Object Detection Using YOLO Algorithm

[1]Anugrah C Biju, [2]Amal K George, [3]Vignesh K H,

[1,2,3,] Department of Computer Science [PG], Kristu Jayanti College, Bengaluru.

[1]anugrahmartin599@gmail.com, [2]amalkomathgeorge@gmail.com, [3]vigneshkh09@gmail.com

Abstract

The aim of this study is to find a stable system that can detect the object in fraction of seconds from different sources like image, surveillances, bus, car etc. There are different types of algorithms to find the object with the help of frame. This study is to recognize the object classification and localization. The trained weights should have the maximum confident level possible when classifying objects from external events or detecting multiple objects from an image.

To detect the object in the existing system, we use to split the image into different class and focus on the specific region or the subject in the frame. Our proposal using the YOLO algorithm model detects and recognizes the objects, and the improved model will examine the entire image. The YOLO model splits the image into regions and maps the confidence probability using a neural network on the image.

Keywords: *Object detection, YOLO Algorithm, YOLO Versions, Neural network.*

1. Introduction

We take the help of eyes and ears to observe and see everything, it captures the information in the surrounding and sends it to our brain to decode. Well, it sounds so simple, right? We understand what we see and what the object we're looking at. There are lots of information to processes but the brain does it for us. This ability of brain led the researchers to think what if machine can also process like brain, with this task. Once the machine starts to recognize the object in its surroundings it can interact better with humans and another machine. The main aim of YOLO algorithm is to improve the machine to make better judgements and makes it friendly which makes it more human.

In pursuit of this we have big hurdle. The main question arises here is "how to make the machine identify an object?" This is what give rise to the domain of computer vision that what we call "Object detection". It is a field of image processing and computer vision that deals with the detecting instance of various object like bus, car, group of people etc [1][14].Object recognition is a vast topic which divides into many sub-domains like image annotation, face detection, activity recognition and more. It is used in various application like surveillance, Tracking, self-driving cars, robotics etc [2][3].We mainly focus on YOLO Algorithm as it gives better architectural performance to the object

detection. It is a regression-based algorithm, rather than selecting different parts of the image, in one run of the algorithm, it guesses the image's class and bounding box[15].

1.1. The various challenges faced by the object detection are:

- Different number of objects.
- Aspect ratio and multiple scales.
- Data limitation.
- Modeling.
- Real-time detection.

2. YOLO Algorithm

It's mainly associated with different process that blends with computer vision and Latino Institute for Development Education Responsibility (LIDER) to generate a technology with multidimensional representation of the object with its attributes [4]. First published in the seminar paper in 2015 by Joseph Redmon[5]. In this article the authors introduced the object detection concept, the one of the open-source implementations of the YOLO algorithm: darknet[16][17].

3. Architecture

The project "Object Detection System Using Machine Learning" detects objects efficiently using the YOLO algorithm, which is applied to image and video data. As the name implies, the input only passesthrough the network once, and the result of a detected object with Bounding Boxes and Labels is obtained [6][18][20](Fig 1). YOLO is among the fastest detection techniques, with excellent accuracy and good real-time performance. With YOLO-V1, YOLO-V2, and YOLO-V3, ever since it was proposed, it has been enhanced. There are two completely linked layers and 24 convolutional layers in YOLOv1 [7][8].

3.1. The three strategies used by the YOLO algorithm are as follows:

- Bounding box regression
- Residual blocks
- Intersection Over Union (IOU)

