**PROJECT REPORT**

****

**Praxis Business School - Kolkata**

**Team 3 (Fall Batch 2021-22):**

1. Dhrubajyoti Chakraborty
2. Suvajyoti Chakraborty
3. Peyal Bhattacharjee
4. Ankit Singh
5. Gautam Kharbanda

**Abstract**

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advent of machine learning and deep learning techniques, the accuracy for object detection has increased drastically. The project aims to incorporate state-of-the-art technique for image object detection with the goal of achieving high accuracy with a real-time performance.

In this project, we use a completely machine learning with Matplotlib, SciPy, OpenCV and SSD (Single Shot Detector) in deep learning based approach to solve the problem of object detection. The network is trained on the sample dataset taken for ten (10) images captured from the Camera at different distances. The model outcome is executing fast and accurate results by creating Bounding Box for the detected object and resulting distance of the object accurate from the camera.

1. **Problem Statement:**

As we move towards more complete image understanding, having more precise and detailed object recognition becomes crucial. In this context, we not only look forward about image identification, but also about precisely estimating the class of object from the trained model.

One of the major problems was that of image classification, which is defined as predicting the class of the image. A slightly complicated problem is that of image localization, 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 complicated problem of object detection involves both classification and localization. It has the following objectives:

* Object detection, segmentation, location, and recognition
* Object tracking

1. **Application:**

**Clock Clock**

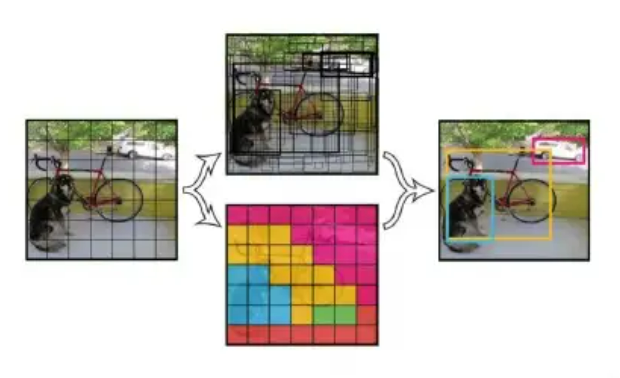


Figure 1.1 . Process of Image Detection in Computer Vision – In Figure 1.1, as it shows the process of image detection in computer vision. The first step is image input. The next steps involve the application of image processing techniques and extraction of the object. Using machine learning algorithm i.e., SSD and OpenCV, image recognition takes place and Bounding box is created. Finally, a description of the object is obtained as ‘clock’ and calculates distance from camera. In this process, image processing is a very important step. It involves converting the original image data to digital data for the use of the computer

1. **Object Detection Methodology:**

Using convolutional feature maps from later layers of the network, run another network over these feature maps to predict class scores and bounding box. The steps are mentioned below:

1. Train a CNN with regression and classification objective.
2. Gather activation from later layers to infer classification 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 filter multiple boxes around the same object.



Neural Network Deep learning used by the network has been constantly improving, in addition to the changes in the network structure, the more is to do some tune based on the original network or apply some trick to make the network performance to enhance. The more well-known algorithms of object detection are a series of algorithms based on CNN using OpenCV along-with SSD (Single Shot Detector) mainly in the following.

* **CNN**

CNN (Convolutional Neural Network) is the state-of-art methodology used for Image. One of the most noteworthy points of this paper is that the CNN is applied to the candidate box to extract the feature vector, and the second is to propose a way to effectively train large CNNs.

* **SSD**

The SSD is a single-shot detector. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. To improve accuracy, SSD introduces: small convolutional filters to predict object classes and offsets to default boundary boxes.

The network used in this project is based on Single shot detection (SSD).

