Project Phase IV Report

On

Flex Lens

Submitted for the requirement of

Project course

BACHELOR OF TECHNOLOGY

ELECTRONICS & BIOMEDICAL ENGINEERING

Submitted to:
Project Teacher:

Submitted By:

NAME:	**ROLL NO:**
Rohan Ghosh	**3180700554102**

Co Supervisor Signature

ELECTRONICS AND BIOMEDICAL ENGINEERING
GURU JAMBHESWAR UNIVERSITY OF SCIENCE AND TECHNOLOGY, HISAR, INDIA
NOVEMBER 2020

NAME: ROHAN GHOSH
ROLL NO: 3180700554102

ABSTRACT

Object detection is one of the most basic and central tasks in computer vision. Its task is to find all the interested objects in the image, and determine the category and location of the objects. Object detection is widely used and has strong practical value and research prospects. Applications include face detection, pedestrian detection and vehicle detection. In recent years, with the development of convolutional neural network, significant breakthroughs have been made in object detection. This paper describes in detail the classification of object detection algorithms based on deep learning. The algorithms are mainly divided into one-stage object detection algorithm and two-stage object algorithm, and the general data sets and performance indicators of object indicators.

Handwritten character recognition (HCR) is the detection of characters from images, documents and other sources and changes them in machine-readable shape for further processing. The accurate recognition of intricate-shaped compound handwritten characters is still a great challenge. Recent advances in convolutional neural network (CNN) have made great progress in HCR by learning discriminatory characteristics from large amounts of raw data. In this paper, CNN is implemented to recognize the characters from a test dataset. The main focus of this work is to investigate CNN capability to recognize the characters from the image dataset and the accuracy of recognition with training and testing. CNN recognizes the characters by considering the forms and contrasting the features that differentiate among characters. Our CNN implementation is experimented with the dataset NIST to obtain the accuracy of handwritten characters

TABLE OF CONTENT

USE OF MODERN TOOLS IN DESIGN AND ANALYSIS

Computer Vision is the branch of the science of computers and software systems which can recognize as well as understand images and scenes. Computer Vision is consisting of various aspects such as image recognition, object detection, image generation, image super-resolution and many more. Object detection is widely used for face detection, vehicle detection, pedestrian counting, web images, security systems and self-driving cars.

Handwriting digits and character recognitions have become increasingly important in today's digitized world due to their practical applications in various day to day activities. It can be proven by the fact that in recent years, different recognition systems have been developed or proposed to be used in different fields where high classification efficiency is needed. Systems that are used to recognize Handwriting letters, characters, and digits help people to solve more complex tasks that otherwise would be time-consuming and costly.

Software and Libraries Requirement:

- Python 3.7.2
- Anaconda navigator
- Jupyter Notebook
- Tensorflow
- Matplotlib
- Numpy

1. Anaconda navigator:

   Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

2. Jupyter Notebook:

   The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

3. Tensorflow:

   TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

4. Matplotlib:

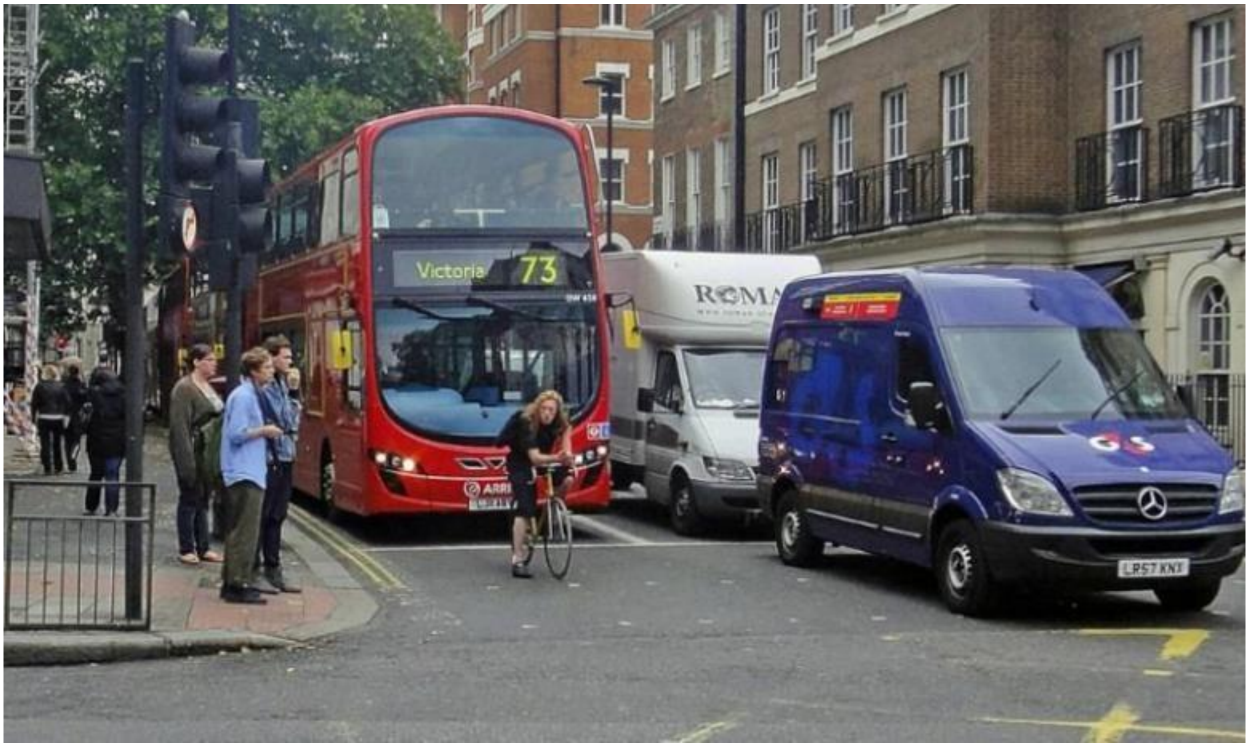
   Matplotlib is a Python programming language plotting library and its NumPy numerical mathextension. It provides an object-oriented API to use general-purpose GUI toolkits suchasTkinter,wxPython, Qt, or GTK+ to embed plots into applications.

