**OBJECT DETECTION USING YOLO**

***Report submitted to***

***Rajiv Gandhi University of Knowledge Technologies,***

***Srikakulam. for the fulfillment of mini project***

***Of***

**Bachelor of Technology**

**in Computer Science and Engineering**

***by***

**M.Tejeswari Anusha(S160572)**

**R.Likhitha(S160602)**

**D.Divya (S160842)**

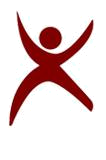
**Y.Naga Saraswati(S160409)**



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, SRIKAKULAM MAY 2021**

© 2021. All rights reserved.

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE** **TECHNOLOGIES**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**(A.P. Government Act 18 of 2008)**

**RGUKT-Srikakulam, Srikakulam Dist – 532402**

**Declaration**

We certify that

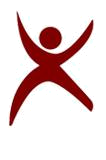
1. The work contained in this report is original and has been done by us under the guidance of my supervisor(s).
2. The work has not been submitted to any other Institute for any degree or diploma.
3. We have followed the guidelines provided by the Institute in preparing the report.
4. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever We have used materials (data, theoretical analysis, figures, and text) from other sources, We have given due credit to them by citing them in the text of the report and giving their details in the references. Further, We have taken permission from the copyright owners of the sources, whenever necessary.

M.Tejeswari Anusha(S160572)

R.Likhitha(S160602)

D.Divya(S160842)

Y.Naga Saraswathi(S160409)

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE** **TECHNOLOGIES**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**(A.P. Government Act 18 of 2008)**

**RGUKT-Srikakulam, Srikakulam Dist – 532402**

**Certificate**

This is to certify that the Dissertation Report entitled, “Object Detection Using YOLO ” submitted by **Ms. M.Tejeswari Anusha, Ms. R.Likhitha, Ms. D.Divya, Ms. Y.Naga Saraswathi,** to Rajiv Gandhi university of Knowledge Technologies, Srikakulam, India, is a record of bonafide Project work carried out by him/her under my/our supervision and guidance and is worthy of consideration for the fulfillment of mini-project of Bachelor of Technology in computer Science and Engineering of the Institute.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr.Jagadeesh Examiner**

Project Supervisor Project Examiner

Faculty Dept. of CSE Faculty Dept. of CSE

**RGUKT IIIT Srikakulam RGUKT IIIT Srikakulam**

Date:

**ACKNOWLEDGEMENT**

We would like to express our profound gratitude and deep regards to our guide ***Mr.Jagadeesh*** for his exemplary guidance, monitoring and constant encouragement to usthroughout this semester. We shall always cherish the time spent with him during the course of this work due to the invaluable knowledge gained in the field of Deep Learning.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **“Object Detection using YOLO”.**

We express gratitude to Mr.Ch.Satish Kumar (HOD of CSE) and other faculty members for being source of inspiration and constant encouragement which helped us in completing the project successfully.

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

M.Tejeswari Anusha (S160572)

R.Likhitha(S160602)

D.Divya (S160842)

Y.Naga Saraswathi (S160409)

**ABSTRACT**

The Objective is to detect of objects using You Only Look Once (YOLO) approach. This method has several advantages as compared to other object detection algorithms. In other algorithms like Convolutional Neural Network, Fast Convolutional Neural Network the algorithm will not look at the image completely but in YOLO the algorithm looks the image completely by predicting the bounding boxes using convolutional network and the class probabilities for these boxes and detects the image faster as compared to other algorithms*.*

***Keywords:*** *YOLO* model, Image processing, Object recognition, Object Localization, Bounding boxes.

**CONTENTS**

|  |  |
| --- | --- |
| Title Page | i |
| Declaration | ii |
| Certificate by Supervisor | iii |
| Acknowledgement | iv |
| Abstract | v |
| Contents | vi |
| Chapter 1 Introduction  1.1 Problem Statement  1.2 Motivation of the Project  1.3 Limitations of the Project  1.4 Existing System  1.5 Proposed System | 1  1  1  1  1  2 |
| Chapter 2 Literature Review  2.1 Study of Related Papers | 2  3 |
| Chapter 3 Requirement Specification  3.1 Functional Requirements  3.2 Non-Functional Requirements  3.3 System Requirements  3.3.1 Hardware Requirements  3.3.2 Software Requirements | 3  3  4  5  5  6 |
| Chapter 4 System Design  4.1 Introduction  4.2 Sequence Diagram | 6  6  7 |
| Chapter 5 Implementation | 11 |
| 5.1 Chapter Overview | 11 |
| 5.2 Implementation tools | 11 |
| 5.3 Packages | 11 |
| 5.3.1 openCV | 11 |
| 5.3.2 Numpy | 12 |
| 5.3.3 Argparse | 13 |
| 5.3.4 Time | 13 |
| 5.3.5 OS | 13 |
| 5.4 Object Localization | 16 |
| 5.5 Object Detection | 16 |
|  |  |
| Chapter 6 Sample code and Result  6.1 Imported libraries  6.2 Code for yolov3 algorithm  6.3 Running on command prompt  6.4 Output | 18  18  19  20  20 |
| Chapter 7 Advantages and application | 21 |
| Chapter 8 Conclusion | 21 |
| Reference | 21 |

**1. INTRODUCTION**

**1.1 Problem Statement**

The project "object detection" is the problem of finding and classifying a variable number of objects on an image, detects objects efficiently based on YOLO algorithm and apply the algorithm on image data. The output of object detection is variable in length, since the number of objects detected may change from image to image.

**1.2 Motivation of the Project**

The motive of object detection is to recognize and localize all known objects in a scene. The goal of object detection is to determine whether there are any instances of objects from given categories (such as humans, cars, bicycles, dogs or cats) in an image and, if present, to return the spatial location and extent of each object instance.Object detection is an important computer vision task used to detect instances of visual objects of certain classes.

**1.3 Limitations of the Project**

One of the biggest difficulties of object detection is that an object viewed from different angles may look completely different.

It also makes more localization errors though the accuracy of output is good.

**1.4 Existing System**

Previously people were using sliding window object detection means independently classify all image patches as being object or non object so, only the image classification is done here later on object localization was introduced. Sliding window classification is the dominant paradigm in object detection. It is one of the most noticeable success of computer vision

**1.5 Proposed System**

The YOLO algorithm “you only look once” at the input image that needs only one image forward propagation pass through the network to make the predictions. Yolo is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy.It has been used in various applications to detected traffic signals, people,parking meters, and animals. Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected things.

