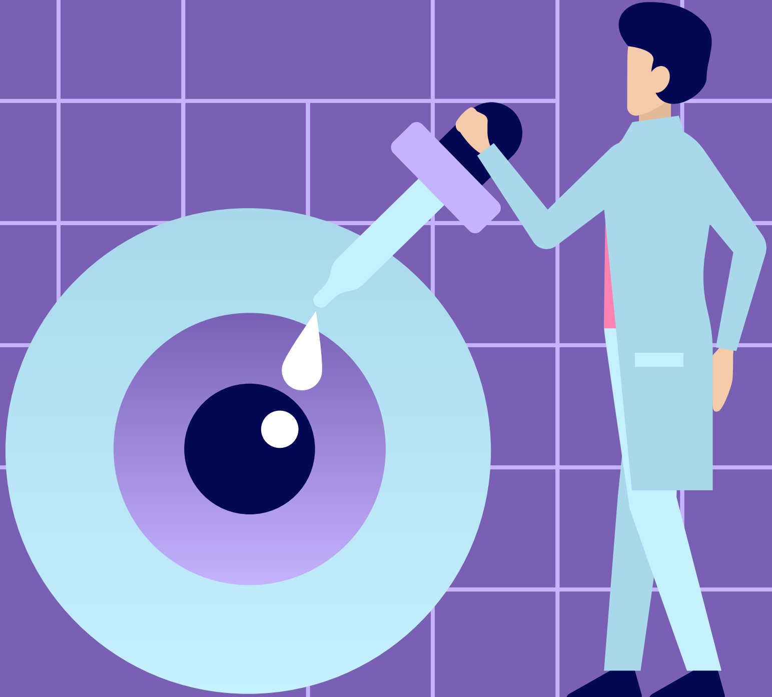
Model Implementation



05

EVALUATION

Performance metrics and Model results



Specificity & Sensitivity

$$\frac{TN}{TN + FP}$$

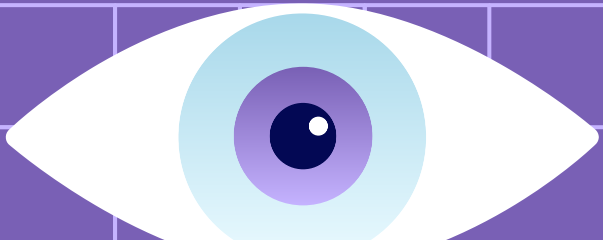
Specificity

focuses on the model's ability to correctly detect negative cases (true negatives) and avoid false positives (positive cases incorrectly classified as negative).

Sensitivity

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

focuses on the model's ability to correctly detect positive cases (true positives) and avoid false negatives (negative cases incorrectly classified as positive).



Accuracy & F1-Score

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy

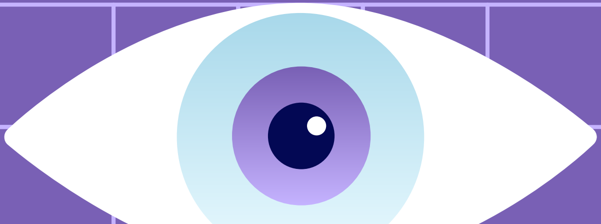
Accuracy is typically presented as a **percentage**, where higher values indicate **better performance.** However, accuracy can be misleading when dealing with **imbalanced** datasets, as a high accuracy value may be achieved by simply predicting the majority class all the time.