Figure 1. Architecture of YOLO

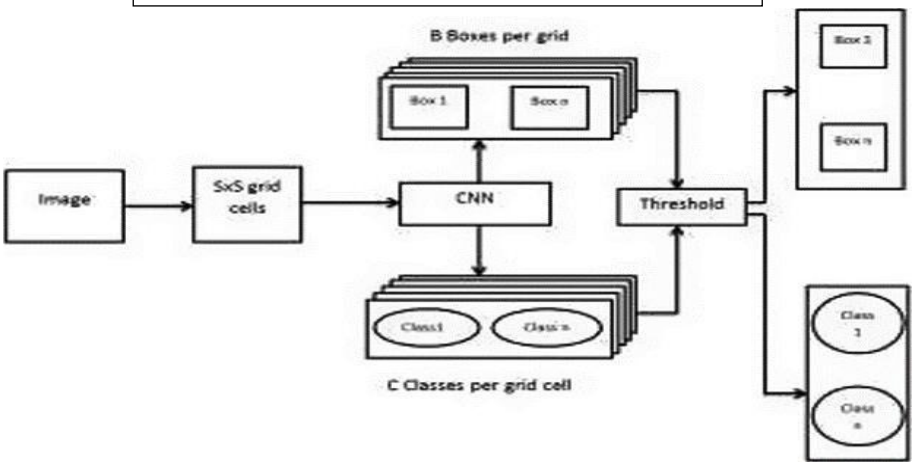


Figure 2. CNN Network Design

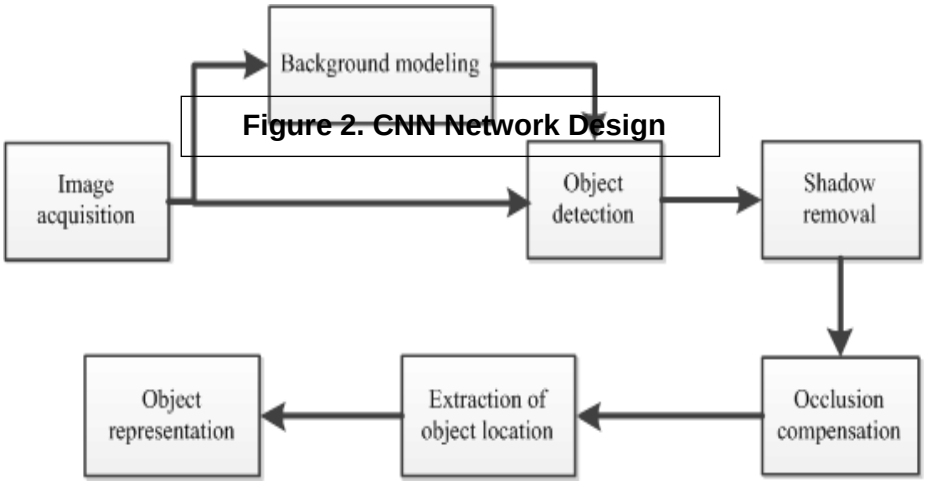


Figure 1 represents the architecture of YOLO algorithm and in Figure 2 the images are divided into SXS grid cells before sending it to the Convolutional NeuralNetwork (CNN). As a result of the Convolutional Neural Network, bounding boxes per grid are generated around all detected objects in the image [9]. The Convolutional Neural Network, on the other side, classifies the Classes, to which the objects belong, yielding C Classes per grid. The Object Detection is then given a threshold [10][19]. The lower the Threshold value, The more bounding boxes that show in the result, the clumsier it becomes[11].

4. Neural Networks:

Because the use of computing systems with interconnected nodes, corresponding to neurons, Neural Networks are closely linked to the organization of the cerebral cortex. Algorithms are used to group and classify raw data in order to find unseen designs and correlations, resulting in continuous growth and innovation [12][20].

- RNNs (Recurrent Neural Network)
- CNNs (Convolutional Neural Network)
- FNNs (Feed Forward Neural Network)
- AENNs (Autoencoder Neural Network)

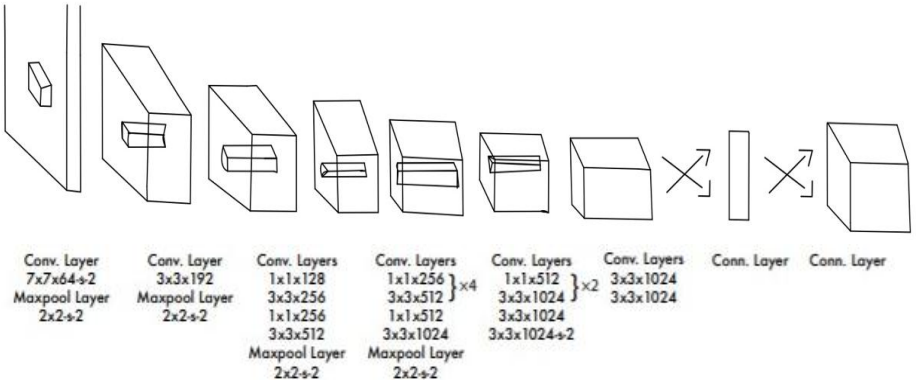


Figure 3. YOLO Convolutional Network Diagram

source: You Only Look Once: Unified, Real-Time Object detection

In Figure 3, it represents the YOLO detection network, which is made up of 24 convolutional layers and 2 fully connected layers. The features space from preceding layers is reduced by alternating 1 × 1 convolutional layers. On the ImageNet classification challenge, the persisted convolutional layers were used at half resolution (224 × 224 input picture) and subsequently at double resolution for detection[19].