   pip install matplotlib – command

5. Numpy:

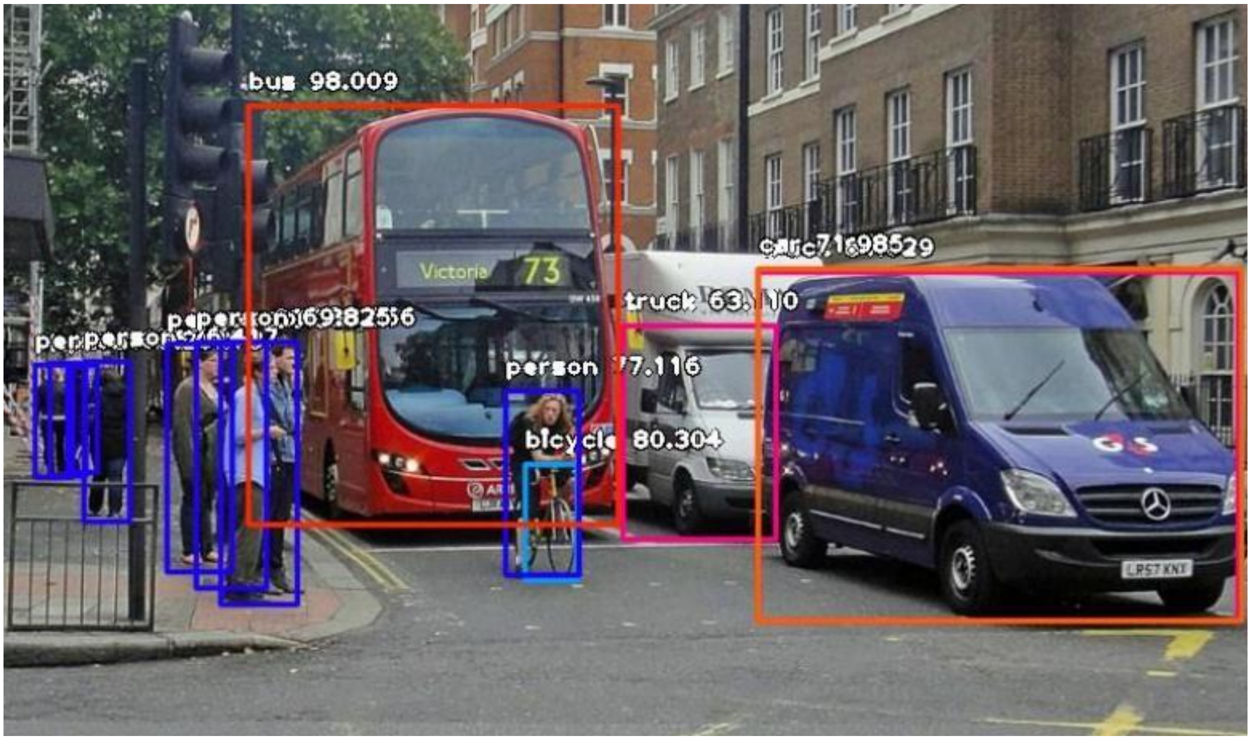
   NumPy is library of Python programming language, adding support for large, multidimensional array and matrices, along with large collection of high-level mathematical function to operate over these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several developers. In 2005 Travis Olphant created NumPy by incorporating features of computing Num array into Numeric, with extension modifications. NumPy is open-source software and has many contributors.

