Object Detection and Localization

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Engineering

by

Deep Dama
Roll No. 07
Durwankur Gursale
Roll No. 17
Anuja Jadhav
Roll No. 22

Under the Supervision of
Prof. A.Palsodkar

DEPARTMENT OF INFORMATION TECHNOLOGY
KONKAN GYANPEETH COLLEGE OF ENGINEERING
KARJAT-410201
May 2021

Certificate

This is to certify that the project entitled Object Detection and Localization is a bonafide work of DEEP DAMA (Roll No.07), ANUJA JADHAV (Roll No.22), DURWANKUR GURSALE (ROLL No.17) submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of Undergraduate in DEPARTMENT OF INFORMATION TECHNOLOGY.

.

Supervisor/Guide

Professor A.Palsodkar

Department of Information Technology

Head of Department

Department of Information Technology

Principal

Dr. M.J. Lengare

Konkan Gyanpeeth College of Engineering

Project Report Approval for B.E.

This project report Object Detection and Localization by DEEP DAMA (Roll No.07), ANUJA JADHAV (Roll No.22), DURWANKUR GURSALE (ROLL No.17) is approved for the degree of DEPARTMENT OF INFORMATION TECHNOLOGY.

.

Examiners

1.....

2.....

Date.

Place.

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data /fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature
DEEP DAMA
Roll No. 07

Signature
DURWANKUR GURSALE
Roll No. 17

Signature
ANUJA JADHAV
Roll No. 22

Date.

Abstract

Object localization refers to identifying the location of one or more objects in an image and drawing a bounding box around their extent. Image classification involves predicting the class of one object in an image. Object detection combines these two tasks and localizes and classifies one or more objects in an image. Object detection is one of the areas of computer vision that is maturing very rapidly. Today, there is a plethora of pre-trained models for object detection (YOLO, RCNN, Fast RCNN, Mask RCNN, Multi-box etc.). So, it only takes a small amount of effort to detect most of the objects in a video or in an image.

Acknowledgements

Apart from the efforts of the team, the success of this mini project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We are highly indebted to our project guide Prof. A. Palsodkar, for his guidance and constant support. We can't thank enough for his tremendous support and help. We feel motivated and encouraged every time we attended his meeting. Without his encouragement and guidance this project would not have materialized.

We are thankful to (Dr. M.J. Lengare), Principal KGCE, for his encouraging attitude.

Finally, yet importantly, we would like to express my heartfelt thanks to our beloved parents for their blessings, and friends and all others for their help and wishes for the successful completion of this mini project.

Contents

Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
YOLO	**Y**ou **O**nly **L**ook **O**nce
mAP	**m**ean **A**verage **P**recision
IOU	**I**ntersection **O**ver **U**nion

Chapter 1

INTRODUCTION

1.1 Introduction

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs. Image classification involves predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. Object detection combines these two tasks and localizes and classifies one or more objects in an image.

We can see that "Single-object localization" is a simpler version of the more broadly defined "Object Localization," constraining the localization tasks to objects of one type within an image, which we may assume is an easier task. The performance of a model for image classification is evaluated using the mean classification error across the predicted class labels. The performance of a model for single-object localization is evaluated using the distance between the expected and predicted bounding box for the expected class. Whereas the performance of a model for object recognition is evaluated using the precision and recall across each of the best matching bounding boxes for the known objects in the image.

Object detection is widely used in many fields. For example, in self-driving technology, we need to plan routes by identifying the locations of vehicles, pedestrians,

roads, and obstacles in the captured video image. Robots often perform this type of task to detect targets of interest. Systems in the security field need to detect abnormal targets, such as intruders or bombs. In object detection, we usually use a bounding box to describe the target location. The bounding box is a rectangular box that can be determined by the xx and yy axis coordinates in the upper-left corner and the xx and yy axis coordinates in the lower-right corner of the rectangle. We will define the bounding boxes of the dog and the cat in the image based on the coordinate information in the above image.

1.2 Objectives

- To detect all instances of objects from a known class, such as people, cars or faces in an image.
- Object detection systems construct a model for an object class from a set of training examples.
- Identifying the type of object in an image and also exact location of the object inside image.
- To analyze scenes in an image or video.

1.3 Purpose, Scope, and Applicability

Purpose scope and application. The description of purpose scope and application are given below:

1.3.1 Purpose

The main purpose of object detection is to detect all instances of objects from a known class such as people cars or faces in an image. Object detection is a key ability required by most computer and robot vision systems. The latest research

on this area has been making great progress in many directions. In many computer vision systems, object detection is the first task being performed as it allows to obtain further information regarding the detected object and about the scene.

