Journal of Engineering Sciences
Vol 11, Issue 6,June/2020
ISSN NO:0377-9254

A Review on YOLO (You Look Only One) - An Algorithm for Real Time Object Detection

Mukesh Chandra Arya[1], Anchal Rawat[2]
[1]Department of Computer Science & Engineering, [2]Department of Computer Science & Applications
GBPEC, Pauri Garhwal 246194, Uttarakhand
Mukesharya808@gmail.com, Aanchal752@gmail.com

ABSTRACT
Object detection is part of image processing method. The compare or match of any object or image to another object or image we use the object detection. In this paper we discuss about the YOLO (you only look one) algorithm. Various detection algorithm in present to use of image detection or object detection the Yolo is one of them. Yolo use in the real time system object detection not a region based object detection
Keywords- *YOLO, object detection, bounding box, image classification.*

1. INTRODUCTION

Yolo is an object detection algorithm.it detects multiple objects present in an image and create a bounding box around them Yolo brings a unified neural network architecture to the table, single architecture which does bounding box prediction and also gives out class probabilities. In other architecture like RCNN, they fast generate potential bounding box in image and run then run a classifier on these proposed boxes. After classification post processing refines the bounding boxes, eliminate duplicate detection and rescore the boxes based on these object in the scene.

In Yolo a single convent simultaneously predict multiple bounding boxes and also the class probabilities for those boxes. This allows Yolo to optimize. Yolo is fast and it reason about the image globally while making predictions example. It makes less than half number of background error compared to fast RCNN.

2. UNIFIED DETECTION

Here the separate components for detection of objects are merged into a single neural network. For prediction of each bounding box, the features from entire image are used. It also predicts bounding boxes for image at the same time. The design of YOLO allows end- to-end training and real-time rate while keeping high level average precision. Here the system separate the input image into S_S grid. If the object's centre drops into grid cell then the grid cell manages object detection. Every grid cell estimates B bouding boxes and then confidence scores related to them. The collected confident scores shows whether the model is confident or not for having an object and also how precise it believes the box is that it estimates. Formally confidence is defined as Pr(Object) _ IOUtruthpred . The confidence score should be zero if there is no object in the cell. Otherwise the confidence score should be equal to the intersection over union (IOU) joining the ground truth and the predicted box. Every bounding box have 5 predictions: x, y, w, h, and confidence. The coordinates (x, y) shows the center of box respective to the boundaries of the grid cell. The height and width are estimated with respect to the entire image. Overall the confidence estimates the IOU among prediction box and ground truth box. Every grid cell also estimates C conditional class possibilities, Pr(ClassijObject). These possibilities are controlled on the grid cell having an object. We only estimate single set of class possibilities in each grid cell, ignoring the quantity of boxes B. At the time of testing, we multiply the conditional class possibilities and the individual box confidence predictions, Pr(ClassijObject) _ Pr(Object) _ IOUtruthpred = Pr(Classi) _ IOUtruthpred (1) that provides us confidence scores for each box. These scores encode the possibility of the class showing in the box and how exactly the object fits into the estimated box.

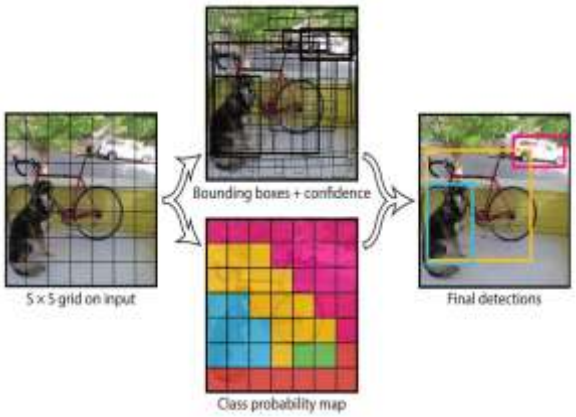


Fig1. Object detect system

3. WORKING OF YOLO ALGORITHM

There are a few different algorithms for object detection and they can be split into two groups:

1. Algorithms based on classification. They are implemented in two stages. First, they select regions of interest in an image. Second, they classify these regions using convolutional neural networks. This solution can be slow because we have to run predictions for every selected region. A widely known example of this type of algorithm is the Region-based convolutional neural network (RCNN) and its cousins Fast-RCNN, Faster-RCNN and the latest addition to the family: Mask-RCNN. Another example is Retina Net.
2. Algorithms based on regression – instead of selecting interesting parts of an image, they predict classes and bounding boxes for the whole image in one run of the algorithm. The two best known examples from this group are the YOLO (You Only Look Once) family algorithms and

SSD (Single Shot Multibox Detector). They are commonly used for real-time object detection as, in general, they trade a bit of accuracy for large improvements in speed.

To understand the YOLO algorithm, it is necessary to establish what is actually being predicted. Ultimately, we aim to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:

1. center of a bounding box (bxby)
2. width (bw)
3. height (bh)
4. value cis corresponding to a class of an object (such as: car, traffic lights, etc.).

BOUNDING BOX PREDICTIONS

YOLO algorithm is used for estimating the precise bounding boxes from the image. The image separates into S x S grids by estimating the bounding boxes for every grid and class possibilities. Both classification of image and localization of object techniques are used for every grid of the image and every grid are provided with a label. The algorithm then verifies every grid individually and spots the label having an object and also spots its bounding boxes. The labels of grid without any object are spotted as zero.

Fig 2: Example image with 3x3 grids

In the given example, an image is divided in 3 x 3 matrixes. Every grid is labeled and undergoes image classification and objects localization techniques. The label is examined as Y. Y having 8 values.

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

Fig 3: Elements of label Y

Pc – shows if an object is available in the grid or not. If available, pc=1 else 0. The bounding boxes of the objects (if present) are bx, by, bh, bw and c1, c2, c3 – are the classes. If the object is a car then the value of c1 and c3 will be 0 and c2 will be 1. In example image, the first grid has no proper object. So it is shown as:

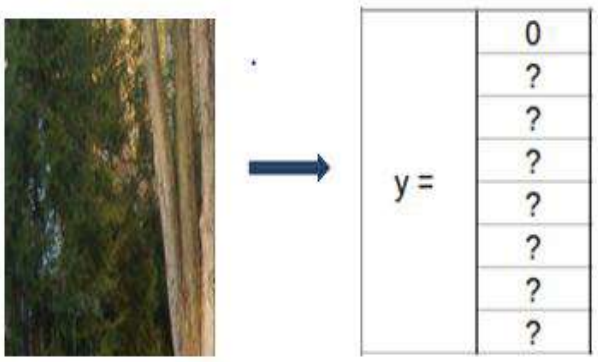
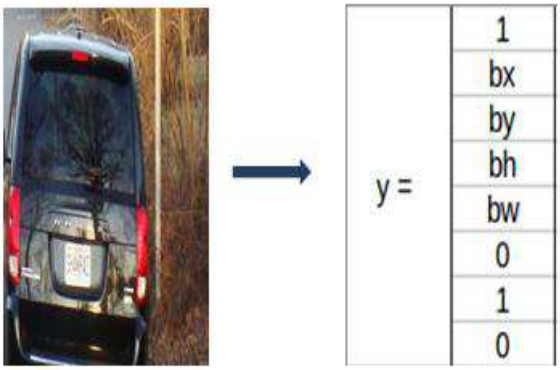


Fig 4: Bounding box and Class values of grid 1

In this grid, there is no proper object so the value of pc is 0. And remaining values doesn't matter because there is no object. So, it is represented by considering a grid having an object. Both 5th and 6th grid of the image has an object. Now the 6th grid, it is represented as.



the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Fig 5: Bounding box and Class values of grid 6.

