Netra.ai

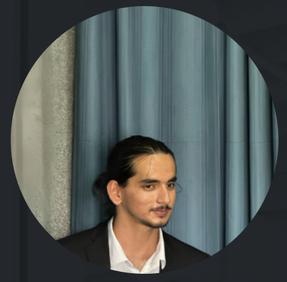
Presented By

Aarya Pathak
Arogya Poudel
Umanga Rayamajhi
Rupak Raj Karki

Our Team

AARYA PATHAK

MACHINE LEARNING ENGINEER



AROGYA POUDEL

DATA SCIENTIST

UMANGA RAYAMAJHI

PYTHON DEVELOPER

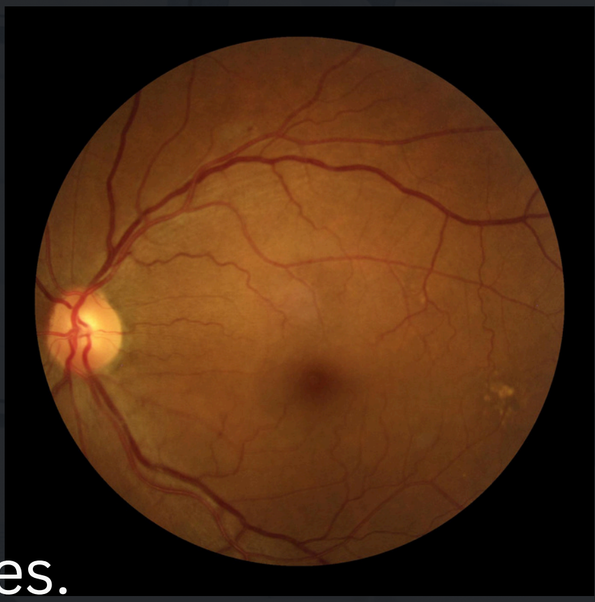
RUPAK RAJ KARKI

DATA ANALYST

Introduction

Netra.ai is a web based AI application that leverages the concept of transfer learning. It lets the user upload their fundus photograph and our application let them know if they have any potential eye disease.

Dataset



- Kaggle (6392 Preprocessed Images).
- 6392 retinal fundus images of right and left eyes.
- Covers 8 eye conditions (e.g., diabetic retinopathy, glaucoma, cataract)

Preprocessing

Pre-processed Data

The dataset consists of eye images, which were resized and normalized to ensure consistency and improve model performance.

Augmentation

Data augmentation is applied to artificially increase the dataset size. Techniques like rescaling helped the model generalize better and avoid overfitting. Additionally we normalized pixel values between 0 and 1 ensuring faster convergence during training.

Training and testing

We split the dataset into 80% training and 20% testing to ensure the model learns effectively while having enough data for evaluation.

