Object Detection using Python OpenCV

PROJECT REPORT

GALGOTIAS
UNIVERSITY

SUBMITTED TO

Guide: Dr.J.N.Singh

PANEL: Dr.Ajay Shanker Singh

SUBMITTED BY

Amisha Saxena

(16SCSE105091)

(1613105010)

ABSTRACT

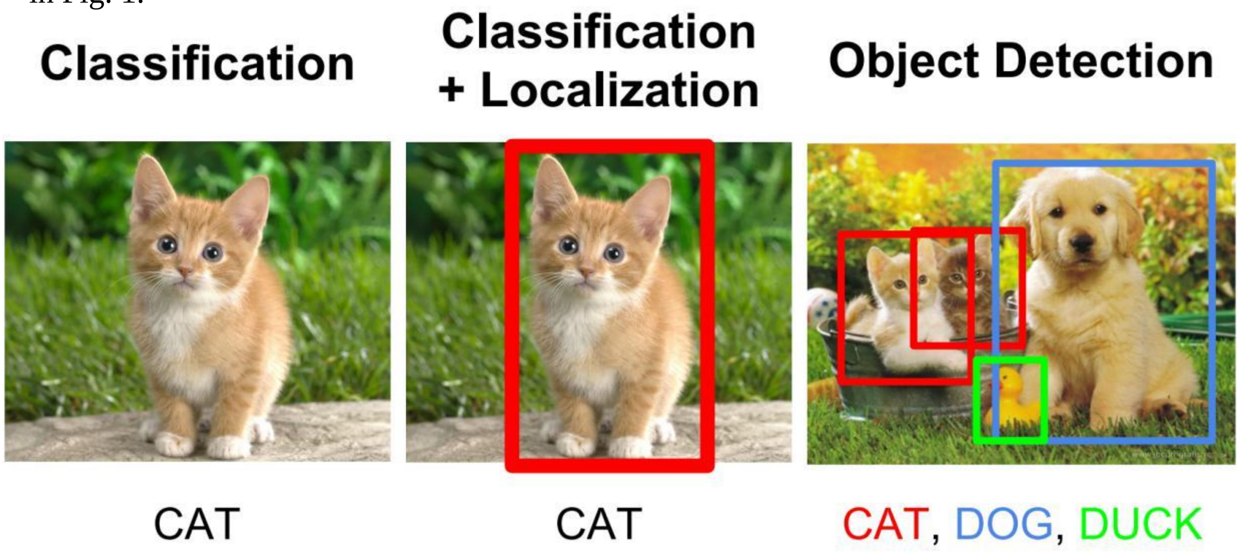
Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advent of deep learning techniques, the accuracy for object detection has increased drastically. The project aims to incorporate state-of-the-art technique for object detection with the goal of achieving high accuracy with a real-time performance. A major challenge in many of the object detection systems is the dependency on other computer vision techniques for helping the deep learning based approach, which leads to slow and non-optimal performance. In this project, we use a completely deep learning based approach to solve the problem of object detection in an end-to-end fashion. The network is trained on the most challenging publicly available data-set, on which a object detection challenge is conducted annually. The resulting system is fast and accurate, thus aiding those applications which require object detection.

1 INTRODUCTION

Object detection is a well-known computer technology connected with computer vision and image processing.With the advent of deep learning techniques, the accuracy for object detection has increased drastically.It 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 including face detection, character recognition, and vehicle calculator.

1.1 Problem Statement

Many problems in computer vision were saturating on their accuracy before a decade. However, with the rise of deep learning techniques, the accuracy of these problems drastically improved. One of the major problem was that of image classi cation, which is de ned as predicting the class of the image. A slightly complicated problem is that of image localiza-tion, where the image contains a single object and the system should predict the class of the location of the object in the image (a bounding box around the object). The more compli-cated problem (this project), of object detection involves both classi cation and localization. In this case, the input to the system will be a image, and the output will be a bounding box corresponding to all the objects in the image, along with the class of object in each box. An overview of all these problems is depicted in Fig. 1.