**3. REQUIREMENT SPECIFICATION**

Requirements-Determination is the process, by which analyst gains the knowledge of the organization and apply it in selecting the right technology for a particular application. A Software Requirements Specification (SRS) is a complete description of the behaviour of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as quality standards).

**3.1 Functional Requirements**

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

**Functional Requirements for Present Project**

1. Input: Image of the Objects
2. Output: Identifies object name displays.
3. Process: system checks the current image with the images in the database if it founds a match detection is done.

**3.2 Non-Functional Requirements**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour. This should be contrasted with functional requirements that define specific behaviour or functions In general; functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements," and "non-behavioral requirements." Qualities, that is, non-functional requirements, can be divided into two main categories one is Execution qualities, such as security and usability, which are observable at run time and Evolution qualities, such as test ability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

**Non-Functional Requirements for Present Project**

Easy detection of images is done. The user need not know the programming language very well that is used to develop a software and the output should be simple and clear.

**3.3 System Requirements**

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer.

Install Python on your computer system

1. Install dependencies like Numpy, OpenCV, etc.

2. Download the Object Detection model file(Yolo)

**3.3.1 Hardware Requirement:**

1. Processor: Intel Core I5/ I7
2. RAM: 4GB
3. Storage: 20GB
4. Monitor with 1024\*720 resolution

**3.3.2 Software Requirement**

1. Operating System: Windows 8,9,10
2. Languages: Python

**4. SYSTEM DESIGN**

**4.1 Introduction**

Grady Booch, James Raumbaugh and Ivar Jacobson have collaborated to combine the best features of their individual object oriented analysis and design methods into a unified method the unified modeling language, the version 1.0 for the Unified Modeling was released in January 1997 the main parts of UML are based on the Booch, OMT and OOSE methods. The goals of UML are:

1. To model systems using object-oriented concepts
2. To establish an explicit coupling between conceptual as well as executable
3. To address the issues of scale inherent in complex, mission critical system
4. To create a modeling language usable by both humans and machines

**Basic Building Blocks of UML**

The basic building blocks in UML are things and relationships; these are combined in different ways following different rules to create different types of diagrams. In UML there are nine types of diagrams, below is a list and brief description of them. The more in depth descriptions in the document, will focus on the first five diagrams in the list, which can be seen as the most general, sometimes also referred to as the UML core diagrams.

**Use case Diagram:** shows a set of use cases, and how actors can use them.

**Class Diagram:** describes the structure of the system, divided in classes with different connections and relationships

**Sequence Diagram:** shows the interaction between a set of objects, through the messages that may be dispatched between them.

**State chart Diagram:** state machines, consisting of states, transitions, events and activities.

**Activity Diagram:** shows the flow through a program from a defined start point to an end point.

**Object Diagram:** A set of objects and their relationships, this is a snapshot of instances of the things found in the class objects.

**Collaboration Diagram:** Collaboration diagram emphasize structural ordering of objects that send and receive messages.

**Component Diagram:** shows organizations and dependencies among a set of components. These diagrams address the static implementation view of the system.

**Deployment Diagram:** show the configuration of run-time processing nodes and components that live on them.

**4.2 Sequence Diagram**

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. The purpose of sequence diagram is to show the flow of functionality through a use case. In other words, we call it a mapping process in terms of data transfers from the actor through the corresponding objects. The key points are:

1. The main purpose is to represent the logical flow of data with respect to a process
2. A sequence diagram displays the objects and not the classes. Messages are written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages.

When an object destroyed (removed from memory), an X is drawn on top of the lifeline, and the dashed line ceases to be drawn below it (this is not the case in the first example though). It should be the result of a message, either from the object itself, or another.