How it works?

- Image classification:- : Predict the type or class of an object in an image. Input: An image with a single object, such as a photograph. Output: A class label denoting the object type.
- Object localization:- Locate the presence of objects in an image and indicate their location with a bounding box. Input: An image with one or more objects, such as a photograph. Output: One or more bounding boxes (Eg:Defined by a point, width, and height)
- Object detection:- Locate the presence of objects with a bounding box and types or classes of the located objects in an image. Input: An image with one or more objects, such as a photograph. Output: One or more bounding boxes (Eg: defined by a point, width, and height), and a class label for each bounding box(Fig 4).

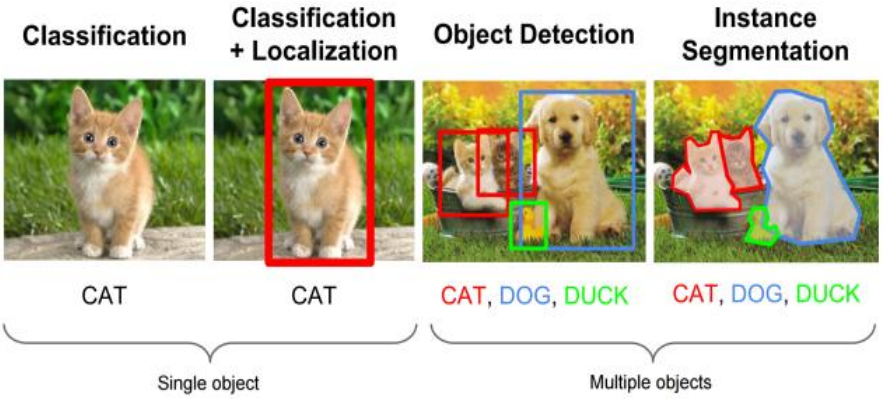


Figure 4. Instance segmentation process

Source: appsilon.com/object-detection-yolo-algorithm

5. Understanding of YOLO Algorithm:

To understand this algorithm, we must know what we are predicting exactly. Ultimately, we are aiming to predict the object, class and bounding box for better description [13].

5.1. Each bounding box can describe by this descriptor:

- bxby center of the bounding box.
- bw width.
- bh height.
- cis value corresponding to a class of an object.

5.2. The object in the bounding box:

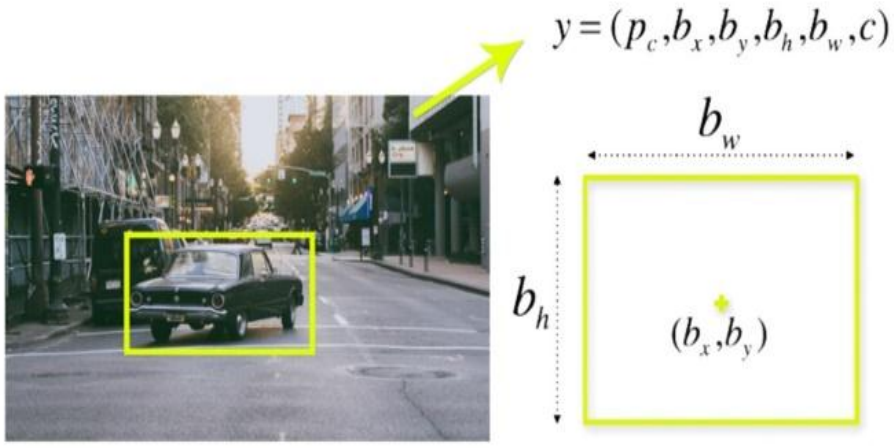


Figure 5. Bounding box probability calculation

Source: appsilon.com/object-detection-yolo-algorithm

The probability that an object exists within the bounding box.When we work with YOLO algorithm, we are not searching for the interesting region in our picture that could contain an object(Fig 5).
During the one pass of forwards propagation, YOLO determines the probability that the cell contains a certain class. The equation for the same is :

$$\text{score}_{c,i} = Pc \times Ci$$

That is the probability that there is an object Pc times object times the probability that the object is a certain class Ci .

