REAL TIME OBJECT DETECTION USING Open CV - Python

By

M. HANUMAN SAI (19BPS1066)

A project report submitted to

AROCKIA SELVAKUMAR

In partial fulfillment of the requirements for the course of
MEE3023 - ROBOTICS AND PROGRAMMING

In
B. Tech. COMPUTER SCIENCE AND ENGINEERING (SPECIALIZATION IN CYBER PHYSICAL SYSTEMS)

Vandalur – Kelambakkam Road

Chennai–600127
November2021

TABLE OF CONTENTS

ABSTRACT

Real time object detection has been attracting much interest due to the wide spectrum of applications. The applications of real time object detection include tracking objects, video surveillance, pedestrian detection, people counting, self-driving cars, face detection, ball tracking in sports and many more. If there is a single object to be detected in an image, it is known as Image Localization and if there are multiple objects in an image, then it is Object Detection. This detects the semantic objects of a class in digital images and videos. MobileNetv3 is a representative tool of Deep learning to detect objects using OpenCV(Opensource Computer Vision), which is a library of programming functions mainly aimed at real time computer vision.

PROBLEM DESCRIPTION

To be precise, the problem that object detection seeks to solve involves determining where the object is, and what it is. However, solving this problem is not easy. Unlike the human eye, a computer processes images in two dimensions. Furthermore, the size of the object, its orientation in the space, its attitude, and its location in the image can all vary greatly. Object detection is relatively simpler if the machine is looking for detecting one particular object. However, recognizing all the objects inherently requires the skill to differentiate one object from the other, though they may be of same type. Such problem is very difficult for machines, if they do not know about the various possibilities of objects

PROJECT OBJECTIVE

The goal of “object detection” is to find the type of an object in a given picture accurately and mark the object with the appropriate category.

Imparting intelligence to machines and making robots more and more autonomous and independent has been a sustaining technological dream for the mankind. It is our dream to let the robots take on tedious, boring, or dangerous work so that we can commit our time to more creative tasks. Unfortunately, the intelligent part seems to be still lagging behind. In real life, to achieve this goal, besides hardware development, we need the software that can enable robot the intelligence to do the work and act independently. One of the crucial components regarding this is vision, apart from other types of intelligences such as learning and cognitive thinking. A robot cannot be too intelligent if it cannot see and adapt to a dynamic environment.

MOTIVATION

Blind people do lead a normal life with their own style of doing things. But, they definitely face troubles due to inaccessible infrastructure and social challenges. The biggest challenge for a blind person, especially the one with the complete loss of vision, is to navigate around places. Obviously, blind people roam easily around their house without any help because they know the position of everything in the house. Blind people have a tough time finding objects around them.So I decided to make a REAL TIME OBJECT DETECTION System. We are interested in this project after we went through few papers in this area. As a result I’m highly motivated to develop a system that recognizes objects in the real time environment

METHODOLOGY

To build our deep learning-based real-time object detector with OpenCV. First we’ll need to -
we develop a technique to identify an object considering the deep learning pre-trained model MobileNet for Single Shot Multi-Box Detector (SSD). This algorithm is used for real-time detection, and for webcam feed to detect the purpose webcam which detects the object in a video stream. Therefore, we use an object detection module that can detect what is in the video stream. In order to implement the module, we combine the MobileNet and the SSD framework for a fast and efficient deep learning-based method of object detection.

ANALYSIS

Object detection methods aim to identify all target objects in the target image and determine the categories and position information in order to achieve machine vision understanding. Numerous approaches have been proposed to solve this problem, mainly inspired by methods of computer vision and deep learning. However, existing approaches always perform poorly for the detection of small, dense objects, and even fail to detect objects with random geometric transformations.
By using MobileNetv3 is a representative tool of Deep learning to detect objects using OpenCV(Opensource Computer Vision) will get better and accurate detection. My analysis demonstrates a strong performance on par, or even better, than state of the art methods.

INTRODUCTION

Now a days, object detection is used globally in numerous fields such as, video surveillance, pedestrian displays, defamation detection, self-driving cars and appearance recognition.
Object detection is a technology to detect various objects in digital images and videos too. It is mainly helpful within the self-driving cars, face detection, etc., where the objects are to be continuously monitored.

The algorithm or the technique involved for object detection during this project is Convolutional Neural Networks which is a class of Deep learning. This uses MobileNet SSD technique during which MobileNet is a neural network used for image classification and recognition whereas SSD is a framework that is used to realize the multibox detector. The mixture of both MobileNet and SSD can do object detection. The main advantage or purpose of choosing Deep learning is that we do not need to do feature extraction from data as compared to machine learning.

Humans can easily detect and identify objects present in an image. The human visual system is fast and accurate and can perform complex tasks like identifying multiple objects with little conscious thought. With the availability of large amounts of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy.

Every object class has its own special features that help in classifying the object. Object recognition is that sub-domain of computer vision which helps in identifying objects in an image or video sequence. With more efficient algorithms, objects can even be recognized even when they are partially obstructed from the direct view.