In this table, 1 shows the existence of an object. And bx, by, bh, bw are the bounding boxes for the object present in the 6th grid. As the object is a car so the classes for the grid are (0,1,0). The matrix form is Y=3x3x8. Also the matrix will be slightly similar for the 5th grid with distinct bounding boxes by depending on the position of objects in the relative grid. If more than two grids have same object then object's center point is located and the grid having that point is used. For the precise identification of the object we can use two methods which are Intersection over Union and Non-Max Suppression. In IoU, it will use the actual and estimated bounding box value and computes the IoU of two boxes by the formulae, IoU = Intersection Area / Union Area.

It will be a good estimation if the value of IoU is more than or equal to our threshold value (0.5). The threshold value is a supposed value which can also be some greater value for increasing the accuracy of the object. The next method is Non-max suppression, in which high possibility boxes are used and the boxes having high IoU are suppressed. This is repeated until a box is selected and considered as the bounding box for that object.

Our detection network has 24 convolutional layers and 2 fully connected layers. Alternating 1×1 convolutional layers lowers the features space from preceding layers. We train in advance

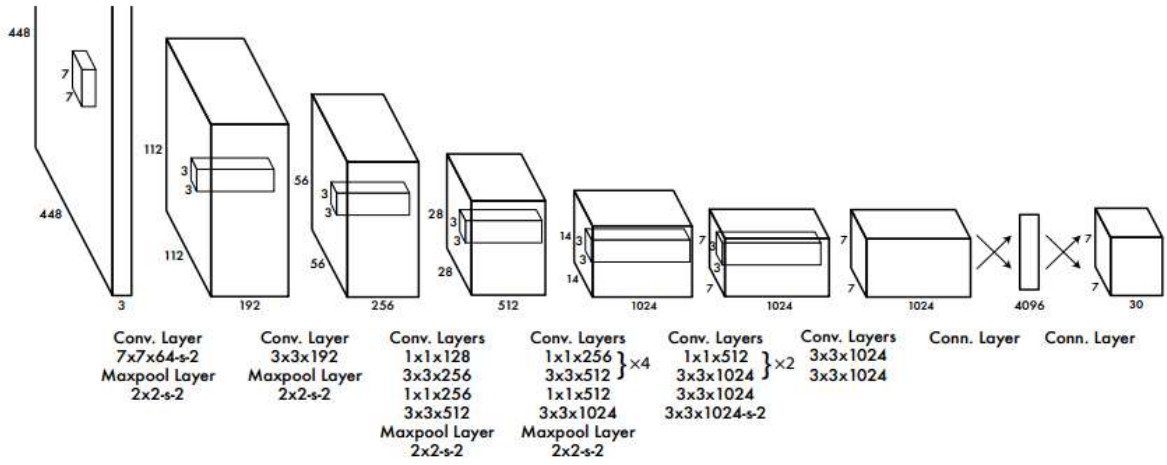


Fig 5: Architecture of YOLO

4. LIMITATIONS OF YOLO

YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds.
Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations. Our model also uses relatively coarse features for predicting bounding boxes since
our architecture has multiple down sampling layers from the input image. Finally, while we train on a loss function that approximates detection performance, our loss function treats errors the same in small bounding boxes versus large bounding boxes. A small error in a large box is generally benign but a small error in a small box has a much greater effect on IOU. Our main source of error is incorrect localizations.

5. CONCLUSION

In this paper we discuss about the YOLO algorithm overview and introductory part of yolo algorithm. Why we use yolo and how to work yolo algorithm. Yolo algorithm used for object detection in real time object. Yolo algorithm is best for the CNN and R-CNN.Comparing to other classifier algorithms this algorithm is much more efficient and fastest algorithm to use in real time.

REFERENCES

1. Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam “Real-Time Object Detection with Yolo” (IJEATVolume-8, Issue-3S, February 2019
2. Joseph Redmon, Ali Farhadi,“YOLO9000: Better, Faster, Stronger”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.
3. Jifeng Dai, Yi Li, Kaiming He, Jian Sun, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, published in: Advances in Neural Information Processing Systems 29 (NIPS 2016).
4. M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In Computer Vision– ECCV 2008, pages 2–15. Springer, 2008.
5. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In International Conference on Computer Vision (ICCV), 2009.
6. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
7. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.