   pip install numpy -command



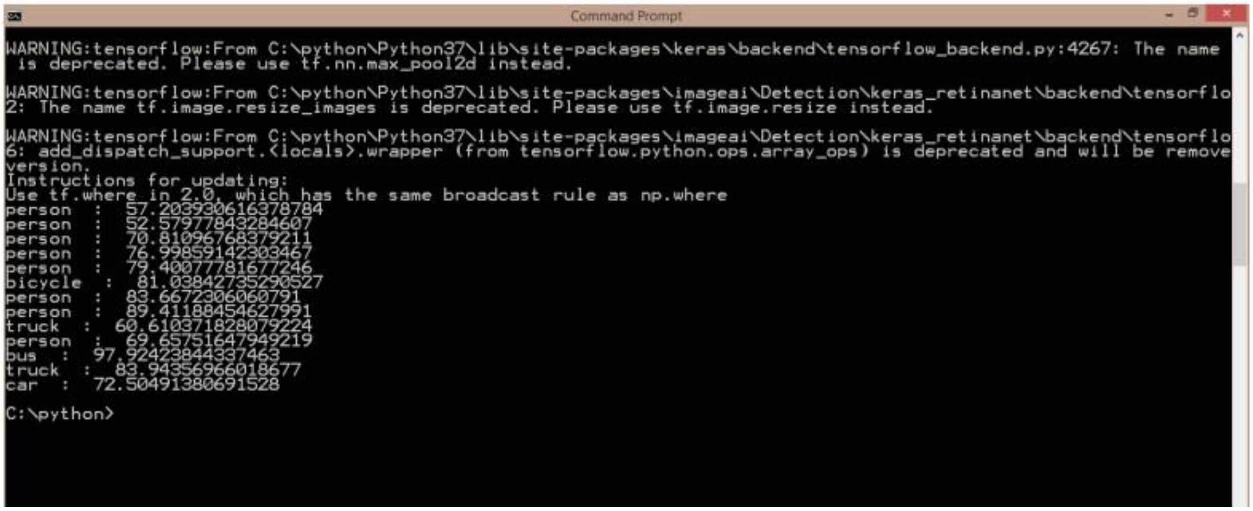
Before Detection

This is a sample image we feed to the algorithm and expect our algorithm to detect and identify objects in the image and label them according to the class assigned to it



After Detection

As expected, our algorithm identifies the objects by its classes and assigns each object by its tag and has dimensions on detected image.



```
WARNING:tensorflow:From C:\python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:4267: The name
 is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\python\Python37\lib\site-packages\imageai\Detection\keras_retinanet\backend\tensorflo
2: The name tf.image.resize_images is deprecated. Please use tf.image.resize instead.

WARNING:tensorflow:From C:\python\Python37\lib\site-packages\imageai\Detection\keras_retinanet\backend\tensorflo
6: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be remove
version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
person  :  57.203930616378784
person  :  52.57977843284607
person  :  70.81096768379211
person  :  76.99859142303467
person  :  79.40077781677246
bicycle  :  81.03842735290527
person  :  83.6672306060791
person  :  89.41188454627991
truck  :  60.610371828079224
person  :  69.65751647949219
bus  :  97.92423844337463
truck  :  83.94356966018677
car  :  72.50491380691528

C:\python>
```

Console result for above image

ImageAI provides many more features useful for customization and production capable deployments for object detection tasks. Some of the features supported are:

-Adjusting Minimum Probability: By default, objects detected with a probability percentage of less than 50 will not be shown or reported. You can increase this value for high certainty cases or reduce the value for cases where all possible objects are needed to be detected

-Custom Objects Detection: Using a provided Custom Object class, you can tell the detection class to report detections on one or a few numbers of unique objects

- Detection Speeds: You can reduce the time it takes to detect an image by setting the speed of detection speed to “fast”, “faster” and “fastest”

-Input Types: You can specify and parse in file path to an image, Numpy array or file stream of an image as the input image.

-Output Types: You can specify that the detect Objects from Image function should return the image in the form of a file or Numpy array

DISCUSSION OR REPORT/RESULT ANALYSIS

Specifically, the contributions towards this project work area that outweighs existing system are as follows:

- An object recognition system is developed, that recognizes the two-dimensional and threedimensional objects
- The feature extracted is sufficient for recognizing the object and marking the location of the object. x the proposed classifier is able to recognize the object in less computational cost
- The proposed global feature extraction requires less time, compared to the traditional feature extraction method
- The performance of the SVM-kNN is greater and promising when compared with the BPN and SVM
- The performance of the One-against-One classifier is efficient.
- Global feature extracted from the local parts of the image
- Local feature PCA-SIFT is computed from the blobs detected by the Hessian-Laplace detector.
- Along with the local features, the width and height of the object computed through projection method is used