We split our picture into 19x19 grids, each cell is representing the prediction of 5 bounding box (if there are more than one object in the single cell). Finally, we come across a huge number of 1805 bounding boxes [21](Fig 6).

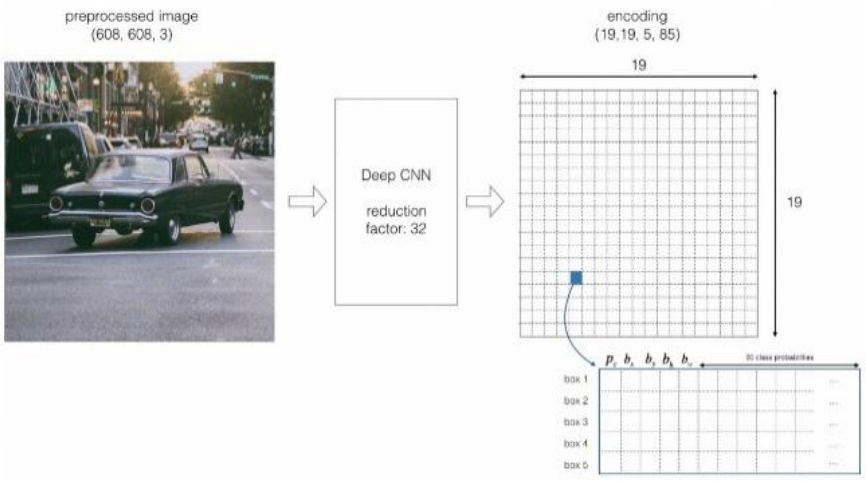


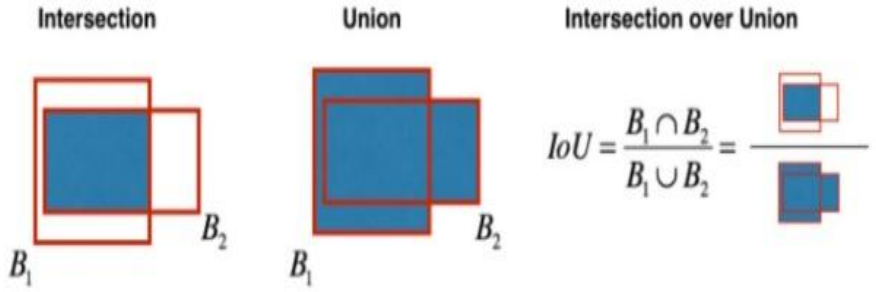
Figure 6. Image size reduction

Source: appsilon.com/object-detection-yolo-algorithm

Figure 7. Representation of Non-max suppression

Source: appsilon.com/object-detection-yolo-algorithm

Suppression that isn't maximal in the process, it removes the boxes with low probability and the bounding box with the maximum shared [13](Fig 7).
Also, the most important parameter of the Algorithm, its Loss function is shown below(Fig8). YOLO simultaneously learns about all the four parameters it predicts.



$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

Figure 8. Loss function for YOLO

Source: You Only Look Once: Unified, Real-Time Object detection

6. YOLO Algorithm versions:

6.1. YOLOv1

Fast YOLO processed images at 155 frames per second in real time, twice the mAP (mean average precision) of comparable real-time detectors(Fig 9).