1.3.2 Scope

- The scope of this project is to detect all instances of objects from a known class such as people cars or faces in an image.
- Once an object instance has been detected (e.g., a face), it is be possible to obtain further information, including: to recognize the specific instance (e.g., to identify the subject's face), to track the object over an image sequence (e.g., to track the face in a video), and to extract further information about the object (e.g., to determine the subject's gender)
- The system developed in this project is such that it will add a bounding box to locate an object in an image once it is detected.

1.3.3 Applicability

Object detection has immense areas of applicability, we list some of them:

- Crowd counting
- Self-driving cars
- Face Detection
- Anomaly detection
- Video Surveillance

1.4 Organisation of Report

The Report is organized into eight chapters. After this introductory chapter, Chapter 2 lists all research papers (Literature Survey) referred and their methodologies. Chapter 3 gives a brief description about technologies and methods which we will be using to develop our system. Chapter 4 contains information about the model we will be using. Chapter 5 presents problem statement and all the required hardware and software specification. It Chapter 6 Consists of Diagrams. Chapter 7 Consists of User Interface Design and testing results. Finally Chapter 8 conclude the report.

Chapter 2

LITERATURE SURVEY

In this chapter we survey previous research done on object detection and localization, we have studied about following papers published by some experts.

1) Object Detection Algorithm based on improved YOLO v3
(Author: Liquan Zhao, Shuaiyang Li)

The 'You Only Look Once' v3 (YOLOv3) method is among the most widely used deep learning-based object detection methods. It uses the k-means cluster method to estimate the initial width and height of the predicted bounding boxes. With this method, the estimated width and height are sensitive to the initial cluster centers, and the processing of large-scale datasets is time-consuming. In order to address these problems, a new cluster method for estimating the initial width and height of the predicted bounding boxes has been developed. Firstly, it randomly selects a couple of width and height values as one initial cluster center separate from the width and height of the ground truth boxes. Secondly, it constructs Markov chains based on the selected initial cluster and uses the final points of every Markov chain as the other initial centers. In the construction of Markov chains, the intersection-over-union method is used to compute the distance between the selected initial clusters and each candidate point, instead of the square root method. Finally, this

method can be used to continually update the cluster center with each new set of width and height values, which are only a part of the data selected from the datasets.

2) You Only Look Once: Unified, Real-Time Object Detection

(Author: Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi)

Authors present YOLO, an approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. The unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second.

A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state- of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

3) YOLO9000: Better, Faster, Stronger

(Author: Joseph Redmon, Ali Farhadi)

In this authors introduce YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories. First they propose various improvements to the YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL VOC and COCO. Using a novel, multi-scale training method the same YOLOv2 model can run at varying sizes, offering an easy tradeoff between

speed and accuracy. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP, outperforming state-of-the-art methods like Faster RCNN with ResNet and SSD while still running significantly faster. Finally they propose a method to jointly train on object detection and classification. Using this method they train YOLO9000 simultaneously on the COCO detection dataset and the ImageNet classification dataset. Their joint training allows YOLO9000 to predict detections for object classes that don't have labelled detection data. They validate their approach on the ImageNet detection task. YOLO9000 gets 19.7 mAP on the ImageNet detection validation set despite only having detection data for 44 of the 200 classes. On the 156 classes not in COCO, YOLO9000 gets 16.0 mAP. But YOLO can detect more than just 200 classes; it predicts detections for more than 9000 different object categories. And it still runs in real-time.

Chapter 3

SURVEY OF TECHNOLOGIES

In this chapter Survey of Technologies we demonstrate our awareness and understanding of Available Technologies related to the topic of our project. Given below are the details of all the related technologies that are necessary to complete our project.

3.1 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and predictive maintenance. Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm a data scientist chooses to use depends on what type of data they want to predict.

- Supervised learning. In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

- Unsupervised learning. This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. Both the data algorithms train on and the predictions or recommendations they output are predetermined.

- Semi-supervised learning. This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

- Reinforcement learning. Reinforcement learning is typically used to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

3.2 Convolutional Neural Network

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a

specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis. A neural network contains layers of interconnected nodes.

3.3 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object- relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

3.4 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. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now.

3.5 NumPy

NumPy (is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. [5] The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Chapter 4

MODELS

4.1 YOLO

- The "You Only Look Once," or YOLO, family of models are a series of end-to-end deep learning models designed for fast object detection, developed by Joseph Redmon, et al. and first described in the 2015 paper titled "You Only Look Once: Unified, Real-Time Object Detection."

