Improvised Object Detection using YOLO v3

Thimmaraju Abhiram[1], Gadamsetty Hemanth Venkata Sai[2], M Pavithra[3]

[1]Student, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India
[2]Student, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India
[3]Assistant Professor, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India

---***---

Abstract - *Nowadays, Deep learning has gradually become the popular object detection algorithm because of its powerful feature extraction. However, to be precise the main challenge is the speed and accuracy of the algorithm in the field of computer vision. Currently, YOLO is the best of all algorithms as compared to other detectors. YOLO Stands for You Only Look Once, it is a real-time object detection algorithm. The YOLOv3 is very fast and it can be adjusted between accuracy and speed by changing some parameters and main interesting thing is that there is no re-training of the models is required*

***Key Words*: COCO, Deep Learning, Object Detection, OpenCV, YOLOv3 method.**

1. INTRODUCTION

Over a few decades, technology has been evolving rapidly in terms of computational power as well as machine learning algorithms to solve real-life problems and make humans work faster and more efficiently. Today, the algorithms are updating to achieve the best results in less time complexity and space complexity. Computers are solving complex problems in a fraction of seconds with the use of CPUs and GPUs.

The problem arises, how can the computers detect an object with an image captured by a camera. This problem is solved by the hard work of many developers by introducing new algorithms like Faster R-CNN, Histogram of Oriented Gradients (HOG), Single Shot Detector (SSD), Spatial Pyramid Pooling (SPP-net), and YOLO (You Only Look Once). Here, we considered YOLO version 3 since it is faster and accurate compared to other detectors.

1.1 Objectives

The main objective of 'Improvised Object Detection using YOLO v3' is to find the best algorithms present in the current world and to detect objects using fewer lines of code, error-free and to detect faster and accurate using a laptop/desktop or by using small devices like Raspberry pi with a webcam so that it can be implemented in schools, universities, homes, public places, etc., at cheaper cost.

1.2 Scope of the Project

We aim to improve the algorithm by changing the parameters, adding new features, and work it with a faster FPS (Frames Per Second) Using Online/Offline CPUs/GPUs and using it in the real-time scenario.

1.3 Existing system

YOLOv3 detects as a regression problem. It divides the image into an s✕s grid and the cell predicts bounding boxes with the confidence of those boxes, and based on the class probability map then it will produce final detection

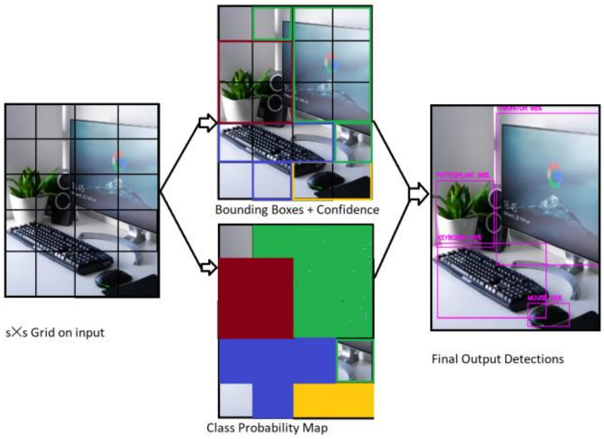

Fig.1.1: Model

It has 24 convolutional layers followed by two fully connected layers. Then we relate these layers on ImageNet classification as half of the resolution as 224✕224 input image and then double the resolution for detection and by using Non-Max Suppression it will remove all the bounding boxes for threshold less than 0.2 so that we can see bounding boxes clearly.

1.4 Drawbacks of the Existing system

It struggles to detect an object in some rare cases and some compatibility issues.

2. LITERATURE SURVEY

2.1 Literature Survey

AUTHOR	TITLE	FINDINGS
Joseph Redmon, Santosh Divvala, Ross Girshick , Ali Farhadi	YOLO v1 (2015)	YOLO v1 performs about 45 Frames Per Second (FPS) which is higher than other real-time object detecors.
Joseph Redmon, Santosh Divvala, Ross Girshick , Ali Farhadi	YOLO v2 (2016)	YOLO v2 performs about 40 Frames Per Second (FPS) and accuracy better than YOLO v1 and other models such as SSD and R-CNN.
Joseph Redmon ,, Ali Farhadi	YOLO v3 (2017)	YOLO v3 is faster and accurate as Single Shot Multibox(SSD). It can recognize 80 different objects from real-time video feed or images.Here it has two main versions YOLO-320 is little bit slower but accurate and YOLO Tiny is extreamly fast but it is less accurate.
Alexey Bochkovskiy	YOLO v4 (2020)	YOLO v4 is faster speeds at 65 Frames Per Second. It is incremental model of YOLO v3 and it has YOLO v4 Tiny also.