**4.2.1 Sequence Diagram**

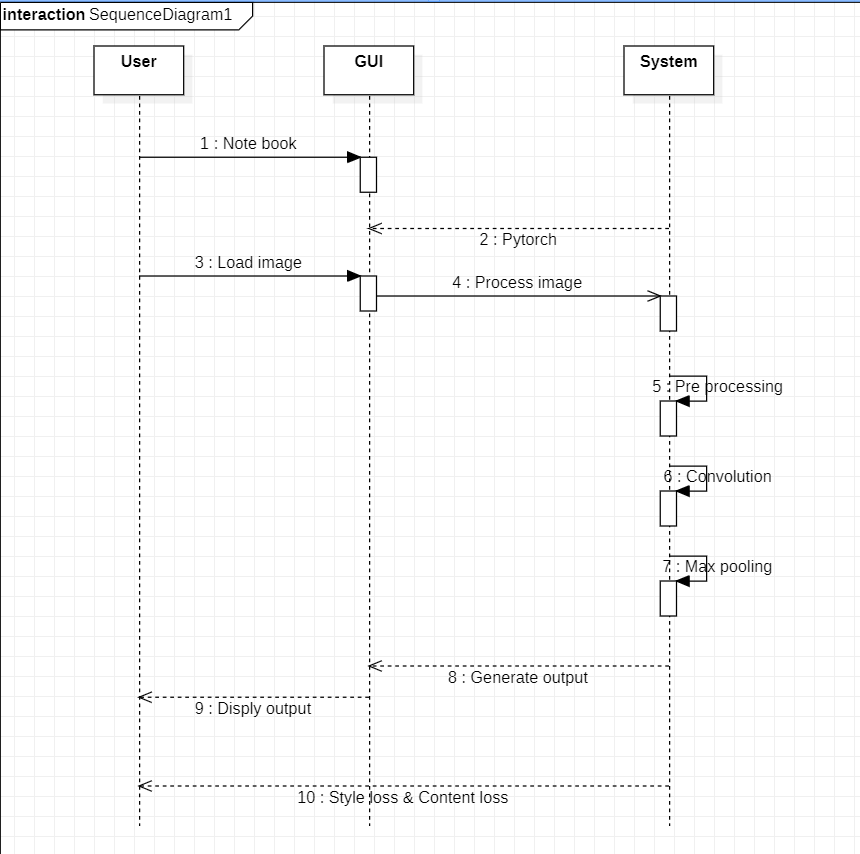
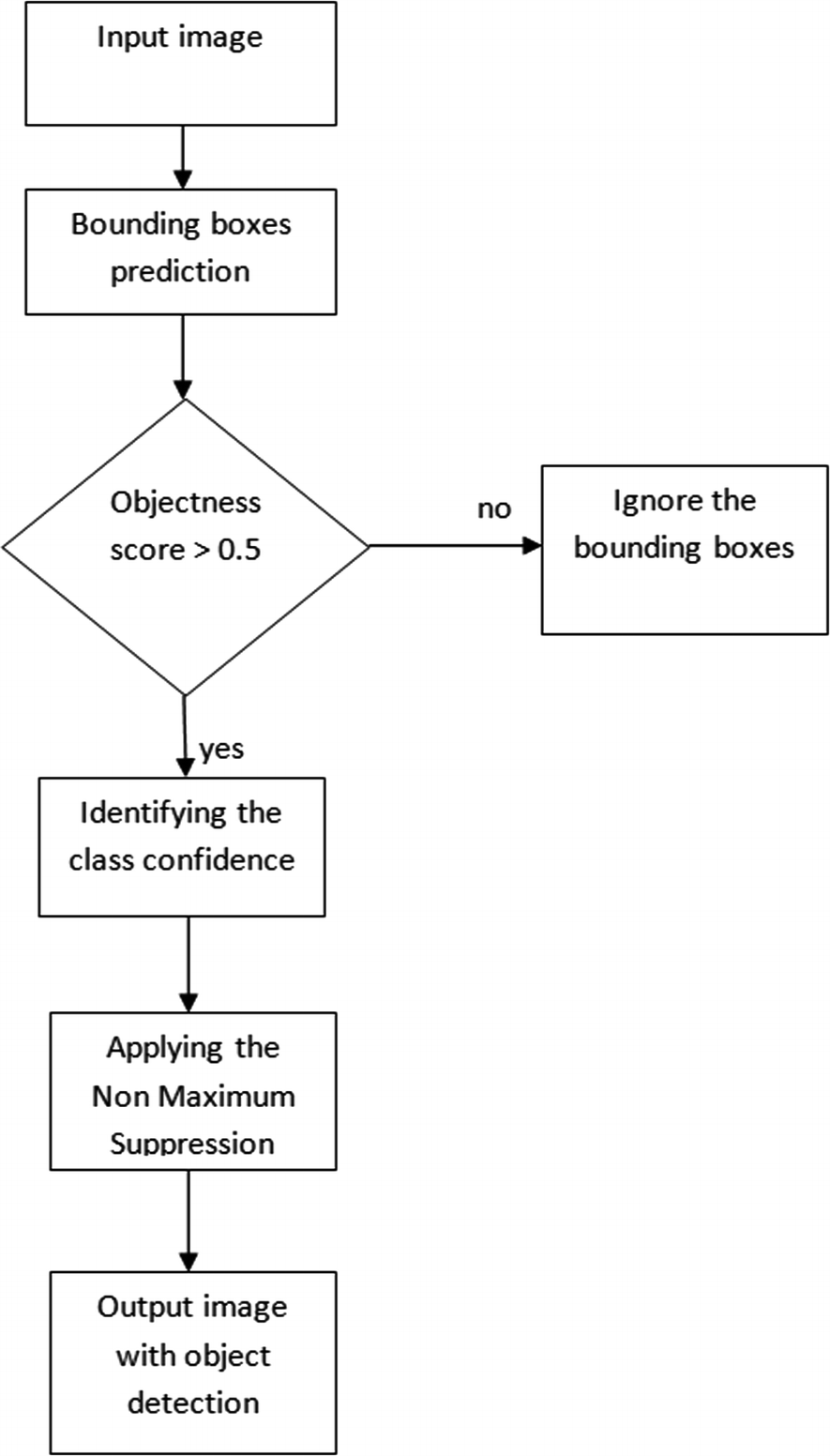
****

Fig 4.2.1: Sequence Diagram

**Flow Chart**

****

**5. Implementation**

**5.1 Chapter overview:**

This chapter presents the process of implementing the proposed object detection system with YOLO v3 algorithm. It firstly introduces the tools that were used and the data set, followed by a detailed description of the implementation of the functionality and the interface.

**5.2** **Implementation tools:**

As the goal of the project was highly targeted towards research, the entire system was

developed in Python, a high-level language and scientific environment.

**5.3 packages:**

**5.3.1.openCv:**

openCv(open source computer vision) is a library of programming functions mainly aimed at real time computer vision. In simple language it is a library used for Image Processing .It is mainly based to all the operation related to images.

**What it can do:**

1. Read and write Images.
2. Detection of faces and its features.
3. Detection of shapes like Circle,rectangle etc

**5.3.2.Numpy:**

NumPy(Numerical Python) is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. It has in-built functions for linear algebra and random number generation. The most important object defined in NumPy is an N-dimensional array type called ndarray.

**What it can do:**

1. Mathematical and logical operations on arrays.
2. Fourier transforms and routines for shape manipulation.
3. Operations related to linear algebra.

**5.3.3.Argparse:**

Python Argparse is a command-line parsing module that is recommended to work with the command line argument. The Argparse module is the better replacement of the Python getopt and optparse module.

**What it can do:**

1. It helps to create a program in a command-line-environment.
2. It allows us to customize the prefix chars.
3. It supports variable numbers of parameters for a single option.
4. It supports sub commands.

**5.3.4.Time:**

The Python time module provides many ways of representing time in code such as objects, numbers and strings. It also provides functionality other than representing time like waiting during code execution and measuring the efficiency of your code.

**What it can do:**

* **Time():** This function is used to count the number of seconds elapsed since the epoch

**5.3.4.OS:**

It is possible to automatically perform many operating system tasks. You first need to import the os module to interact with the underlying operating system. So, import it using the import os statement before using its functions.

**What it can do:**

1. It provide functions for creating and removing a directory(folder).
2. It helps in fetching its contents.
3. It allows to changing and identifying the current directory.

