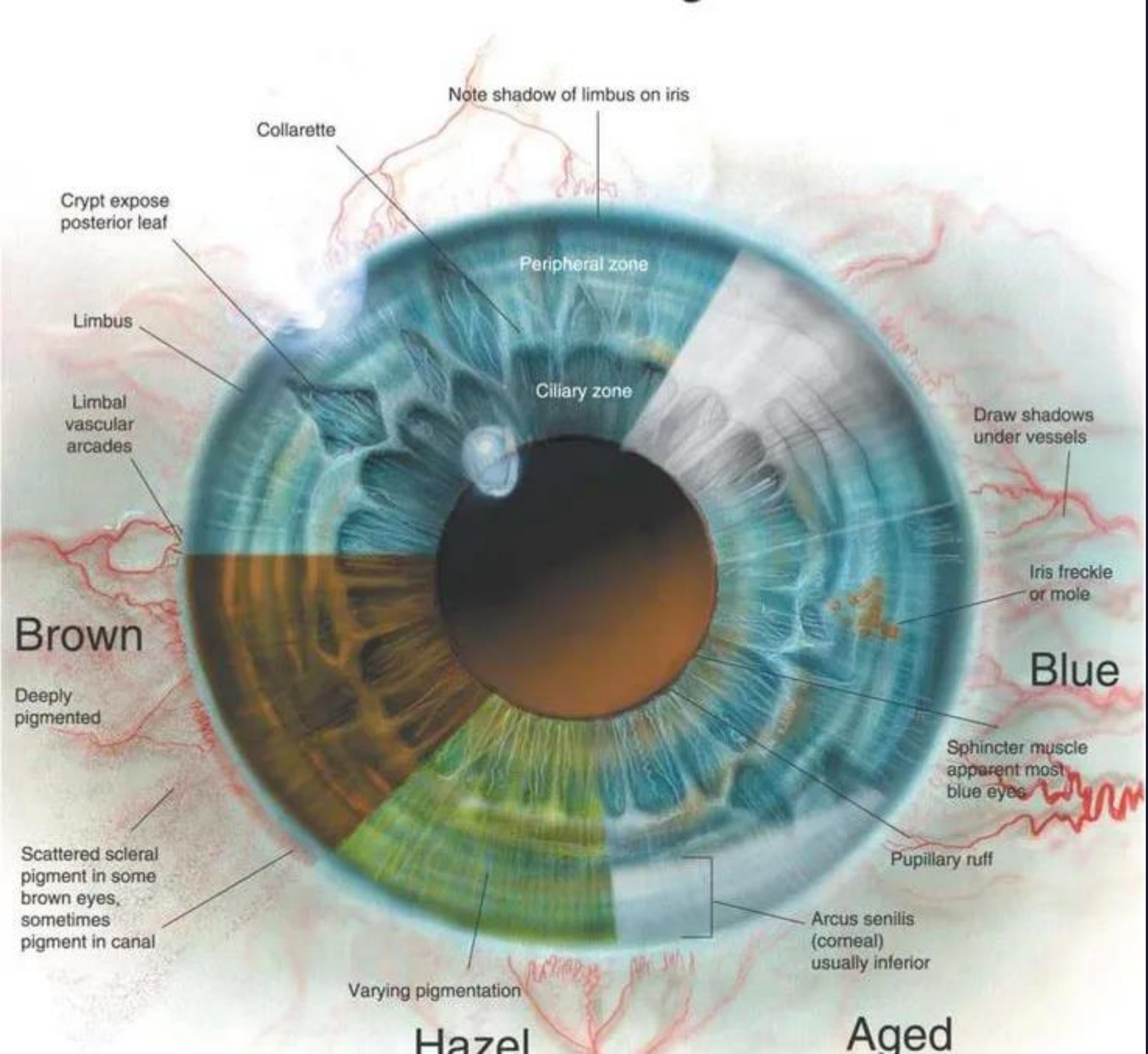


DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINNERING

IRIS DETECTION AND RECOGNITION

EEL 6825

Presenter By:

Amit Kumar

Introduction

Iris recognition is considered one of the most robust identification techniques due to the unique pattern of the iris for each individual.

In this project, state-of-the-art detection and recognition models are employed for iris detection and recognition.

The aim of this project is to identify and recognize individuals by learning the unique features of their iris and comparing them with others.