Open CV

OpenCV stands for Open supply pc Vision Library is associate open supply pc vision and machine learning software system library. The purpose of creation of OpenCV was to produce a standard infrastructure for computer vision applications and to accelerate the utilization of machine perception within the business product . It becomes very easy for businesses to utilize and modify the code with OpenCV as it is a BSD-licensed product. It is a rich wholesome libraby as it contains 2500 optimized algorithms, which also includes a comprehensive set of both classic and progressive computer vision and machine learning algorithms. These algorithms is used for various functions such as discover and acknowledging faces. Identify objects classify human actions. In videos, track camera movements, track moving objects. Extract 3D models of objects, manufacture 3D purpose clouds from stereo cameras, sew pictures along to provide a high-resolution image of a complete scene, find similar pictures from a picture information, remove red eyes from images that are clicked with the flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality

OpenCV's application areas include :

♣ 2D and 3D feature toolkits

♣Egomotion estimation

♣ Facial recognition system

♣ Gesture recognition

♣ Human–computer interaction (HCI)

♣ Mobile robotics

♣ Motion understanding

♣ Object identification

♣ Segmentation and recognition

♣ Stereopsis stereo vision: depth perception from 2 cameras

♣ Structure from motion (SFM)

♣ Motion tracking

♣ Augmented reality

MOBILENETS

To build lightweight deep neural networks MobileNets are used. It is based on a streamlined architecture that uses depth-wise separable convolutions. MobileNet uses 3×3 depth-wise separable convolutions that uses between 8 times less computation than standard convolution at solely a little reduction accuracy. Applications and use cases including object detection, fine grain classification, face attributes and large scale-localization.

TENSOR FLOW

Tensor flow is an open source software library for high performance numerical computation. It allows simple deployment of computation across a range of platforms (CPUs, GPUs, TPUs) due to its versatile design also from desktops to clusters of servers to mobile and edge devices. Tensor flow was designed and developed by researchers and engineers from the Google Brain team at intervals Google’s AI organization, it comes with robust support for machine learning and deep learning and the versatile numerical computation core is used across several alternative scientific domains.

To construct, train and deploy Object Detection Models TensorFlow is used that makes it easy and also it provides a collection of Detection Models pre-trained on the COCO dataset, the Kitti dataset, and the Open Images dataset. One among the numerous Detection Models is that the combination of Single Shot Detector (SSDs) and Mobile Nets architecture that is quick, efficient and doesn't need huge computational capability to accomplish the object Detection.

LITERATURE SURVEY

Bhumika Gupta (2017) et a;. proposed object detection is a well-known computer technology connected with computer vision and image processing that focuses on detecting objects or its instances of a certain class (such as humans, flowers, animals) in digital images and videos. There are various applications of object detection that have been well researched including face detection, character recognition, and vehicle calculator. Object detection can be used for various purposes including retrieval and surveillance. In this study, various basic concepts used in object detection while making use of OpenCV library of python 2.7, improving the efficiency and accuracy of object detection are presented.

Kartik Umesh Sharma (2017) et al, proposed an object detection system finds objects of the real world present either in a digital image or a video, where the object can belong to any class of objects namely humans, cars, etc. In order to detect an object in an image or a video the system needs to have a few components in order to complete the task of detecting an object, they are a model database, a feature detector, a hypothesiser and a hypothesiser verifier. This paper presents a review of the various techniques that are used to detect an object, localise an object, categorise an object, extract features, appearance information, and many more, in images and videos. The comments are drawn based on the studied literature and key issues are also identified relevant to the object detection. Information about the source codes and online datasets is provided to facilitate the new researcher in object detection area. An idea about the possible solution for the multi class object detection is also presented. This paper is suitable for the researchers who are the beginners in this domain.

Mukesh Tiwari (2017) et al. presented object detection and tracking is one of the critical areas of research due to routine change in motion of object and variation in scene size, occlusions, appearance variations, and ego-motion and illumination changes. Specifically, feature selection is the vital role in object tracking. It is related to many real time applications like vehicle perception, video surveillance and so on. In order to overcome the issue of detection, tracking related to object movement and appearance. Most of the algorithm focuses on the tracking algorithm to smoothen the video sequence. On the other hand, few methods use the prior available information about object shape, color, texture and so on. Tracking algorithm which combines above stated parameters of objects is discussed and analyzed in this research. The goal of this paper is to analyze and review the previous approach towards object tracking and detection using video sequences through different phases. Also, identify the gap and suggest a new approach to improve the tracking of object over video frame.

Aishwarya Sarkale (2018) et al. proposed humans have a great capability to distinguish objects by their vision. But, for machines object detection is an issue. Thus, Neural Networks have been introduced in the field of computer science. Neural Networks are also called as ‘Artificial Neural Networks’. Artificial Neural Networks are computational models of the brain which helps in object detection and recognition. This paper describes and demonstrates the different types of Neural Networks such as ANN, KNN, FASTER R-CNN, 3D-CNN, RNN etc. with their accuracies. From the study of various research papers, the accuracies of different Neural Networks are discussed and compared and it can be concluded that in the given test cases, the ANN gives the best accuracy for the object detection.

Karanbir Chahal (2018) et al. proposed Object detection is the identification of an object in the image along with its localization 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 light weight convolutional base architectures.

Ultimately, a rigorous review of the strengths and weaknesses of each detector leads us to the present state of the art.