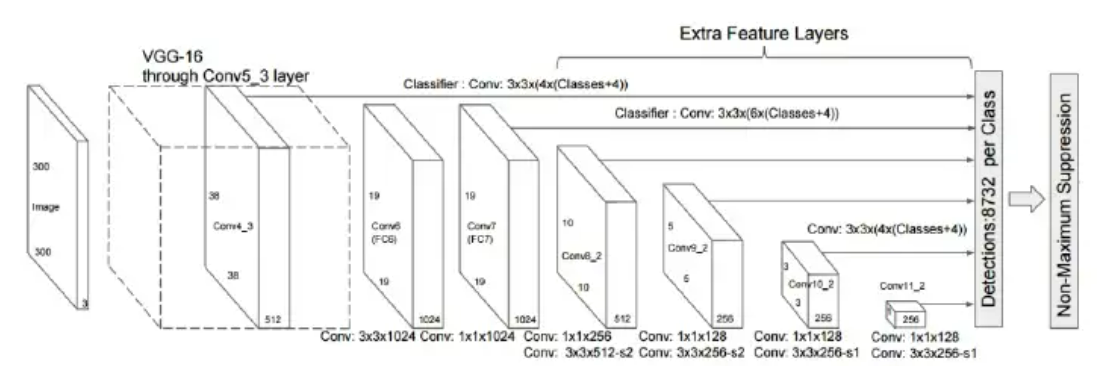


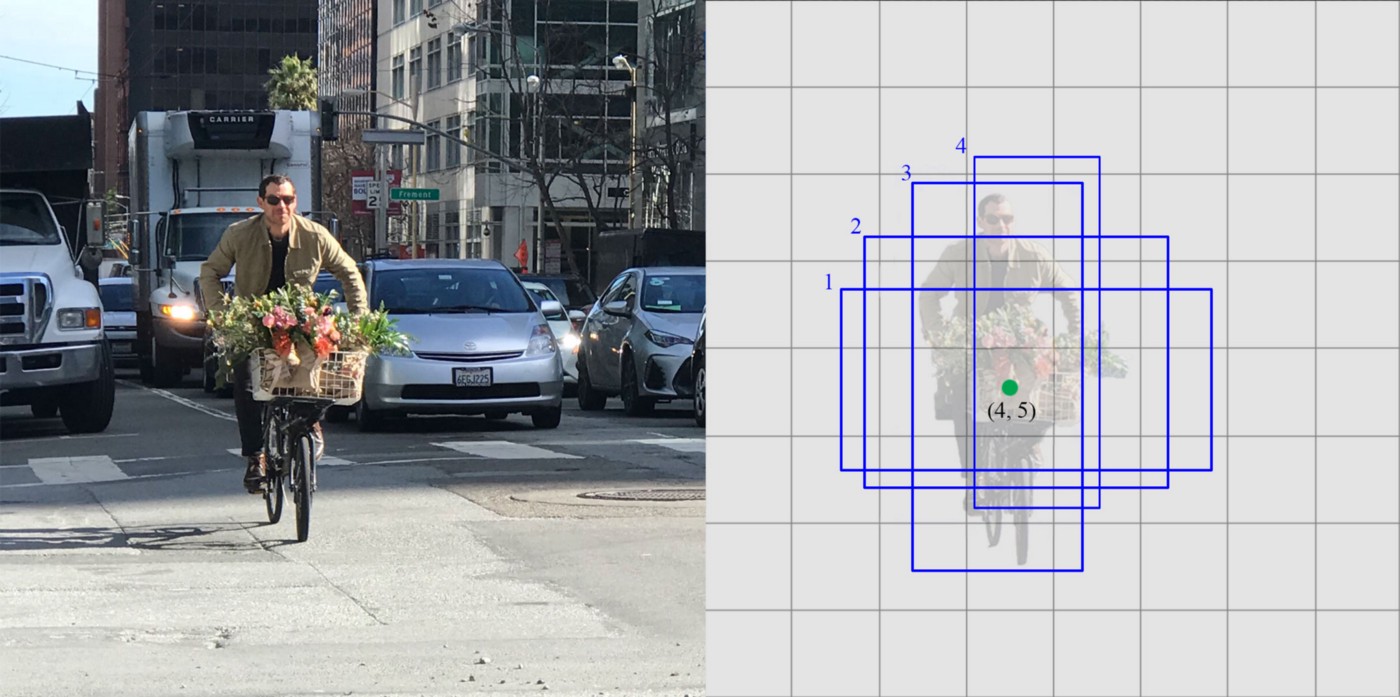
Figure 3.1: SSD Architecture

SSD normally starts with a VGG model, which is converted to a fully convolutional network. Then attach some extra convolutional layers are attached that help to handle bigger objects. The output at the VGG network is a 38x38 feature map. The added layers produce19x19, 10x10, 5x5, 3x3, 1x1 feature maps. All these feature maps are used for predicting bounding boxes at various scales (later layers responsible for larger objects).

SSD is designed for object detection in real-time. Faster R-CNN uses a region proposal network to create boundary boxes and utilizes those boxes to classify objects. While it is considered the start-of-the-art in accuracy, the whole process runs at 7 frames per second. Far below what real-time processing needs. SSD speeds up the process by eliminating the need for the region proposal network. To recover the drop in accuracy, SSD applies a few improvements including multi-scale features and default boxes. These improvements allow SSD to match the Faster R-CNN’s accuracy using lower resolution images, which further pushes the speed higher. According to the following comparison, it achieves the real-time processing speed and even beats the accuracy of the Faster R-CNN. (Accuracy is measured as the mean average precision mAP: the precision of the predictions.)