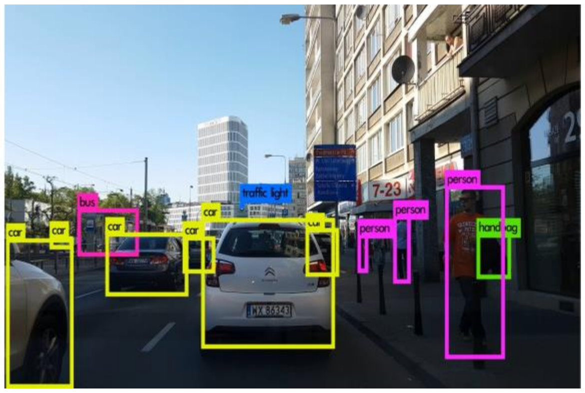


1.2 Applications

A well known application of object detection is face detection, that is used in almost all the mobile cameras. A more generalized (multi-class) application can be used in autonomous driving where a variety of objects need to be detected. Also it has a important role to play in surveillance systems. These systems can be integrated with other tasks such as pose estimation where the rst stage in the pipeline is to detect the object, and then the second stage will be to estimate pose in the detected region. It can be used for tracking objects and thus can be used in robotics and medical applications. Thus this problem serves a multitude of application

(a) Surveillance

(b) Autonomous vehicles



1.3 Challenges

The major challenge in this problem is that of the variable dimension of the output which is caused due to the variable number of objects that can be present in any given input image. Any general machine learning task requires a xed dimension of input and output for the model to be trained. Another important obstacle for widespread adoption of object detection systems is the requirement of real-time (>30fps) while being accurate in detection. The more complex the model is, the more time it requires for inference; and the less complex the model is, the less is the accuracy. This trade-o between accuracy and performance needs to be chosen as per the application. The problem involves classi cation as well as regression, leading the model to be learnt simultaneously. This adds to the complexity of the problem.

2 <u>PROPOSED SYSTEM</u>

One of the important fields of Artificial Intelligence is Computer Vision the science of computers and software systems that can recognize and understand images and scenes. Computer Vision is also composed of various aspects such as image recognition, object detection, image generation, image super-resolution and more. Object detection is probably the most profound aspect of computer vision due the number of practical use cases. Object detection refers to the capability of software systems to locate objects in an image/scene and identify each object. It has been widely used for face detection, vehicle detection, pedestrian counting, web images, security systems and driverless cars. There are many ways object detection can be used as well in many fields of practice. Like every other computer technology, a wide range of creative and amazing uses of object detection will definitely come from the efforts of computer programmers and software developers.

Getting to use modern object detection methods in applications and systems, as well as building new applications based on these methods is not a straight forward task. Early implementations of object detection involved the use of classical algorithms, like the ones supported in OpenCV, the popular computer vision library. However, these classical algorithms could not achieve enough performance to work under different conditions.

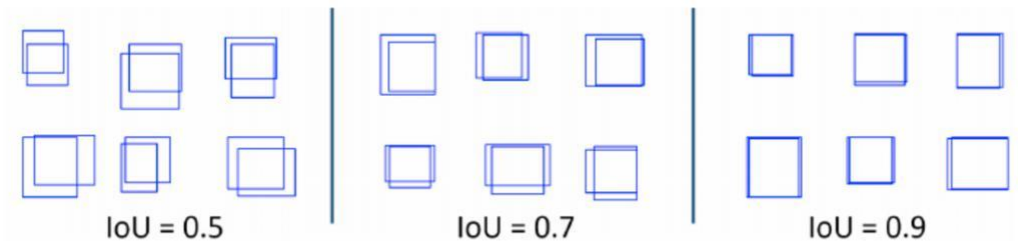
This project took use of several software libraries, packages and programs to utilize machine learning. Python was the choice of programming language, and TensorFlow was used for the deep learning computations, which in turn has a list of dependencies. TensorFlow offers a version for CPU usage and another for GPU, this project used the GPU version. Said version requires extra programs from the GPU designer NVIDIA, such as CUDA Toolkit, cuDNN and their GPU drivers. So far NVIDIA is the leading GPU designer for deep learning (also crypto mining and other similar high complex tasks) since they also write programs that are compatible with their cards that enable much of this capacity. The card used for this project was a GeForce GTX 990 TI.