F1-score

$$2 \cdot \frac{precision \cdot recall}{precision + recall}$$

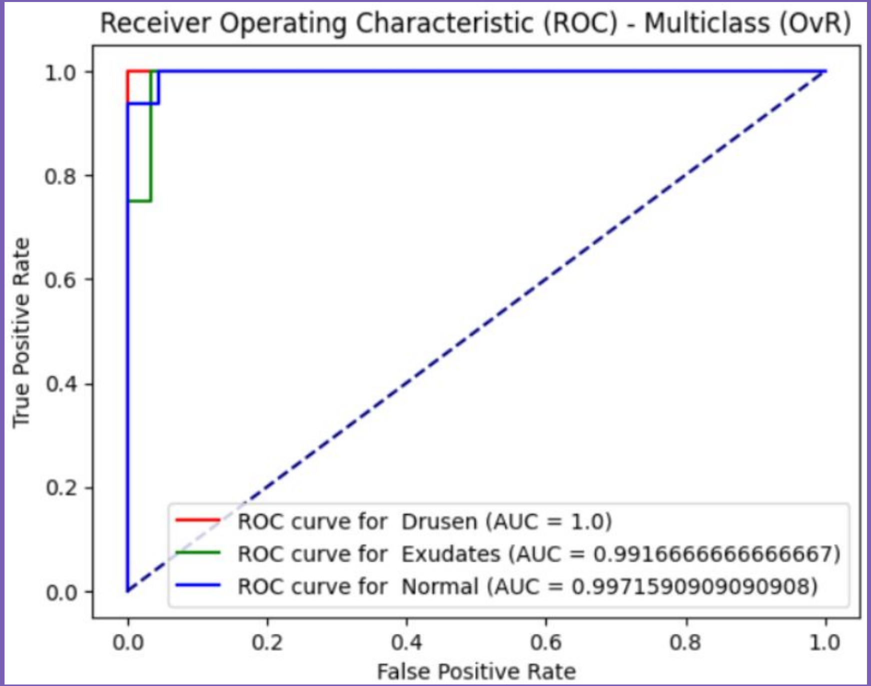
particularly useful when the data is imbalanced, meaning the classes are unevenly represented.

F1-score ranges from **0 to 1,** where a value of 1 indicates **perfect precision and recall**, while a value of 0 indicates **poor performance**.

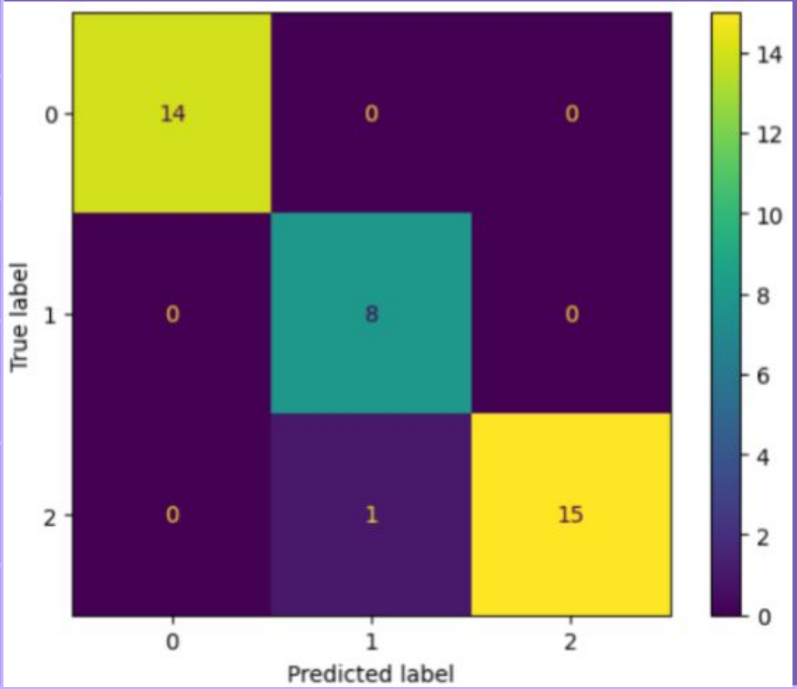


Model -> VGG16

14,714,688	14,714,688	0	block5_conv3
Total Params	Trainable Params	Non-trainable Params	Output Layer


Receiver Operating Characteristic (ROC) - Multiclass (OvR)
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve for Drusen (AUC = 1.0)
ROC curve for Exudates (AUC = 0.9916666666666667)
ROC curve for Normal (AUC = 0.9971590909090908)