- The approach involves a single deep convolutional neural network (originally a version of GoogLeNet, later updated and called DarkNet based on VGG) that splits the input into a grid of cells and each cell directly predicts a bounding box and object classification.

- There are three main variations of the approach, at the time of writing; they are YOLOv1, YOLOv2, and YOLOv3. The first version proposed the general architecture, whereas the second version refined the design and made use of predefined anchor boxes to improve bounding box proposal, and version three further refined the model architecture and training process.

- YOLO can work well for multiple objects where each object is associated with one grid cell. But in the case of overlap, in which one grid cell actually contains the centre points of two different objects, we can use something called anchor

boxes to allow one grid cell to detect multiple objects.

YOLO in action

Chapter 5

REQUIREMENTS AND ANALYSIS

5.1 Problem Definition:

To build a system that will detect all instances of objects from a known class such as people cars or faces in an image.

Sub-problem:

1. To detect objects from several different classes
2. To classify multiple objects from a single image.
3. To create a bounding box for the images detected.

5.2 Requirements Specification

In this phase we define the requirements of the system. The Requirements Specification describes the things in the system and the actions that can be done on these things.The requirements of the system are:

- The image from the dataset and the dataset should have minimum two Phases including one for training and the second for testing.
- A system or model to train and test the dataset.

5.3 Software and Hardware Requirements

Hardware: A computer system having a multi-corer, minimum of 8GB RAM. Storage of minimum 500 GB and input and output peripherals.
Software: 1) Python 2) Anaconda 3) Jupyter Notebook

5.4 Evaluation Metrics

Precision and recall: Precision – It is used to measure the correct predictions. Recall – it is used to calculate the true predictions from all correctly predicted data.

Intersection over Union(IOU): IOU is a metric that finds the difference between ground truth annotations and predicted bounding boxes. This metric is used in most state of art object detection algorithms. In object detection, the model predicts multiple bounding boxes for each object, and based on the confidence scores of each bounding box it removes unnecessary boxes based on its threshold value.

Average Precision(AP): To evaluate the detection commonly we use precision-recall curve but average precision gives the numerical values it is easy to compare the performance with other models. Based on the precision-recall curve AP it summarises the weighted mean of precisions for each threshold with the increase in recall. Average precision is calculated for each object.

Mean Average Precision(mAP): Mean average precision is an extension of Average precision. In Average precision, we only calculate individual objects but in

mAP, it gives the precision for the entire model. To find the percentage correct predictions in the model we are using mAP.

Variations among mAP: In most of the research papers, these metrics will have extensions like mAP iou = 0.5, mAP iou = 0.75, mAP small, medium, large.