Motivation

- Security and Privacy is one of the major concerns nowadays.
- With the growing technology it getting harder and harder to make sure of our privacy and security.
- To tackle this problem Biometric Identification is one of the most robust ways to ensure your privacy.
- Features like Fingerprint and Iris recognition are one of the most trusted biometric identification techniques because of there distinctive feature.
- Iris recognition is considered more accurate compared to fingerprint recognition because of its harder to implement

Existing Approach

Existing approaches to iris recognition have primarily relied on handcrafted feature extraction and matching techniques.

These approaches require prior knowledge of iris anatomy and are time-consuming to develop. Additionally, they may not perform well when the images are captured under different lighting and angle conditions.

With the advent of deep learning techniques, it is now possible to train end-to-end models that can learn features directly from the raw images.

This approach has shown promising results in improving the accuracy and robustness of iris recognition systems.

Dataset

Object Detection dataset:

- Roboflow was used to obtain the dataset with Iris images.
- The dataset was labelled using bounding box feature of Roboflow.
- Augmentation were introduced to the dataset using Roboflow’s dataset preparation feature.

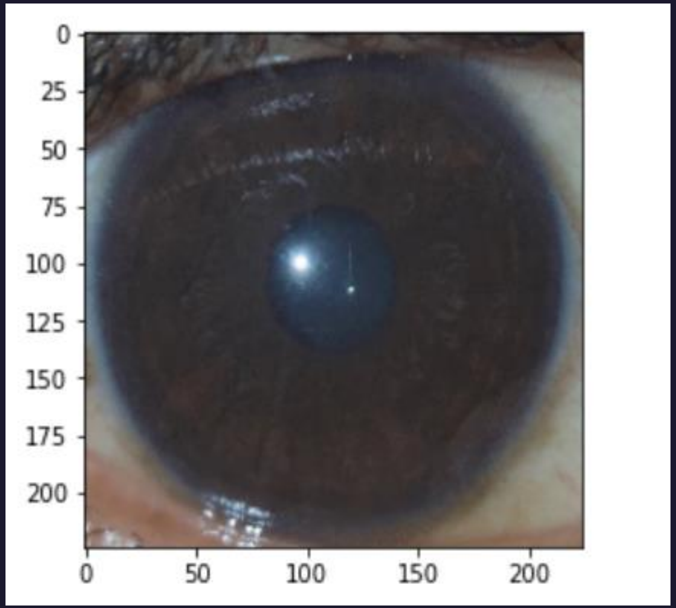


Figure 2: Shows the cropped and enhanced image for recognition.

Recognition Dataset:

- The dataset was obtained from video of Iris frame by frame using a mobile phone for Anchor and Positive.
- The dataset for Negative was taken from Object detection dataset.
- The dataset is then cropped and enhanced using OpenCV library using the coordinates form detection model.

Figure 1: Shows a labelled image on Roboflow using bounding box.

YOLOv8

- YOlOv8 is most recent model from ultralytics in the series of YOLO models. YOLO models are widely used for detection task due to there high accuracy and speed.
- YOLOv8 can detect multiple objects in a single image and provide their location and class probability scores.
- YOLO (You Only Look Once) is a real-time object detection algorithm that works by dividing an input image into a grid of cells and predicting bounding boxes and class probabilities for each cell.