3 EXISTING SYSTEM

There has been a lot of work in object detection using traditional computer vision techniques (sliding windows, deformable part models). However, they lack the accuracy of deep learning based techniques. Among the deep learning based techniques, two broad class of methods are prevalent: two stage detection (RCNN [1], Fast RCNN [2], Faster RCNN [3]) and uni ed detection (Yolo [4], SSD [5]). The major concepts involved in these techniques have been explained below.

3.1 Bounding Box

The bounding box is a rectangle drawn on the image which tightly ts the object in the image. A bounding box exists for every instance of every object in the image. For the box, 4 numbers (center x, center y, width, height) are predicted. This can be trained using a distance measure between predicted and ground truth bounding box. The distance measure is a jaccard distance which computes intersection over union between the predicted and ground truth boxes as shown in Fig.



3.2 Classification + Regression

The bounding box is predicted using regression and the class within the bounding box is predicted using classi cation. The overview of the architecture is shown in Fig. 4

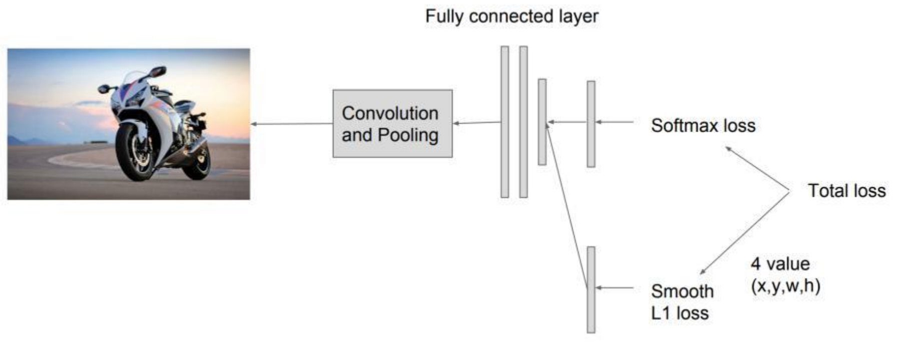
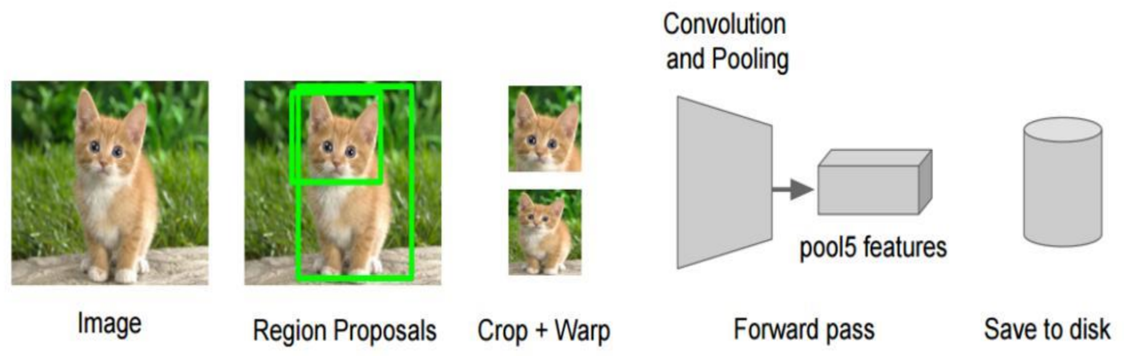