PROJECT MANAGEMENT AND PROFESSIONAL COMMUNICATION

Flex Lens

Object Detection

&

Handwriting Recognition

System

SLIDE 1

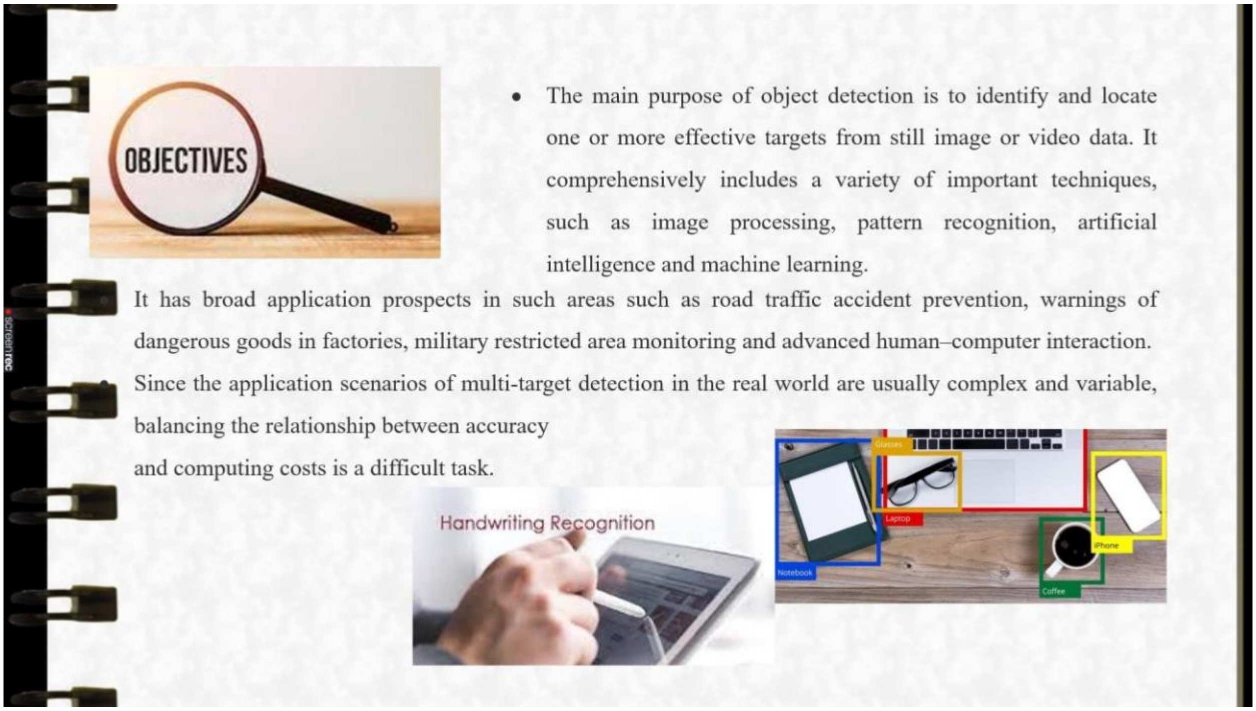
<>

"Flex Lens" can recognize handwritten stuffs and digitize them for use in various purpose in this superfast evolving environment, where Metaverse is pep talk. And object detection and recognition for faster processing of intermediatory tasks. It has a widefield application in security cameras, for developing systems for physically disabled peoples, etc.

Conclusively, "Flex Lens" is a smarter, faster and better system for the flexibility in our day-to-day life. Hence, making the earth a better place.

Introduction to
Object Detection
& Handwriting Recognition

SLIDE 2

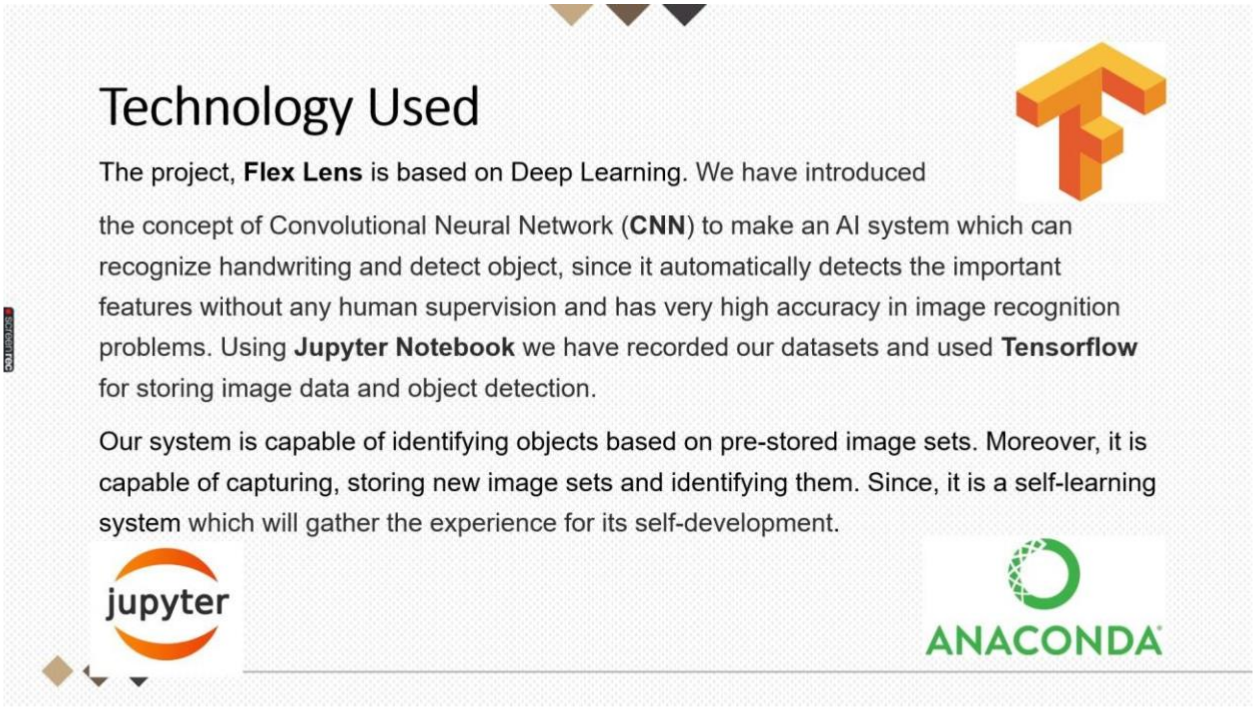


- The main purpose of object detection is to identify and locate one or more effective targets from still image or video data. It comprehensively includes a variety of important techniques, such as image processing, pattern recognition, artificial intelligence and machine learning.

It has broad application prospects in such areas such as road traffic accident prevention, warnings of dangerous goods in factories, military restricted area monitoring and advanced human–computer interaction.

- Since the application scenarios of multi-target detection in the real world are usually complex and variable, balancing the relationship between accuracy and computing costs is a difficult task.