The SSD object detection composes of 2 parts:

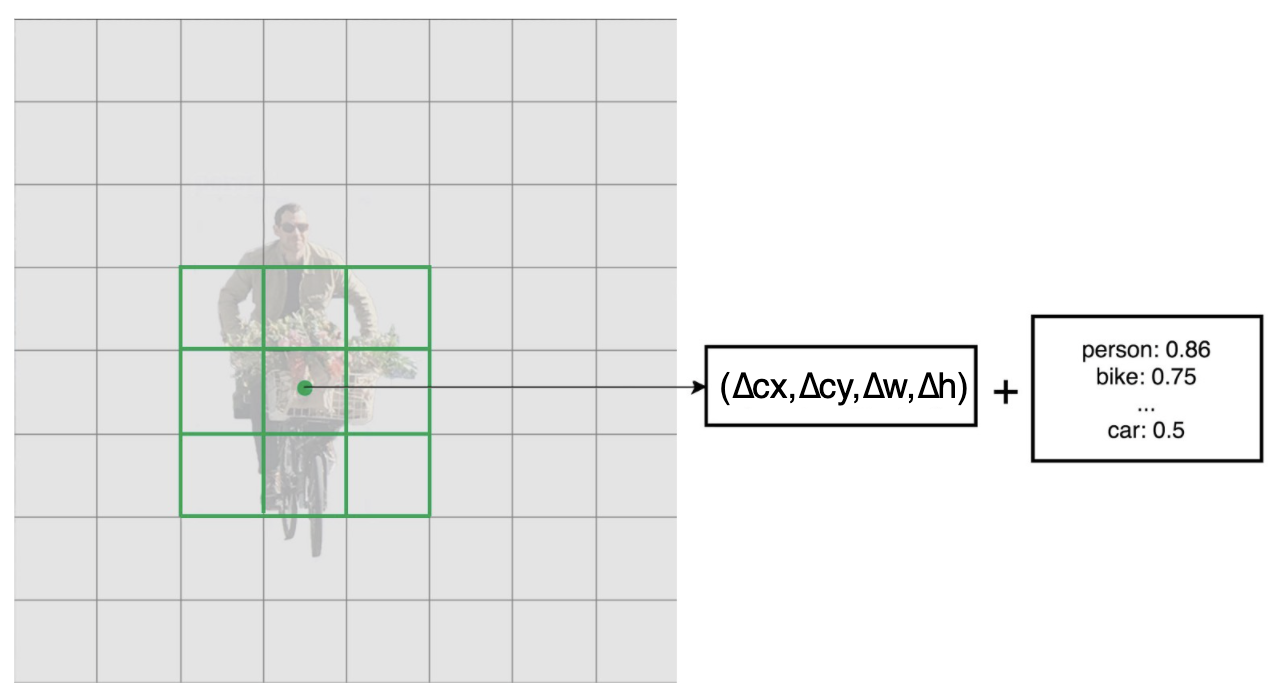
* Extract feature maps, and
* Apply convolution filters to detect objects.

Example – In below image, the original image is on the left side and predictions at each cell are on the right side.

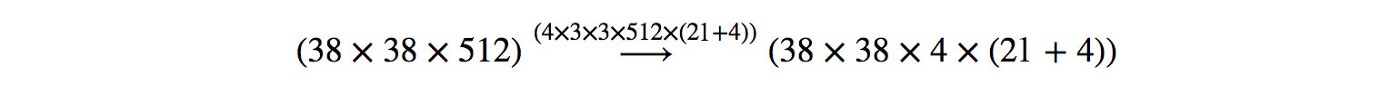
Each prediction composes of a boundary box and 21 scores for each class (one extra class for no object), and we pick the highest score as the class for the bounded object. Conv4\_3 makes a total of 38 × 38 × 4 predictions: four predictions per cell regardless of the depth of the feature maps. As expected, many predictions contain no object. SSD reserves a class “0” to indicate it has no objects.

**Convolutional predictors for object detection:**

SSD does not use a delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using small convolution filters. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter outputs 25 channels: 21 scores for each class plus one boundary box (detail on the boundary box later).



For example, in Conv4\_3, we apply four 3 × 3 filters to map 512 input channels to 25 output channels.



