Project Report

On

FEATURE BASED IMAGE RETRIEVAL

Submitted by

ID NO: R170227, N.Sai Pradeepa,

ID NO: R171142, S.Jyosthna

Under the guidance of

RATNAKUMARI CHALLA

Assistant professor of computer science engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

R.K.Valley, Kadapa, Andhra Pradesh

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

R.K.Valley, Kadapa(Dist), Andhra Pradesh, 516330.

CERTIFICATE

This is to certify that the project work titled "**FEATURE BASED IMAGE RETRIEVAL**" is a bonafide project work submitted by **S.JYOSTHNA, N.SAI PRADEEPA** in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfillment of requirements for the award of degree of Bachelor of Technology in Computer science and engineering for the year 2021-2022 carried out the work under the supervision

GUIDE

RATNAKUMARI CHALLA

HEAD OF THE DEPARTMENT

P HARINADHA

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.We are extremely grateful to our respected Director,Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.We also express my sincere gratitude to our respected Head of the Department Mr. P HARINADHA for his encouragement, overall guidance in viewing this project as a good asset and effort in bringing out this project.We would like to convey thanks to our guide at college Ms.RATNA KUMARI CHALLA for her guidance, encouragement, cooperation and kindness during the entire duration of the course and academics.Our sincere thanks to all the members who helped us directly and indirectly in the completion of project work. We express our profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

In this work, an afford is made to detect and identify the persons or suspects faces for police investigation purpose. In this, we are mainly focused in detecting the persons face with reasonably good accuracy. Firstly we input image from the camera to detect and it returns a output with border line around persons face with respective name under the border box and in addition to that the accuracy is also displayed.

INTRODUCTION

What is Object Detection?

Object detection is a supervised machine learning problem, which means you must train your models on labelled examples. Each image in the training dataset must be accompanied with a file that includes the boundaries and classes of the objects it contains. There are several open-source tools that create object detection annotations.

The object detection network is trained on the annotated data until it can find regions in images that correspond to each kind of object.

In this, we use this to detect faces from trained images for "control rooms"

It is widely used in computer vision tasks such as image annotation, vehicle counting, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person in a video.

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e the centre) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin colour and distance between eyes can be found.

APPLICATIONS

- Object detection in retail
- Autonomous Driving.
- Animal detection in Agriculture.
- People detection in Security

- Vehicle detection with AI in Transportation

PURPOSE

The main objective of object detection is to identify and locate one or more effective targets from image or video data. It comprehensively includes a variety of important techniques, such as image processing, pattern recognition, artificial intelligence and machine learning.

The main purpose of this is to identify the given video data from already trained model. It is used to retain the images from the database.

CONTRIBUTION

In this we used Custom Object detection using yolo algorithm to provide the information from already trained images. We have trained data set of three different persons and stored information of configuration and their weights in files. We allow our camera to record the image and we use code to compare with the images from the data set and return their name with the reasonable accuracy.

LITERATURE

Why build such a system?

Object detection is the identification of an object in the image along with its localisation and classification. It has wide spread applications and is a critical component for vision based software systems. This paper seeks to perform a rigorous survey of modern object detection algorithms that use deep learning. As part of the survey, the topics explored include various algorithms, quality metrics, speed/size trade offs and training methodologies. This paper focuses on the two types of object detection algorithms- the SSD class of single step

detectors and the Faster R-CNN class of two step detectors. Techniques to construct detectors that are portable and fast on low powered devices are also addressed by exploring new lightweight convolutional base architectures. Ultimately, a rigorous review of the strengths and weaknesses of each detector leads us to the present state of the art.

In various fields, there is a necessity to detect the target object and also track them effectively while handling occlusions and other included complexities. Many researchers attempted for various approaches in object tracking. The nature of the techniques largely depends on the application domain.

PRELIMINARIES

YOLO ALGORITHM

YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.