Richard Socher (2018) et al. proposed recent advances in 3D sensing technologies make it possible to easily record color and depth images which together can improve object recognition. Most current methods rely on very welldesigned features for this new 3D modality. We introduce a model based on a combination of convolutional and recursive neural networks (CNN and RNN) for learning features and classifying RGB-D images. The CNN layer learns low-level translationally invariant features which are then given as inputs to multiple, fixed-tree RNNs in order to compose higher order features. RNN scan be seen as combining convolution and pooling into one efficient, hierarchical operation. Our main result is that even RNNs with random weights compose powerful features. Our model obtains state of the art performance on a standard RGB-D object data set while being more accurate and faster during training and testing than comparable architectures such as two-layer CNNs.

Yordanka Karayaneva (2018) et al. presented schools in many parts of the world use robots as social peers in order to interact with children and young students for a rich experience. Such use has shown significant enhancement of children’s learning. This project uses the humanoid robot NAO which provides object recognition of colours, shapes, typed words, and handwritten digits and operators. The recognition of typed words provides performance of the corresponding movements in the sign language. Five classifiers including neural networks are used for the handwritten recognition of digits and operators. The accuracy of the object recognition algorithms are within the range of 82%-92% when tested on images captured by the robot including the movements which represent words in the sign language. The five classifiers for handwritten recognition produce highly accurate results which are within the range of 87%-98%. This project will serve as a promising provision for an affective touch for children and young students.

Abdul Muhsin M (2019) et al. proposed everybody deserve to live independently, especially those who disabled, with the last decades, technology gives attention to disabled to make them control their life as possible. In this work, assistive system for blind is suggested, to let him knows what is around him, by using YOLO for detecting objects within images and video streams quickly based on deep neural network to make accurate detection, and OpenCV under Python using Raspberry Pi3. The obtained results indicated the success of the proposed model in giving blind users the capability to move around in unfamiliar indoor outdoor environment, through a user friendly device by person and object identification model.

Geethapriya. S (2019) et al. proposed the Objective is to detect of objects using You Only Look Once (YOLO) approach. This method has several advantages as compared to other object detection algorithms. In other algorithms like Convolutional

Neural Network, Fast Convolutional Neural Network the algorithm will not look at the image completely but in YOLO the algorithm looks the image completely by predicting the bounding boxes using convolutional network and the class probabilities for these boxes and detects the image faster as compared to other algorithms.

R. Sujeetha (2019) et al. proposed object detection and tracking could be a immense, vivacious however inconclusive and trending area of computer vision. Due to its immense use in official surveillances, tracking modules applied in security and lots of others applications have made researchers to devise a lot of optimized and specialized methods. However, problems are faced in implementing object detection and tracking in real-time; like tracking in real time and giving appropriate optimized results, over dynamic computation to find the efficient performance with respect to time factor, or multiple objects tracking create this task more difficult. Though, several techniques are devised but still lies a lot of scope of improvement, however during this literature review we've seen some illustrious and multiple ways of object detection and tracking. In this method we will be using Tensor Flow and Open CV library and CNN algorithm will be used and we will be labelling the detected layers with accuracy being checked at the same time .For validation purpose live input video will be taken for the same where objects will be getting detected and it can be simulated same for real-time through external hardware added. In the end we see the proper optimized and efficient algorithm for object tracking and detection.

STEPS INVOLVED IN OBJECT DETECTION

Let us start with an image (im.jpg) and detect various objects in it.

Install OpenCV-Python

Below Python packages are to be downloaded and installed to their default location - Python-2.7.x, NumPy and Matplotlib. Install all packages into their default locations. Python will be installed to C/Python27/. Open Python IDLE. Enter import NumPy and make sure NumPy is working fine. Download OpenCV from Sourceforge. Go to OpenCV/build/python/2.7 folder. Copy cv2.pyd to C:/Python27/lib/site-packages.

Read an Image

Use the function CV2.imread() to read an image. The image should be in the current working directory otherwise, we need to specify the full path of the image as the first argument.

The second argument is a flag which specifies the way image should be read.

1. CV2.IMREAD_COLOR: This function is used to load a color image. Transparency of image, if present will be neglected. It is the default flag.

2. CV2.IMREAD_GRAYSCALE: Loads image in grayscale mode
3. CV2.IMREAD_UNCHANGED: Loads image as such including alpha channel.

Feature detection and description

• Understanding features (What are the main features in an image? How can finding those features be useful to us?)
• Corner detection (Okay, Corners are good features? But how do we find them)
• Feature matching (We know a great deal about feature detectors and descriptors. So let us now learn to match different descriptors. OpenCV provides two techniques, Brute-Force matcher, and FLANN based matcher.)
• Homography (As we are aware of feature matching, so let us now blend it with Camera calibration and 3D reconstruction (calib3d module) to find objects for description in a complex image.)