**5.4 Object Localization:**

Read the data:

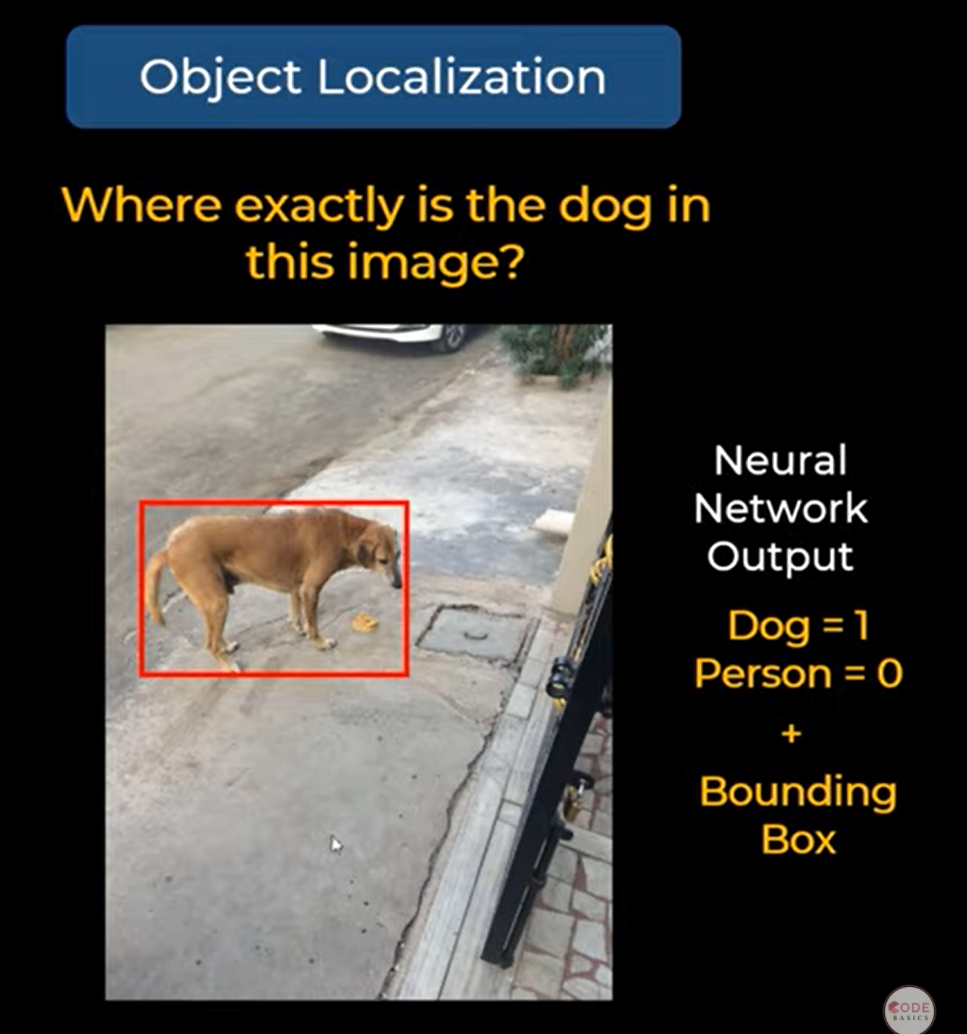
This deals with understanding the data. The model will preprocess the data sets.

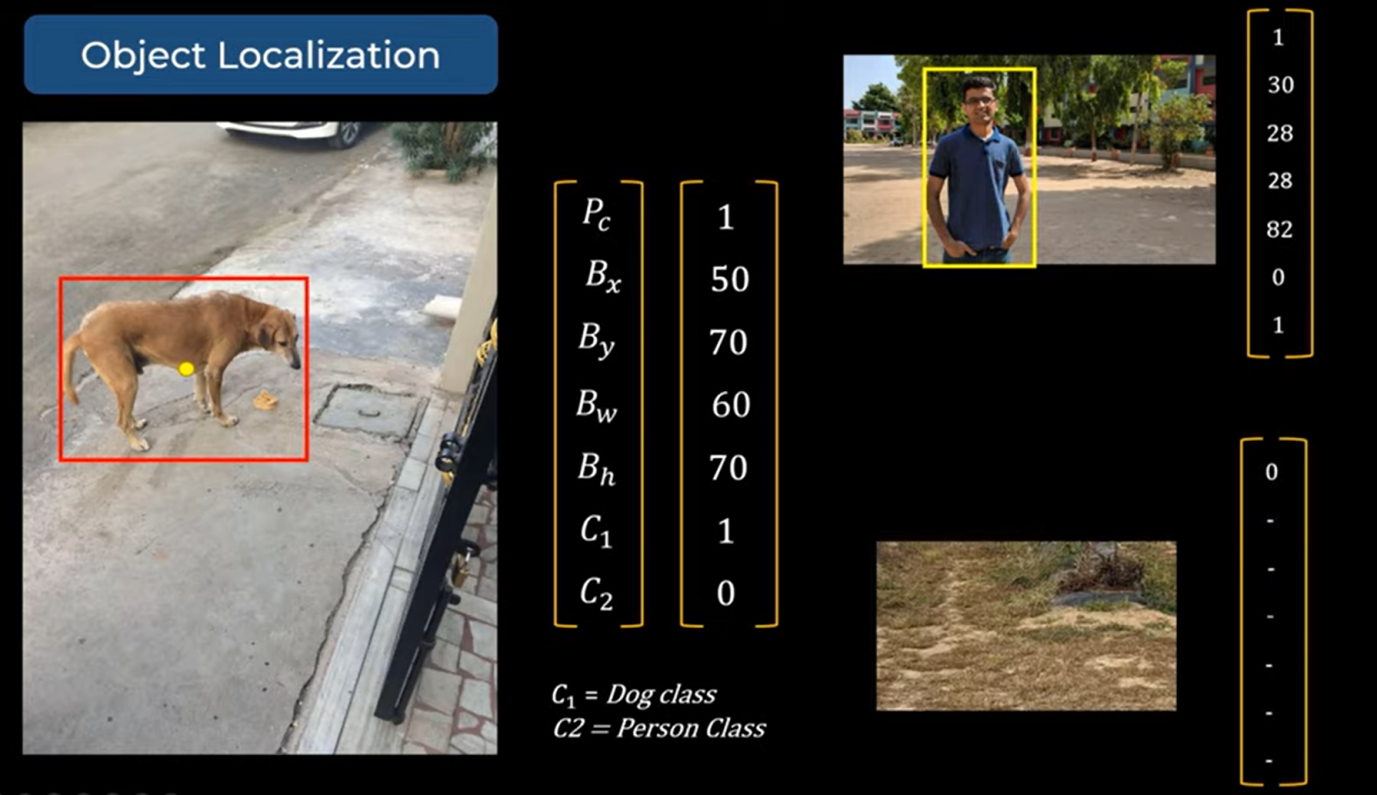
Load the Data: The model will resize, re-scale, convert your labels into one-hot encoding vectors and split up your data in training and validation sets.