YOLO is an algorithm used for 2 types of detection

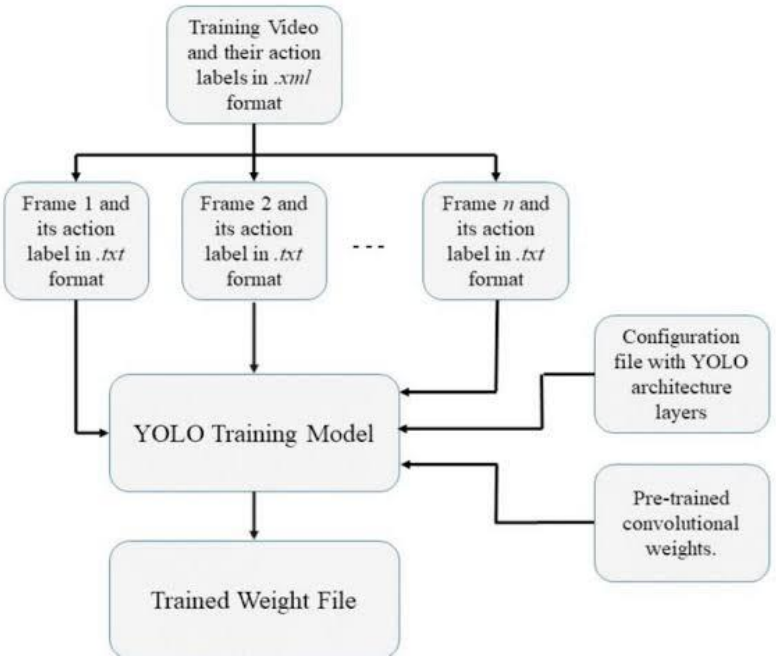
1.object detection

2.face detection

In this project we use object detection yolo algorithm to process and return the image with its name under the border and with the reasonable accuracy.

Applications of YOLO Algorithm:

