International Journal for Modern Trends in Science and Technology, 6(12): 360-364, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061267
Available online at: http://www.ijmtst.com/vol6issue12.html

Real-Time Object Detection Model

Akash Kumar[1] | Dr. Amita Goel[2] | Prof. Vasudha Bahl[3] | Prof. Nidhi Sengar[4]

[1]B-tech scholar, Department of IT Maharaja Agrasen Institute of Technology, Delhi, India.
[2]Professor, Department of IT Maharaja Agrasen Institute of Technology, Delhi, India.
[3]Assistant Professor, Department of IT Maharaja Agrasen Institute of Technology, Delhi, India.
[4]Assistant Professor, Department of IT Maharaja Agrasen Institute of Technology, Delhi, India

To Cite this Article
Akash Kumar, Dr. Amita Goel, Prof. Vasudha Bahl and Prof. Nidhi Sengar, "Real-Time Object Detection Model", *International Journal for Modern Trends in Science and Technology*, 6(12): 360-364, 2020.

Article Info
Received on 12-November-2020, Revised on 05-December-2020, Accepted on 11-December-2020, Published on 15-December-2020.

ABSTRACT

Object Detection is a study in the field of computer vision. An object detection model recognizes objects of the real world present either in a captured image or in real-time video where the object can belong to any class of objects namely humans, animals, objects, etc.
This project is an implementation of an algorithm based on object detection called You Only Look Once (YOLO v3).
The architecture of yolo model is extremely fast compared to all previous methods. Yolov3 model executes a single neural network to the given image and then divides the image into predetermined bounding boxes. These boxes are weighted by the predicted probabilities. After non max-suppression it gives the result of recognized objects together with bounding boxes. Yolo trains and directly executes object detection on full images.

KEYWORDS: *Object Detection, Neural Network, Yolov3, Bounding Box*

I. INTRODUCTION

In the past decades unlike humans computers couldn't immediately know what and where are the objects in the image. But in recent years, things have changed drastically in the field of computer vision. Now there are many methods available for object detection in computers however they are not as fast and accurate as human's eye but still a very progressive step towards this very important and challenging field in computer vision.
Object detection is applied in several jobs for example in securities to check if no one is carrying any forbidden objects, movement acknowledgement, In CCTV to identify any movement and objects, etc.

Some recent methods like R-CNN, instead of selecting huge number of regions in an image this method use selective search to extract just 2000 regions. To identify those 2000 regions, the R-CNN uses its selective search algorithm for classification and when the classification is done it starts post-processing to eradicate duplicate detections and to refine the bounding box. This approach is slow and hard to optimize because each individual component has to be trained separately.

In This project we use different approach in which we reframe object detection as a single regression problem.
Yolo is very refreshing and simple. It requires single convolutional network to predict multiple

bounding boxes and class probability of predicted bounding boxes simultaneously.

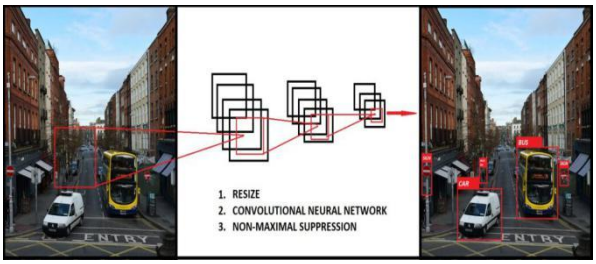

Figure 1. Steps in YOLOv3 model

YOLO trains on full images and directly advances to detection performance.

II. Theory

As it follows only one look at the image slidingwindows is not the efficient approach instead it divides the entire image into grid of S X S dimensions. After splitting each cells are liable for predicting numerous things.

First step, each cell will predict confidence value for each bounding box. Each confidence score or objectness score can be obtained as follows:

$$C^{j}_{i} = Pi, j(Object) * IOU^{truth}_{pred}$$

Where C^{j}_{i} is objectness score or confidence score of j^{th} bounding box in i^{th} cell. Pi, j(Object) is just a function of object. IOU^{truth}_{pred} is the intersection of union between the predicted box and the ground truth box

For the final result we multiply the probabilities with the confidence values and from this we get bounding boxes weighted by their probabilities for containing that object.

