Object Detection and Localization

Submitted in partial fulfilment of the requirements

For the degree of

Bachelor of Engineering

Synopsis Report

By

Deep Dama

Roll No: - 07

Anuja Jadhav

Roll No: - 22

Durwankur Gursale

Roll No: - 17

Under the Supervision of

Prof. A.Palsodkar

DEPARTMENT OF INFORMATION TECHNOLOGY
KONKAN GYANPEETH COLLEGE OF ENGINEERING,
KARJAT-410201
November 2020

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
Prof.A.Palsodkar
Department of Information Technology

Head of Department	Principal
Prof. J.P.Patil	Dr. M.J. Lengare
(Department of Information Technology)	(Konkan Gyanpeeth College of Engineering)

Project Report Approval

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 I 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

ANUJA JADHAV

(Roll No.22)

Signature
DURWANKUR GURSALE
(Roll No.17)

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.

Acknowledgement

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

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

I take this opportunity to convey our sincere thanks to (Prof. J.P. Patil), (Head of Department), for providing guidance and whole-hearted cooperation. I am thankful to (Dr. M. J. Lengare), Principal KGCE, for his encouraging attitude.

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

Signature
DEEP DAMA
(Roll No.07)

Signature

ANUJA JADHAV

(Roll No.22)

Signature
DURWANKUR GURSALE
(Roll No.17)

CONTENTS

Contents

Abbreviations

CNN: Convolutional Neural Network

YOLO: You Only Look Once

Chapter 1

INTRODUCTION

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.

OBJECTIVES:

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

PURPOSE, SCOPE AND APPLICATION:

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

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.

SCOPE:

1. 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.
2. 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)
3. 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

APPLICABILITY:

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

1. Crowd counting
2. Self-driving cars
3. Face Detection
4. Anomaly detection
5. Video Surveillance

Chapter 2
LITERATURE SURVEY

Literature Survey

Literature Survey: Is the process of analysing, summarizing, organizing and presenting novel conclusions from the results of technical review of large number of recently published scholarly articles. In this chapter we survey previous research done on object detetction and localization, we have studied about following papers published by some experts.

1) OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks

(Author: Pierre Sermanet David Eigen Xiang Zhang Michael Mathieu Rob Fergus Yann LeCun)

In this paper, the author presented an integrated framework for using Convolutional Networks for classification, localization and detection. They also show how a multiscale and sliding window approach can be efficiently implemented within a ConvNet, also introduce a novel deep learning approach to localization by learning to predict object boundaries. Bounding boxes are then accumulated rather than suppressed in order to increase detection confidence. They show that different tasks can be learned simultaneously using a single shared network. This integrated framework is the winner of the localization task of the ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013) and obtained very competitive results for the detection and classifications tasks. In post-competition work, they establish a new state of the art for the detection task. Finally, release a feature extractor from their best model called OverFeat.

2) Region-based Convolutional Networks for Accurate Object Detection and Segmentation

(Author: Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik)

In this paper, authors propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50% relative to the previous best result on VOC 2012—achieving a mAP of 62.4%. Their approach combines two ideas:

(1) one can apply high-capacity convolutional networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data are scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, boosts performance significantly. Since they combine region proposals with CNNs, they call the resulting model an R-CNN or Region-based Convolutional Network.

3) Fast R-CNN

(Author: Ross Girshick)

This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN helps efficiently classify object proposals using deep convolutional networks. Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3× faster, tests 10× faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License

4) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

(Author: Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun)

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, authors introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. They further merge RPN and Fast R-CNN into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look

5) 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.

6) SSD: Single Shot MultiBox Detector

(Author: Wei Liu1, Dragomir Anguelov2, Dumitru Erhan3, Christian Szegedy3, Scott Reed4, Cheng-Yang Fu1, Alexander C. Berg)

Authors present a method for detecting objects in images using a single deep neural network. Their approach, named SSD, discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Experimental results on the PASCAL VOC, COCO, and ILSVRC datasets confirm that SSD has competitive accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference.