PYTHON VS OTHER LANGUAGES FOR OBJECT DETECTION

Object detection is a domain-specific variation of the machine learning prediction problem. Intel‟s OpenCV library that is implemented in C/C++ has its interfaces available in a number of programming environment such as C#, Matlab, Octave, R, Python etc.
Some of the benefits of using Python codes over other language codes for object detection are • More compact and readable code.
• Python uses zero-based indexing.
• Dictionary (hashes) support is offered.
• Simple and elegant Object-oriented programming

Software Specifications

- Python 3.7
- TensorFlow
- Anaconda Software
- Machine Learning Libraries
- Jupyter Notebook
- Opencv

why you need object detection

One of the best examples of why you need object detection is the high-level algorithm for autonomous driving:

• In order for a car to decide what to do next: accelerate, apply brakes or turn, it needs to know where all the objects are around the car and what those objects are

• That requires object detection

• You would essentially train the car to detect known set of objects: cars, pedestrians, traffic lights, road signs, bicycles, motorcycles, etc

Applications

Here are a some of the applications and future scope of object detection.

1. **Face detections and recognition**: 59 Face detection perhaps be a separate class of object detection. We wonder how some applications like Facebook, Faceapp, etc., detect and recognize our faces. this is often a sample example of object detection in our day to day life. Face detection is already in use in our lifestyle to unlock our mobile phones and for other security systems to scale back rate .
2. **Object tracking**: Object detection is additionally utilized in tracking objects like tracking an individual and his actions, continuously monitoring a ball within the game of Football or Cricket. As there's an enormous interest for people in these games, these tracking techniques enables them to know it during a better way and obtain some additional information. Tracking of the ball is of maximal importance in any ball-based games to automatically record the movement of the ball and adjust the video frame accordingly
3. **Self-driving cars**: this is often one among the main evolutions of the planet and is that the best example why we'd like object detection. so as for a car to travel to the specified destination automatically with none human interference or to form decisions whether to accelerate or

to use brakes and to spot the objects around it. this needs object detection.

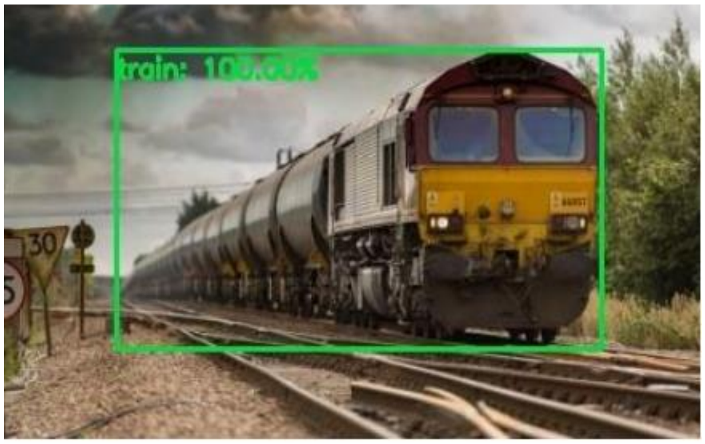
4. **Security**: Identification of unwanted or suspicious objects in any particular area or more specifically object detection techniques are used for detecting bombs/explosives. It is also even used for personal security purpose.
5. **Surveillance**: Objects can be recognized and tracked in videos for security purpose. Object recognition is required so that the suspected person or vehicle can be tracked

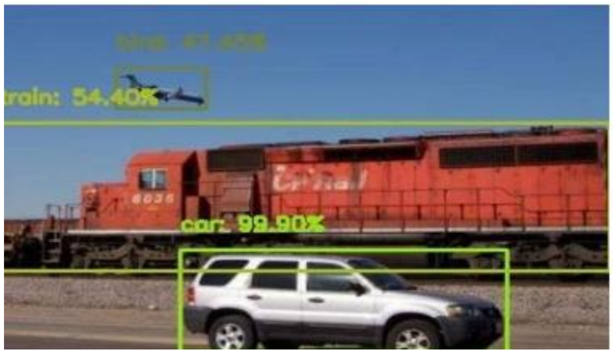
CHALLENGES

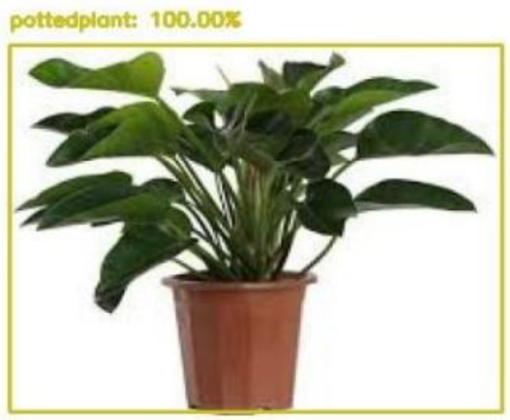
The main purpose is to recognize a specific object in real time from a large number of objects. Most recognition systems are poorly scalable with many recognizable objects. Computational cost rises as the number of objects increases. Comparing and querying images using color, texture, and shape are not enough because two objects might have same attributes. Designing a recognition system with abilities to work in the dynamic environment and behave like a human is difficult. Some main challenges to design object recognition system are lighting, dynamic background, the presence of shadow, the motion of the camera, the speed of the moving objects, and intermittent object motion weather conditions etc.