1. Autonomous driving
2. Wildlife
3. Security



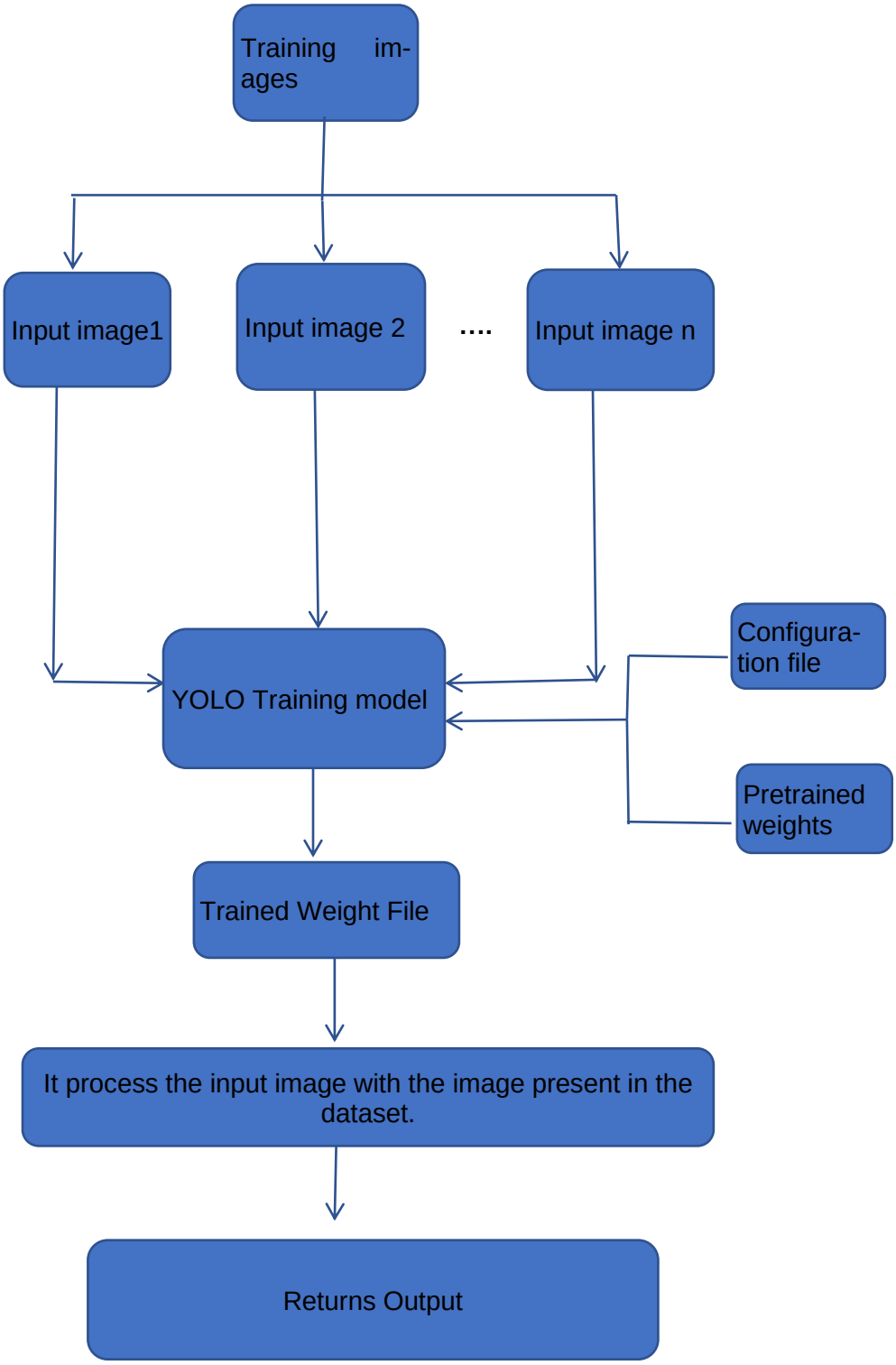
WORKING MODEL

It contains three steps

Step-1: To train the Data set.

Step-2: To Activate environment using Anaconda prompt.

Step3: To run the code for the output.


Training images
Input image1
Input image 2
....
Input image n
Configuration file
YOLO Training model
Pretrained weights
Trained Weight File
It process the input image with the image present in the dataset.
Returns Output

In Step1 we have to collect the data set that is nothing but the images from Chrome. Then we installed Labelling tool for recognizing and resizing the downloaded images for the data set. We run a simple code to rename all the resized images and use for the calculation of the weights.

Filename: rename.py

```
import os
count=0
for i in os.listdir();
      os.rename(i,str(count)+ '.'+ i.split('.')[-1])
      count+=1
```

In Step2 we have to activate environment to run our code in Anaconda prompt,for that purpose we have to use commands those are

>>conda env list

base C:\Users\Pradeepa\anaconda3

tf C:\Users\Pradeepa\anaconda3\envs\tf

torch C:\Users\Pradeepa\anaconda3\envs\torch

>>activate tf //For activating tf environment

>jupyter notebook

In Step3 we upload files in jupyter notebook which are related to dataset and we run the each and every piece of code in jupyter notebook.After running all commands the output of this code will redirect to open a camera and we have to show the object or a face of a person for detection it will detect the person's name or object's name in which we have trained already.