7) Feature Pyramid Networks for Object Detection

(Author: Tsung-Yi Lin1, Piotr Dollar , Ross Girshick , Kaiming He1 , Bharath Hariharan, and Serge Belongie)

In this Feature pyramids are a basic component in recognition systems for detecting objects at different scales. But recent deep learning object detectors have avoided pyramid representations, in part because they are compute and memory intensive. In this paper, the authors exploit the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. A topdown architecture with lateral connections is developed for building high-level semantic feature maps at all scales. This architecture, called a Feature Pyramid Network (FPN), shows significant improvement as a generic feature extractor in several applications. Using FPN in a basic Faster R-CNN system, this method achieves state-of-the-art singlemodel results on the COCO detection benchmark without bells and whistles, surpassing all existing single-model entries including those from the COCO 2016 challenge winners. In addition, this method can run at 6 FPS on a GPU and thus is a practical and accurate solution to multi-scale object detection.

8) 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.

9) Focal Loss for Dense Object Detection

(Author: Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollar)

The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN, where a classifier is applied to a sparse set of candidate object locations. In contrast, one-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far.

In this paper, we investigate why this is the case. Authors discover that the extreme foreground-background class imbalance encountered during training of dense detectors is the central cause. They propose to address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. The novel Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of their loss, they design and train a simple dense detector they call RetinaNet. Their results show that when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors.

10) Objects as Points

(Author: Xingyi Zhou, Dequan Wang, Philipp Krahenb ¨ uhl)

Detection identifies objects as axis-aligned boxes in an image. Most successful object detectors enumerate a nearly exhaustive list of potential object locations and classify each. This is wasteful, inefficient, and requires additional post-processing. In this paper, authors take a different approach. They model an object as a single point — the center point of its bounding box. Their detector uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and even pose. Their center point based approach, CenterNet, is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box based detectors. CenterNet achieves the best speed-accuracy trade-off on the MS COCO dataset, with 28.1% AP at 142 FPS, 37.4% AP at 52 FPS, and 45.1% AP with multi-scale testing at 1.4 FPS. They use the same approach to estimate 3D bounding box in the KITTI benchmark and human pose on the COCO keypoint dataset. Their method performs competitively with sophisticated multi-stage methods and runs in real-time.

11) 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.

Paper Comparison :

Sr. No	Paper Name	Author Names	Description
1)	OverFeat: Integrated Recognition, Localization and Detection	Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu Rob, Fergus Yann LeCun	The author presented an integrated framework for using Convolutional Networks for classification, localization and detection. They also show how a multiscale and sliding window approach can be efficiently implemented within a ConvNet. Bounding boxes are then accumulated rather than suppressed in order to increase detection confidence.
2)	Region-based Convolutional Networks for Accurate Object Detection and Segmentation	Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik	In this paper, authors propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50% relative to the previous best result on VOC 2012—achieving a mAP of 62.4%.
3)	Fast R-CNN	Ross Girshick	Fast R-CNN helps efficiently classify object proposals using deep convolutional networks. Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy.
4)	Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks	Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun	Authors introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position.
5)	You Only Look Once: Unified, Real-Time Object Detection	Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi	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.

6)	SSD: Single Shot MultiBox Detector	Wei Liu1, Dragomir Anguelov2, Dumitru Erhan3, Christian Szegedy3, Scott Reed4, Cheng-Yang Fu1, Alexander C. Berg	SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape.
7)	Feature Pyramid Networks for Object Detection	Tsung-Yi Lin1, Piotr Dollar , Ross Girshick , Kaiming He1 , Bharath Hariharan, and Serge Belongie	Feature Pyramid Network (FPN), shows significant improvement as a generic feature extractor in several applications. Using FPN in a basic Faster R-CNN system, this method achieves state-of-the-art singlemodel results on the COCO detection
8)	YOLO9000: Better, Faster, Stronger	Joseph Redmon, Ali Farhadi	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
9)	Focal Loss for Dense Object Detection	Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollar	Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of their loss, they design and train a simple dense detector they call RetinaNet.
10)	Objects as Points	Xingyi Zhou, Dequan Wang, Philipp Krahenb ¨ uhl	In this paper the authors model an object as a single point — the center point of its bounding box. Their detector uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and even pose
11)	Object Detection Algorithm based on improved YOLO v3	Liquan Zhao, Shuaiyang Li	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

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.

- ### 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.

• **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.

• **Tensorflow**

TensorFlow is a framework created by Google for creating Deep Learning models. Deep Learning is a category of machine learning model. Machine Learning has enabled us to build complex applications with great accuracy. Whether it has to do with images, videos, text or even audio, Machine Learning can solve problems from a wide range. Tensorflow can be used to achieve all of these applications.

• 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.

