Object Detection using YOLOv5

Deep Learning 18MXBF

Author:
Ajith Sam Raj S
19MX202
Department of Computer Applications
PSG College of Technology

Abstract

Object detection, which combines object categorization and object location within an input, is regarded as one of the most difficult challenges in the field of computer vision. Object detection is the ability of computer and software systems to locate and identify items in an image as in Fig 1.1. Deep neural networks (DNNs) have recently been shown to outperform previous approaches in object detection, with the YOLO family (You Only Look Once model) being one of the most advanced DNN-based object detection methods in terms of both speed and accuracy.Object detection offers a wide range of real-time applications, including object tracking, automated CCTV surveillance, person detection, vehicle detection and so on. In this investigation, an object detector based on the YOLOv5 algorithm is used. A living chicken is the object to be discovered. The model has an accuracy of 63%. The findings show that the proposed model outperforms competing models in terms of prediction, accuracy, and sensitivity, indicating its utility in real-time object detection.

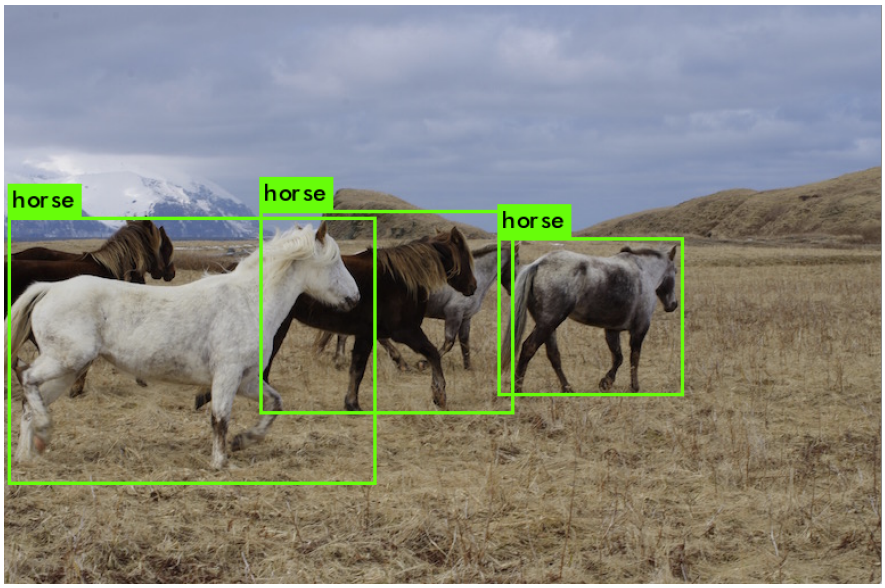


Fig 1.1 Object Detection example

Keywords

Deep Learning, CNN, YOLOv5, Object Detection, Bounding Box.

Introduction

Every day, new algorithms in the field of Deep Learning are being developed. For security, CCTV cameras are now installed in every home, office, and public places. However, in order to detect a person or an object, the CCTV camera requires human involvement. However, it can detect objects with the help of object detection algorithms. Let's see the evolution of Object Detection algorithms. "Viola Jones Detector", created by Paul Viola and Michael Jones in 2001, allows for the real-time detection of human faces. It uses sliding windows to examine all potential places and scales in a picture for the presence of a human face.

"Histogram of Oriented Gradients" (HOG) is an enhancement on the scale invariant feature transform and shape contexts of its time, as proposed by N. Dalal and B. Triggs in 2005. HOG uses blocks, a dense pixel grid in which gradients are formed by the amount and direction of change in the brightness of pixels within the block (similar to a sliding window).

P. Felzenszwalb first proposed the "Deformable Part-based Model" (DPM) in 2008 as an expansion of the HOG detector. R. Girshick later improved the design in a number of ways. By detecting its window, body, and wheels, the difficulty of detecting a "vehicle" can be broken down into a "divide and conquer" method. This is a strategy that DPM employs. The training procedure entails learning how to properly decompose an object, while inference is putting together detections from various object elements.Convolutional neural networks were reborn in 2012, when deep convolutional networks were successful in learning stable and high-level feature representations of an image. The concept of Regions with CNN features (RCNN) for object detection broke the object detection deadlocks in 2014.

"Region based Convolutional Neural Networks" (RCNN) begins with a selective search to extract a set of object proposals (object candidate boxes). Each suggestion is then rescaled to a fixed size image and fed into a CNN model that has already been trained to extract features. Finally, linear SVM classifiers are utilised to predict the presence of an object and recognise object types inside each region. The three variants of RCNN are Fast RCNN, Faster RCNN and Masked RCNN.

To locate the object within the image, all of the prior object detection techniques used areas. The network does not examine the entire image; instead, it examines portions of it that have a high probability of having the object. YOLO improves detection performance by training on entire photos. A single CNN predicts multiple bounding boxes and class probabilities for those boxes using "YOLO". It also predicts all bounding boxes for an image across all classes at the same time.