5.5 Planning and Scheduling

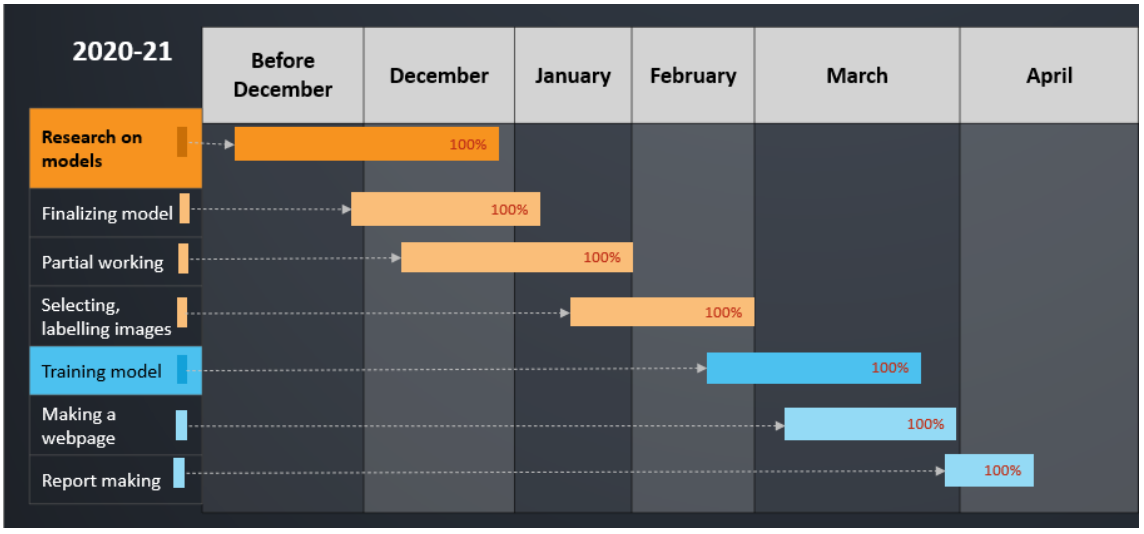

FIGURE 5.1: Gantt Chart

Chapter 6

SYSTEM DESIGN

6.1 Basic Modules

A module is a collection of source files and build settings that allow the division of the project into discrete units of functionality. The project can have one or many modules and one module may use another module as a dependency. Each module can be independently built, tested, and debugged. Additional modules are often useful when creating code libraries within the project or when we want to create different sets of code and resources for different device types, such as phone and wearable, but keep all the files scoped within the same project and share some code.