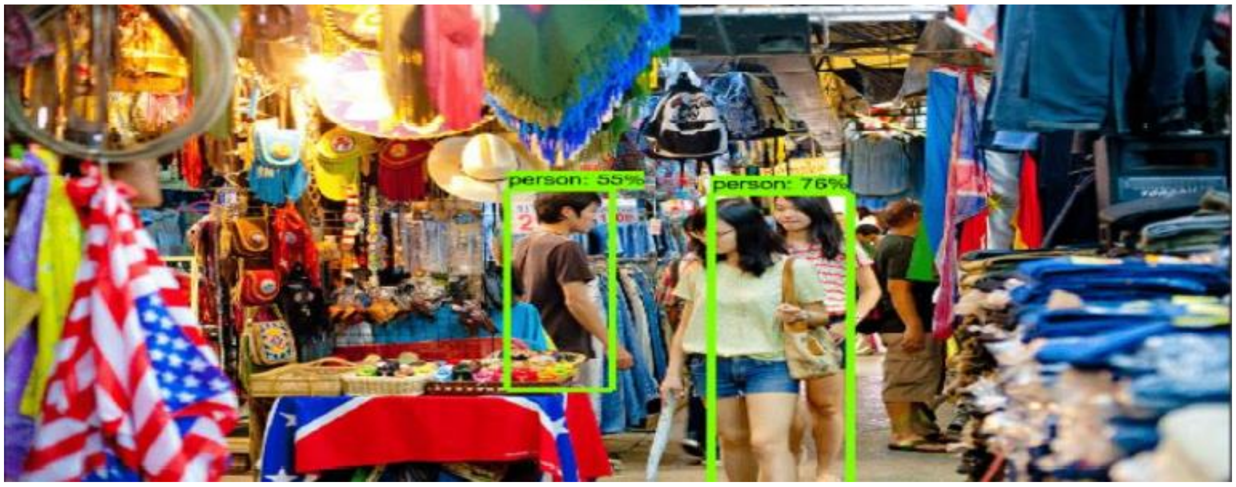
EXPERIMENTAL RESULTS

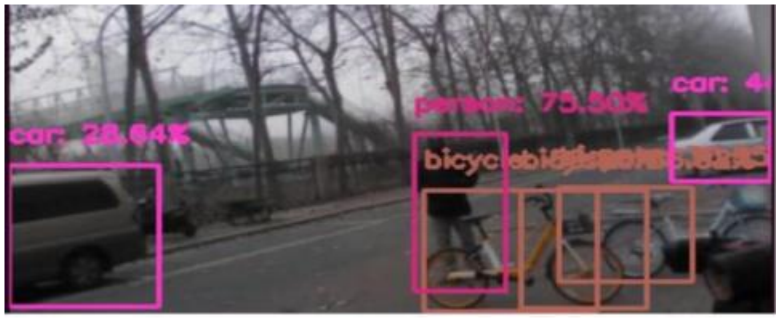
Here, in this project I’ve considered around 80 objects to be detected during the training. Some of those include ‘person’, ‘car’, ‘train’, ‘bird’, ‘sofa’, ‘dog’, ‘’plant’, ‘aero plane’, ‘bicycle’, ‘bus’, ‘motorbike’, etc. The output of this project displays the objects detected with a rectangular box around the object with a label indicating it’s name and therefore the exactness with which the object has been detected on the top of it. It can dig out any number of objects existing during a single image with certainty.