Figure : Architecture overview

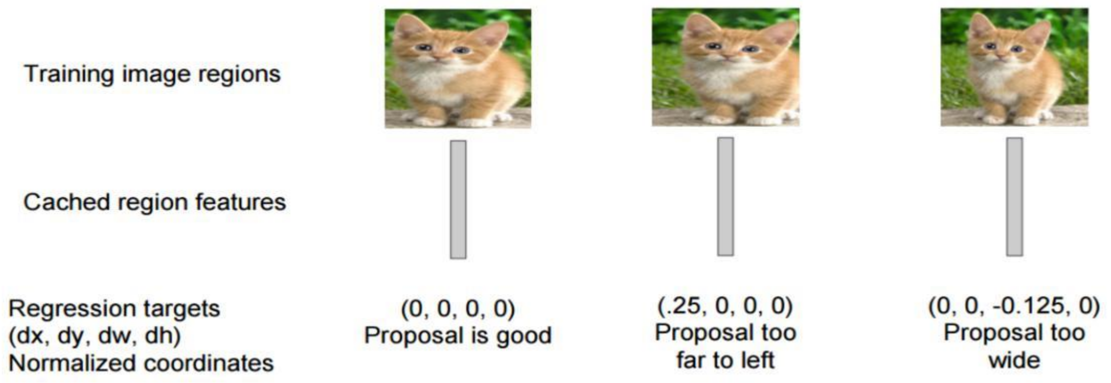
3.3 Two-stage Method

In this case, the proposals are extracted using some other computer vision technique and then resized to xed input for the classi cation network, which acts as a feature extractor. Then an SVM is trained to classify between object and background (one SVM for each class). Also a bounding box regressor is trained that outputs some some correction (o sets) for proposal boxes. The overall idea is shown in Fig. 5 These methods are very accurate but are computationally intensive (low fps).
sta

(a) Stage 1.



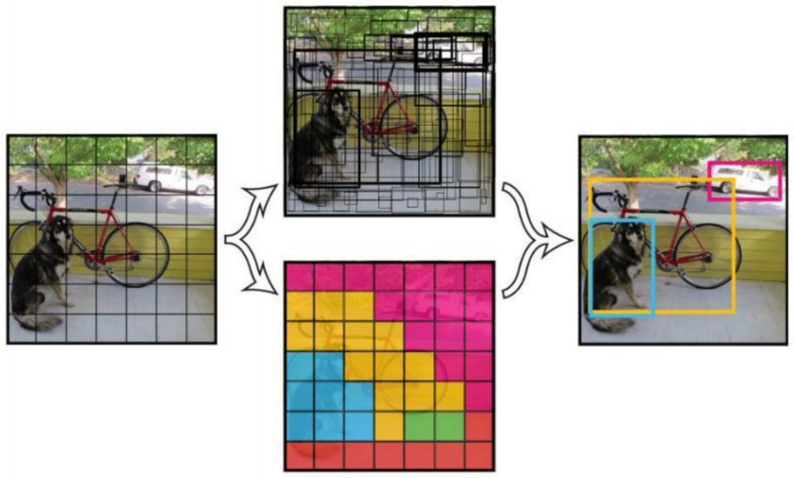
(b) Stage 2



3.4 Unified Method

The difference here is that instead of producing proposals, pre-de ne a set of boxes to look for objects. Using convolutional feature maps from later layers of the network, run another network over these feature maps to predict class scores and bounding box o sets. The broad idea is depicted in Fig. 6. The steps are mentioned below:

1. Train a CNN with regression and classi cation objective.
2. Gather activation from later layers to infer classi cation and location with a fully connected or convolutional layers.
3. During training, use jaccard distance to relate predictions with the ground truth.
4. During inference, use non-maxima suppression to lter multiple boxes around the same object.



The major techniques that follow this strategy are: SSD (uses di erent activation maps (multiple-scales) for prediction of classes and bounding boxes) and Yolo (uses a single ac-tivation map for prediction of classes and bounding boxes). Using multiple scales helps to achieve a higher mAP(mean average precision) by being able to detect objects with di erent sizes on the image better. Thus the technique used in this project is SSD.

4 <u>IMPLEMENTATION AND RESULTS</u>

4.1 Implementation Details

The project is implemented in python 3. Tensor ow was used for training the deep network and OpenCV was used for image pre-processing.

The system speci cations on which the model is trained and evaluated are mentioned as follows: CPU - Intel Core i7-7700 3.60 GHz, RAM - 32 Gb, GPU - Nvidia Titan Xp.