Analyse the data : Even though you know the dimension of the images by now, it's still worth the effort to analyze it through model.

**Object Localization**:

Object localization refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. Object localization is one of the image recognition tasks along with image classification and object detection. Though object detection and object localization are sometimes used interchangeably, they are not the same.





**5.5 Object detection**

Convolutional Neural Network (CNN): is used to detect objects in real time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

Bounding box :

Bounding box regression

A bounding box is an outline that highlights an object in an image.

Every bounding box in the image consists of the following attributes:

Width (bw)

Height (bh)

Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.

Bounding box center (bx, by)

Intersection over union (IOU)

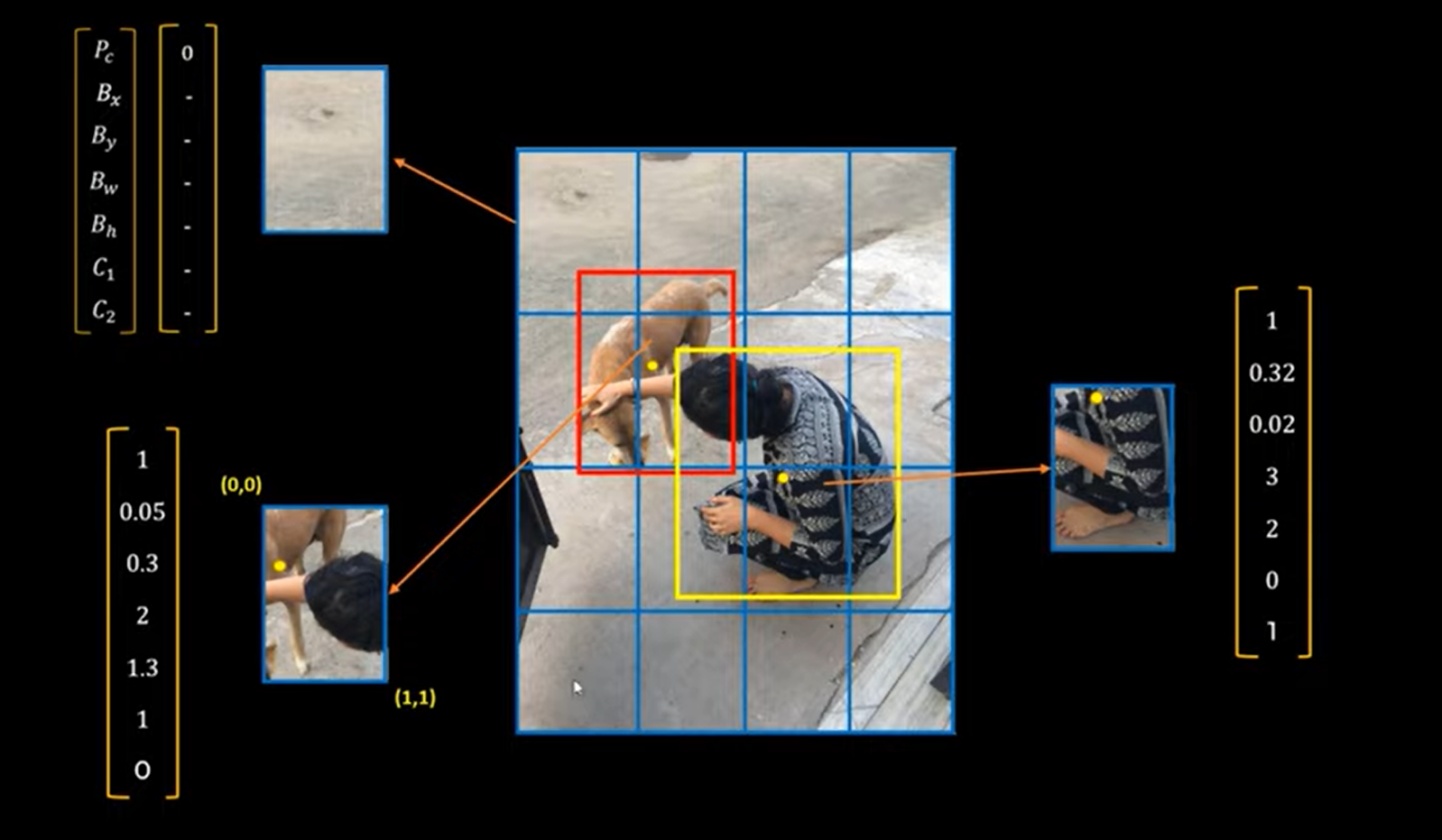
Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly.