YOLO

One of the most prominent object detection convolutional neural networks is YOLO (You Only Look Once). Following the publication of their first YOLO article by Joseph Redmon et al. in 2015, they published further versions in 2016, 2017, and 2020 by Alexey Bochkovskiy. Internally YOLO employs Convolutional Neural Network (CNN) to predict the objects and Bounding box to distinguish the objects. Before feeding the images into the algorithm, they must be pre-processed.

In computer vision, bounding boxes are the most prevalent sort of annotation. Bounding boxes are rectangular boxes that define where the target object is located. The x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle can be used to determine them. Object recognition and localization tasks frequently employ bounding boxes. This process is called Annotation.

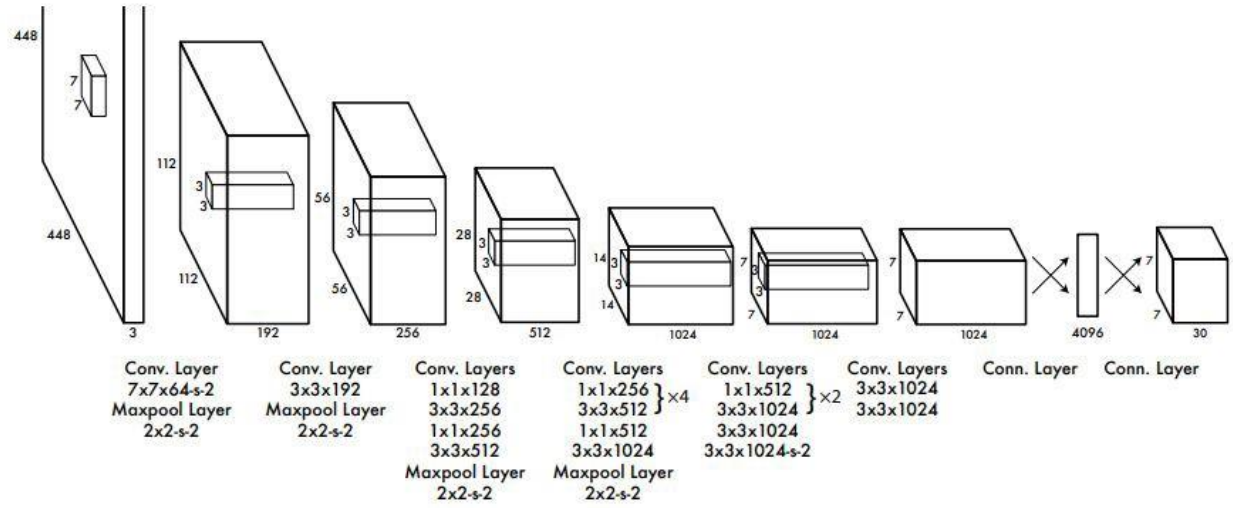


Fig 1.2 Model Architecture

From Fig 1.2 we can infer that YOLO has a feature extractor with 24 convolution layers. They are followed by two fully connected layers that are responsible for object classification and bounding box regression. The result is a tensor with dimensions of 7 x 7 x 30. YOLO CNN, like VGG19, is a simple single path CNN. With inspiration from Google's Inception version 1 CNN, YOLO uses 1x1 convolutions followed by 3x3 convolutions. Except for the final layer, all layers use leaky ReLU activation. A linear activation function is used in the last layer. The optimizer employed is Stochastic Gradient Descent (SGD).

Dataset

The object to be detected is a living Chicken. There is no specific data set for living chickens. The custom has been prepared with 22 images for the train and 8 images for testing.

Working

The image is first separated into several grids. The dimensions of each grid are S x S. Fig 1.3 shows how a grid is created from an input image.

Fig 1.3 Image to Grid

There are several grid cells of identical size in Fig 1.3. Objects that appear within grid cells will be detected by each grid cell. If an item centre emerges within a specific grid cell, for example, that cell will be responsible for detecting it.

A bounding box is an outline that draws attention to a certain object in a picture.The attributes present in every bounding box are Length (bw), a certain height (bh), class (for example, person, car, traffic light, etc.), The letter c is used to indicate this Center of the bounding box (bx,by). See an example of the attributes in Fig 1.4.