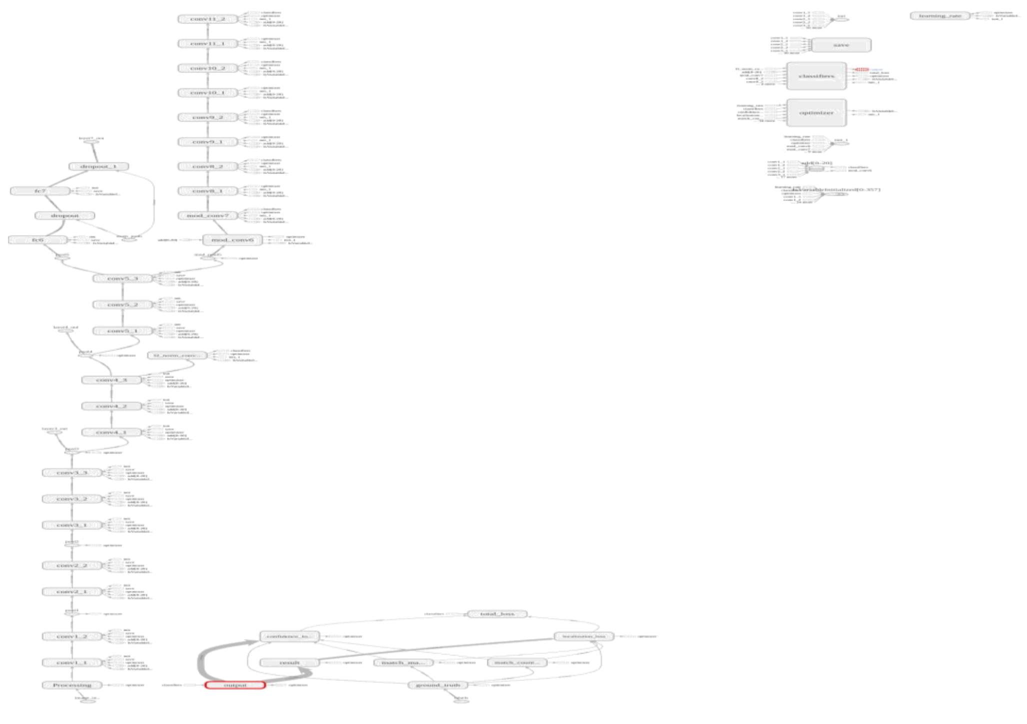
4.2 Pre-processing

The annotated data is provided in xml format, which is read and stored into a pickle le along with the images so that reading can be faster. Also the images are resized to a xed size.

4.3 Network

The entire network architecture is shown in Fig. The model consists of the base network derived from VGG net and then the modi ed convolutional layers for ne-tuning and then the classi er and localizer networks. This creates a deep network which is trained end-to-end on the data