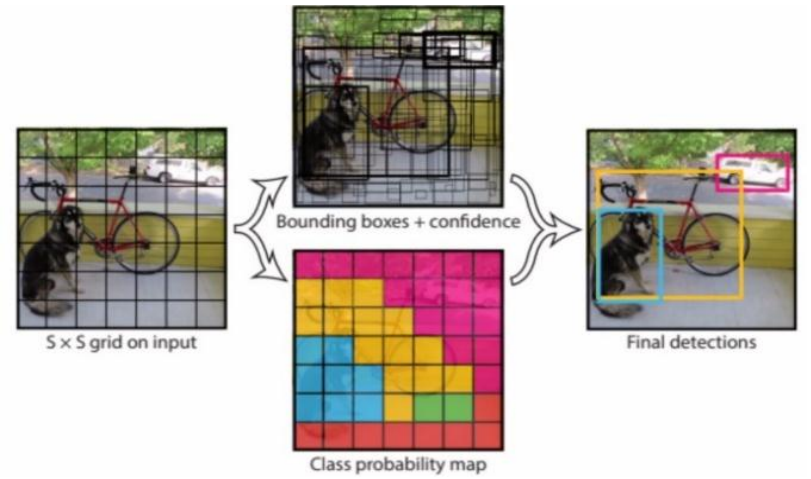


Figure 9. YOLOv1

Source:YOLOV3: An incremental improvement

6.2. YOLOv2

YOLOv2 is also known as YOLO9000 as the name indicate the model's ability to predict 9000 different object categories and run in real time(Fig 10).
Article - "YOLO9000; Better, Faster, Stronger"

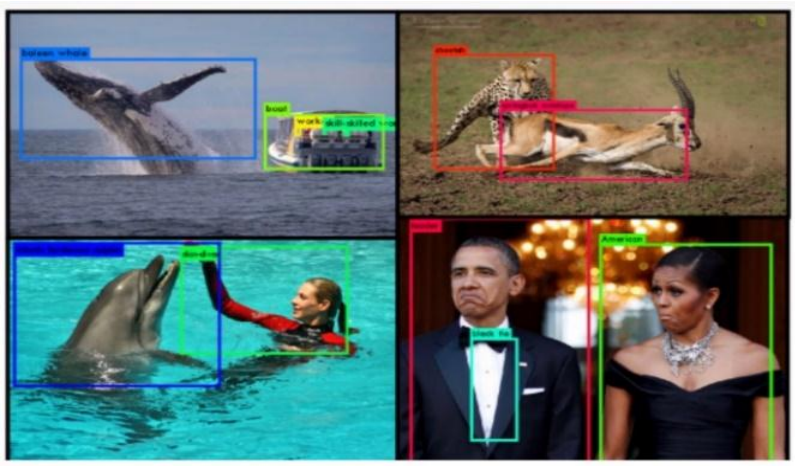


Figure 10. YOLOv2

Source:YOLOV3: An incremental improvement

6.3. YOLOv3

Article – "YOLOv3: An incremental"

It is based on the Darknet53 architecture(Fig 11); it is open source neural network framework which is essential for real-time prediction.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 11. YOLOv3: Darknet53 architecture

Source:YOLOV3: An incremental improvement

6.4. YOLOv4

Weighted Residual Connections, Cross-Stage-Partial Connections, and cross mini-batch normalization were among the unique innovations included in the SPDarknet53 design. Article - "YOLOv4: Optimal and Accuracy of Object Detection"

6.5. YOLOv5

Here we use PyTorch deep learning framework. Fig 12 depicts the YOLOv5 subversions.