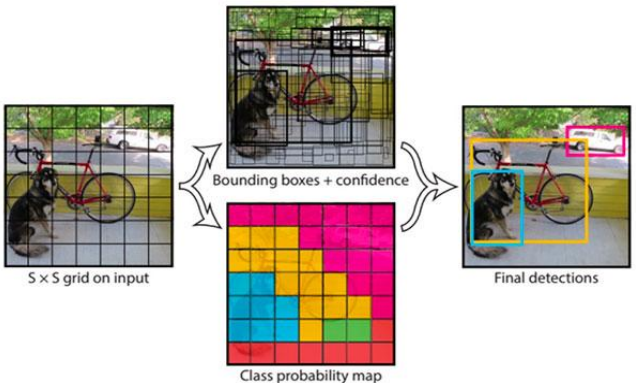

Figure 2. Yolov3 model working

Now to get rid of all low confidence value predictions simple thresholding will help.

Even after applying thresholding duplicates can still be present and to get rid of them we apply non-maximum suppression.

This process will take a bounding box with the highest probability and then check other closest bounding box. And the ones with the highest overlap highest IoU will be suppressed. Because everything is done in single pass it is extremely fast with good accuracy.

III. Literature Survey

In this section we will go through the history of object detection in multiple aspects.

The progress object detection has gone through two historical periods: one without deep learning (before 2014) and the other with deep learning (after 2014)

Voila Jones Detector

Before deep learning traditional methods were used to detect objects. It all started 19 years ago when **P. Voila** and **M. Jones** achieved a milestone in object detection. They detected the human face without any constraints.

It was running on a 700MHZ Pentium III CPU. The detector followed the basic approach for the detection i.e.; sliding windows: to go through every possible location and scales the image to see if any window contains human face. It may sound easy but the calculations behind it was beyond the power of computers at that time.

Before 2014 many traditional approaches came for example in 2005 **N Dalal** and **B Triggs** came up with HOG detector and in 2008 as an extension of HOG detector with variety of improvement **R. Girshick** gave us a detector called DPM (Deformable Part-based Model).

Although today's object detectors have far surpassed these detectors in terms of accuracy and time but many of them are still deeply influence by these detectors because of their valuable insights.

R-CNN

In 2013 three authors proposed a model that uses selective search for generating region proposals by uniting similar pixels into regions. The three author names are Alex Krizhevsky, Geoff Hinton, and Ilya Sutskever.

R-CNN follows the idea given above, it starts with the extraction of object bounding boxes by selective search.

Working of Selective Search:

1. First initiate initial sub-segmentation, initiate many candidate regions
2. Use greedy approach to recursively merge similar regions into larger ones.

3. Use the initiated regions to produce the final candidate proposals.

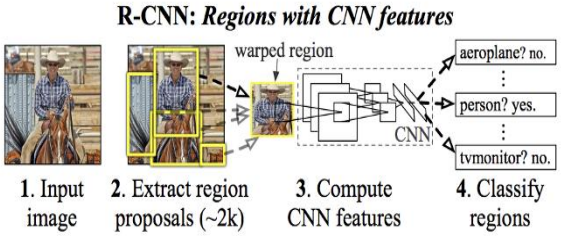

Figure 3. Regions with CNN Features

The success R-CNN had was amazing and it made big progress but it had drawbacks: because of the large number of overlapped proposals it leads to extremely slow detection speed. It takes 14 secs per image with GPU.
To solve this problem Fast R-CNN came.

FAST R-CNN

This came in 2015 just after SPPNet it was further improvement of R-CNN and SPPNet. R. Girshik proposed the model which enables us to train a detector simultaneously with bounding box regression under same network. It increased the map up to 70% and had 200 times faster detection speed compared to R-CNN. But it's detection speed was still limited.

FASTER R-CNN

Shortly after the Fast R-CNN a new model was introduced named FASTER R-CNN. As it name sounds it was faster than any other previous versions of R-CNN model. Although having speed fast in detection of objects it still had limitations like: there is still computation redundancy at subsequent detection stage.

FEATURE PYRAMID NETWORKS

In 2017, A method came to know which works on basis of Faster R-CNN called Feature Pyramid Networks (FPN). FPN worked on a top down architecture with lateral connections. It was for building high level semantics. FPN has now become building block of today's detectors.

YOU ONLY LOOK ONCE

In 2017 a new model named YOLO was proposed. The person behind this proposal was R. Joseph.
It follows completely different approach compared to other detectors. From its name we can tell it takes only one look at the image to detect object. Yolo is the very first one-stage detector of deep learning era.

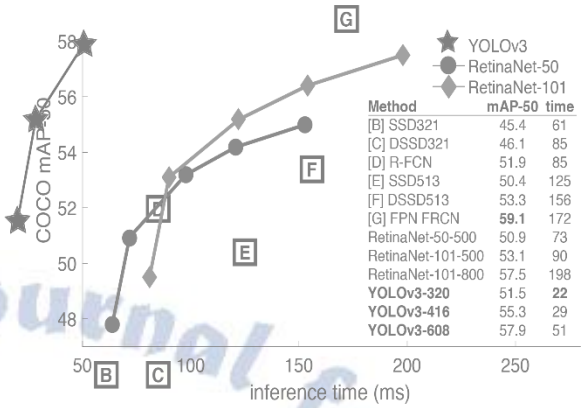

Figure 4. Comparison of other models with Yolov3

The authors have abandoned the previous techniques like "proposal detection + verification". Instead the yolo model just apply single neural network on the full image. This network splits the image in bounding boxes and predicts the probability for each region simultaneously.
R. Joseph has further improved his model by launching yolo v2, v3 editions.
The speed yolo gives for the detection is outstanding but this reduces its accuracy compared to the two staged detectors. However, R. Joseph is improving the accuracy of Yolo model without compromising its speed. Now time will tell how much yolo improves its accuracy.

IV. YOLOv3 ARCHITECTURE AND TRAINING OF MODEL

The network architecture of yolo v3 model is inspired by the Google Net model for image classification. This architecture consists 24 convolutional layers followed by 2 completely connected layers. Yolo model simply uses 1 x 1 reduction layers and 3 x 3 convolutional layers. The full network is show in above figurethe abbreviation "Fig." even at the beginning of a sentence.

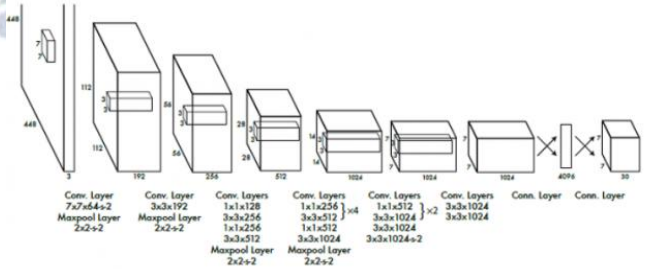

Figure 5. Network Architecture of YOLO v3 Model

A yolov3 model takes input with size of 416 X 416 and feature map of three types 1) 13 x 13 x 69 2) 26 x 26 x 69 3) 52 x 52 x 69 as the output image.

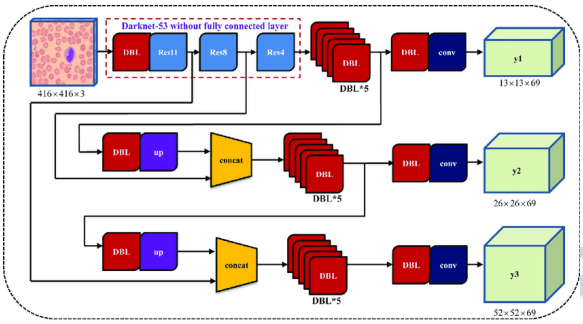

Figure 6. pipeline of yolov3 model with input size of 416 x 416

Training with custom dataset

The very first process for training of custom dataset is to have datasets of image we want to detect. In this project darknet is used for training the object detector. Darknet provides an open source neural network and

it is written in C and CUDA. First we have to set up our GPU machine before installing darknet. After installation we collect the dataset and start annotation, Annotation process makes txt file for each image which contains coordination and class object for every image. In this project **labellimg** tool is used for annotation. The format is '(object-id) (x-centre) (y-centre) (width) (height)'. Change GPU from 0 to 1 and OpenCV to 0 to 1 as well. After collecting the labelled dataset save them in the root directory of Darknet

Now split dataset in two text files train.txt and test.txt one contains the path and the other contains the test set. After this prepare the configuration files some Modification is needed to in the yolo model for preparation of cfg files. Object.names and training.data these are two files required to be created in which Object.names contains name of the classes and training.data contains parameters for the training. Download pretrained convolutional weights of yolov3 put it in main directory i.e.; darknet. Now after all of this we can start training our dataset.

". /darknet detector train cfg/obj.data cfg/yolov3-tiny.cfg darknet53. conv.74"

when average loss is less than 0.06 stop the training process. After training we now have .weights files. Now the training is done we can detect objects we trained from our custom dataset.

V. RESULTS

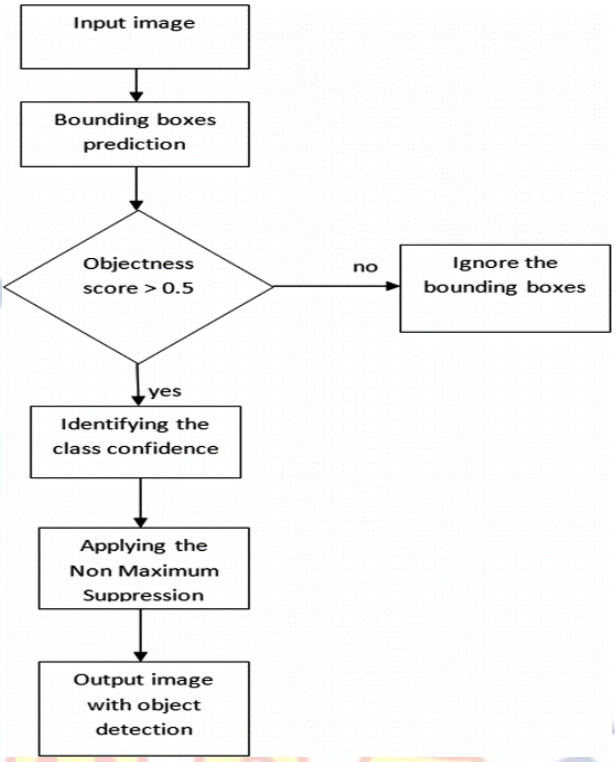

Figure 7. A DFD diagram of yolo v3 model

We built our object detector and used yolov3 model in it. It can detect objects in real-time and as well as in still images. We created our own custom dataset to help the community in this time of corona virus, it can help in public places or places where mask is mandatory to detect if a person has worn a mask or not.

Below are result of our object detector. We can see the bounding box around the face of person, by looking at results it is found that this model is working perfectly fine and providing almost accurate results.

Figure 8. Results (in fig 1. The object detector identifies that the person has worn mask in fig 2. The detector identifies that the person has not worn mask)

VI. Conclusions

The ability to recognize objects in real-time is useful in many applications: from security to self-driving cars, and in health care as well. We used this object detection to help in the time of pandemic to detect if a person has worn a mask or not. Until the vaccine is not ready for covid-19 virus and everything don't get to normal this object detector can help in maintaining rules where not wearing mask is prohibited. It would help in slowing down the spreading of virus.

In this project we used a different model, a unified model for object detection, compared to others this model is simple to construct and easy to understand. Yolo model is trained on loss function. It is extremely fast with good accuracy. The algorithm is much more efficient to use in real time.

VII. References

[1] P. Viola and M. Jones. "Robust real-time object detection. International Journal of Computer Vision", 4:34–47, 2001.

[2] YOLO model Juan Du1," Understanding of Object Detection Based on CNN Family", New Research, and Development Center of from china

[3] Joseph Redmon research paper on unified, real-time object detection in YOU LOOK ONLY ONCE.

[4] Zheng, Z. Wang, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. 2019, Available online.