Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

Object Detection and Classification using YOLOv3

Dr. S.V. Viraktamath
Dept. of Electronics and Communication Engineering
SDM College Of Engineering and Technology
Dharwad, India

Madhuri Yavagal
Dept. of Electronics and Communication Engineering
SDM College Of Engineering and Technology
Dharwad, India

Rachita Byahatti
Dept. of Electronics and Communication Engineering
SDM College Of Engineering and Technology
Dharwad, India

***Abstract*—Autonomous driving will increasingly require more and more dependable network-based mechanisms, requiring redundant, real-time implementations. Object detection is a growing field of research in the field of computer vision. The ability to identify and classify objects, either in a single scene or in more than one frame, has gained huge importance in a variety of ways, as while operating a vehicle, the operator could even lack attention that could lead to disastrous collisions. In attempt to improve these perceivable problems, the Autonomous Vehicles and ADAS (Advanced Driver Assistance System) have considered to handle the task of identifying and classifying objects, which in turn use deep learning techniques such as the Faster Regional Convoluted Neural Network (F-RCNN), the You Only Look Once Model (YOLO), the Single Shot Detector (SSD) etc. to improve the precision of object detection. YOLO is a powerful technique as it achieves high precision whilst being able to manage in real time. This paper explains the architecture and working of YOLO algorithm for the purpose of detecting and classifying objects, trained on the classes from COCO dataset.**

Keywords— YOLO, Convolutional Neural Network, Bounding Box, Anchor Box, Fast Region Based Convolutional Neural Network, Intersection over Union, Non-Max Suppression, COCO Dataset.

I. INTRODUCTION

Quick, exact calculations for object detection would permit computer to drive vehicles without particular sensors, empower assistive gadgets to pass on constant scene data to human clients, and open the potential for universally useful, responsive automated frameworks [1]. Object discovery includes identifying locale of interest of object from given class of picture [2]. There are basically two algorithms for object discovery and they can be arranged into two kinds:

1. Classification-dependent algorithms are performed in two steps. First, they define and select areas of significance for an image. Second, these regions are organized into convolutional neural networks. The above-mentioned arrangement is mild, since it is required to make estimates for all chosen regions. A commonly recognized case of this type of algorithm is the Regional Convolutional Neural Network (RCNN) and Medium RCNN, Faster RCNN, and the most recent: Mask RCNN[2].
2. Algorithms based on regression – rather than selecting a field of interest for an image, they estimate groups and bounding boxes for the whole picture in one run of the algorithm.The two most common models in this set are the YOLO family algorithms which provides maximum speed and precision for multiple object detection in a single frame [3] and the SSD this algorithms that are typically used to track objects in real-time.

To understand the YOLO algorithm, it is important to determine what is currently expected. It varies from the majority of the neural network models because it uses a single convolutional network that predicts bounding boxes and the resulting probabilities. The bounding boxes are weighted by the probabilities and the model makes their detection dependent on the final weights. Thus, end-to-end output of the model can be directly maximized and, as a result, images can be produced and processed at a rapid pace[4]. Every bounding box can be represented using four descriptors:

1. Centre of a bounding box (bx, by)
2. Width (bw)
3. Height (bh)
4. Value ‘c’ refers to an object class

The pc value also needs to be predicted, that indicates the likelihood that there is an object in the bounding box [5].

II. METHODOLOGY

YOLO takes an input image first and this input image is then divided into grids (say 3 X 3 grid) as shown in Fig 1

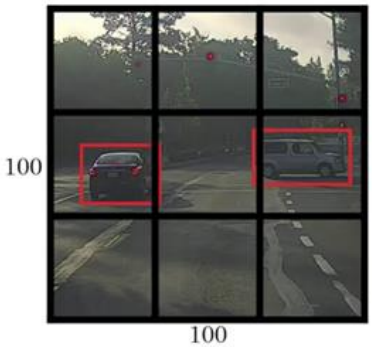

Fig. 1: Input image divided into 3 X 3 grid [6]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

On every grid, image classification and localization are applied. The bounding boxes and their equivalent class probabilities for objects are then predicted by YOLO.

In order to train it, the labelled data needs to be transferred to the model. Suppose the image is divided into a 3 X 3 grid and there is an aggregate of 3 classes in which the objects need to be categorised. Suppose that the classes are people, cars, and trucks, the y label is an 8-dimensional vector for each grid cell as shown in Table 1

Table 1: 8-dimensional Y label

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

Here,

- The probability of an object being present in the grid is described by **pc**
- **bx, by, bh, bw** defines the bounding box coordinates if an entity is present.
- The classes are reflected by **c1, c2, c3**.

Consider the first grid from Fig 1 as shown in Fig 2

Fig. 2: Grid having no object [6]

- Since there is no object in this grid as shown in Fig 2, pc will be zero and all the other entries in the y label will be '?' i.e., As there is no entity in the grid, it doesn't matter what the bx, by, bh, bw, c1, c2, and c3 values are.

  There are two objects (2 cars) in Fig 1; YOLO takes the mid-point of these two objects and assigns these objects to the grid containing the mid-point of these objects. The y label for the left-cantered grid with the car is as shown in Table 2.

Table 2: Y label of left centred grid in Fig 1

y =	1
	bx
	by
	bh
	bw
	0
	1
	0

Since in this grid there is an entity, pc will be to 1 and bx, by, bh, bw will be determined according to the same grid cell with which we are dealing. Therefore, since the car is the 2nd class, c2 = 1 and c1 and c3 = 0. An 8-dimensional output vector for each of the 9 grids is outputted. This performance will have a dimension to the (3 X 3 X 8).

The object could be assigned to a solitary grid where its mid-point is found, regardless of whether an entity spreads to more than one grid. By increasing the number of grids, we can reduce the odds of different objects occurring in a similar grid cell.

III. SPECIFICATIONS AND SYSTEM ARCHITECTURE

YOLO, in a single glance, takes the entire image and predicts for these boxes the bounding box coordinates and class probabilities. YOLO's greatest advantage is its outstanding pace, it's extremely fast, and it can handle 45 frames per second [1]. Amongst the three versions of YOLO, version-3 is fastest and more accurate in terms of detecting small objects.

The proposed algorithm, YOLO version-3 consists of total 106 layers [10]. The architecture is made up of 3 distinct layer forms. Firstly, the residual layer which is formed when activation is easily forwarded to a deeper layer in the neural network. In a residual setup, outputs of layer 1 are added to the outputs of layer 2. Second is the detection layer which performs detection at 3 different scales or stages. Size of the grids is increased for detection. Third is the up-sampling layer which increases the spatial resolution of an image. Here image is up sampled before it is scaled. Also, concatenation operation is used, to concatenate the outputs of previous layer to the present layer. Addition operation is used to add previous layers. In the Fig 3, the pink colored blocks are the residual layers, orange ones are the detection layers and the green are the up-sampling layers. Detection at three different scales is as shown Fig 3.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

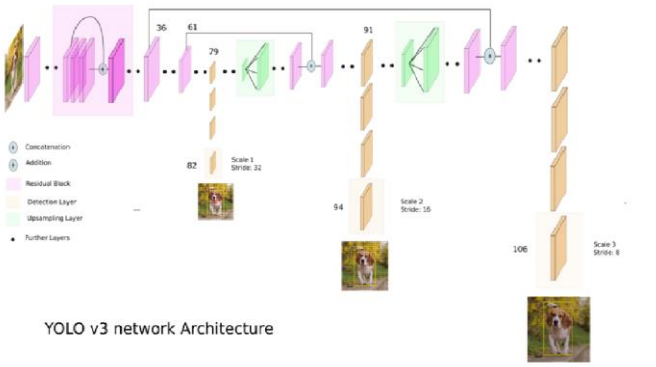

Fig. 3: Architecture of YOLO v3 [4]

YOLO v3 predicts 3 different scales of prediction. The detection layer is used to detect feature maps of three different sizes, with strides 32, 16, 8 respectively. This means that detections are made on scales of 13 x 13, 26 x 26 and 52 x 52 with an input of 416 x 416.

The working of YOLO is better explained in sections from A to I.

A. Encoding Bounding Boxes

Fig. 4: Right-centred grid [6]

Consider the right-cantered grid that includes a car, according to this grid, only bx, by, bh, and bw are determined. This grid will have the y label as shown in Table 2.

B. Making Predictions

The following formulae explain how the network output is converted to obtain bounding box predictions. The x, y centre co-ordinates, width and height of our prediction are bx, by, bw, bh. The output of the network is bx, by, bw, bh. The top-left coordinates of the grid include cx and cy. pw and ph are the dimensions of the anchor boxes [9].

$$b_x = \sigma(tx) + cx \quad \text{-- (1)}$$
$$b_y = \sigma(ty) + cy \quad \text{-- (2)}$$
$$b_w = pw \times e^{tw} \quad \text{-- (3)}$$
$$b_h = ph \times e^{th} \quad \text{-- (4)}$$

pc = 1(In Fig 4) since this grid has an object and it is a car, so c2 = 1. In YOLO, the grid coordinates in the form of bx, by are the x & y coordinates of the midpoint of the grid object . It will be (approximately) bx = 0.3 in this case & by = 0.4

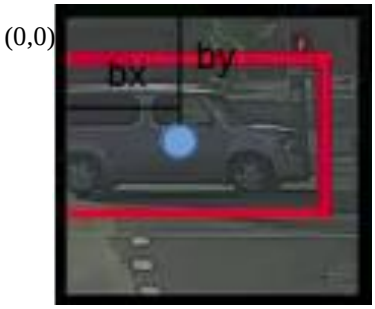

Fig. 5: x & y coordinates of the mid-point of the object [6]

bh is the ratio of the height of the bounding box (the red box in Fig 5) to the height of the corresponding cell of the grid, which is about 0.8 in Fig 5. bw is the ratio between the width of the bounding box and the width of the cell of the grid. This grid (Fig 5) will have a y label, as shown in Table 3.

Table 3: Y label of grid shown in Fig 5

y =	**1**
	0.3
	0.4
	0.8
	0.4
	0
	1
	0

bx and by will continuously lie in the 0 and 1 range while the midpoint in the grid will continually remain in the grid, bh and bw may be greater than 1 in the event that the dimensions of the bounding box are greater than the grid dimension.

C. Intersection over Union and Non-Max Suppression

It measures the intersection of the ground-truth bounding box and the expected bonding box over the union. Consider the ground-truth and the bounding boxes expected for a vehicle, shown in Figure 6.

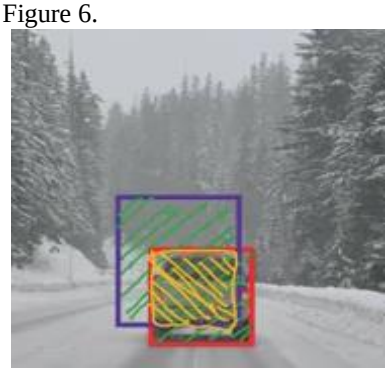

Fig. 6: Intersection over Union [6]

The red box in Fig 6 is the bounding box for ground-truth and the purple box is the anticipated one. The intersection area

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

over the union of these two boxes is determined by IoU. i.e., the shaded yellow region as shown in Fig 6.
In general,

IoU = Area of the intersection / Area of the union
i.e.,
IoU = Area of yellow box / Area of green box.

If the IoU is greater than 0.5, then the estimate is reasonable. 0.5 is an arbitrary limit, can be changed as explicit problems suggests. Instinctively, the higher the limit is raised, the better the predictions become. There is one more strategy that can radically boost YOLO's yield - Non-Max Suppression. One of the most well-known problems with algorithms for object detection is that they can recognize it on multiple occasions instead of detecting an object only a single time.

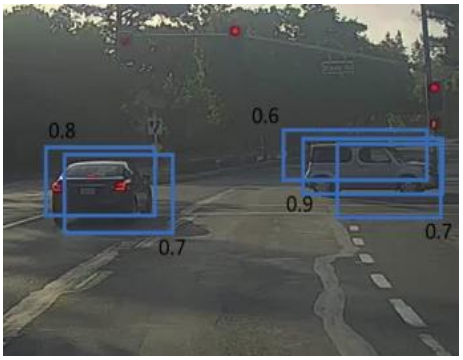

Fig. 7: Vehicles recognised more than once [6]

Cars are recognised more than once, as seen in Fig 7. To get a lone detection of any object, the Non-Max Suppression is employed

1. It looks at the probability of each detection first and selects the highest. In Fig 7, 0.9 is the greatest likelihood, so the 0.9 probability box is selected first.
2. The method then glances at the other boxes in the frame. Boxes that have a high IoU with the currently chosen box will be removed. The boxes with 0.6 and 0.7 probabilities are suppressed.
3. After the boxes have been removed, the next box is chosen from all of the most possible boxes, which in this situation is 0.8.
4. Once again, it looks at the IoU of the box and compresses boxes with a high IoU.
5. The algorithm repeats steps from 1 to 4, until the final bounding boxes are selected or compressed Non-Max Suppression is a way to take the boxes and suppress the close-by boxes of non max possibilities with the max likelihood, as shown in Figure 8 single detection of objects is obtained after applying the above-mentioned steps.

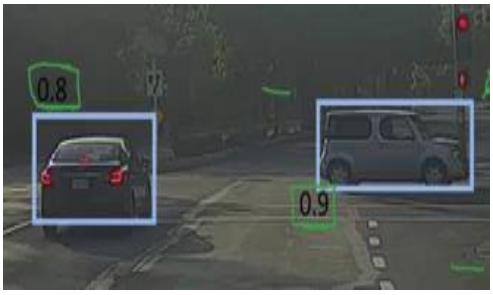

Fig. 8: Vehicles recognised only once [6]

D. Anchor Boxes

When multiple objects are in a single grid, it leads to the concept of the anchor boxes. Consider Fig 9, split into the grid of 3 X 3, an object is allocated to a grid by taking its midpoint and it is allocated to the corresponding grid according to its position. The midpoint of both objects is in the same grid in Fig 9.

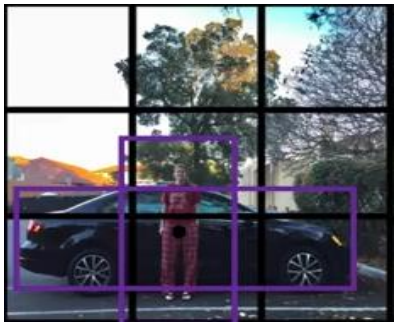

Fig. 9: Midpoint of 2 objects in same grid [6]

Two separate types, called anchor boxes or anchor box shapes, are predefined for production of both boxes. For each grid, instead of one output, two outputs are obtained.

Anchor box 1: Anchor box 2:

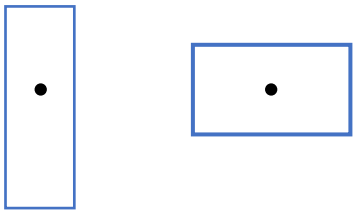

Fig. 10: Two outputs of one grid

The y label of YOLO with the anchor boxes is as shown in Table 4.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

Table 4: Y label with 2 anchor boxes

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3
	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

The first 8 rows of the y label represent the anchor box 1 and the next 8 represent the 2nd anchor box. The objects are allocated to the boxes on the basis of the similarities between the bounding box and the shape of the box. Since the shape of anchor box 1 is identical to the person's bounding box, the person is allocated to the box 1 and the car to box 2. Instead of 3 X 3 X 8 (using 3 X 3 grid and 3 classes), the output for this situation is 3 X 3 X 16 (since 2 anchors are utilized).

E. Flowchart

Flowchart of that steps that the YOLO algorithm follows

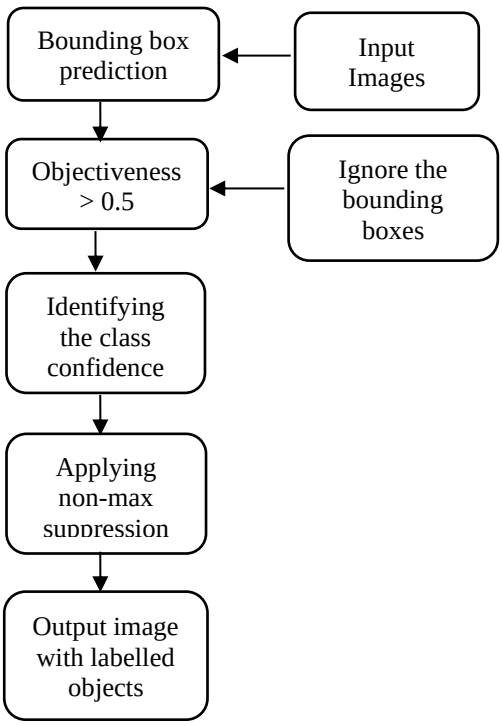

Fig. 11: Flow chart showing working of YOLO V3

F. Testing

The new picture will be split into the same number of grids that is selected during the training. The model predicts a 3 x 3 x 16 output for each grid. The 16 values in this prediction are in the same format as the training label.

G. Training

The input for training the model will be images and their respective y labels. Fig 1 is divided into 3 X 3 grid with two grid anchors, with 3 separate classes of objects. The corresponding y labels has the shape of 3 X 3 X 16. Training takes the form of an image and maps it into a target 3 X 3 X 16.

H. Implementation of YOLO

- **Darknet: T**his algorithm is implemented using an open-source neural network framework i.e., Darknet which was developed in C Language and CUDA technology to render speedy calculations on a GPU necessary for real-time predictions.
- **DNModel.py:** Darknet Model file is a computer vision code used for building the model using the configuration file and it appends each layer.
- **Util.py:** Contains all the formulas used.
- **imageprocees.py:** Required to perform the image processing task. It takes all the input images to resize them and perform Up-sampling, also performs transpose function.
- **detect.py:** The main code which is run to perform object detection. This code uses all the above-mentioned files to perform object detection. Performs all the functions according to the YOLO concept

I. COCO Dataset

COCO basically means that the data set images are daily objects recorded on everyday scenes and provides the labelling of multi-objects, annotations of segmentation masks, image captioning, key-point detection and panoptic segmentation annotations with a total of 81 categories, making it a very flexible and polyvalent dataset [8].

IV. APPLICATIONS

1. **Vehicle detection**

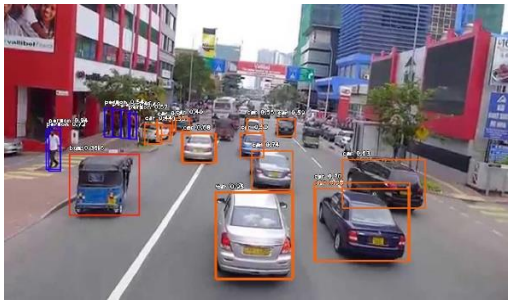

Fig. 12: Vehicle Detection

Different vehicle types, i.e., cars, vans, bikes, busses, training vehicles, vessels, bicycles, flights are all detected by YOLO picture model and all-in real time. While any of the above vehicles are detected, a bounding area is formed around which the vehicle is detected. Type of vehicle is also shown above the bounding box.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

2. **Crowd detection**

Fig. 13: Crowd Detection

This system uses a collection of rules for object detection to screen the flow of humans in various places. The system can control information in real time and track the position of unusual crowds.

3. **Optical character recognition**

Fig. 14: optical character recognition

The mechanical or electronic translation of printed, hand-written or published text images, into machine-coded textual material, regardless of whether the scanned report is, the photograph of a report, or a real time state, is the optical recognition of character regularly abbreviated by the term OCR.

4. **Image fire detection**

Fig. 15: Image fire detection

Real-time vision identification of fireplace has now been enabled for monitoring equipment, guaranteeing a first-class trend in the field of fireplace security.

V. CONCLUSION

YOLO is one of the best-known, most powerful object detection models, dubbed "You Only Look Once." YOLO is the first option for every real-time identification of objects. Both input images are divided into the SXS grid structure by YOLO algorithms. For object detection, any grid is responsible. Now these grid cells forecast the observed object boundary boxes. We have five principal attributes for each box, including x and y for coordinates, w and h for object width and height and an insight into the possibility that the box holds the object.

In recent years deep learning-based object identification has become a hot spot for analysis due to its powerful study skills and scale transition. This paper suggests a series of YOLO rules to classify objects using a single neural network for the purpose of detection. The rules are easy to create and can be instantly comprehensively photographed. Limit the classifier to a particular area through position concept techniques. In the prediction of limits, YOLO accesses the whole photograph. Moreover, in history regions it expects fewer false positives. This algorithm "only looks once" as in it requires only one forward propagation to cross through the network to make estimations.

REFERENCES

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.91

[2] Chandan G, Ayush Jain, Harsh Jain, and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV" Proceedings of the International Conference on Inventive Research in

[3] Computing Applications (ICIRCA 2018) IEEE Xplore Compliant Part Number:CFP18N67-ART; ISBN:978-1-5386-2456-2

[4] Chethan Kumar B, Punitha R, and Mohana, "YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications" Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020) IEEE Xplore Part Number: CFP20P17-ART; ISBN: 978-1-7281-5821-1

[5] 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

[6] Pulkit Sharma, "A Practical Guide to Object Detection using the Popular YOLO Framework – Part III" DECEMBER 6, 2018.

[7] Nikhil Yadav , Utkarsh , "Comparative Study of Object Detection Algorithms", IRJET, 2017.

[8] Viraf, "Master the COCO Dataset for Semantic Image Segmentation", May 2020.

[9] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", University of Washington.

[10] Karlijn Alderliesten, "YOLOv3 — Real-time object detection", May 28 2020.

[11] Arka Prava Jana, Abhiraj Biswas, Mohana, "YOLO based Detection and Classification of Objects in video records" 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.

[12] Akshay Mangawati, Mohana, Mohammed Leesan, H. V. Ravish Aradhya, "Object Tracking Algorithms for video surveillance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0676-0680.