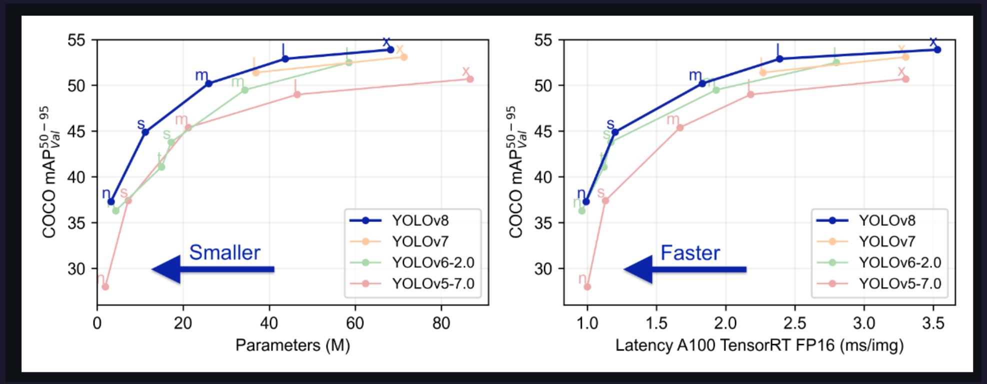


Figure 3: Shows performance of YOLO models

Figure 4: Shows grid structure similar to YOLO

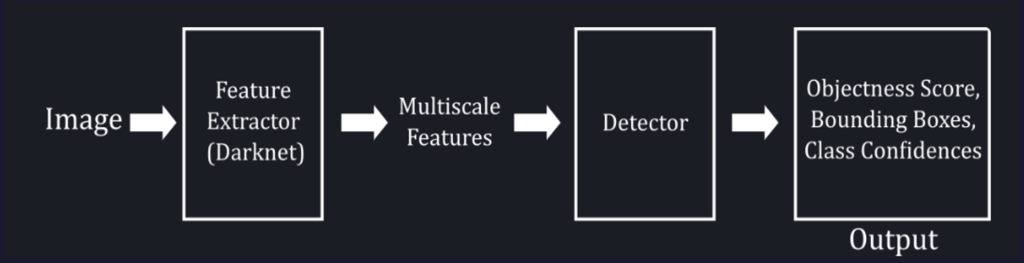


Figure 5: General Architectue of YOLOv8

Dataset preprocessing:

For the recognition dataset

- Utilizing OpenCV the images were cropped using the Coordinates obtained form detection library.
- Enchantments like Brightness, Noise reduction and sharpening were used on the images.

```python
alpha = 1.5
# control brightness by 50
beta = 50
image2 = cv2.convertScaleAbs(cropped_image, alpha=alpha, beta=beta)

# Apply median blur
ksize = 5  # kernel size, should be an odd number
image_blur = cv2.medianBlur(image2, ksize)

# Create the sharpening kernel
kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
```