2.2 Problem Statement

There are different algorithms for real-time object detection but the problem comes that is which algorithm is more efficient in terms of accuracy and speed as well as community support for the algorithm.

2.3 System Requirements

2.3.1 Software Environment:

●Python 3.8.2 (Above 3 version)

●OPEN CV 4.4.0 (Above 4 version)

●opencv-contrib-python 4.5.1.48 (Above 4.2 version)

● COCO (Common Objects in Context) Dataset (coco.names) file.

●Configuration(.cfg) and

Model weight(.weight) files from YOLO Website.

2.3.2 Hardware Environment:

●Windows Laptop/ Desktop Intel I3 & above

●Webcam (Above 5MP)

3. PROPOSED METHOD

The main aim is to improve the algorithm and work it with a faster FPS (Frames Per Second) Using Online/Offline CPUs/GPUs and to use it in real-time scenarios.

First, import the necessary libraries in a python file. Then by using the OpenCV video capture method get the input video feed via webcam/mp4 or image (mostly jpg recommended) and load the class files, model configuration file, and model weight file.

Then define the confidence threshold for detecting an image with the desired accuracy and non-max suppression to reduce the overlapping of the bounding boxes.

Next, by defining a function to find the objects if found the objects then it will be sent to the Bounding boxes function else it will try to detect the object. In the Bounding Box function, It will get x,y,w,h then by using the cv.rectangle function we will draw the boxes for every object it detects. By using the cv.putText function for labelling the rectangle.

This will be in the new output window here if the user presses the ESC button the code terminates the program and closes the window.