SLIDE 3

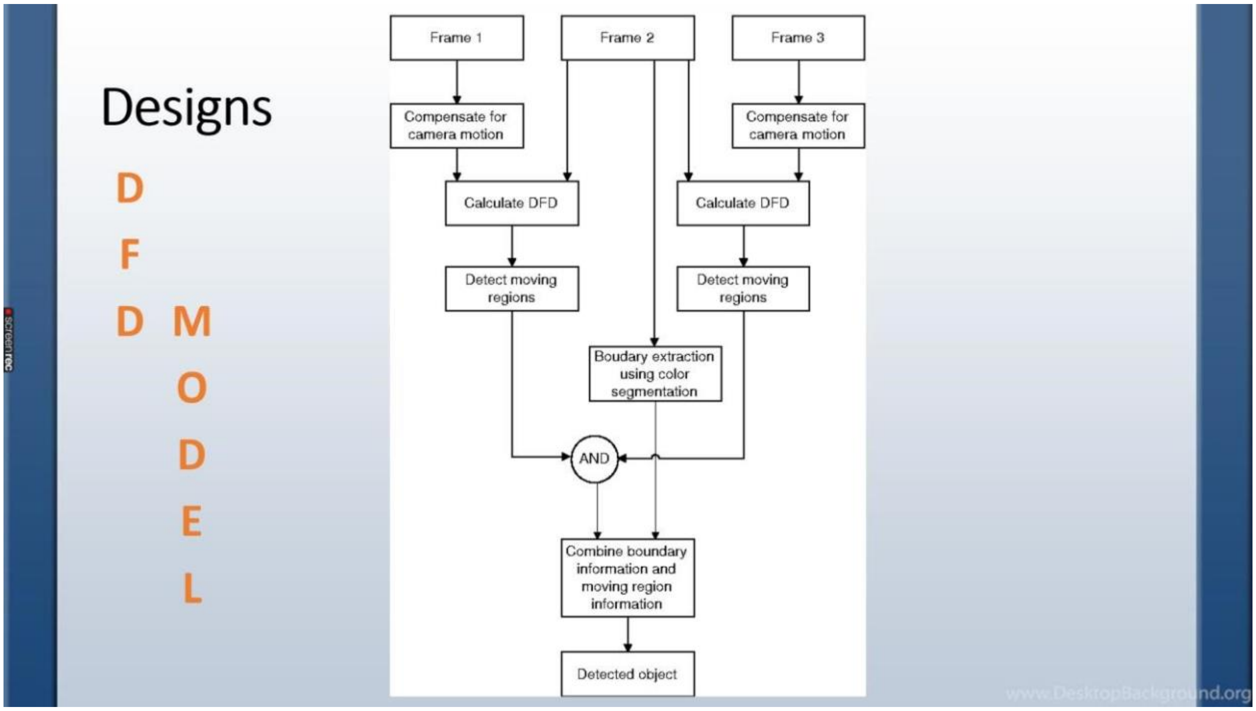


Technology Used

The project, **Flex Lens** is based on Deep Learning. We have introduced the concept of Convolutional Neural Network (**CNN**) to make an AI system which can recognize handwriting and detect object, since it automatically detects the important features without any human supervision and has very high accuracy in image recognition problems. Using **Jupyter Notebook** we have recorded our datasets and used **Tensorflow** for storing image data and object detection.

Our system is capable of identifying objects based on pre-stored image sets. Moreover, it is capable of capturing, storing new image sets and identifying them. Since, it is a self-learning system which will gather the experience for its self-development.