Figure 6: Shows the code used for enchancement

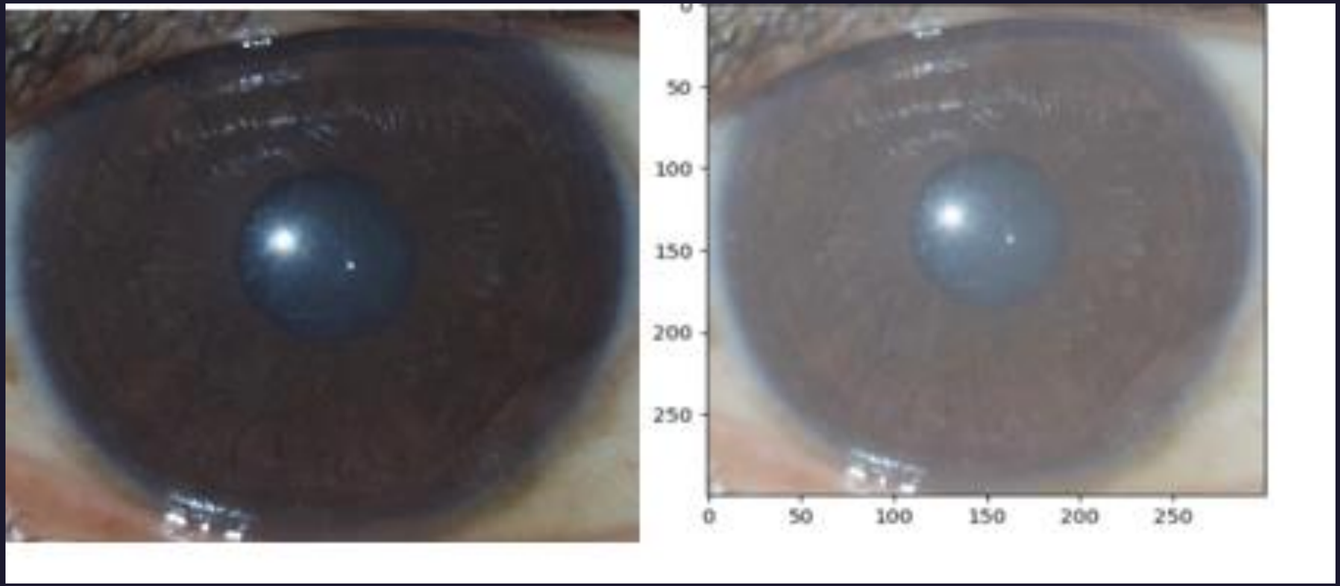


Figure 7: Shows Image before and after enhancement

Siamese Network

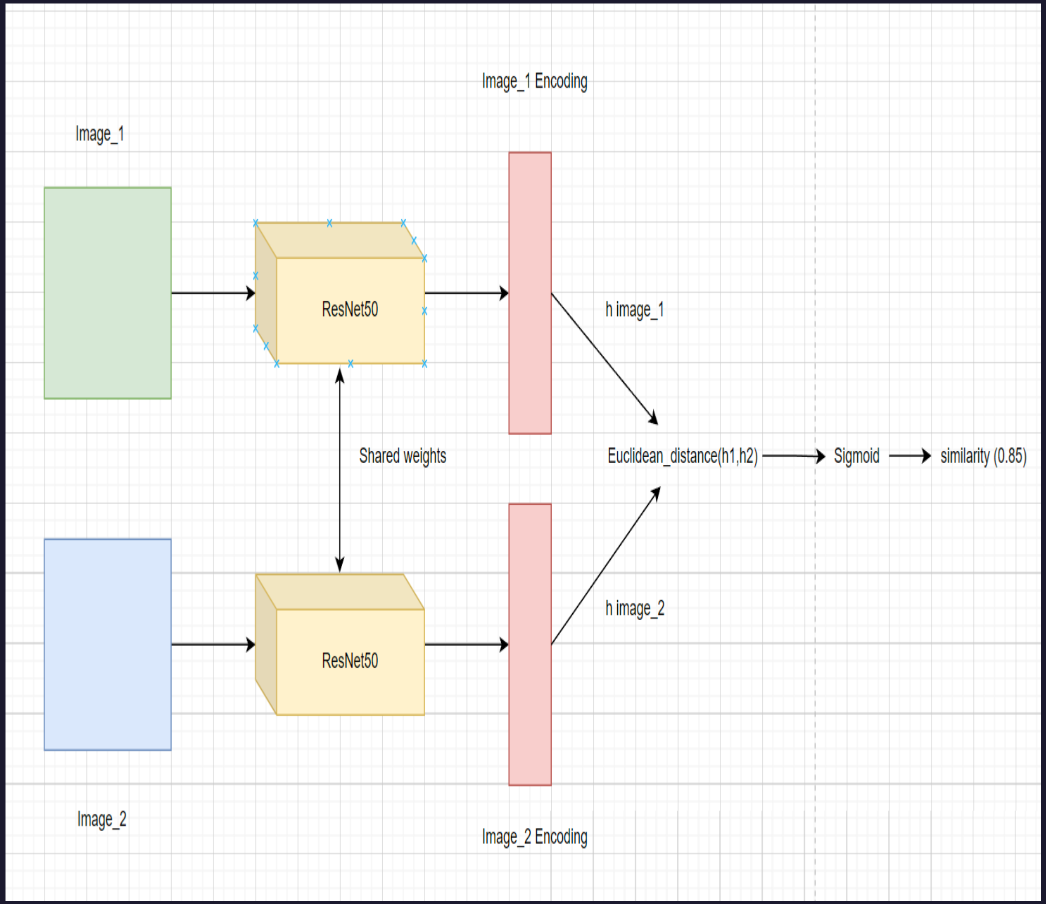


Figure 8: Shows the architecture of Siamese network

Siamese Networks are a type of neural network architecture that learn similarity between two inputs.

The network consists of two identical subnetworks that share the same weights and architecture. The two subnetworks take in two separate inputs and calculate their feature vectors.

The distance or similarity between the feature vectors is then calculated to determine if the inputs are similar or dissimilar.

This project used a Siamese Network to learn the unique features of the iris and compare them to other irises for recognition task.

ResNet50 was used as embedding architecture.

Methodology

Iris Detection

The YOLOv8 model is trained

- Epochs =100, Batch size =16, Workers =8,Optimizer=SGD

Dataset Enhancement

Input Image is cropped	Image Enhancement

Iris Recognition

Resnet50 model as Embedding	Siamese model is Trained • Epochs =500, workers = 8, Optimizer = Adam(lr= 1e-4)

Results

Precision, recall	Similarity

Performance Evaluation:

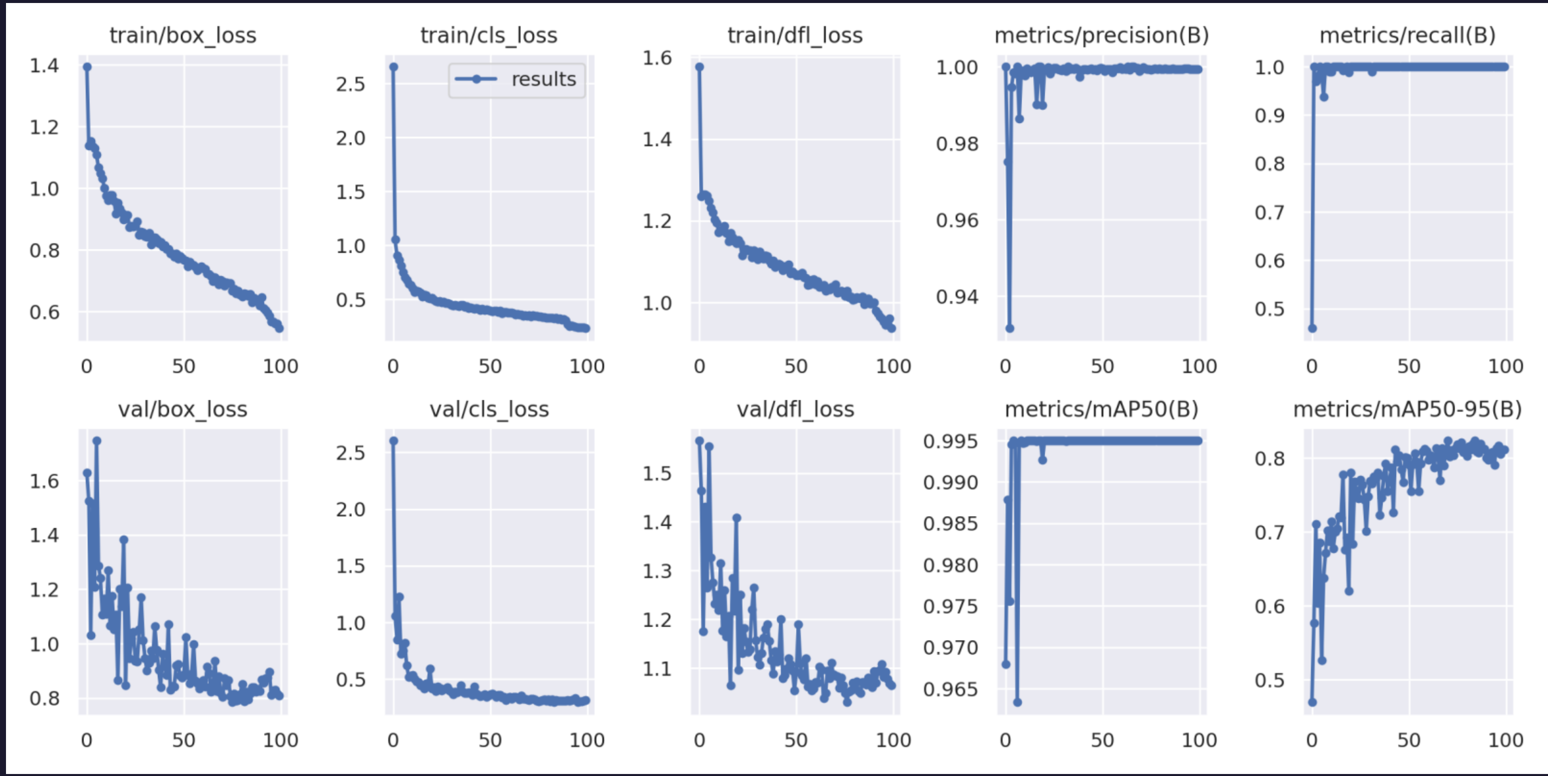


Figure 9: Shows the graph of different measure while training the YOLOv8 Model

Performance Evaluation:

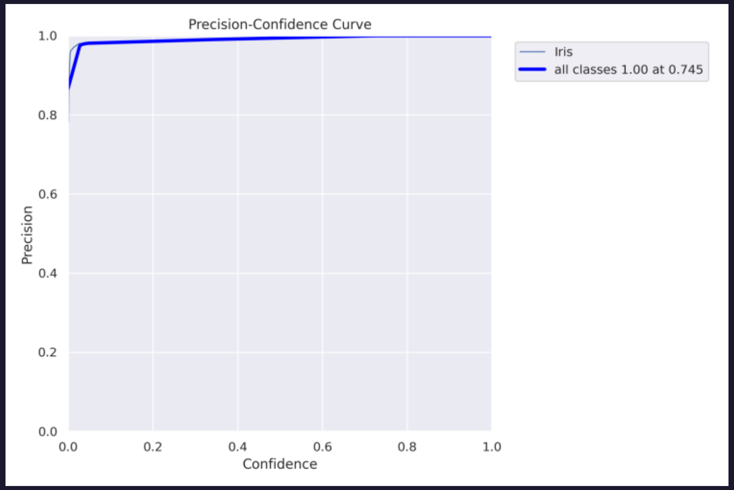
Figure 10: Shows Graph of performance of YOLOv8 model while training.