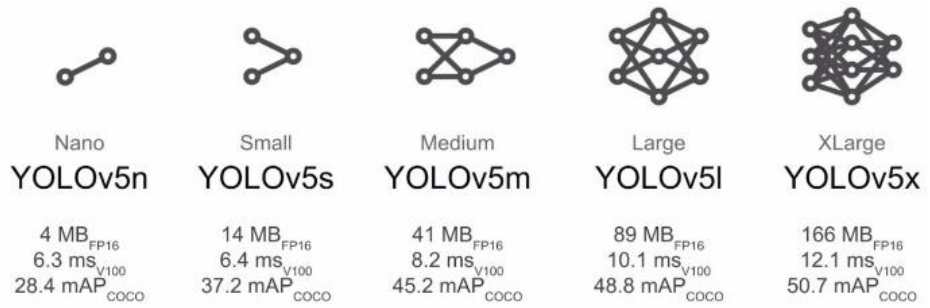


Figure 12. YOLOv5 subversions

Source:YOLOV3: An incremental improvement

7. Comparison between the versions:

7.1. YOLOv5 performance:

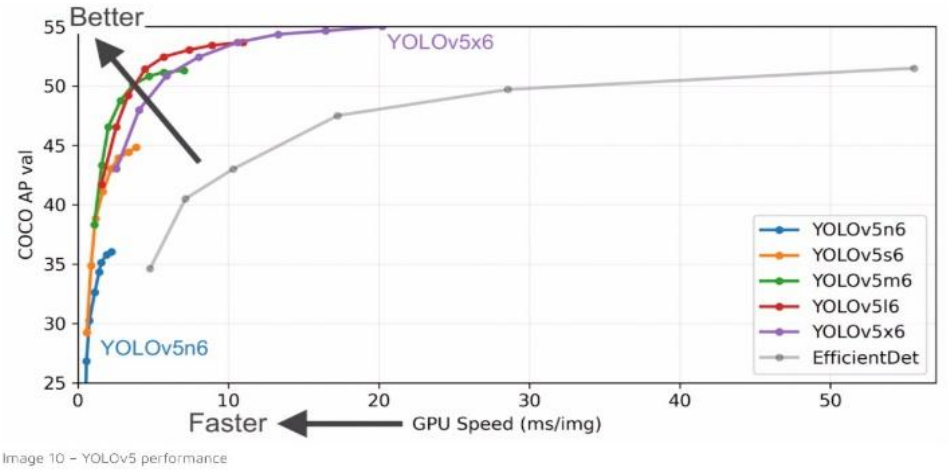


Figure 13. YOLOv5 performance

Source:YOLOV3: An incremental improvement

7.2. YOLOR performance:

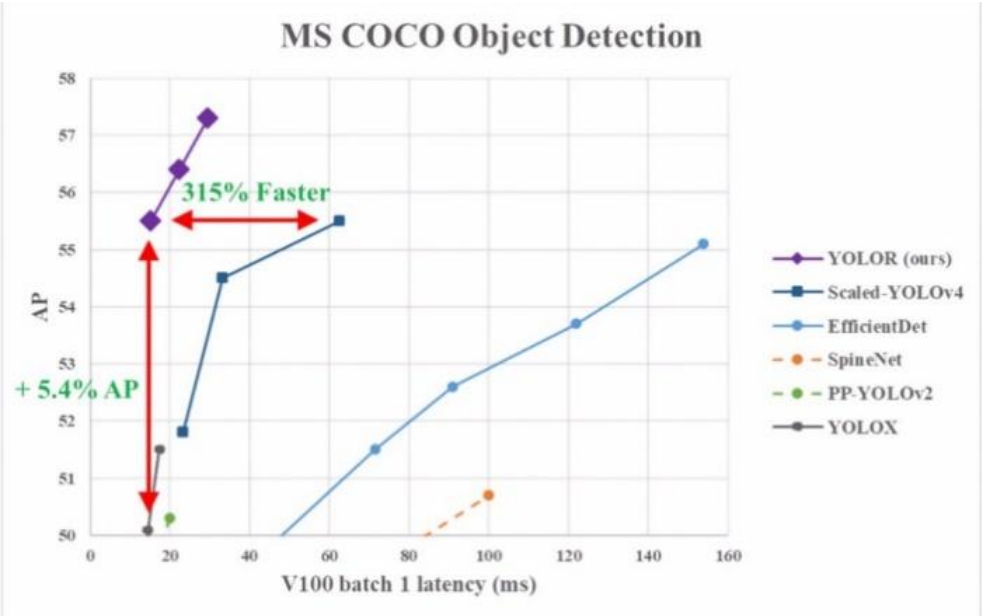


Figure 14. YOLOR

Source:YOLOV3: An incremental improvement

7.3. Other Model:

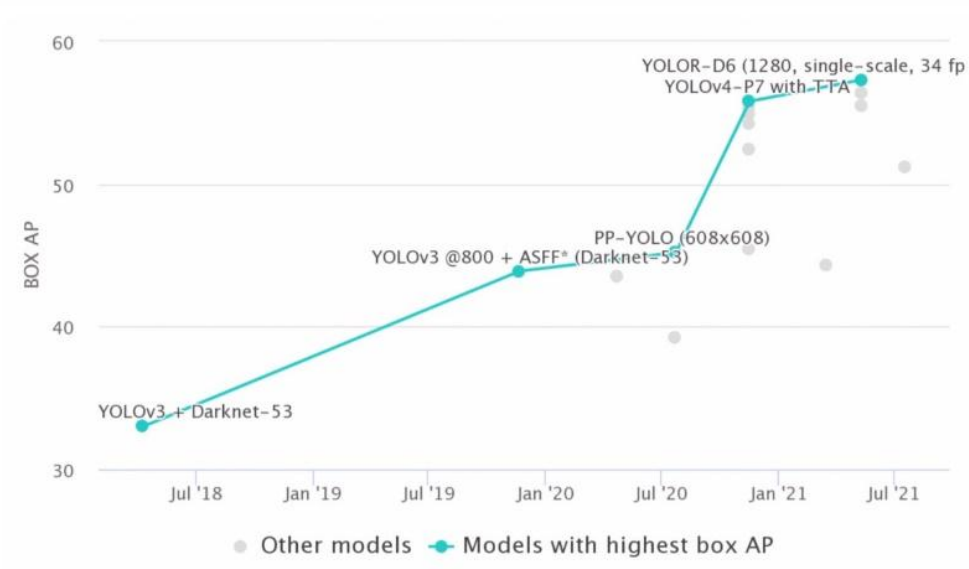


Figure 15. Comparison between YOLO versions

source: You Only Look Once: Unified, Real-Time Object detection

8. Conclusion:

In this paper we have conducted a study on YOLO algorithm and how it is used in object detection. When compared to other object detection techniques such as Fast R-CNN and Retina-Net, this technique provides better detection results. The approach has been generalized from natural photos to many domains, and it outperforms several strategies. The algorithm is easy to develop, and it can be trained on a whole image. The classifier is limited to specific region by region proposal strategies. When predicting boundaries, YOLO uses the entire image. It also means that there will be fewer background false positives. Comparing to other classifier algorithms this algorithm is much more efficient and fastest algorithm to use in real time.

References:

[1] Redmon and A. Farhadi, ‘‘YOLOV3: An incremental improvement,’’ (**2018**), *arXiv:1804.02767* [Online].

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ‘‘Densely connected convolutional networks,’’ in *Proc. CVPR*, (**2017)**, vol. 1, no. 2, pp. 4700 4708.
[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, ‘‘Object detection with discriminatively trained part-based models,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, , Sep. (**2009**), pp. 1627 1645
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classi ca-tion with deep convolutional neural networks,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, (**2012)**, pp. 1097 1105.

[5] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (**2017)**, pp. 7263-7271.