Confusion Matrix



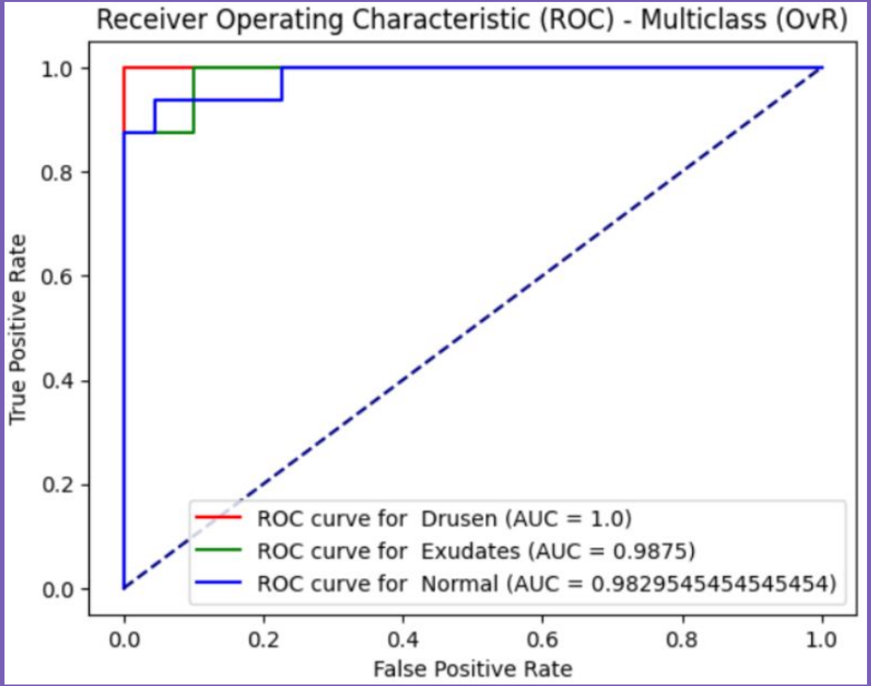
Sensitivity	0.98
Accuracy	0.97
F1-score	0.97

Specificity

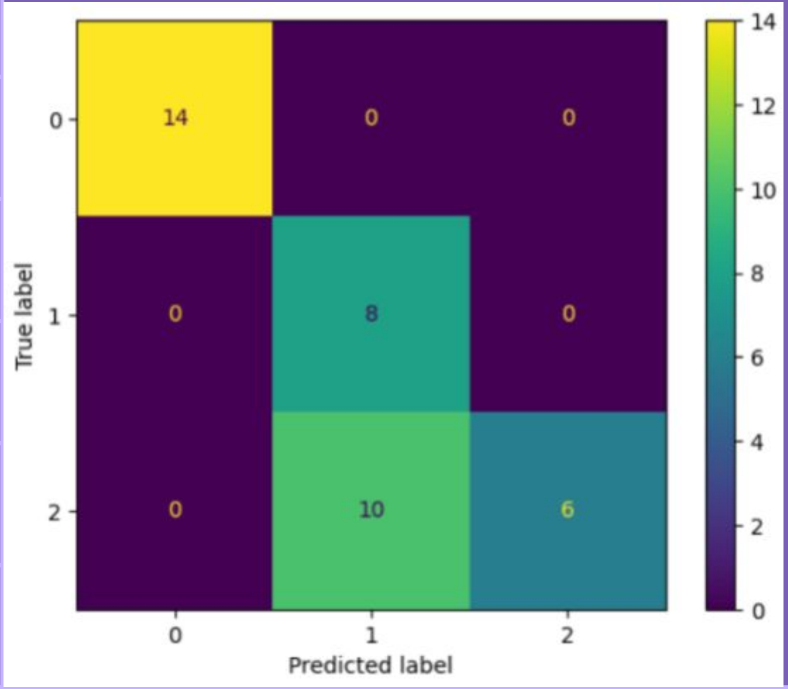
Drusen	1.00
Exudates	0.97
Normal	0.96

Model -> inceptionV3

Total Params	Trainable Params	Non-trainable Params	Output Layer
9,639,968	9,620,384	19,584	conv2d_168


Receiver Operating Characteristic (ROC) - Multiclass (OvR)
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve for Drusen (AUC = 1.0)
ROC curve for Exudates (AUC = 0.9875)
ROC curve for Normal (AUC = 0.9829545454545454)