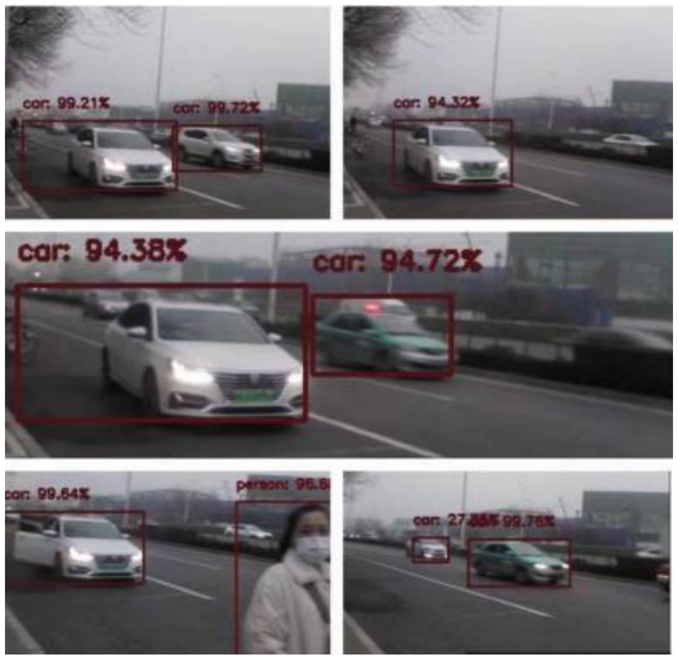




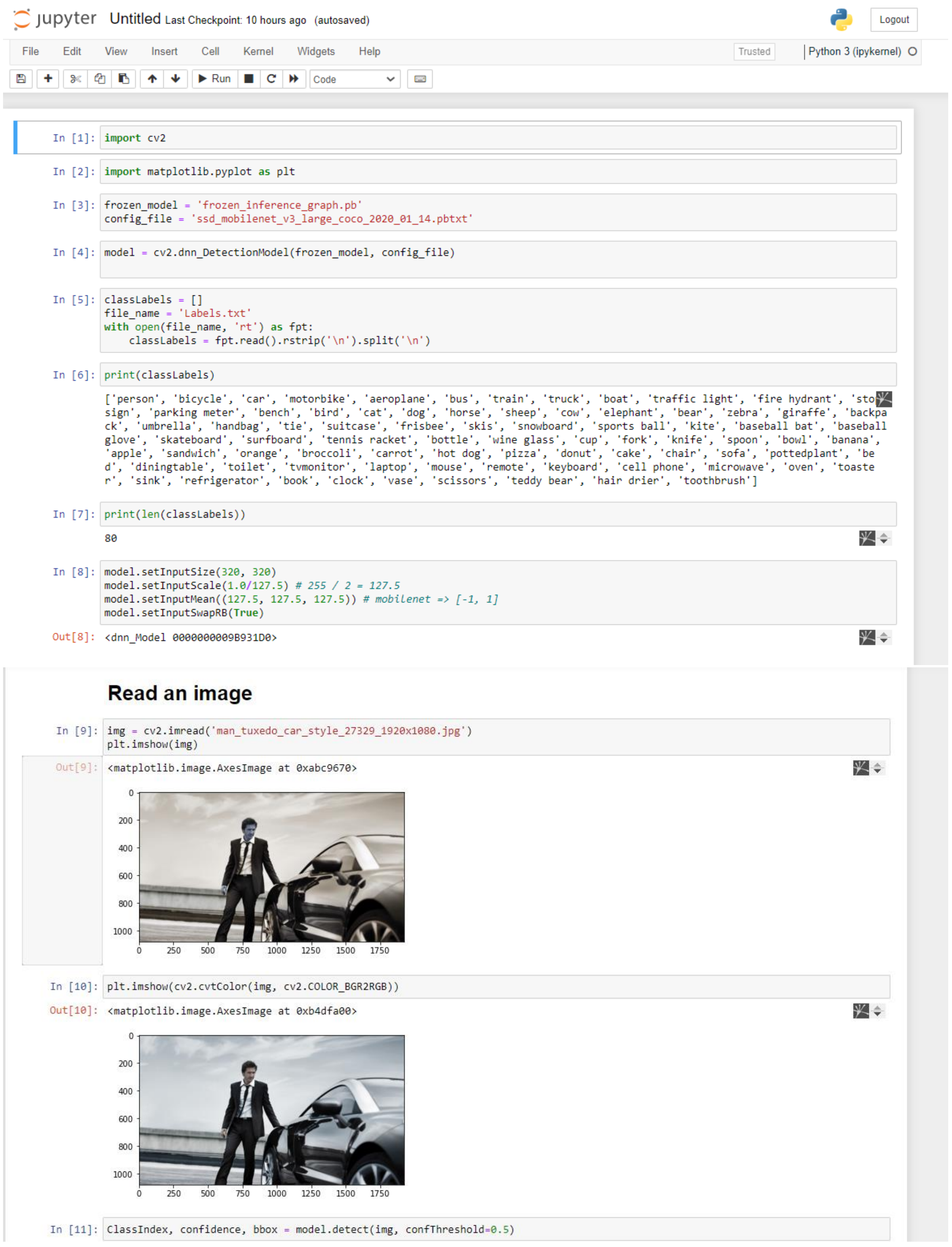

pottedplant: 100.00%


person: 55%
person: 76%


car: 28.64%
person: 75.50%
car: 4
bicyc


car: 99.21%
car: 99.72%
car: 94.32%
car: 94.38%
car: 94.72%
car: 99.64%
person: 96.5
car: 27.88% 99.76%

CODE


jupyter Untitled Last Checkpoint: 10 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