1. **Experiments and Results**

**4.1 Dataset-**

We evaluate the performance of the proposed approach on the dataset which contains fifty (50) test images captured over different five (5) distances. We first tried out to train our model on YOLO (You only look once (YOLO) and used a pre-trained model for object detection. However, when images were converted in greyscale, to calculate the Bounding Box, it was found that YOLO was not able to execute bounding box for small objects. Hence, it was not effective for all types of images dataset and Model accuracy was low. YOLO Model was dropped.

Then approach was set to apply SSD (Single Shot Detector) model. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. To improve accuracy, SSD introduces: small convolutional filters to predict object classes and offsets to default boundary boxes.

At test time an algorithm was trained with Images dataset of ten (10) different images which were captured from different set of distances i.e., 2.0 Feet, 2.5 Feet, 3 Feet, 3.5 Feet and 4.0 Feet. The distance measurement was considered 2.0 – 4.0 Feet in accordance to train our model with all size of objects well fitted in the model.

**4.2 Implementation Details**:

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

Object Detection Model Execution Steps:

* Import Matplotlib and CV2 - Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations. CV2 is Imported for OpenCV to solve computer vision problem.
* Apply SSD model – Using SSD Mobilenet and Tensorflow Object Detection, we use COCO dataset and Frozen Model to detect a single class from pretrained object classes
* Read images from CV2 and Plotting the Images – This method loads an image from the specified file and to read images are plotted during the execution to feed model of object detection
* Feed the model to create Bounding box – Using CV2, Bounding box is created in the images output from model processing
* Class identification and Confidence threshold – To check class identification from class index of pre-trained model and to check the confidence threshold
* Mapping the Area in Bounding Box – Using Puttext from CV2, Bounding box is creating with value generated of Object name from class index along-with Bounding Box is generated.
* Image Output with Object Name, Class Index, Area from Object Bounding Box, these details are collected and created Data-frame
* Calculating distance from Linear Interpolation using values from Data-frame

The identified object results from the above model and dataset are mapped in Table with area measured against the Distance. Dataset with Area Outcome as follows:

Now, we get the area outcome of each image at corresponding distances measured. We use the linear Interpolation using results in data-frame by detecting the object against the measured distance for each image.

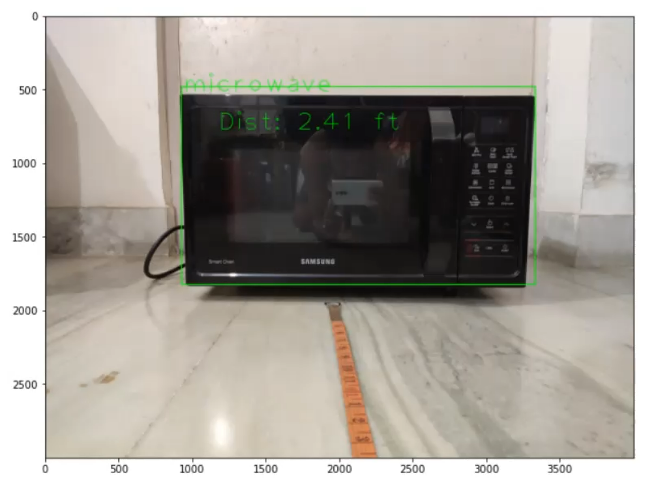
**Linear interpolation** involves estimating a new value by connecting two adjacent known values with a straight line. If the two known values are (x1, y1) and (x2, y2), then the y value for some point x is:

Equation for linear interpolation

**Using a pre-trained model for feature extraction**: Since, we are working with a small dataset, hence we have considered to take advantage of features learned by a model, which is already trained on a larger dataset in the same domain. Hence, we instantiated the pre-trained model. The pre-trained model is "frozen" and only the weights of the classifier get updated during training. In this case, we only trained a classifier that determines the image class given that set of extracted features and the convolutional base extracted all the features associated with each image.

**4.3 Result Analysis**

As we apply the above model by feeding images and run the algorithm, at first, class Index is detected and gets matched with image. Basis the match, outcome will index class in Data-frame and will pick corresponding area from data-frame

Using SciPy, on the basis of interpolation, in result image below, in the Bounding Box, we get the and exact distance from camera which is 2.41 Ft, measured by the model on each image feed.

**Conclusion**

An accurate and efficient object detection system has been developed which achieves comparable metrics with the existing system. This project uses recent techniques in the field of computer vision and deep learning. Hence, we conclude the model applicability on the basis accurate and acceptable results.