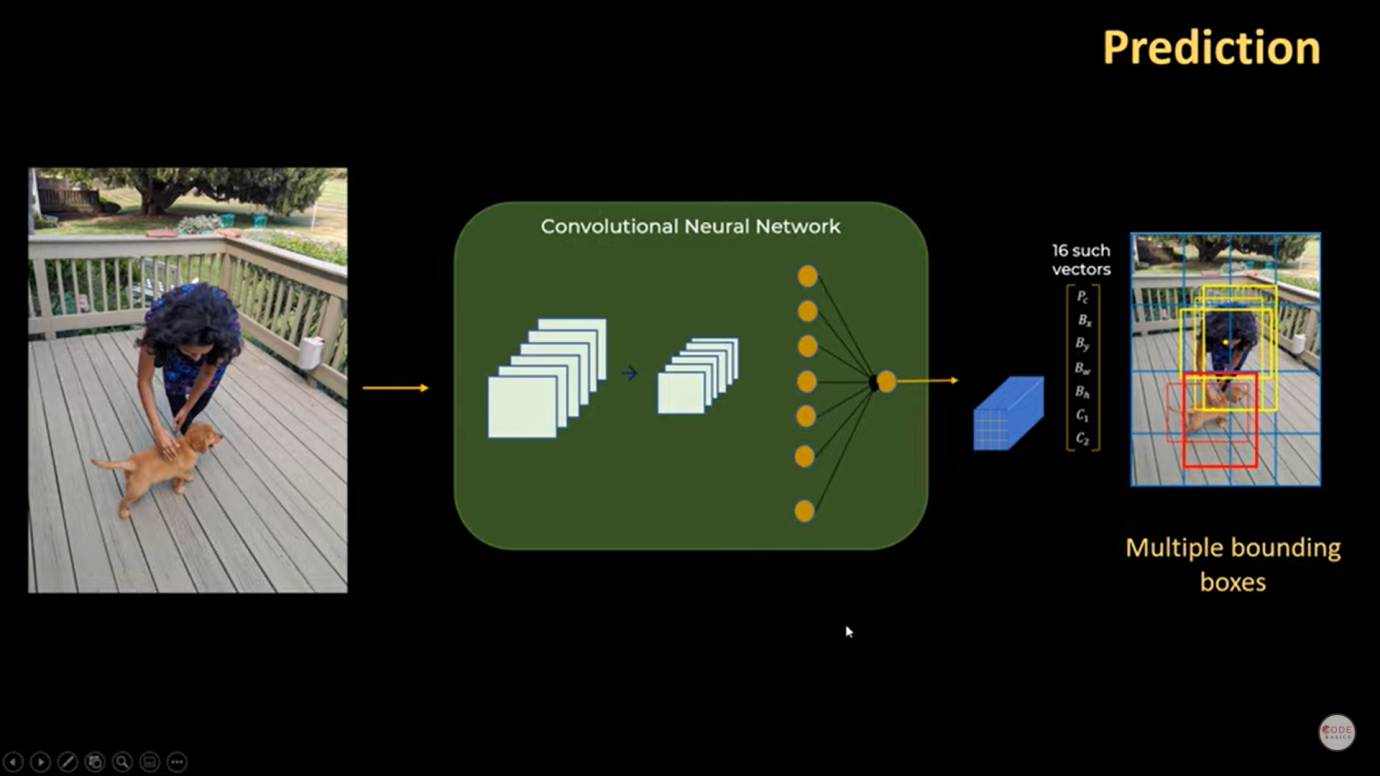
Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to 1 if the predicted bounding box is the same as the real box. This mechanism eliminates bounding boxes that are not equal to the real box.

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object.

For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network.

Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly.





Finally detected image :

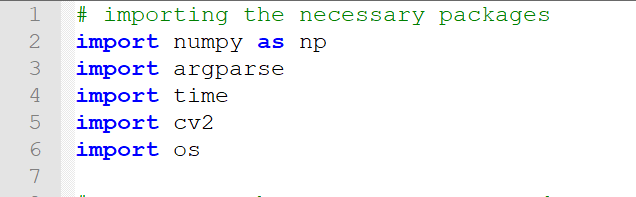
We have gained an overview of object detection and the YOLO algorithm.

We have gone through the main reasons why the YOLO algorithm is important.

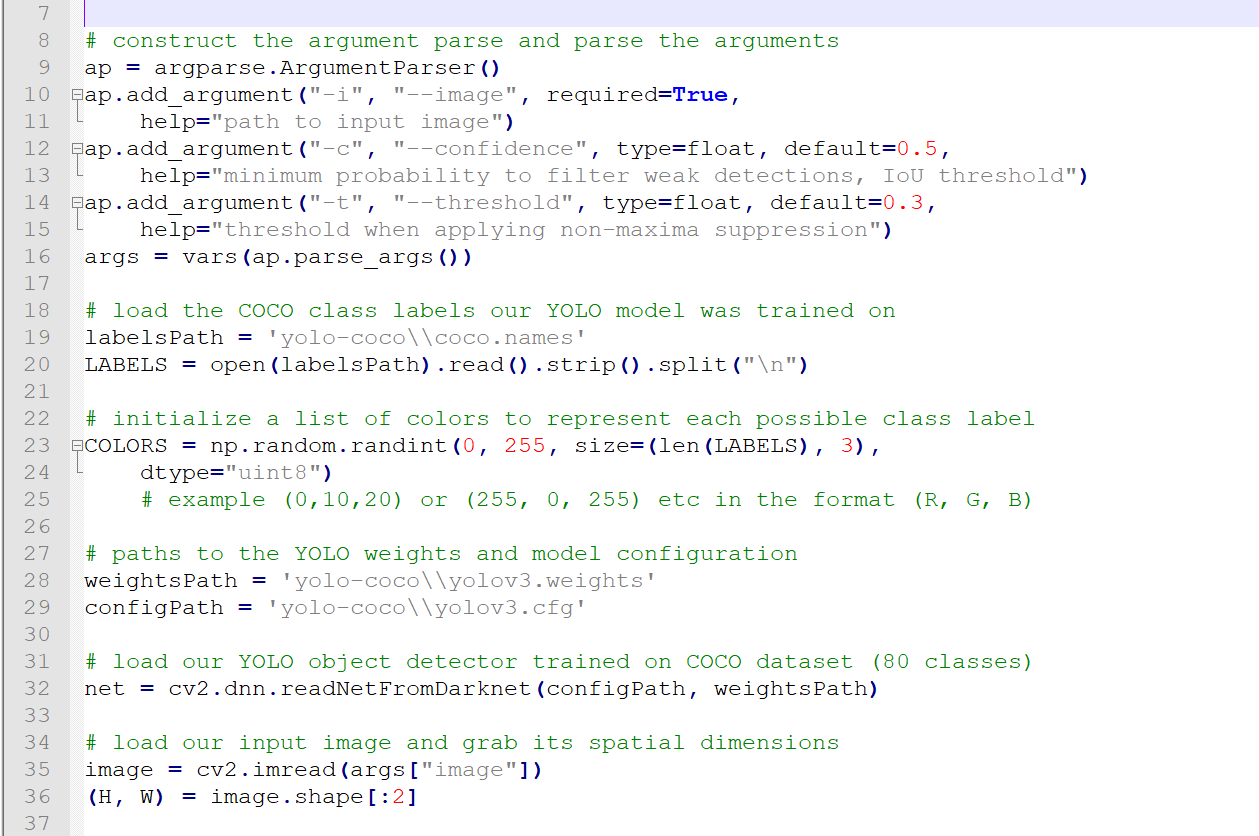
We have learned how the YOLO algorithm works. We have also gained an understanding of the main techniques used by YOLO to detect objects.