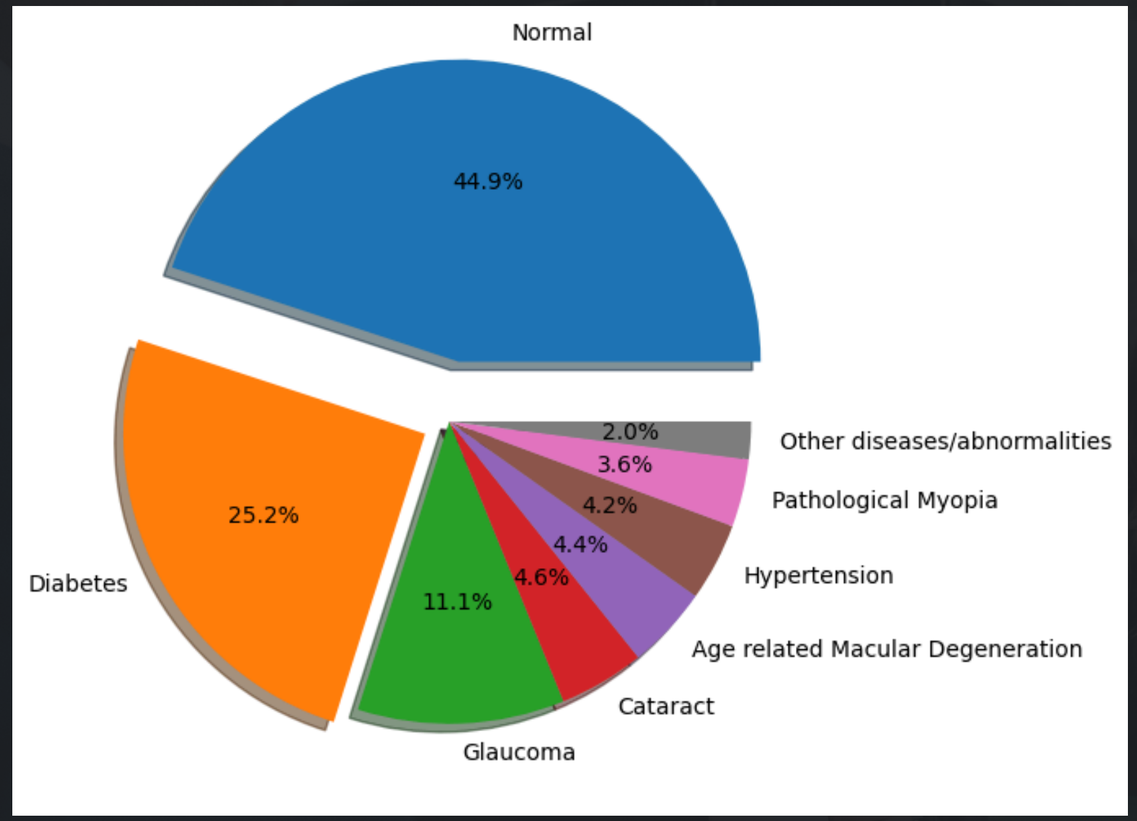
Libraries and Frameworks

- Pandas
- Matplotlib
- Seaborn

Graphical Representation

- Bar Graph
- Pie Chart
- Histograms

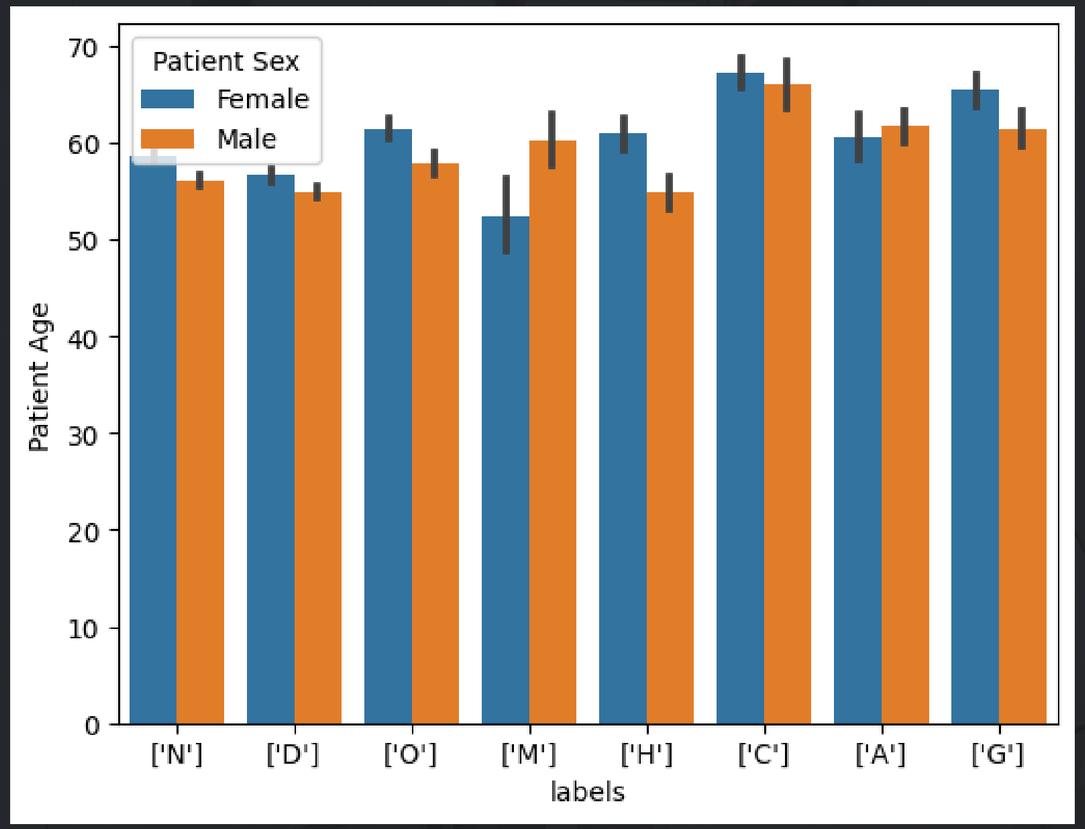
Data Visualization



Our dataset consists of 45% normal eye images and 55% diseased cases, with Diabetes (25.2%) being the most prevalent. Other conditions like Glaucoma (11.1%) and Cataract (4.6%) have moderate representation, while rarer diseases like Pathological Myopia (3.6%) and Other abnormalities (2.0%) require careful handling to prevent model bias.