[6] Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once:Unified, Real-Time Object Detection”, The IEEE Conference on ComputerVision and Pattern Recognition (CVPR), (**2016)**, pp. 779-788.
[7] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large- Scale Image Recognition”, published in Computer Vision and Pattern Recognition (cs.CV).
[8] Matthew B. Blaschko Christoph H. Lampert, “Learning to Localize Objects with Structured Output Regression”, Published in Computer Vision – ECCV (**2008)** pp 2-15.
[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, “SSD: Single Shot MultiBox Detector”, Published in Computer Vision – ECCV (**2016)** pp 21-37.
[10] Lichao Huang, Yi Yang, Yafeng Deng, Yinan Yu DenseBox, “Unifying Landmark Localization with End to End Object Detection”, Published in Computer Vision and Pattern Recognition (cs.CV).
[11] Dumitru Erhan, Christian Szegedy, Alexander Toshev, “Scalable Object Detection using Deep Neural Networks”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (**2014)**, pp. 2147-2154.
[12] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Published in Advances in Neural Information Processing Systems 28 (NIPS **2015**).
[13] 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**).
[14] Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. (**1962)**;160(1):106-54.
[15] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. (**1998)**;86(11):2278-324.
[16] Ranzato MA, Huang FJ, Boureau YL, LeCun Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: 2007 IEEE conference on computer vision and pattern recognition. IEEE; (**2007**), p. 1-8.
[17] Nickolls J, Buck I, Garland M, Skadron K. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? Queue. (**2008**);6(2):40-53.
[18] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. (**2012**);25:1097-105.
[19] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition.
[20] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; (**2014)**, p. 580-7.
[21] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; (**2015)**, p. 1440-8.