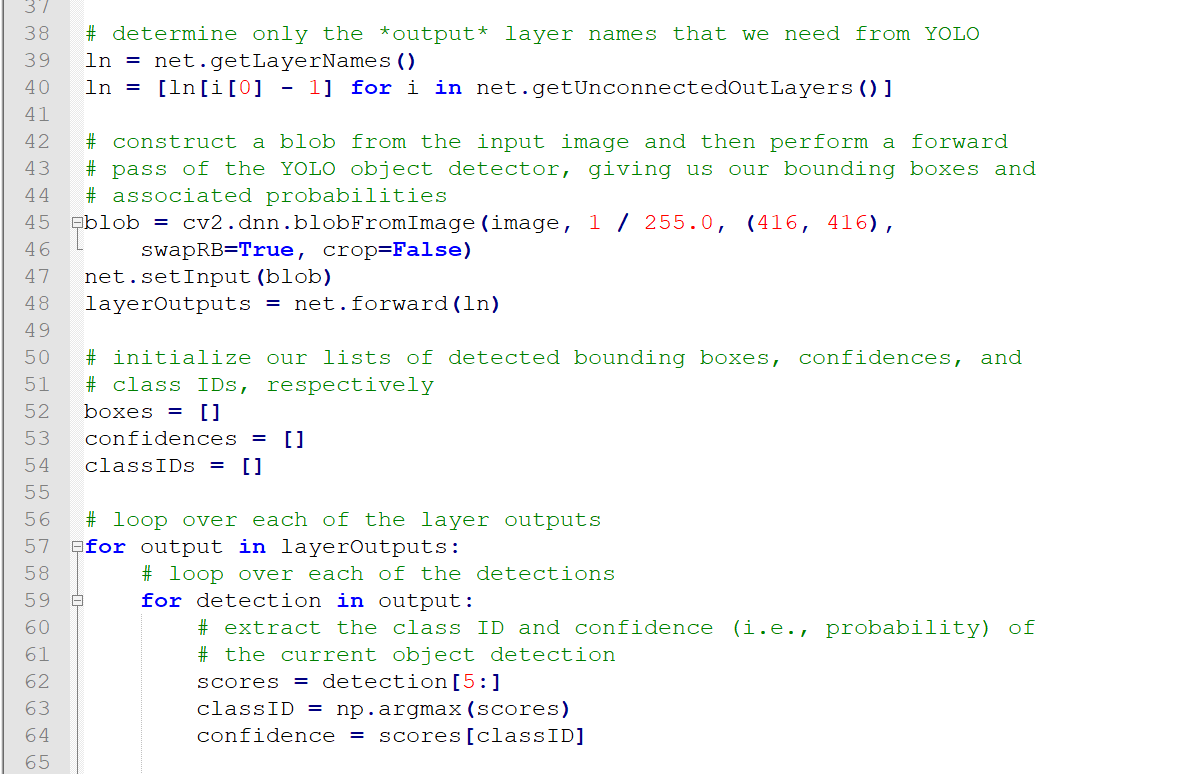
**6. Sample code and Results**

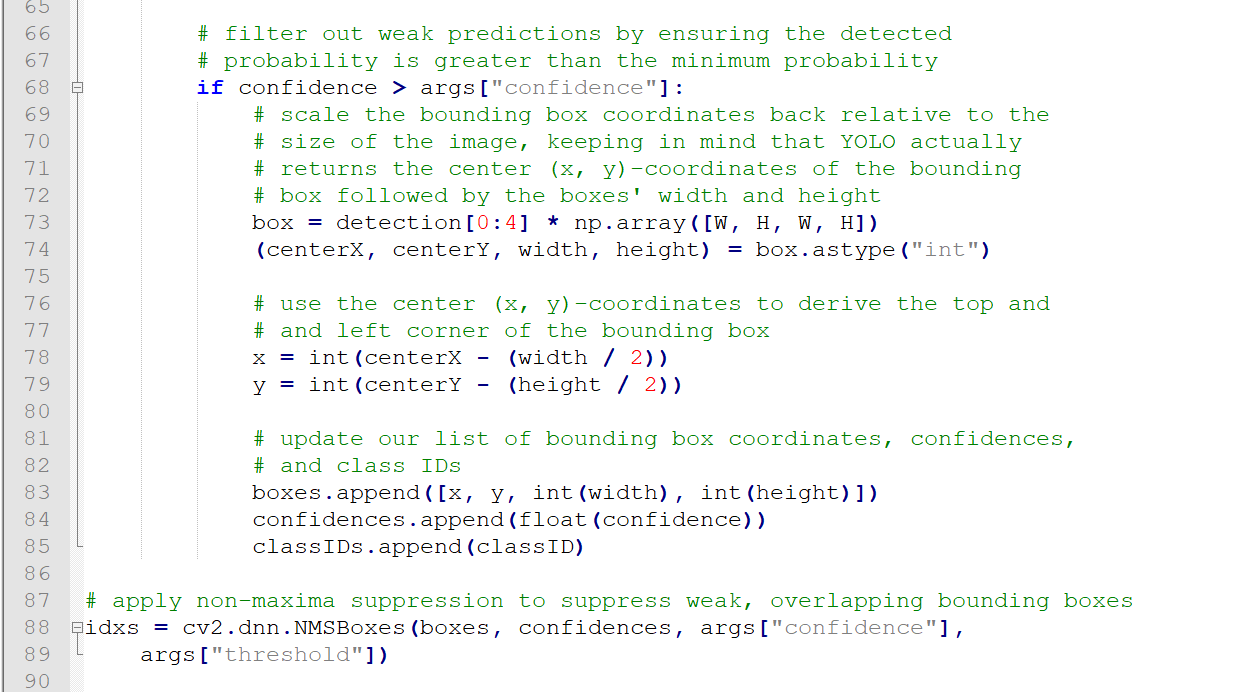
**6.1 Imported libraries**

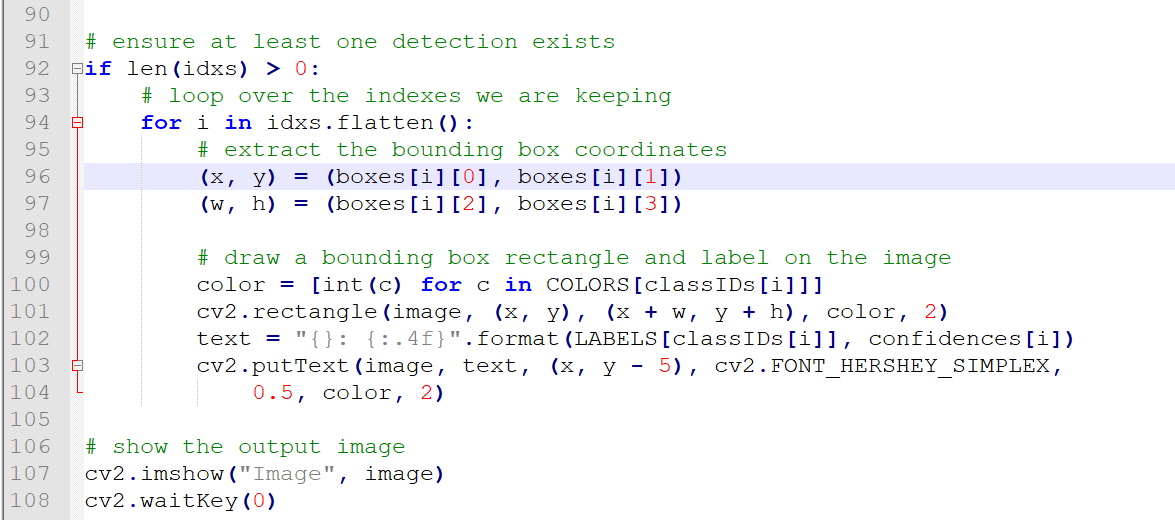


**6.2 Code for yolov3 algorithm**

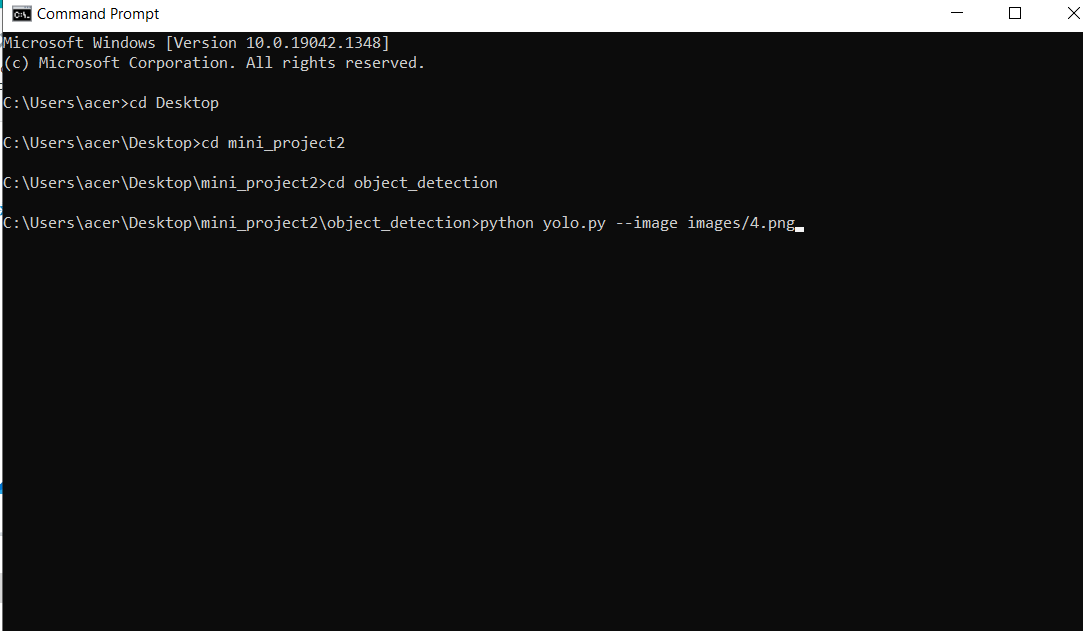
****

****

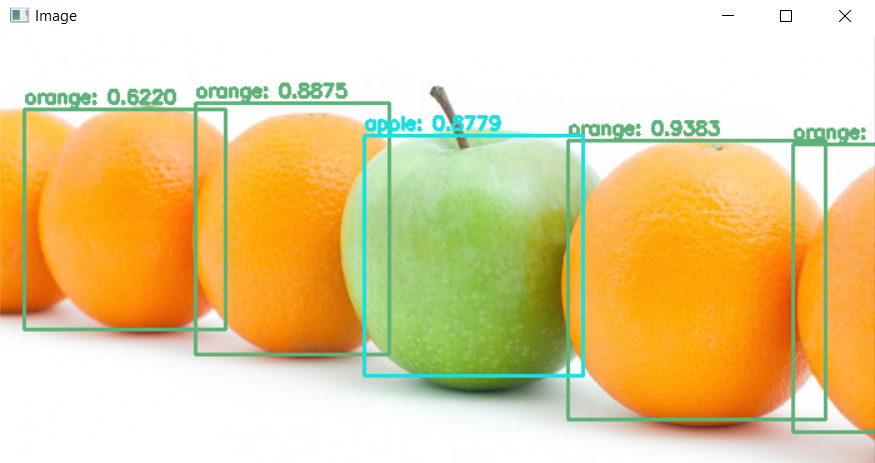
****

****

**6.3 Running in command prompt**

****

**6.4 Output**



**7. Advantages and applications**

**1. Super speed:** It’s incredibly fast and can process 45 frames per second.

**2. Multiple object classification in one go:** the newest technology helps in detecting objects in a single image efficiently.

**3. High Accuracy:** YOLO is a predictive technique that provides accurate results with minimal background errors.

**4. Learning Capabilities:** The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

**5. Autonomous Driving:** YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car.

**6. Simple Algorithm & flowcharts:** YOLO can also be used in security systems to enforce security in an area. Let’s assume that people have been restricted from passing through a certain area for security reasons. If someone passes through the restricted area, the YOLO algorithm will detect him/her which will require the security personnel to take further action.

**8. Conclusion**

This system has been proposed for object detection using YOLO. The main motive behind developing this system is to eliminate all the drawbacks which were associated with previous object detection.

The drawbacks ranging taking more time and process, will completely be eliminated.

Hence, desired results with user friendly interface is expected in the future, from the system.

Reference

<https://pjreddie.com/media/files/papers/yolo.pdf>

<https://www.researchgate.net/publication/342419696_Object_Detection_through_Modified_YOLO_Neural_Network>

<https://arxiv.org/pdf/1807.05511.pdf>

<https://www.youtube.com/watch?v=ag3DLKsl2vk>