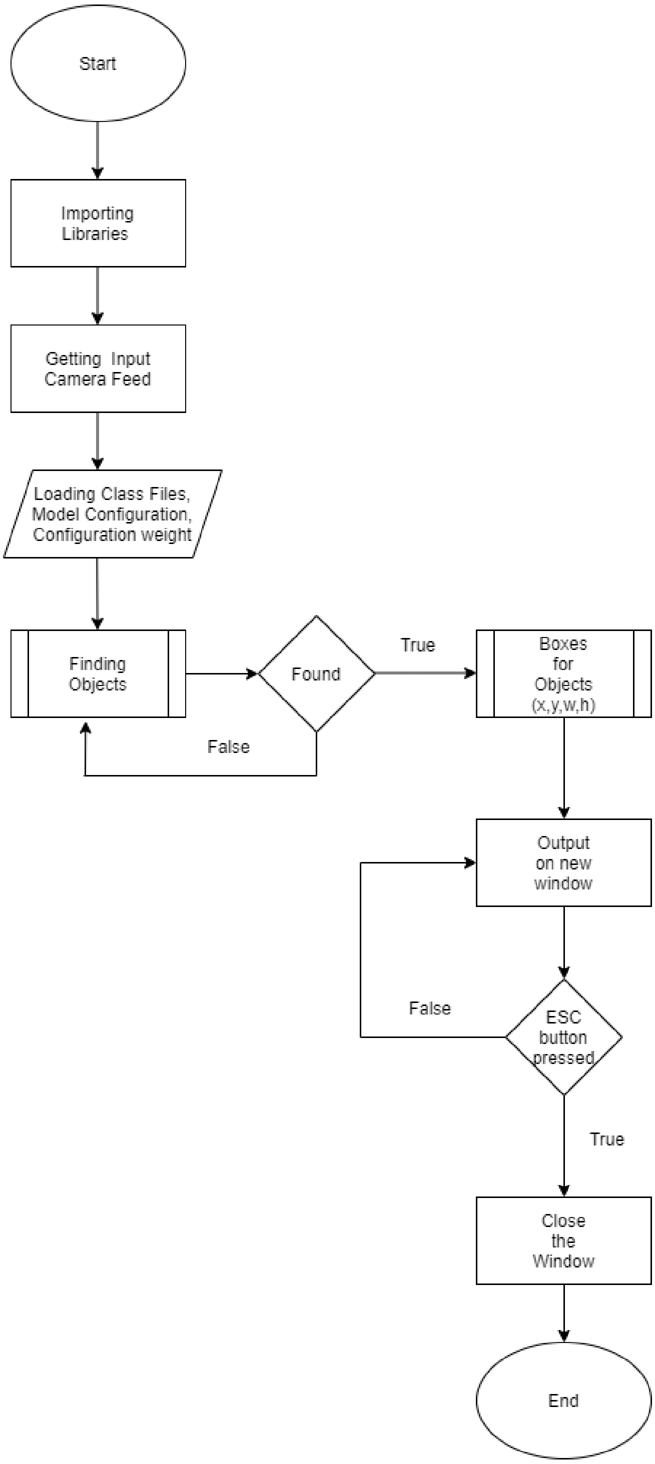

Fig.3.1: Flow Chart

4. CHALLENGES FACED

We faced several problems during the project. The first issue was choosing the right algorithm that works with the CPU that is it has to run in real-time nearly 15-20 FPS (Frames Per Second) we tried with google colab but we couldn't make use of GPU. With the use of Google Colab, we faced an issue with darknet and other issues like we have to use cv2.draw function instead of cv2.imshow since in Jupiter it gives output in graphical format only.

The second issue was an error that occurred while opening the webcam and it has taken more time to solve by using the video capture method. And when we try to give input as an image it runs perfectly fine with the .jpg extension but with .png and other extension has made a big issue.and finally, The third issue was an error caused by cv2.resize function mainly with the images with different resolutions with their higher aspect ratios. Finally, we make sure that we have cleared issues with nearly all the errors but some of them exist.

5. EXPERIMENTAL RESULTS

```
Object Detection Started...

Detected: KEYBOARD

Detected: TVMONITOR

Detected: POTTEDPLANT

Detected: MOUSE

[Finished in 3.6s]
```

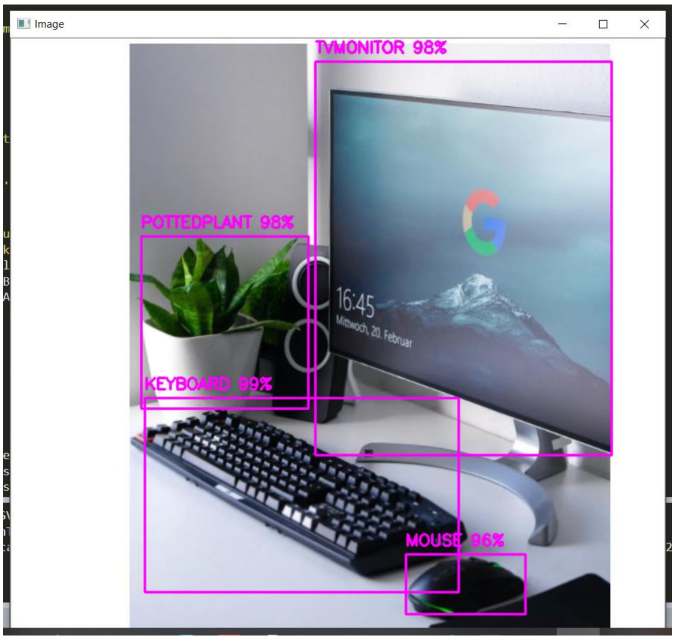

Fig. 5.1: Output Window

We have achieved an accuracy of ~96% by using YOLO v3 algorithm, which is better than other detectors. GPU's (High performance) are highly expensive for most individuals to buy and our algorithm uses a normal laptop with an integrated webcam as a major plus point and it leads to cost-effective implementation with the best accuracy.

6. CONCLUSION

In this report, a functional real-time computer vision-based object detection using YOLO v3 has been successfully performed and we achieved an accuracy of ~96% with the COCO dataset. we can detect nearly 80 different objects from the COCO dataset and we used some of them like the person, chair, laptop, keyboard, mouse, plotted plant in our current work.

REFERENCES

[1] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[2] CyberAILab, A Closer Look at YOLOv3, 2018.

[3] Hassan, N. I., Tahir, N. M., Zaman, F. H. K., & Hashim, H, People Detection System Using YOLOv3 Algorithm 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). doi:10.1109/iccsce50387.2020.9204925

[4] Rafael Padilla; Sergio L. Netto, A Survey on Performance Metrics for Object-Detection Algorithms

[5] A. Vidyavani, K. Dheeraj, M. Rama Mohan Reddy, KH. Naveen Kumar, Object Detection Method Based on YOLOv3 using Deep Learning Networks(IJRCET). Doi: June 2020/IRJET-V7I6576