(a) Precision-Confidence Curve

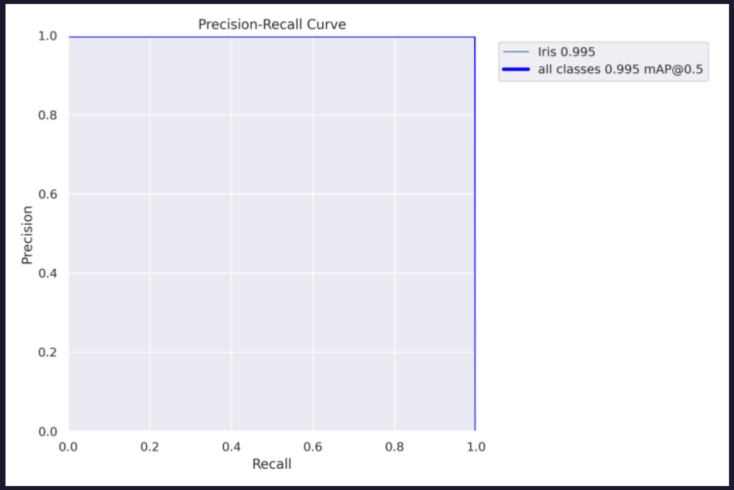
(b) Precision-Recall Curve

(c) F1-Confidence Curve

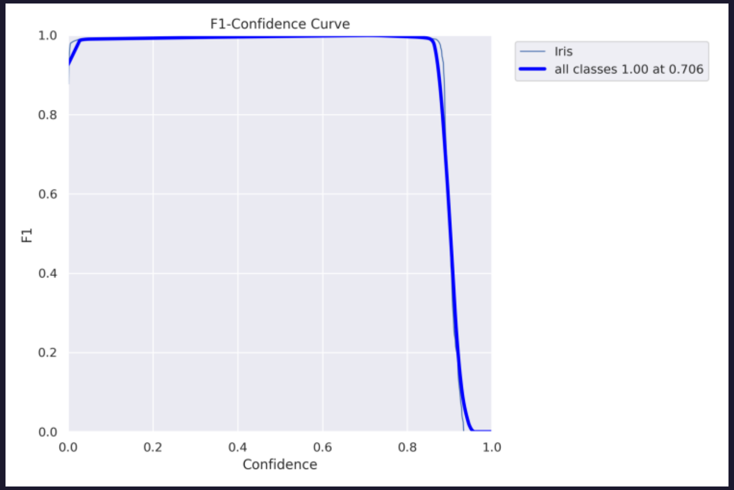
(d) Recall-Confidence Curve



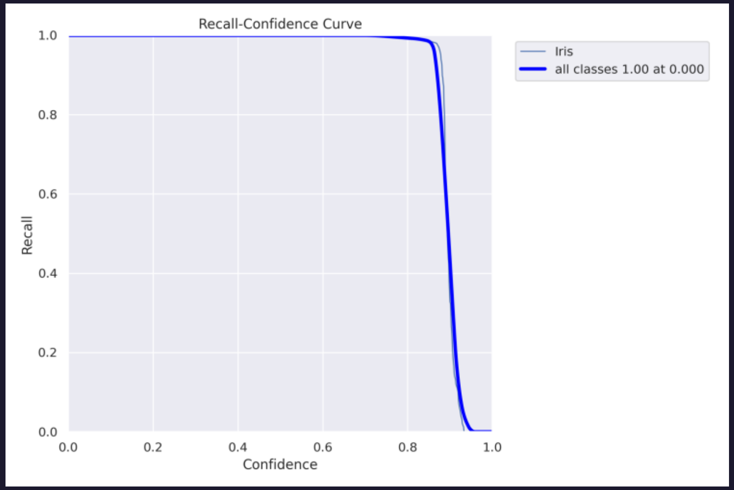
a



b



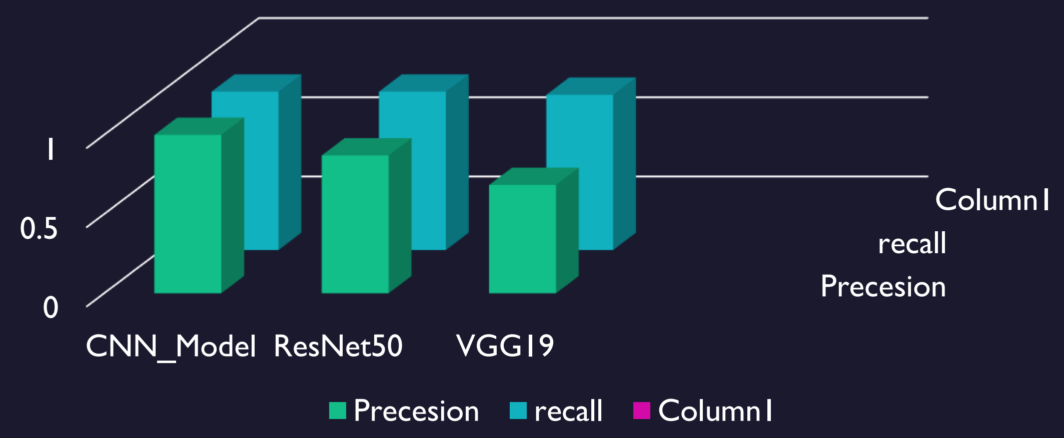