Observation and Insights

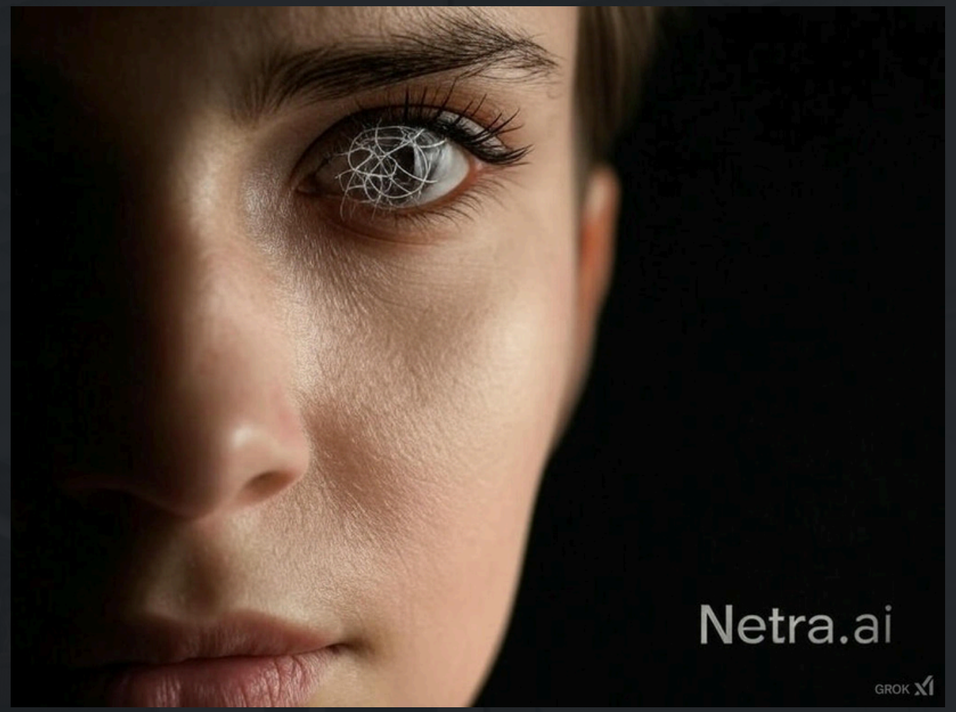
- Dataset includes 6,392 complete ophthalmology patient records.
- Mean patient age ~60 years; females slightly older than males across diagnoses.
- 44.9% normal examinations; diabetes (25.2%) and glaucoma (11.1%) most common conditions.
- Age distribution peaks between 55-70 years.



Deep Learning Approach

Deep Learning Goals

Accurate multi-class eye disease classification



Solution

Advanced CNN architecture with ensemble approach

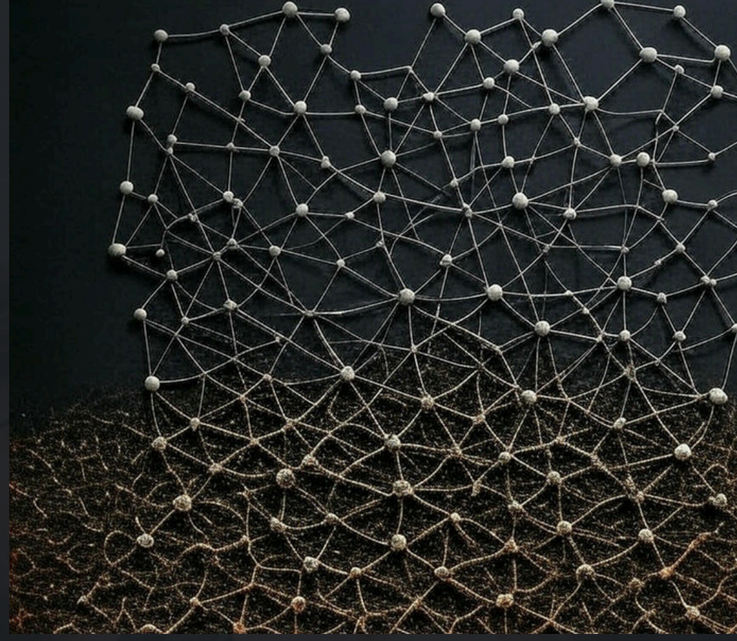
Approach

Started with single pre trained model implementation, discovered benefits of model fusion