Filename: Videocapture.py

```
import cv2
import numpy as np


cap = cv2.VideoCapture(0)

while True:
    _, img = cap.read()
    img = cv2.resize(img,(1280,720)
    cv2.imshow('img',img)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

IMPLEMENTATION AND RESULTS

In this work we are mainly focused in detecting the persons face with reasonable accuracy .YOLO is a popular algorithm for face detection due to its speed and accuracy. It is faster than other algorithms because of its simple architecture.

The implementation of face detection system using yolo is demonstrated and analysed to study the various features regarding detection of the faces.

OpenCV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

In this, we use OpenCV to open the camera for capturing the image from primary camera of our system

Matplotlib:

.

Matplotlib is a cross platform, data visualization and graphical plotting library for python and it's numerical extension numpy. As such, it offers a viable open source alternative to MATLAB. Developers can also used matplotlib's API's to embed plots in GUI applications.

By Using this project and based on experimental results we are able to detect object more precisely and identify objects individually with exact location of an object in the picture in x, y axis

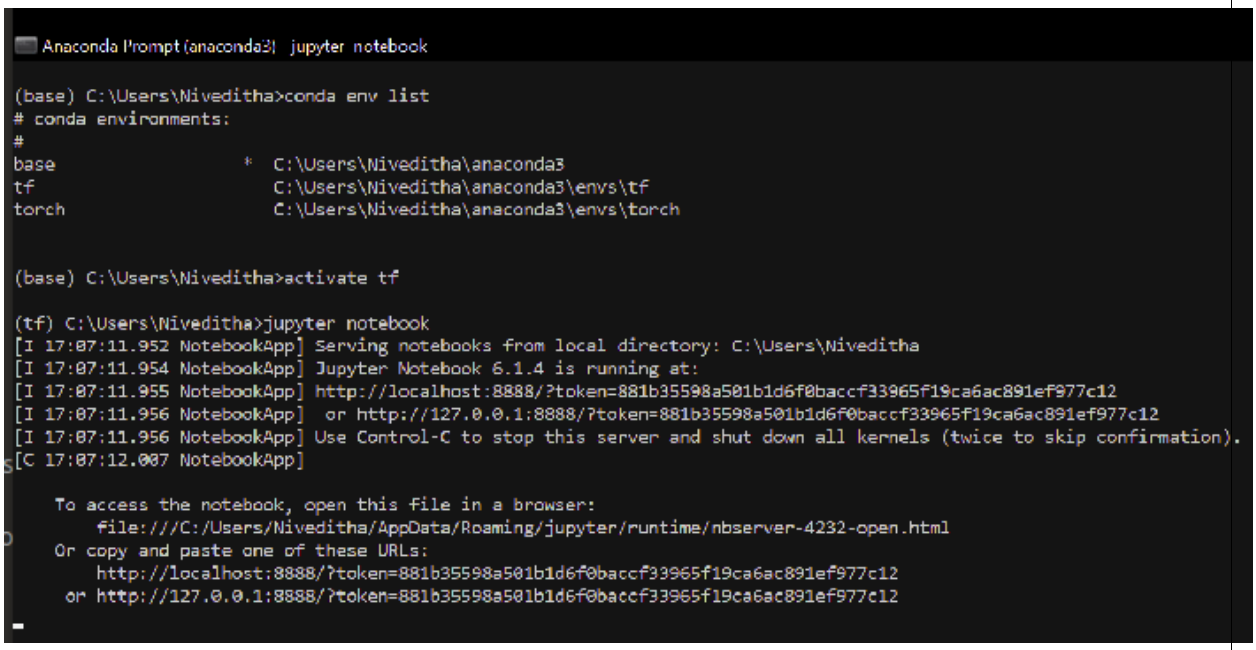