Confusion Matrix



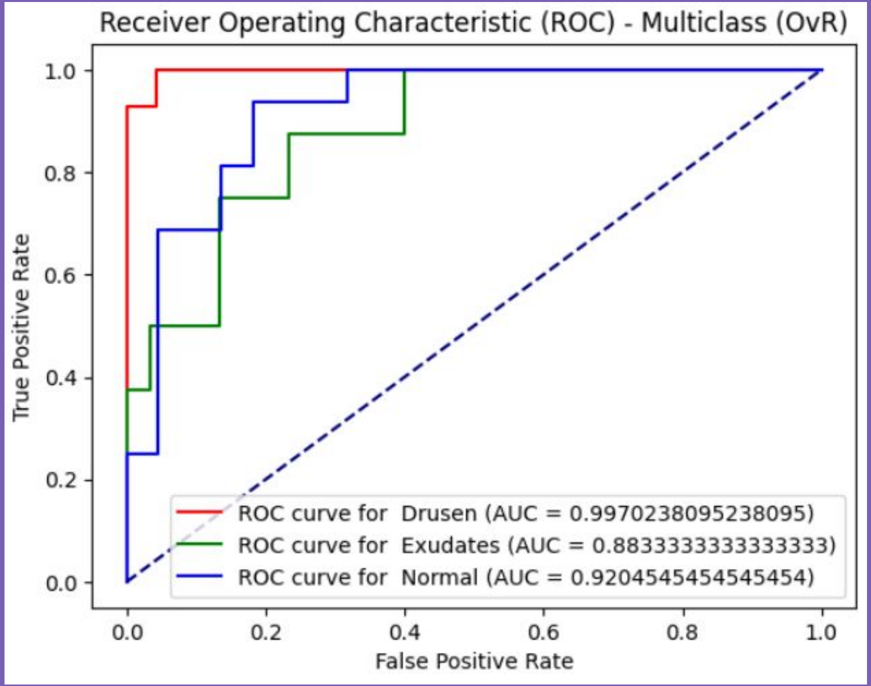
Sensitivity	0.79
Accuracy	0.74
F1-score	0.73

Specificity

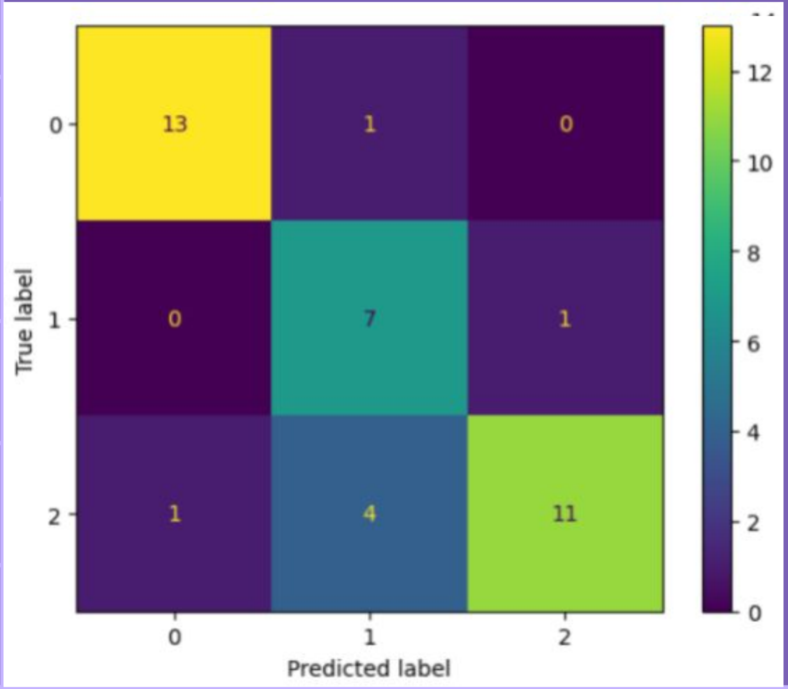
Drusen	1.00
Exudates	0.67
Normal	0.69

Model -> Resnet50

Total Params	Trainable Params	Non-trainable Params	Output Layer
67,968	67,584	384	conv2_block1_3_conv


Receiver Operating Characteristic (ROC) - Multiclass (OvR)
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve for Drusen (AUC = 0.9970238095238095)
ROC curve for Exudates (AUC = 0.8833333333333333)
ROC curve for Normal (AUC = 0.9204545454545454)