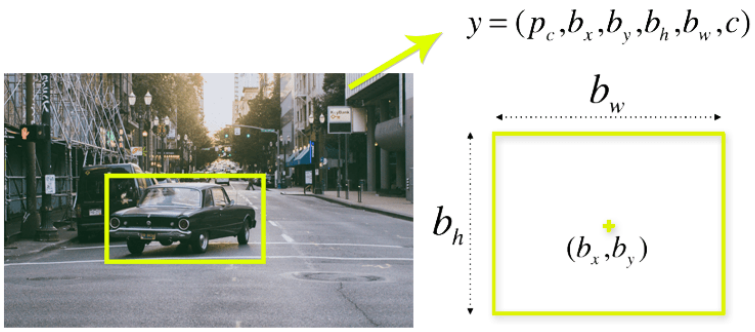


Fig 1.4 Bounding Box attributes

The concept of intersection over union (IOU) illustrates how boxes overlap in object detection. IOU is used by YOLO to create an output box that properly surrounds the items. The bounding boxes and their confidence scores are predicted by each grid cell. If the anticipated and real bounding boxes are identical, the IOU is 1. This approach removes bounding boxes that aren't the same size as the actual box. Fig 1.5 is a simple example of how an IOU works.

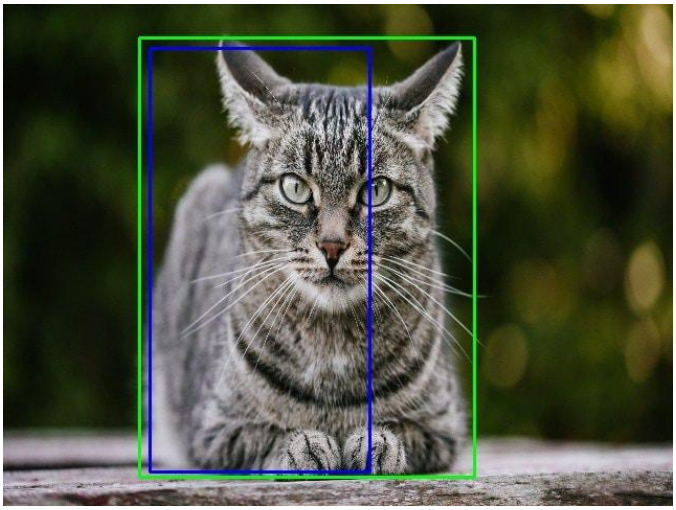


Fig 1.5 IoU

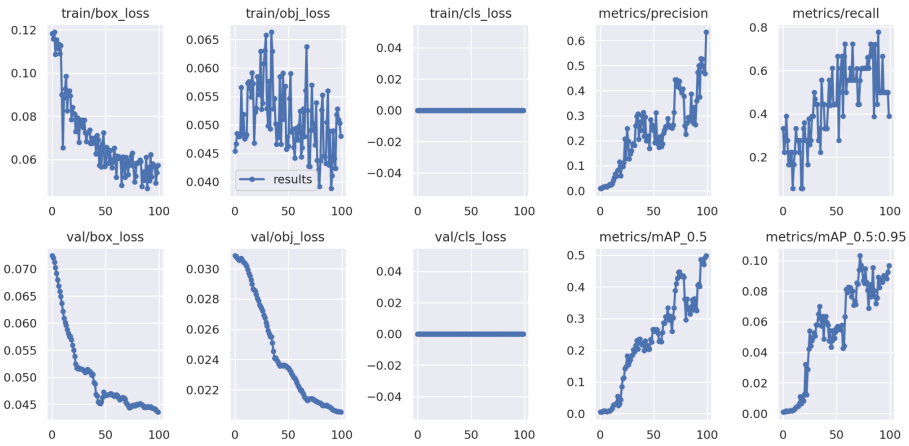
There are two bounding boxes in Fig1.5, one in green and the other in blue. The blue box represents the anticipated box, and the green box represents the actual box. YOLO makes sure the two boundary boxes are the same size.

The image is first subdivided into grid cells. B bounding boxes are forecasted in each grid cell, along with their confidence scores. To determine the class of each object, the cells estimate the class probability. We can see at least three types of objects, for example: a car, a dog, and a bicycle. A single convolutional neural network is used to make all of the predictions at the same time. The predicted bounding boxes are equal to the true boxes of the objects when intersection over union is used. This phenomena gets rid of any extra bounding boxes that don't fit the objects' properties (like height and width).The final detection will be made up of distinct bounding boxes that exactly suit the objects.

Inference

Head over to Ultralytics Git Repository and employ the notebook provided. Annotate the image and export the YOLO format zip file. Upload the files to the colab. As it's a custom object detection i.e.Predicting a living chicken coco file has to be rewritten. Download the coco.yaml file and rewrite the number of classes and class then upload the file again to the colab. Now it's ready for model building. See build report at wandbi.ai .

Train Validation



Metrics

Conclusion

The YOLO algorithm and how it is employed in object detection have been discussed in this work. When compared to other object detection approaches such as Fast R-CNN and Retina-Net, this technique delivers better detection results.

In conclusion, we learned about object detection and the YOLO algorithm. The major reasons for the YOLO algorithm's importance have been discussed. We've learned how to use the YOLO algorithm. We've also learned about the major methods that YOLO employs to detect things.