Steps to run the software

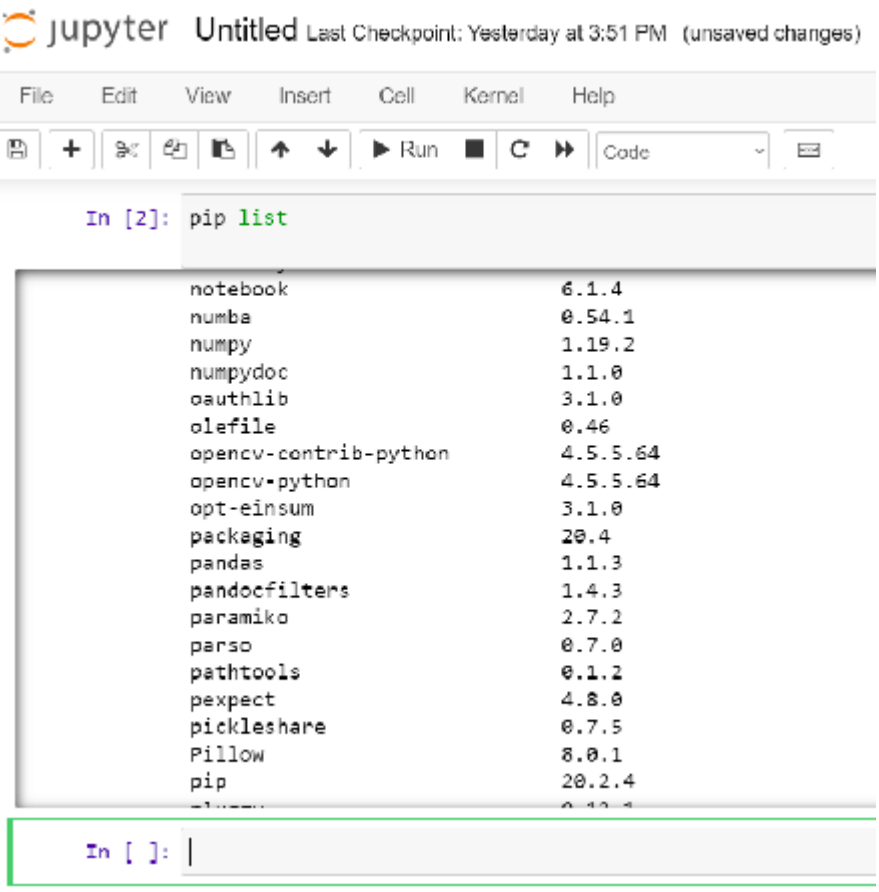
1.Activate jupyter notebook from anaconda prompt

2.Install the necessary packages

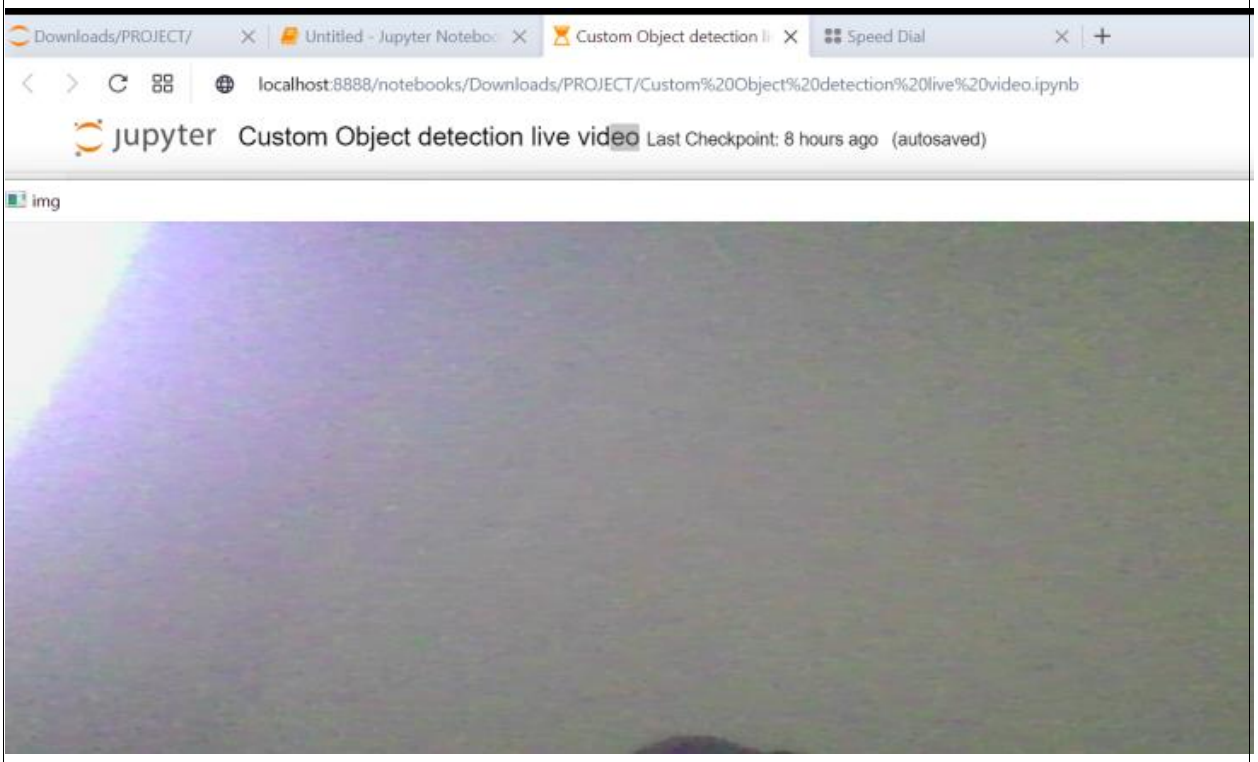
3.Upload and open the necessary files in jupyter notebook.

4.Click on the run button and camera will automatically open as output.


Anaconda Prompt (anaconda3) - jupyter notebook
(base) C:\Users\Niveditha>conda env list
conda environments:
#
base * C:\Users\Niveditha\anaconda3
tf C:\Users\Niveditha\anaconda3\envs\tf
torch C:\Users\Niveditha\anaconda3\envs\torch
(base) C:\Users\Niveditha>activate tf
(tf) C:\Users\Niveditha>jupyter notebook
[I 17:07:11.952 NotebookApp] Serving notebooks from local directory: C:\Users\Niveditha
[I 17:07:11.954 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 17:07:11.955 NotebookApp] http://localhost:8888/?token=881b35598a501b1d6f0baccf33965f19ca6ac891ef977c12
[I 17:07:11.956 NotebookApp] or http://127.0.0.1:8888/?token=881b35598a501b1d6f0baccf33965f19ca6ac891ef977c12
[I 17:07:11.956 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 17:07:12.007 NotebookApp]
To access the notebook, open this file in a browser:
file:///C:/Users/Niveditha/AppData/Roaming/jupyter/runtime/nbserver-4232-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=881b35598a501b1d6f0baccf33965f19ca6ac891ef977c12