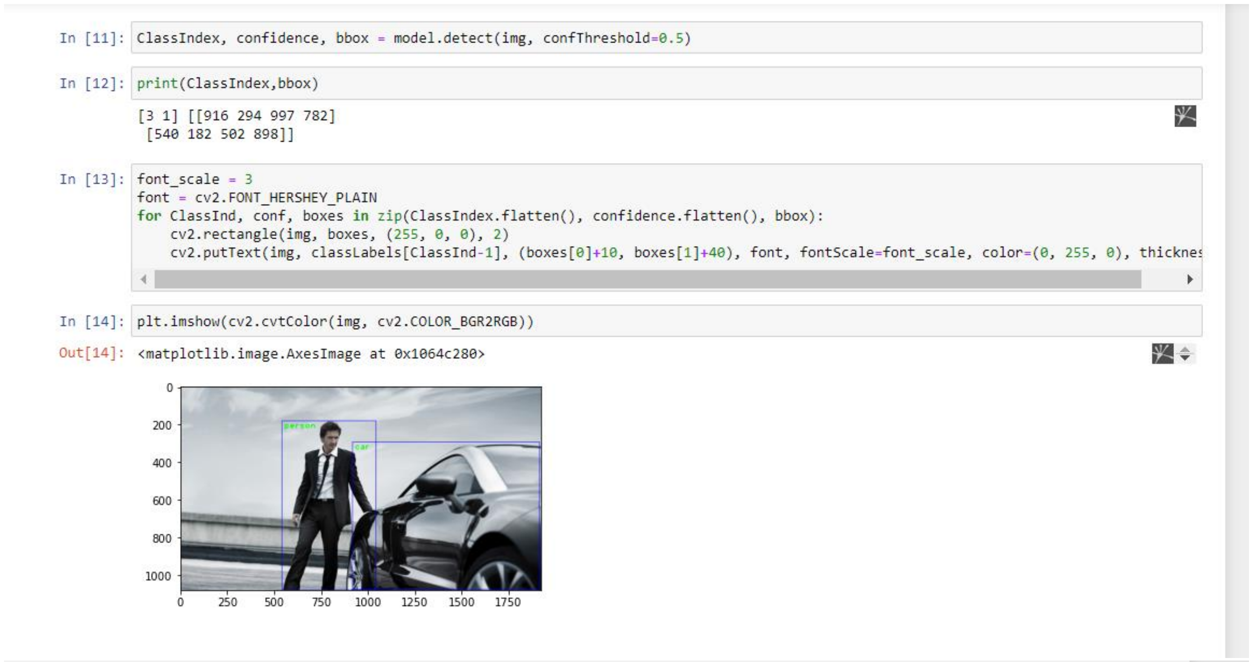
```
In [1]: import cv2

In [2]: import matplotlib.pyplot as plt

In [3]: frozen_model = 'frozen_inference_graph.pb'
        config_file = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'

In [4]: model = cv2.dnn_DetectionModel(frozen_model, config_file)

In [5]: classLabels = []
        file_name = 'Labels.txt'
        with open(file_name, 'rt') as fpt:
            classLabels = fpt.read().rstrip('\n').split('\n')

In [6]: print(classLabels)

        ['person', 'bicycle', 'car', 'motorbike', 'aeroplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'sto
        sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpa
        ck', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball
        glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana',
        'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'sofa', 'pottedplant', 'be
        d', 'diningtable', 'toilet', 'tvmonitor', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaste
        r', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']

In [7]: print(len(classLabels))

        80

In [8]: model.setInputSize(320, 320)
        model.setInputScale(1.0/127.5) # 255 / 2 = 127.5
        model.setInputMean((127.5, 127.5, 127.5)) # mobilenet => [-1, 1]
        model.setInputSwapRB(True)

Out[8]: <dnn_Model 0000000009B931D0>
```

Read an image

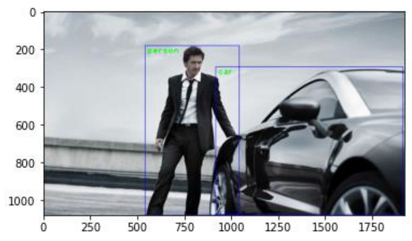
```
In [9]: img = cv2.imread('man_tuxedo_car_style_27329_1920x1080.jpg')
        plt.imshow(img)

Out[9]: <matplotlib.image.AxesImage at 0xabc9670>
```

```
In [10]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

Out[10]: <matplotlib.image.AxesImage at 0xb4dfa00>
```

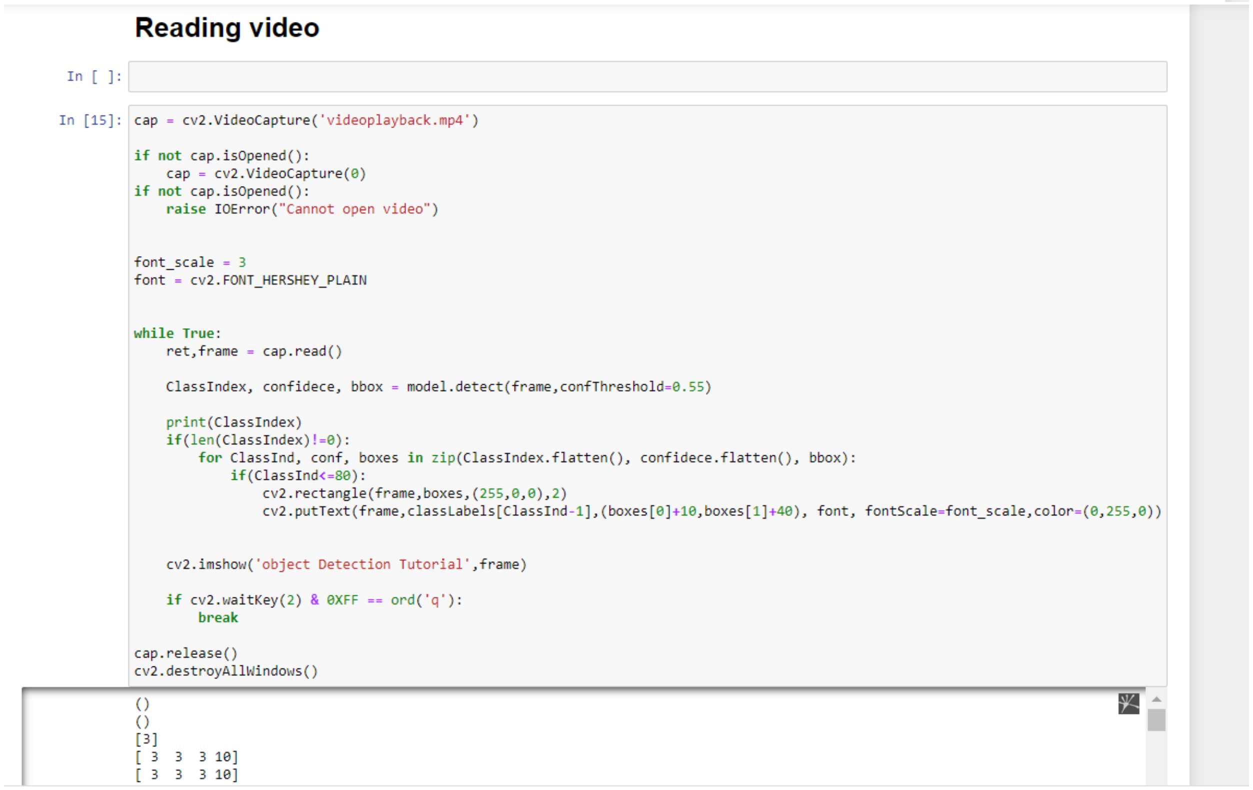
```
In [11]: ClassIndex, confidence, bbox = model.detect(img, confThreshold=0.5)
```



```
In [11]: ClassIndex, confidence, bbox = model.detect(img, confThreshold=0.5)

In [12]: print(ClassIndex,bbox)

[3 1] [[916 294 997 782]
 [540 182 502 898]]

In [13]: font_scale = 3
         font = cv2.FONT_HERSHEY_PLAIN
         for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(), bbox):
             cv2.rectangle(img, boxes, (255, 0, 0), 2)
             cv2.putText(img, classLabels[ClassInd-1], (boxes[0]+10, boxes[1]+40), font, fontScale=font_scale, color=(0, 255, 0), thicknes

In [14]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

Out[14]: <matplotlib.image.AxesImage at 0x1064c280>
```