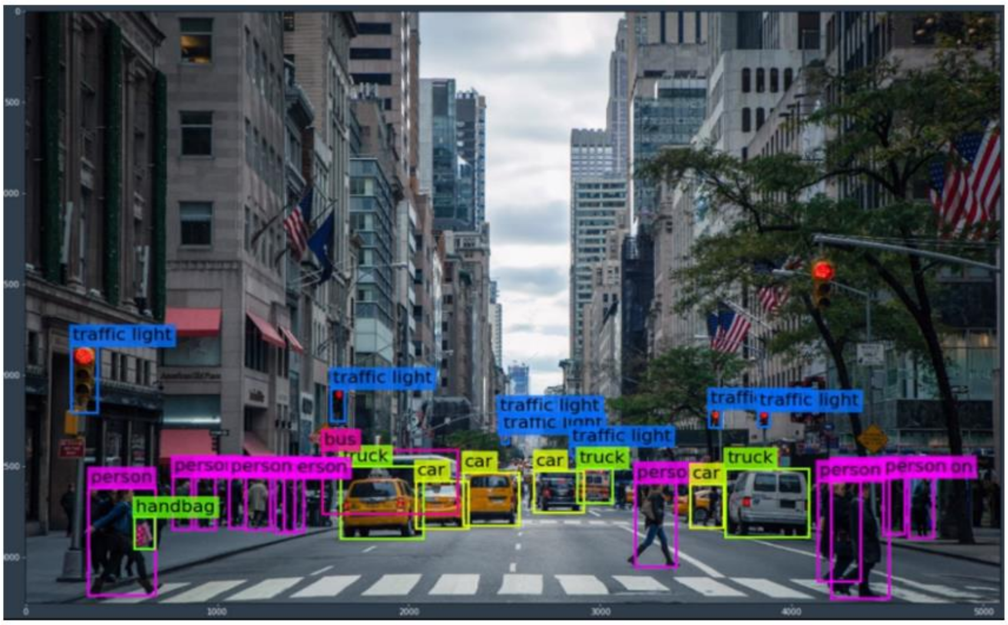
• NumPy

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

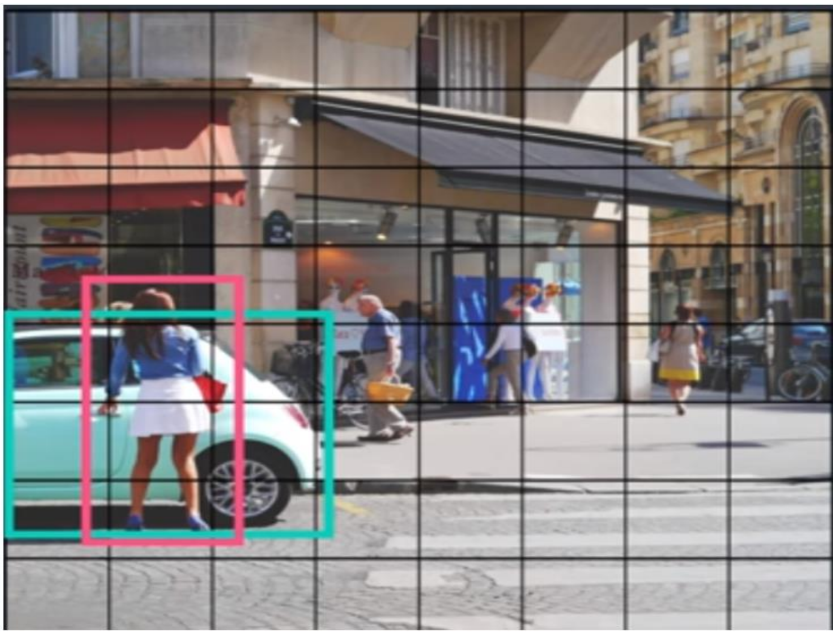
❖ YOLO

- ➢ 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.
- ➢ Although the accuracy of the models is close but not as good as Region-Based Convolutional Neural Networks (R-CNNs), they are popular for object detection because of their detection speed, often demonstrated in real-time on video or with camera feed input.



YOLO in action

- 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.



Anchor Boxes in action

- In image above, we see that we have a person and a car overlapping in the image. So, part of the car is obscured. We can also see that the centres of both bounding boxes, the car, and the pedestrian fall in the same grid cell. Since the output vector of each grid cell can only have one class, then it will be forced to pick either the car or the person.