or http://127.0.0.1:8888/?token=881b35598a501b1d6f0baccf33965f19ca6ac891ef977c12



After running the code it opens camera for taking object photo or person's photo as an input and displays the name if the object is detected.



1.After opening of camera

2.Provide the input to the camera by providing a person's photo.

3.When the trained persons are given as input it detects the person's face and displays name of a particular person.

Figure 1:

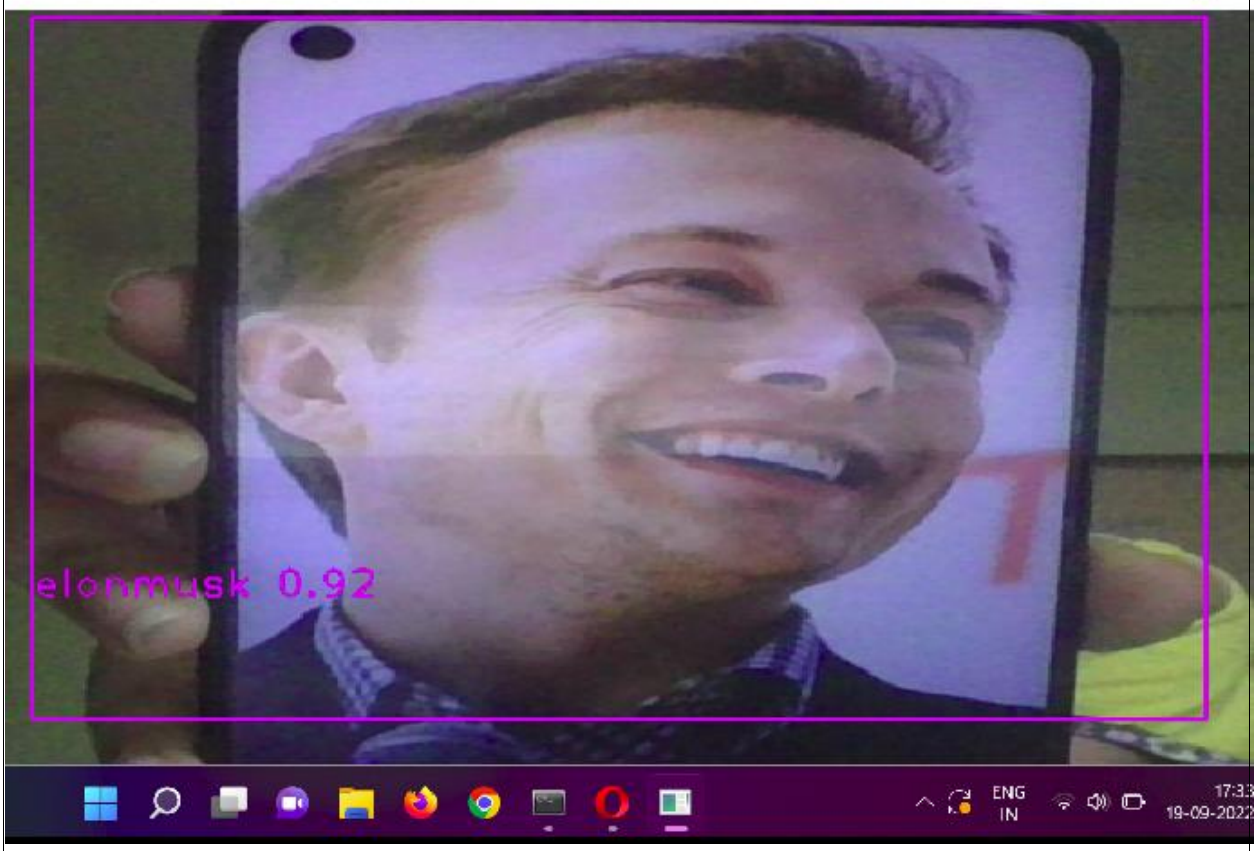


Figure 2:

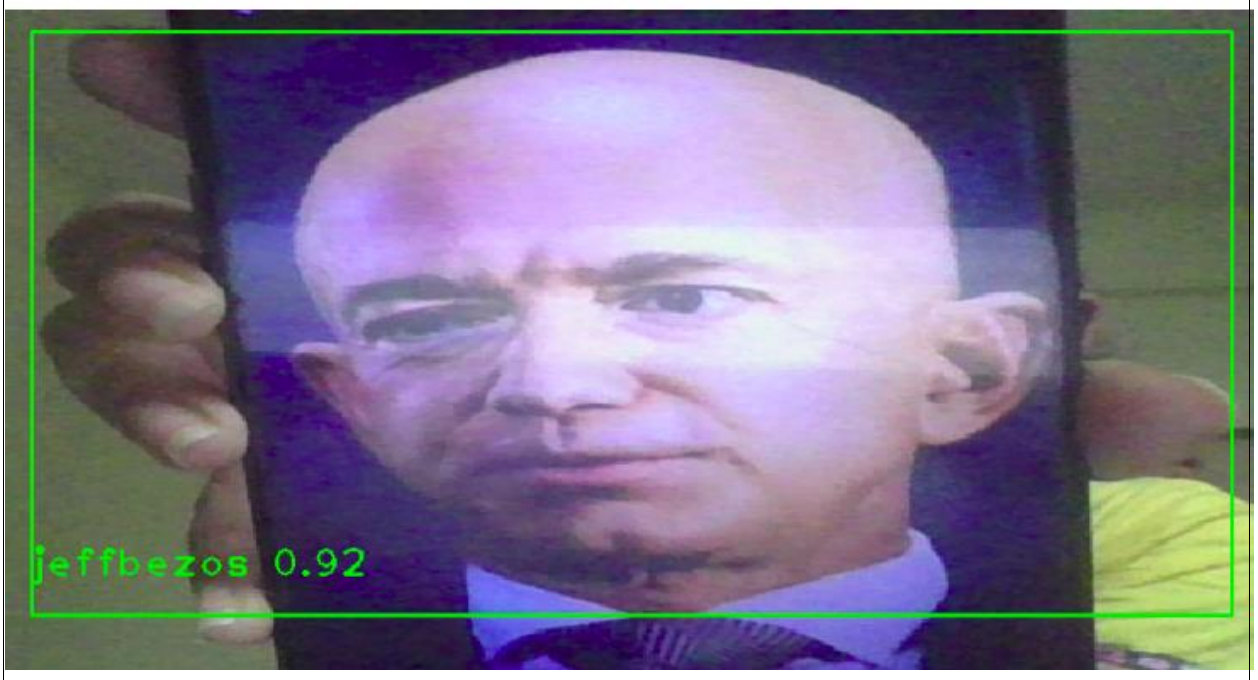


Figure 3:

When the unidentified person or untrained object is given as input then it displays nothing.

FUNCTIONAL TESTING

S. No	Test Cases	Input	Expected Result	Actual Result	Status	Remarks
1.	Enter valid username and password.	Username: Password:	Redirect to next page	Opened next page	Pass	No remarks
2.	Video capture is working or not.	Should enable camera.	Should open camera.	As expected opened camera for recording image.	Pass	No remarks
3.	Displaying boundary box or not.	Show an image to the camera.	It displays boundary box if it is in dataset.	As expected it displays boundary box.	Pass	No remarks
4.	Show valid image from the dataset	Show a valid image to the camera.	It should recognize and display their name.	As expected it displays their name	Pass	No remarks
5.	While showing an image it should display with good accuracy.	Show a trained image to the camera.	It should display accuracy by side of the image name	As expected it displays accuracy	Pass	No remarks
6.	Show an invalid image	Show an in-valid image to the cam-era	It should display nothing.	As expected it does not display anything	Pass	No remarks

CONCLUSION

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. It should be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical

By using this project and based on experimental results we are able to detect object more precisely and identify the objects individually with exact location of an object in the picture in x, y axis. This paper also provide experimental results on different methods for object detection and identification and compares each method for their efficiencies.

REFERENCES

For renaming files

https://github.com/jakkcoder/training_yolo_custom_object_detection_files/blob/main/Rename_files.ipynb

For open cv

https://docs.opencv.org/4.x/d5/de5/tutorial_py_setup_in_windows.html

For darknet

https://github.com/pjreddie/darknet

For label img tool

https://tzutalin.github.io/labelImg/

Other references

https://github.com/jakkcoder/training_yolo_custom_object_detection_files

For Imageye tool

https://bit.ly/3nXfPI8

---THANK YOU---