c



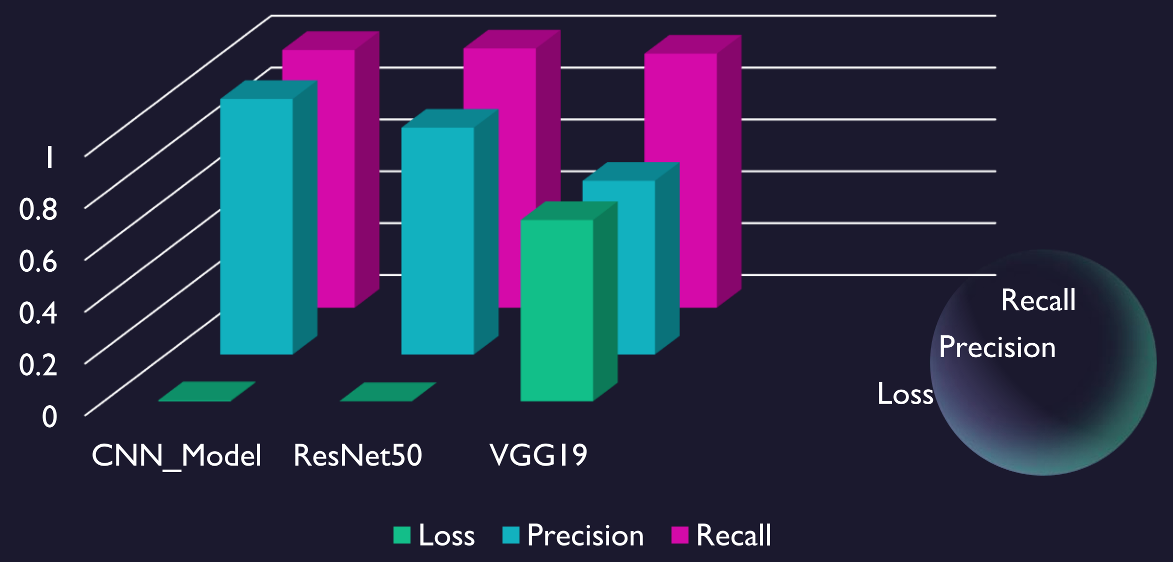
d

Performance Evaluation:

Graph 1: Performance of different CNN Architecture as Embedding to Siamese Network while Validation



Graph 2: Performance of different CNN Architecture as Embedding to Siamese Network while Training



Results:

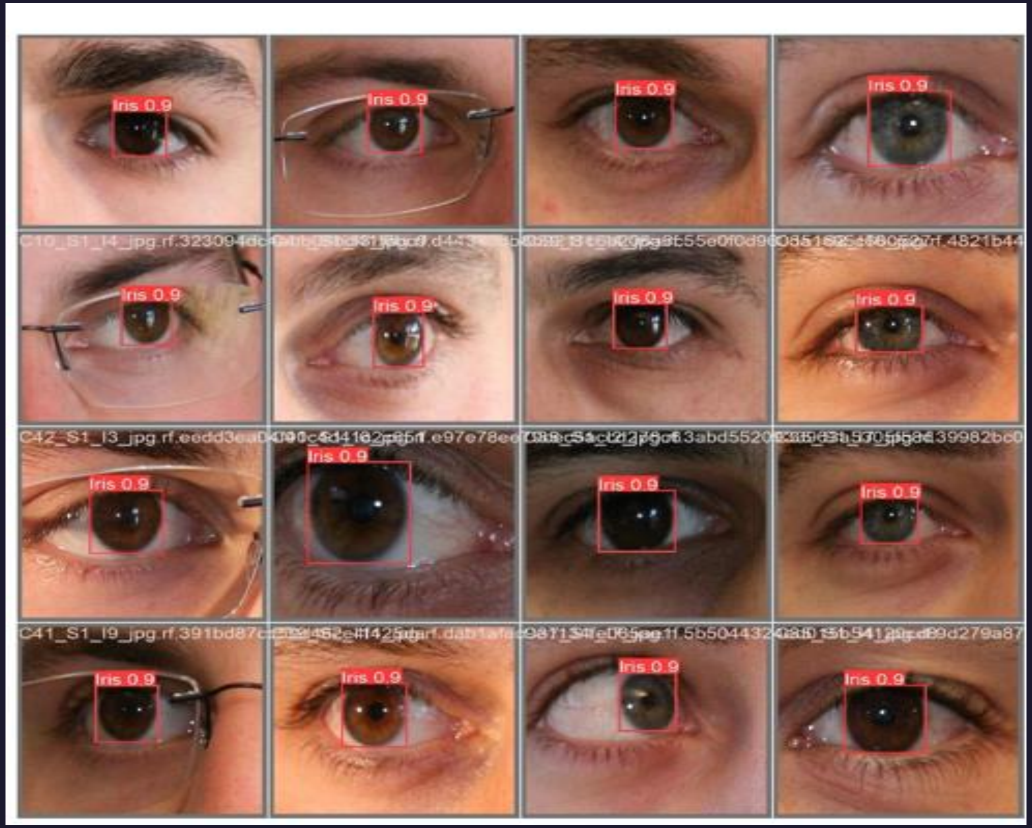


Figure 11: Shows the output of the Detection model

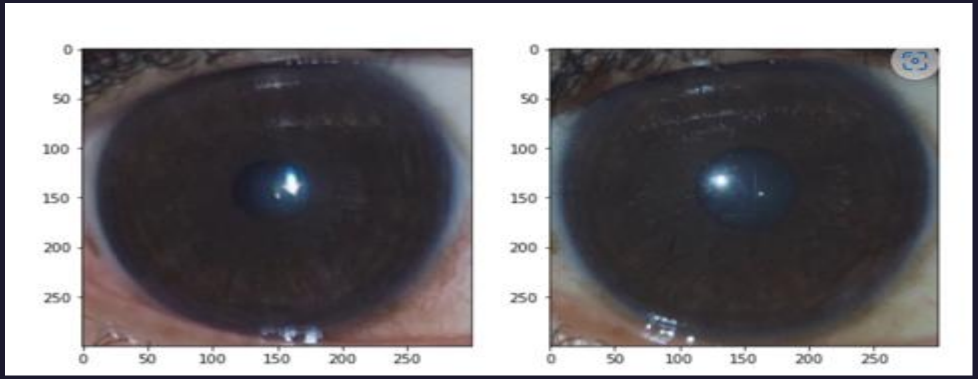


Figure 12: Shows the output when Input Iris image matches the anchor image

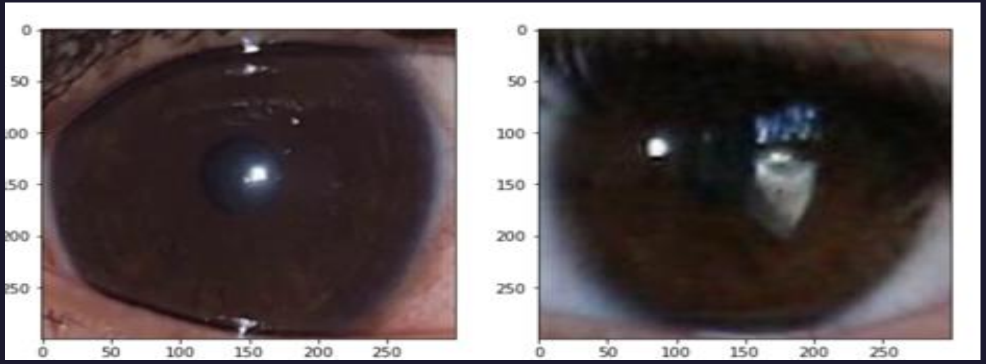


Figure 13: Shows the output when Input Iris image does not match with anchor image.

Conclusion

- The Performance of YOLOv8 model was very good in detecting the Iris as evident form the performance measures in figure 9 and 10

- The Siamese Model performance was also good the CNN model form the original paper performed best.

- ResNet50 model also performed good and was also easier to train and was less expensive compared to Original CNN model and VGG19 model. As Evident form Graph1 and Graph 2

Future work:

Incorporate the detection and image enhancement in the recognition model so that we don't need to create a separate file for final detection data.

Challenges:

Original CNN Model and Efficientnet Model were hard to train, and the Jupyter kernel kept breaking while training these models.

Thank You

Amit Kuma

Kumar.amit1@ufl.edu