SLIDE 4

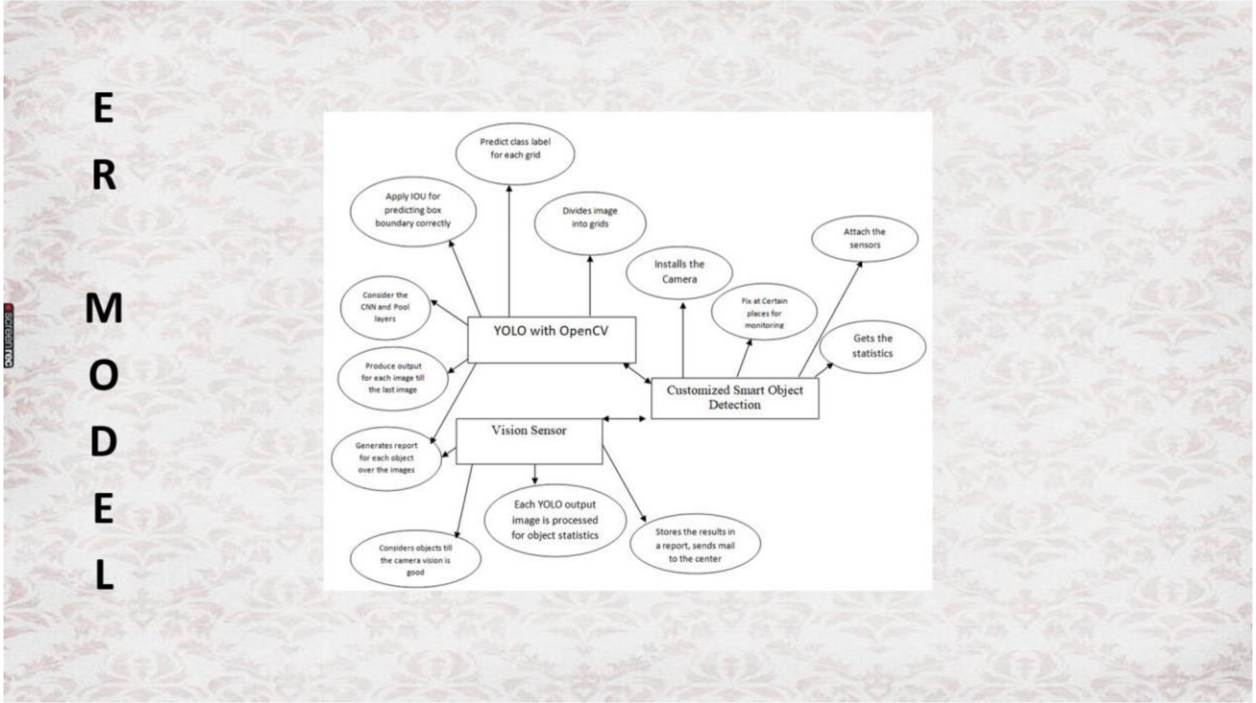


Designs

D
F
D M
O
D
E
L

Frame 1
Frame 2
Frame 3
Compensate for camera motion
Compensate for camera motion
Calculate DFD
Calculate DFD
Detect moving regions
Detect moving regions
Boudary extraction using color segmentation
AND
Combine boundary information and moving region information
Detected object

SLIDE 5



E R MODEL

YOLO with OpenCV

Predict class label for each grid

Apply IOU for predicting box boundary correctly

Divides image into grids

Consider the CNN and Pool layers

Produce output for each image till the last image

Generates report for each object over the images

Customized Smart Object Detection

Installs the Camera

Fix at Certain places for monitoring

Attach the sensors

Gets the statistics

Vision Sensor

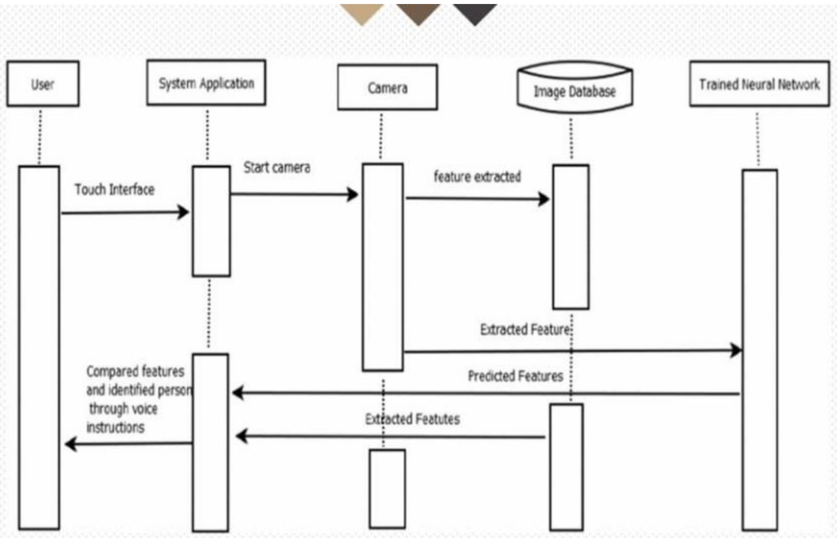
Considers objects till the camera vision is good