Confusion Matrix



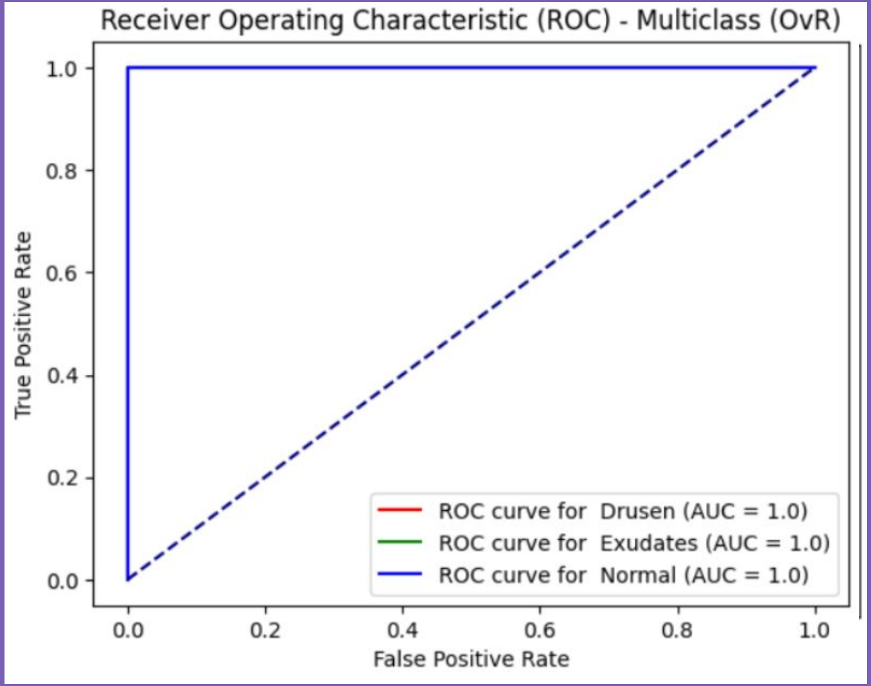
Sensitivity	0.83
Accuracy	0.82
F1-score	0.82

Specificity

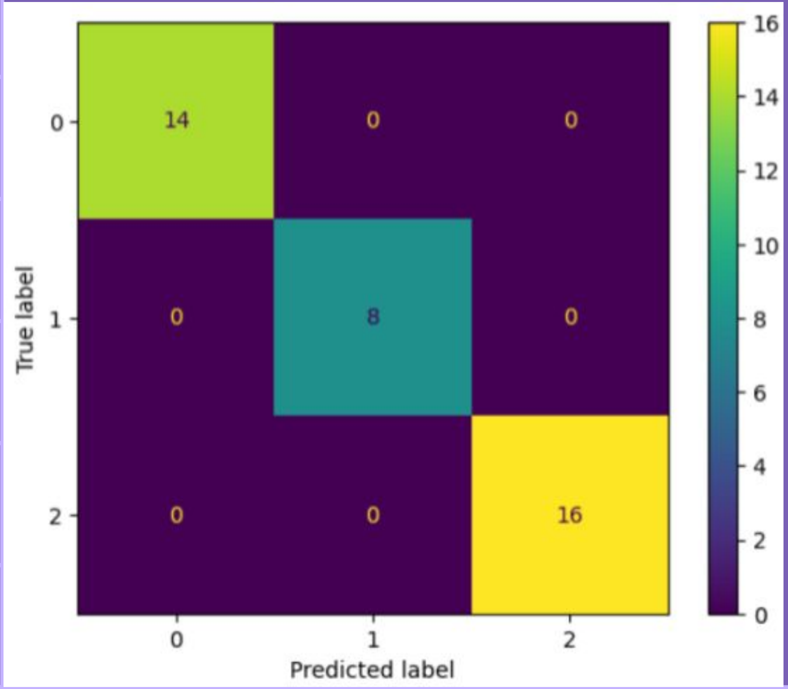
Drusen	0.96
Exudates	0.80
Normal	0.78

Model -> Resnet50

Total Params	Trainable Params	Non-trainable Params	Output Layer
22,526,848	22,478,848	48,000	conv5_block3_2_conv


Receiver Operating Characteristic (ROC) - Multiclass (OvR)
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve for Drusen (AUC = 1.0)
ROC curve for Exudates (AUC = 1.0)
ROC curve for Normal (AUC = 1.0)

Confusion Matrix



Sensitivity	1.00
Accuracy	1.00
F1-score	1.00

Specificity

Drusen	1.00
Exudates	1.00
Normal	1.00

REFERENCES

- Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015 | by Sik-Ho Tsang | Medium

- mawady/DeepRetinalClassification: 3-Class Retinal Classification via Deep Network Features and SVM Classifier (Academic Research Use) (github.com)

- Ibrahim Sadek, Mohamed Elawady, and Abd El Rahman Shabayek. "Automatic Classification of Bright Retinal Lesions via Deep Network Features." [http://arxiv.org/abs/1707.02022].

THANK YOU

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**

Please keep this slide for attribution