Training Methodology & Implementation

Model Training Process

- Dataset split (training/test)
- Basic Tensorflow/Keras pipeline setup
- Number of epochs run
- Fine Tuning



Experimentation

Tested various pre-trained CNN architectures for improved accuracy

Experimentation

A	B	C	D	E	F	G	H
Encoder		Neural Network Architectur	Feature Aggregation	Learning rate	Loss Function	Activation Function	Accuracy
One Hot	ResNet150	ResNet152	Flatten	0.001	Categorial Cross Entropy	Softmax	86%
Label	EfficientNetB4		Global Average Pooling 2d	0.001	Categorial Cross Entropy	Softmax	83%
Label	VGG 16	VGG 19	Global Average Pooling 2d		Categorial Cross Entropy	Softmax	
Label	EfficientNetB4	EfficientNetB6	Global Average Pooling 2d	0.0001	Categorial Cross Entropy	Softmax	85%
Label	IncpetionV3		Global Average Pooling 2d	0.0001	Categorial Cross Entropy	Softmax	85%
Label	IncpetionV3		Global Average Pooling 2d	0.0001	Binary Cross Entropy	Sigmoid	77%

ResNet Architecture Implementation

- Why ResNets

Skip connections address vanishing gradient problem crucial for detailed medical imaging.

- Technical Implementation:

1. ResNet-50: Captures fine-grained retinal features
2. ResNet-152: Identifies broader structural patterns

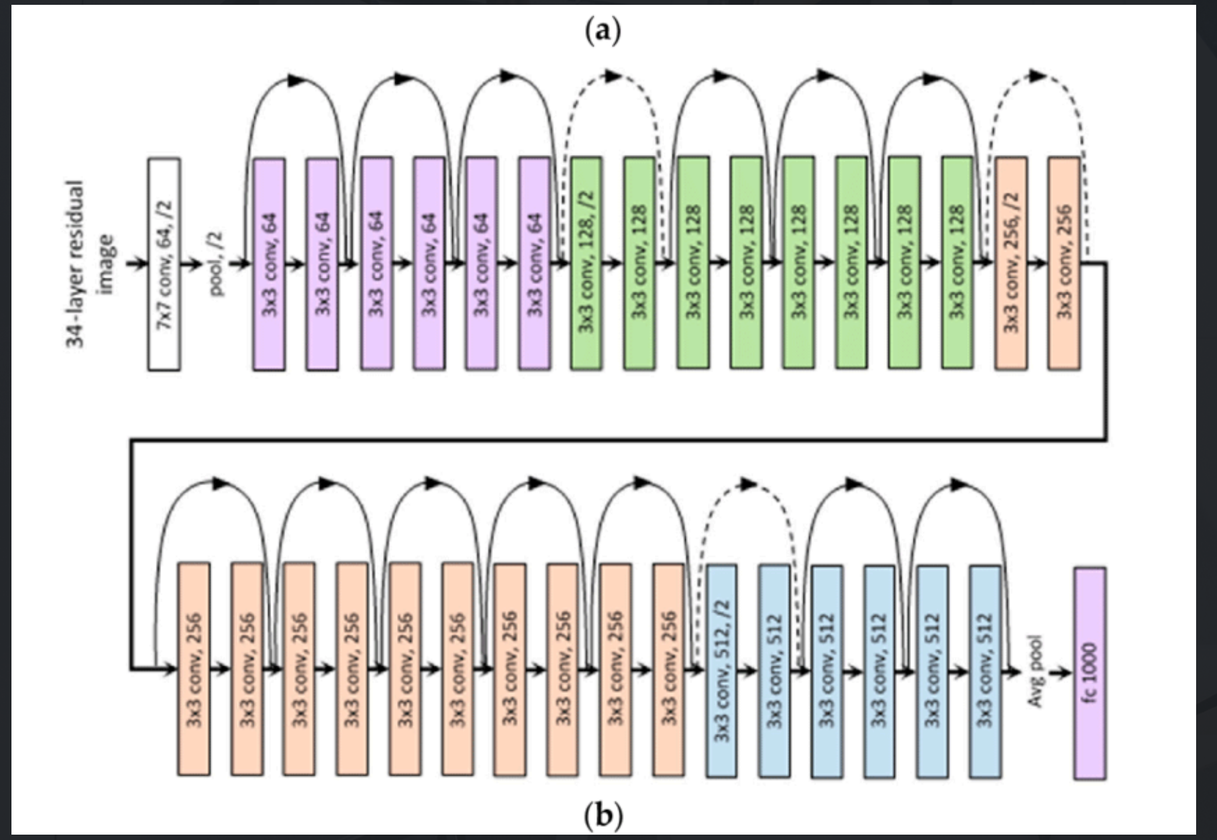


Fig: ResNet Architecture

Performance Results

Precision: 87.06%

Accuracy: 85.76%

Loss: 26.91%

Recall: 84.71%

Key breakthrough: How concatenation improved performance vs. single models

Data Preprocessing in Model

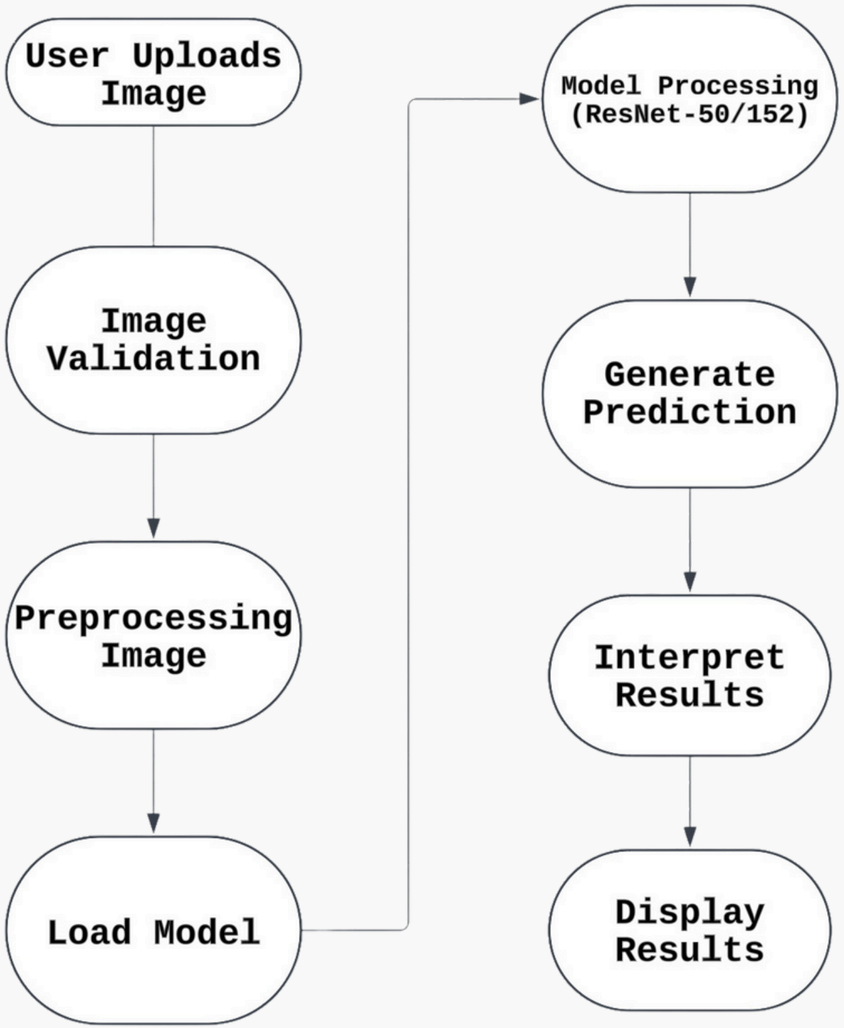
- **User Uploads an Image** — The user selects and uploads a fundus photograph using st.file_uploader(). (JPG, PNG, BMP)
- **Image is Read & Displayed** — Image is Loaded & Displayed using PIL and st.image()
- **Preprocessing** — Converted to RGB, resized (224x224), normalized, and reshaped for the model.

Data Preprocessing in Model

Image Processing & Prediction

- **Load Model** — Load pre-trained ResNet-50/152 and display error if loading fails.
- **Model Processing** — Extract image features through convolutional layers and use residual blocks for better classification.
- **Generate Prediction** — Output probabilities for eye conditions. Picks the highest probability as the result.
- **Interpret & Display** — Map prediction to disease name. Show confidence score and detailed probabilities.

Disease Prediction Workflow



References

- Deep Residual Learning for Image Recognition
  https://arxiv.org/pdf/1512.03385

- GeeksforGeeks
  https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/

- Z. Feng (2022) “An Overview of ResNet Architecture and Its Variants”


Thank
You