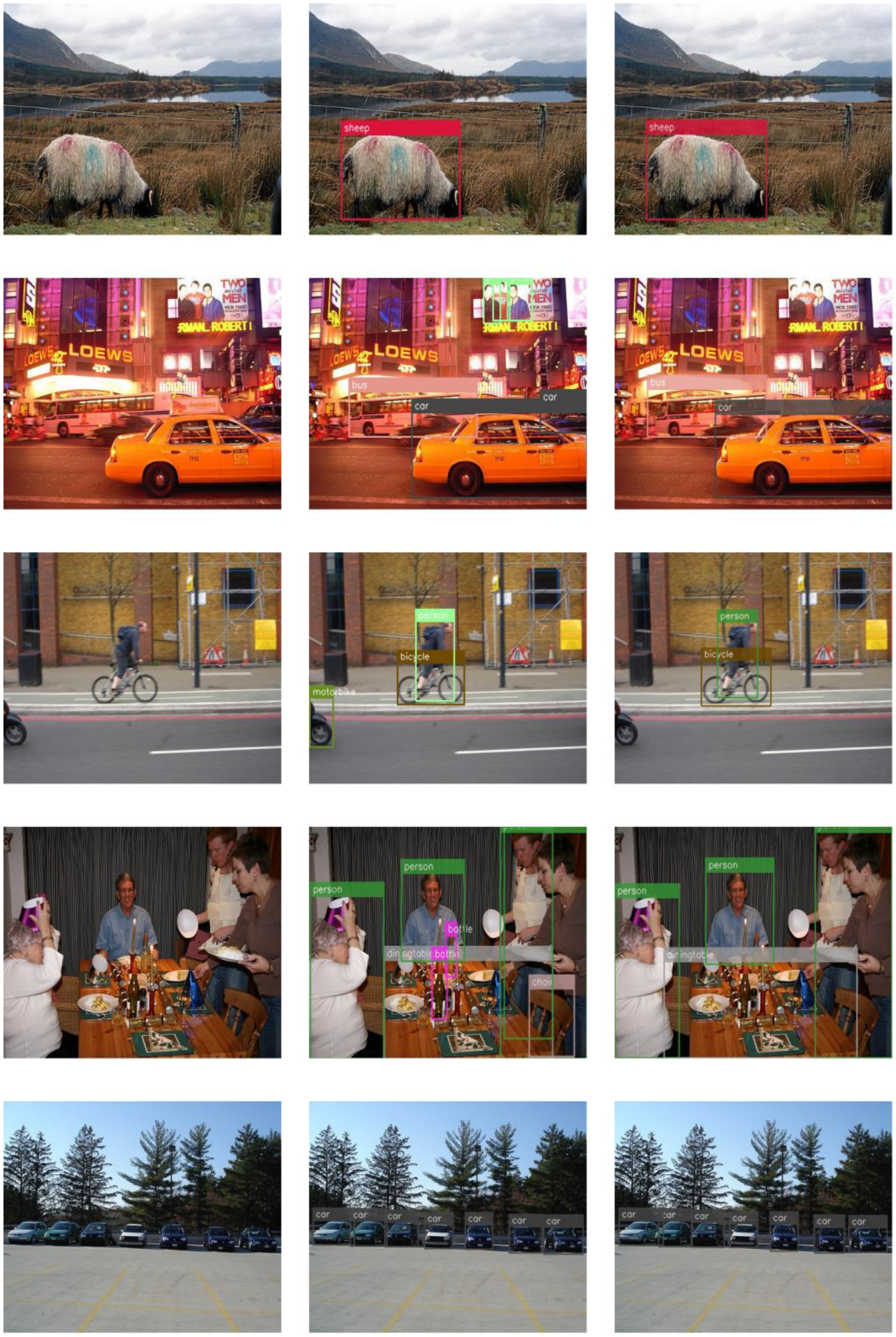
Fig:Network in tensor board



4.4 Qualitative Analysis

The results from the PASCAL VOC dataset are shown in Table.

INPUT **GROUND EARTH** **PREDICTION**



5 CONCLUSION

An accurate and efficient object detection system has been developed which achieves comparable metrics with the existing state-of-the-art system. This project uses recent techniques in the eld of computer vision and deep learning. Custom dataset was created using labelImg and the evaluation was consistent. This can be used in real-time applications which require object detection for pre-processing in their pipeline.

An important scope would be to train the system on a video sequence for usage in tracking applications. Addition of a temporally consistent network would enable smooth detection and more optimal than per-frame detection.

6 FUTURE WORKS

Computer vision is still a developing discipline, it has not been matured to that level where it can be applied directly to real life problems.

After few years computer vision and particularly the object detection would not be any more futuristic and will be ubiquitous. For now, we can consider object detection as a sub-branch of machine learning.

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 CustomObject class, you can tell the detection class to report detections on one or a few number 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

7 REFERENCES

- https://docs.python.org/3.8/library/index.html
- https://www.geeksforgeeks.org/introduction-to-multi-task-learningmtl-for-deep-learning/
- https://www.geeksforgeeks.org/introduction-machine-learning-using-python/
- https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/
- https://github.com/tirthajyoti/Machine-Learning-with-Python/blob/master/Pandas%20and%20Numpy/Numpy_Pandas_Quick.ipynb
- https://github.com/tirthajyoti/Machine-Learning-with-Python/blob/master/OOP_in_ML/Class_MyLinearRegression.ipynb