Each YOLO output image is processed for object statistics

Stores the results in a report, sends mail to the center

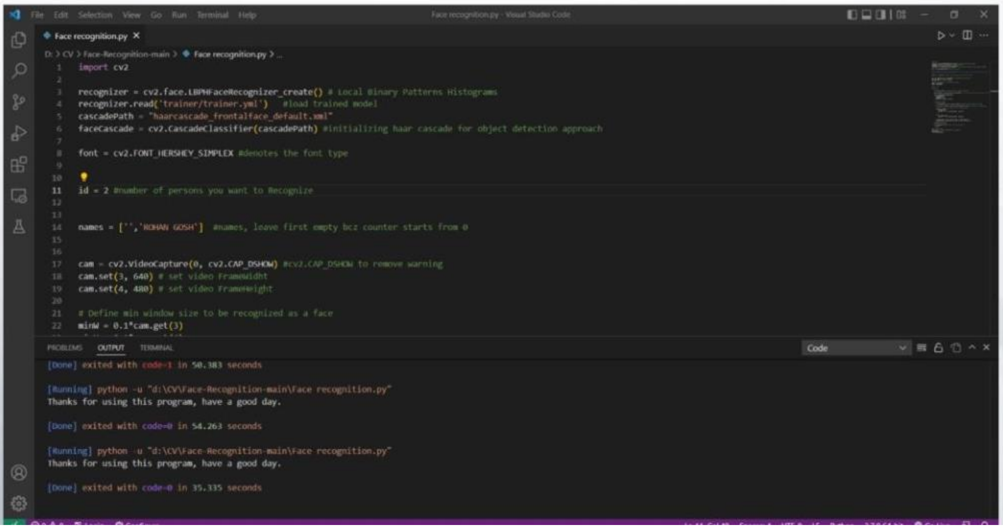
SLIDE 6

SEQUENCE DIAGRAM



SLIDE 7

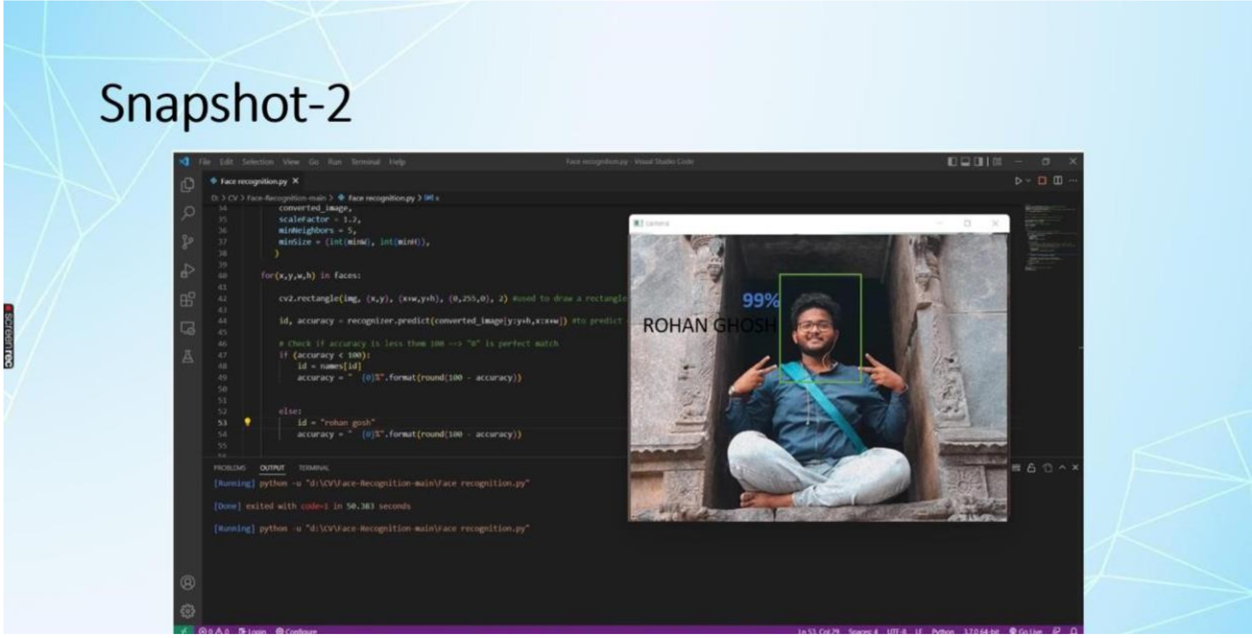
Snapshot-1


File Edit Selection View Go Run Terminal Help
Face recognition.py - Visual Studio Code
Face recognition.py
D: > CV > Face-Recognition-main > Face recognition.py > ...
PROBLEMS OUTPUT TERMINAL
Code

```
import cv2

recognizer = cv2.face.LBPHFaceRecognizer_create() # Local Binary Patterns Histograms
recognizer.read('trainer/trainer.yml')   #load trained model
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath) #initializing haar cascade for object detection approach

font = cv2.FONT_HERSHEY_SIMPLEX #denotes the font type


id = 2 #number of persons you want to Recognize


names = ['','ROHAN GOSH']  #names, leave first empty bcz counter starts from 0


cam = cv2.VideoCapture(0, cv2.CAP_DSHOW) #cv2.CAP_DSHOW to remove warning
cam.set(3, 640) # set video FrameWidht
cam.set(4, 480) # set video FrameHeight

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
```

```
[Done] exited with code=1 in 50.383 seconds

[Running] python -u "d:\CV\Face-Recognition-main\Face recognition.py"
Thanks for using this program, have a good day.

[Done] exited with code=0 in 54.263 seconds

[Running] python -u "d:\CV\Face-Recognition-main\Face recognition.py"
Thanks for using this program, have a good day.

[Done] exited with code=0 in 35.335 seconds
```