Reading video



```
In [ ]:

In [15]: cap = cv2.VideoCapture('videoplayback.mp4')

         if not cap.isOpened():
             cap = cv2.VideoCapture(0)
         if not cap.isOpened():
             raise IOError("Cannot open video")


         font_scale = 3
         font = cv2.FONT_HERSHEY_PLAIN


         while True:
             ret,frame = cap.read()

             ClassIndex, confidece, bbox = model.detect(frame,confThreshold=0.55)

             print(ClassIndex)
             if(len(ClassIndex)!=0):
                 for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidece.flatten(), bbox):
                     if(ClassInd<=80):
                         cv2.rectangle(frame,boxes,(255,0,0),2)
                         cv2.putText(frame,classLabels[ClassInd-1],(boxes[0]+10,boxes[1]+40), font, fontScale=font_scale,color=(0,255,0))


             cv2.imshow('object Detection Tutorial',frame)

             if cv2.waitKey(2) & 0XFF == ord('q'):
                 break

         cap.release()
         cv2.destroyAllWindows()

()
()
[3]
[ 3  3  3 10]
[ 3  3  3 10]
```

CONCLUSION

A high accuracy object detection procedure has been achieved by using the MobileNet and the SSD detector for an object detection, which can push a processing speed to 14 fps, and make it efficient to all camera that can often process at only 6 fps. This system can detect the items within its dataset, such as a car, bicycle, bottle, chair, etc. The dataset can be expanded by adding an unlimited number of items through using images, referred to as deep learning technology. I used OpenCV 3.4.2 and Python programming language for the experiment of SSD algorithm. The goal of this project is to develop an autonomous system where the recognition of objects and scenes helps the community to make the system interactive and attractive. For future work, this work will be primarily deployed to identify the item with better features in the external environment.

The possibilities of using computer vision to solve real world problems are immense. Python has been preferred over MATLAB for integrating with OpenCV because when a Matlab program is run on a computer, it gets busy trying to interpret all that Matlab code as Matlab code is built on Java. OpenCV is basically a library of functions written in C\C++. Additionally, OpenCV is easier to use for someone with little programming background.

REFERENCES

1. Geethapriya S, N. Duraimurugan, S.P. Chokkalingam, “Real-Time Object Detection with Yolo”, International Journal of Engineering and Advanced Technology (IJEAT)

2. Abdul Vahab, Maruti S Naik, Prasanna G Raikar an Prasad S R4, “Applications of Object Detection System”, International Research Journal of Engineering and Technology (IRJET)

3. Hammad Naeem, Jawad Ahmad and Muhammad Tayyab, “Real-Time Object Detection and Tracking”,IEEE

4. Meera M K, & Shajee Mohan B S. 2016, "Object recognition in images", International Conference on Information 60 Science (ICIS).

5. Astha Gautam, Anjana Kumari, Pankaj Singh: "The Concept of Object Recognition", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 3, March 2015

6. Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016,pp. 779- 788

7. V. Gajjar, A. Gurnani and Y. Khandhediya, "Human Detection and Tracking for Video Surveillance: A Cognitive Science Approach," in 2017 IEEE International Conference on Computer Vision Workshops, 2017

8. R. Sujeetha, Vaibhav Mishra , “Object Detection and Tracking using Tensor Flow”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019.

9. Richard Socher, Brody Huval, Bharath Bhat, Christopher D. Manning, Andrew Y. Ng, “Convolutional-Recursive Deep Learning for 3D Object Classification”, International Conference on Computational Intelligence and Communication Networks, 2018.

10. Yordanka Karayaneva and Diana Hintea, “Object Recognition in Python and MNIST Dataset Modification and Recognition with Five Machine Learning Classifiers”, Journal of Image and Graphics, Volume 6, Number 1, June 2018.