6.2 Network Architecture

FIGURE 6.1: Network Architecture

6.3 Flow chart

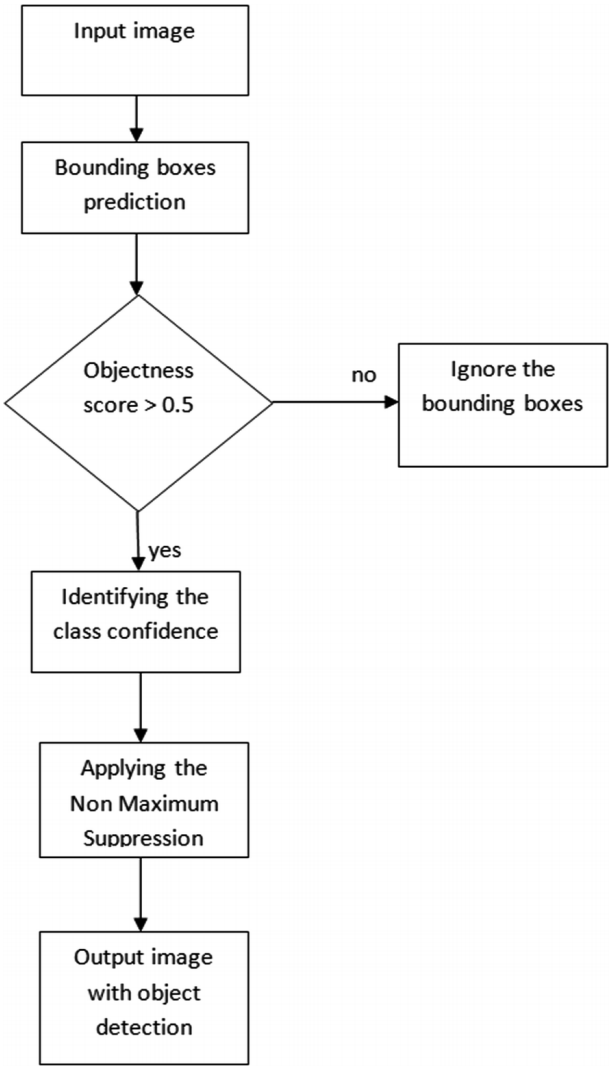

FIGURE 6.2: Flow Chart

Chapter 7

IMPLEMENTATION

Layout

FIGURE 7.1: Home page

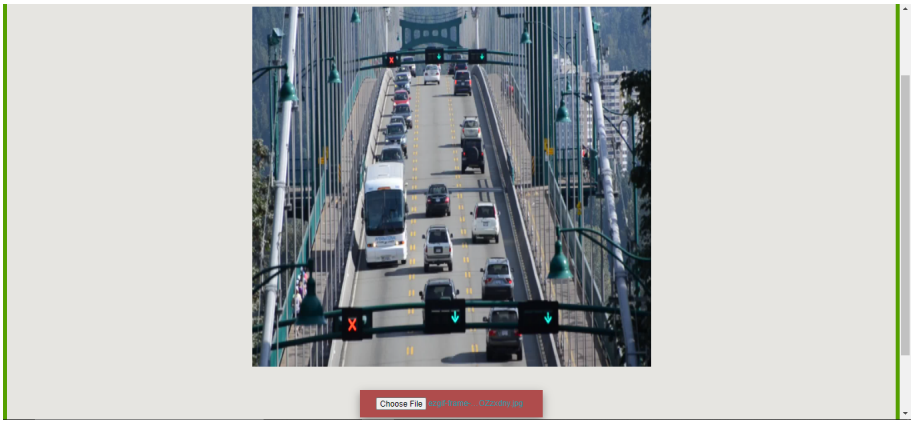

FIGURE 7.2: Uploading a picture

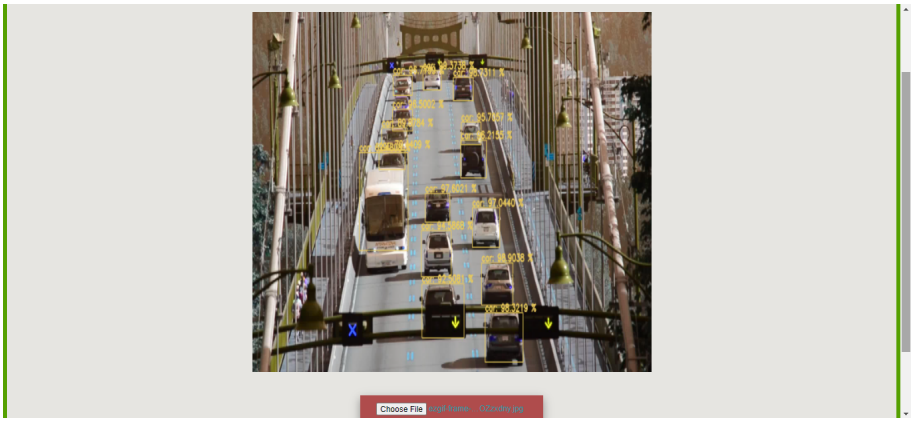

FIGURE 7.3: Output

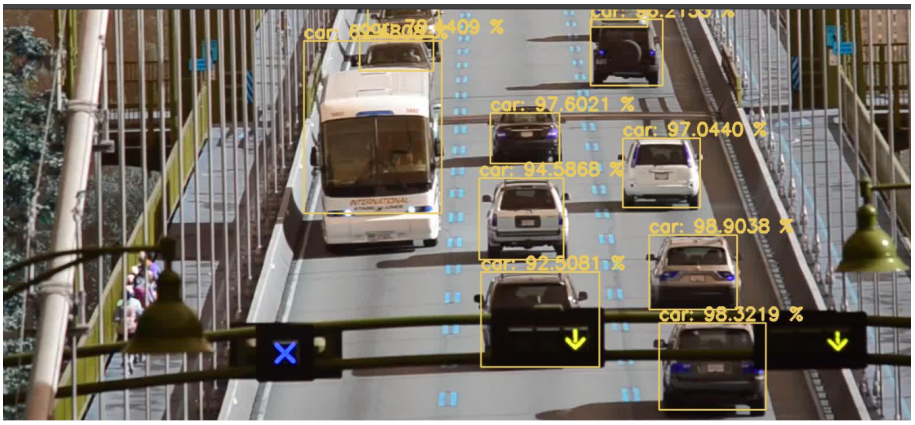

FIGURE 7.4: Output

FIGURE 7.5: Output

FIGURE 7.6: Output

Chapter 8

CONCLUSION

8.1 CONCLUSION

Object detection has many real-world use cases today. For instance, object detection models are capable of tracking multiple people at once, in real-time, as they move through a given scene or across video frames. From retail stores to industrial factory floors, this kind of granular tracking could provide invaluable insights into security, worker performance and safety, retail foot traffic, and more. After researching through various papers related to Object Detection and Localization, we took a basic object detection model and improved its accuracy to detect images from different classes and produce a bounding box around it. Comparing between several models, we decided to use YOLO because of its speed and convenience for real time object detection.

Bibliography

[1] Akansha Bathija M.Tech Student, P. G. S. (2004). Visual object detection and tracking using yolo and sort. M.tech thesis, K J Somaiya College of Engineering Mumbai.

[2] Chakraborty, S. (2006). Real time object detection using yolov3. M.tech thesis, Indian Institute of Technology Kanpur.

[3] Girshick, R. (2015). Fast r-cnn. 112(5):1166–1181.

[4] Joseph Redmon, A. F. (2016a). Yolo v2. 20(1):105–112.

[5] Joseph Redmon, Santosh Divvala, R. G. A. F. (2016b). Yolo v1. 36(22):3915–3932.

[6] Omkar Masurekar, O. J. (2017). Feature pyramid networks for object detection.

[7] Pierre Sermanet, David Eigen, X. Z. M. M. R. F. Y. L. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks.

[8] Shaoqing Ren, Kaiming He, R. G. J. S. (2016). Faster r-cnn. 114(12):2083–2092.

[9] Wei Liu, Dragomir Anguelov, D. E. C. S. S. R. C.-Y. F. A. C. B. (2016). Ssd. Master's thesis.