SLIDE 8



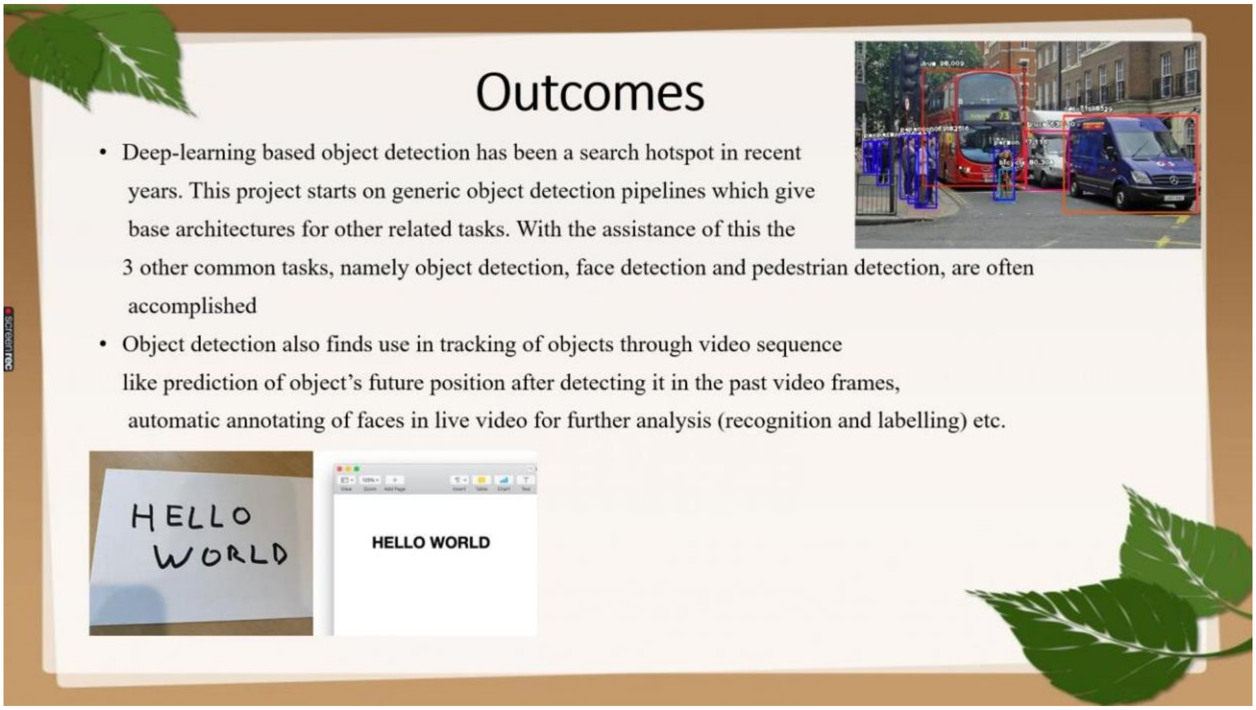
Snapshot-2

File Edit Selection View Go Run Terminal Help
Face recognition.py
D: > CV > Face-Recognition-main > Face recognition.py
99%
ROHAN GHOSH
PROBLEMS OUTPUT TERMINAL

```
            converted_image,
            scaleFactor = 1.2,
            minNeighbors = 5,
            minSize = (int(minW), int(minH)),
           )

        for(x,y,w,h) in faces:

            cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2) #used to draw a rectangle

            id, accuracy = recognizer.predict(converted_image[y:y+h,x:x+w]) #to predict

            # Check if accuracy is less them 100 ==> "0" is perfect match
            if (accuracy < 100):
                id = names[id]
                accuracy = "  {0}%".format(round(100 - accuracy))


            else:
                id = "rohan gosh"
                accuracy = "  {0}%".format(round(100 - accuracy))
```

```
[Running] python -u "d:\CV\Face-Recognition-main\Face recognition.py"

[Done] exited with code=1 in 50.383 seconds

[Running] python -u "d:\CV\Face-Recognition-main\Face recognition.py"
```

SLIDE 9



Outcomes

- Deep-learning based object detection has been a search hotspot in recent years. This project starts on generic object detection pipelines which give base architectures for other related tasks. With the assistance of this the 3 other common tasks, namely object detection, face detection and pedestrian detection, are often accomplished
- Object detection also finds use in tracking of objects through video sequence like prediction of object's future position after detecting it in the past video frames, automatic annotating of faces in live video for further analysis (recognition and labelling) etc.

SLIDE 10

ATTAINMENT OF STATED OUTCOMES

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

Deep-learning based object detection has been a search hotspot in recent years. This project starts on generic object detection pipelines which give base architectures for other related tasks. With the assistance of this the 3 other common tasks, namely object detection, face detection and pedestrian detection, are often accomplished. Authors accomplished this by combing 2 things: Object detection with deep learning and OpenCV and Efficient, threaded video streams with OpenCV. The camera sensor noise and lightening condition can change the result because it can create problem in recognizing the objects. generally, this whole process requires GPU's rather than CPU's. But we've done using CPU's and executes in much less time, making it efficient. Object Detection algorithms act as a mixture of both image classification and object localization. It takes the given image as input and produces the output having the bounding boxes adequate to the number of objects present within the image with the category label attached to every bounding box at the highest. It projects the scenario of the bounding box up the shape of position, height and width

REFERENCES

[1] Abdulllah, M., Agal, A., Alharthi, M., & Alrashidi, M. (2018). Retracted: Arabic handwriting recognition using neural network classifier. Journal of Fundamental and Applied Sciences, 10(4S), 265-270.

[2] Abe, S. (2010). Support Vector Machines for Pattern Classification. Berlin, Germany: Springer Science & Business Media

[3] Aggarwal, C. C. (2018). Neural Networks and Deep Learning: A Textbook. Basingstoke, England: Springer.

[4] Balas, V. E., Roy, S. S., Sharma, D., & Samui, P. (2019). Handbook of Deep Learning Applications. Basingstoke, England: Springer

[5] Boukharouba, A., & Bennia, A. (2017). Novel feature extraction technique for the recognition of handwritten digits. Applied Computing and Informatics, 13(1), 19-26. doi:10.1016/j.aci.2015.05.001

[6] Buckland, M. K. (2006). Emanuel Goldberg and His Knowledge Machine: Information, Invention, and Political Forces. Santa Barbara, CA: Greenwood Publishing Group.

[7] Chandio, A. A., Leghari, M., Hakro, D., AWAN, S., & Jalbani, A. H. (2016). A Novel Approach for Online Sindhi Handwritten Word Recognition using Neural Network. Sindh University Research JournalSURJ (Science Series), 48(1)

[8] Chen, L., Wang, S., Fan, W., Sun, J., & Naoi, S. (2015). Beyond human recognition: A CNNbased framework for handwritten character recognition. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 695-699. doi:10.1109/acpr.2015.7486592

[9] Ding, S., Zhao, H., Zhang, Y., Xu, X., & Nie, R. (2015). Extreme learning machine: algorithm, theory and applications. Artificial Intelligence Review, 44(1), 103-115

[10] Dwivedi, U., Rajput, P., Sharma, M. K., & Noida, G. (2017). Cursive Handwriting Recognition System Using Feature Extraction and Artificial Neural Network. Int. Res. J. Eng. Technol, 4(03), 2202