Chapter 5

REQUIREMENTS AND ANALYSIS

Problem Definition

In this section we define the problem on which we are working in the project. Details are provided of the overall problem and then divided the problem in to sub-problems.

a) 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:

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

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:

1) The image from the dataset and the dataset should have minimum two Phases including one for training and the second for testing.
2) A system or model to train and test the dataset.
3) High level API such as Tensor Flow.

Software and Hardware Requirements:

Hardware:

1) 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

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.

Chapter 6
Implementation

1)

Build a dataset

The dataset contains annotations for clothing items - bounding boxes around shirts, tops, jackets, sunglasses. The dataset is from DataTurks and is on Kaggle.

```
!gdown --id 1uWdQ2kn25RSQITtBHa9_zayplm27IXNC
```

It downloads the data set from google drive.

Output:

```
Downloading...
From: https://drive.google.com/uc?id=1uWdQ2kn25RSQITtBHa9_zayplm27IXNC
To: /content/clothing.json
100% 199k/199k [00:00<00:00, 62.4MB/s]
```

The dataset contains a single JSON file with URLs to all images and bounding box data.

2)
Importing required libraries

```
from pathlib import Path
from tqdm import tqdm
import numpy as np
import json
import urllib
import PIL.Image as Image
import cv2
import torch
import torchvision
from IPython.display import display
from sklearn.model_selection import train_test_split

import seaborn as sns
from pylab import rcParams
import matplotlib.pyplot as plt
from matplotlib import rc
```

```
%matplotlib inline
%config InlineBackend.figure_format='retina'
sns.set(style='whitegrid', palette='muted', font_scale=1.2)
rcParams['figure.figsize'] = 16, 10

np.random.seed(42)
```

3)
Listing Annotation

```
clothing = []
with open("clothing.json") as f:
    for line in f:
        clothing.append(json.loads(line))
clothing[0]
```

Output:

```
{'annotation': [{'imageHeight': 312,
   'imageWidth': 147,
   'label': ['Tops'],
   'notes': '',
   'points': [{'x': 0.02040816326530612, 'y': 0.2532051282051282},
    {'x': 0.9931972789115646, 'y': 0.8108974358974359}]}],
 'content': 'http://com.dataturks.a96-
i23.open.s3.amazonaws.com/2c9fafb063ad2b650163b00a1ead0017/4bb8fd9d-8d52-46c7-aa2a-
9c18af10aed6___Data_xxl-top-4437-jolliy-original-imaekasxahykhd3t.jpeg',
 'extras': None}
```

4)
Listing the categories:

```
categories = []
for c in clothing:
  for a in c['annotation']:
    categories.extend(a['label'])
categories = list(set(categories))
categories.sort()
categories
```

Output:

```
['Jackets',
 'Jeans',
 'Shirts',
 'Shoes',
 'Skirts',
 'Tops',
 'Trousers',
 'Tshirts',
 'sunglasses']
```

Chapter 7
Conclusion

Conclusion:

After researching through various papers related to Object Detection and Localization we have concluded that, a system can be developed that can detect images from different classes and produce a bounding box around it. We will test our system against benchmark datasets and compare our results based on precision, , and the time taken.

BIBLOGRAPHY:

- [Submitted on 21 Dec 2013 (v1), last revised 24 Feb 2014 (this version, v4)] OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun

- Region-based Convolutional Networks for Accurate Object Detection and Segmentation Ross Girshick, Jeff Donahue, Student Member, IEEE, Trevor Darrell, Member, IEEE, and Jitendra Malik, Fellow, IEEE

- *[Submitted on 30 Apr 2015 (v1), last revised 27 Sep 2015 (this version, v2)]*Fast R-CNN Ross Girshick
- *[Submitted on 4 Jun 2015 (v1), last revised 6 Jan 2016 (this version, v3)] Faster R-CNN* Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun

- *Submitted on 8 Jun 2015 (v1), last revised 9 May 2016 (this version, v5)] YOLO v1* Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

- *[Submitted on 8 Dec 2015 (v1), last revised 29 Dec 2016 (this version, v5)] SSD* Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg

- *[Submitted on 9 Dec 2016 (v1), last revised 19 Apr 2017 (this version, v2)]* Feature Pyramid Networks for Object Detection Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie

- *[Submitted on 25 Dec 2016] YOLO V2* Joseph Redmon, Ali Farhadi

- *[Submitted on 7 Aug 2017 (v1), last revised 7 Feb 2018 (this version, v2)* Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollar Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár