Object Detection Using Artificial Intelligence

Implementation Report

Team Members

E. Venkata Srujan(2019250)
M. Gnana Sundar(2019286)

Instructor

Dr. Kusum Kumari Bharti

INTRODUCTION

We have chosen the topic “Object Detection system” and we would like to implement a tool which detects the different objects present in image and classifies them (as dog or human etc.) with a bounding box around them.

We humans mostly can detect the objects we see in an image but for computer vision it is difficult to detect the object. Computer vision involves working with digital images and videos to deduce some understanding of contents within these images and videos. Computer vision cannot perform complex tasks like identifying multiple objects and detect obstacles. But now-a-days with the availability of large amounts of data, faster GPUs, and better algorithms,

There are a few different algorithms for object detection. We have taken one of the best Algorithms that is YOLO (You Only Look Once) .It's an object detector that uses features learned by a deep convolutional neural network to detect an object. It’s is based on regression – instead of selecting interesting parts of an image, we’re predicting classes and bounding boxes for the whole image in one run of the algorithm It's an object detector that uses features learned by a deep convolutional neural network to detect an object.

It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes. The biggest advantage of using YOLO is its superb speed – it’s incredibly fast and can process 45 frames per second. YOLO also understands generalized object representation

Implementation:

We will use yolo Algorithm for object detection using tensorflow. Tensorflow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

Initially it divides the image into many grids to assigned an object to a grid, we took the midpoint of the object and based on its location, assign the object to the corresponding grid,
Each of these grids is responsible for predicting some bounding boxes. A bounding box describes the rectangle that encloses an object .we need to calculate the IoU(Intersection over Union)
We define a box using its two corners (upper left and lower right): (x1, y1, x2, y2). Next, we also need to find the coordinates (xi1, yi1, xi2, yi2) of the intersection of two boxes where: xi1, yi1 is the maximum coordinates of x1, y1 of the two boxes and xi2, yi2 is the minimum coordinates of x2, y2 of the two boxes
Note that to calculate the area a rectangle (or a box) we can to multiply its height (y2 – y1) by its width (x2 – x1).
So to calculate IoU, first calculate the area of intersection by this formula area intersection = (xi2 - xi1)*(yi2 - yi1).
Next, calculate the area of union
area = (area of box 1 + area of box 2) – area intersection
Therefore IoU=area intersection/union area.
Now we implement non-max suppression to find the best box, the steps are: Select the box that has the highest score.

Compute its overlap with all other boxes, and remove boxes that overlap it more than a certain threshold which we call iou threshold. Go back to step 1 and iterate until there are no more boxes with a lower score than the currently selected box
These steps will remove all boxes that have a large overlap with the selected boxes. Only the best boxes remain.

When we divide the image into grids,if the midpoint of both the objects lies in the same grid. We will only be getting one of the two boxes, either for the car or for the person. But if we use anchor boxes, we might be able to output both boxes first, we define many different shapes called yolo anchor or yolo anchor masks. Now, for each grid, instead of having one output, we will have more outputs. We can always increase the number of anchor boxes as well.

After predicting the box around the object we are going to use , one of the open source neural network framework called Darknet in Yolo algorithm to predict the object in the box . Darknet is a convolutional neural network that acts as a backbone for the YOLOv3 object detection approach. Yolo has an advantage of convolution layers (Darknet) to deepen the network structure, which also inserts the residual block to the network.

We use the yolo conv to go through the Darknet layers to using the trained weights files to predict the class of the object in the box based on the class list defined as the class names which contains names of 80 different objects. These trained weights of yolo helps to filter the objects present in the box with the help of residual layers

We will predict the bounding boxes of the objects present in the image using the Darknet network around the object and then draw all the different possible boxes around the object

We can now predict the bounding box around the object and the class of the object using the Yolo we pass these predictions as inputs and then we can draw the output of the prediction on the image and then show that predicted bounding box and the class of object on the image.

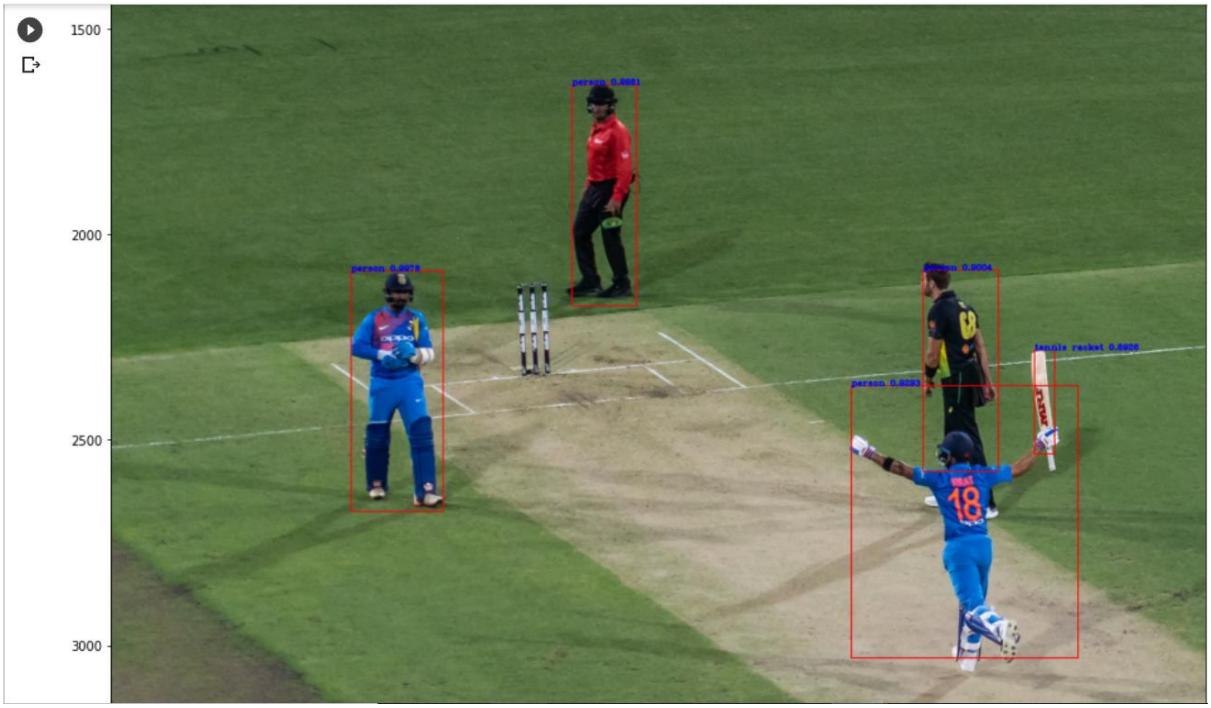
We need to Freeze a layer to control the way the weights are updated. When a layer is frozen, it means that the weights cannot be modified further. We need to transforming each image and normalizing it in range [0,1] for printing the probability of the object that belongs to the given class

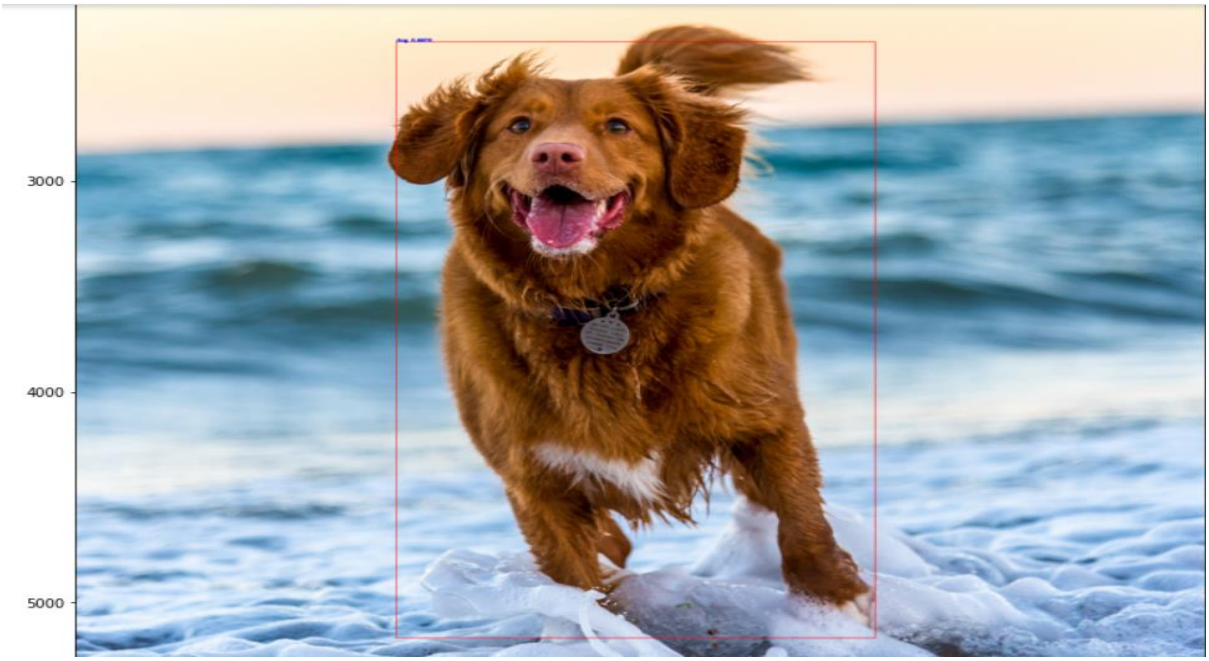
Comparison with other algorithms:

There are mainly three algorithms for object detection Yolo, rcnn, faster rcnn. Among three Yolo is the best.
The biggest advantage of using YOLO is its superb speed – it's incredibly fast. YOLO also understands generalized object representation.
It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes where as other algorithms such as faster rcnn doesnot run on a single instance.
It is more advanced version when compared to other algorithms. It detects object more accurately than the other algorithms
Yolo detects large more precisely when compared to other algorithms.

Results and Conclusion:

This section is about the result obtained after performing the method given above. And after implementing the method, we have tested our model that detects the objects in the picture given. And below are the few examples that we have checked for:





Upgradation to be introduced

After implementing the procedure that we have mentioned above, we thought of upgrading it to detect the objects in videos, in our implementation the objects in the video cannot be detected. And we are also thought of adding a feature of Counting the particular object type in the image. Presently we are working on it

The code of our implementation is hosted on github and google drive:

GITHUB - https://github.com/Sundar1872/Object-Detection

Google Drive – https://drive.google.com/drive/folders/1-0TOcTrnhslTN3utIv-WIgjhrBaUt47P

References:

1. Object Detection with Deep Learning by Zhong-Qiu Zhao, Peng Zheng (research paper) .
2. A Survey of Modern Object Detection Literature using Deep Learning by karanbir Chahal and Kuntal Dey