IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 5, October, 2020, pp. 691-697
Journal homepage: http://iieta.org/journals/isi

Object Detection Using Stacked YOLOv3

Sai Shilpa Padmanabula, Ramya Chowdary Puvvada, Venkatramaphanikumar Sistla*, Venkata Krishna Kishore Kolli

Department of CSE, VFSTR Deemed to be University, Vadlamudi, Guntur 522213, Andhra Pradesh, India

Corresponding Author Email: drsvpk_cse@vignan.ac.in

https://doi.org/10.18280/isi.250517

Received: 27 May 2020
Accepted: 3 August 2020

Keywords:
object detection, YOLOv3, deep neural network, Non-maxima Suppression, class probabilities, unified architecture, transfer learning

ABSTRACT

Object detection is a stimulating task in the applications of computer vision. It is gaining a lot of attention in many real-time applications such as detection of number plates of suspect cars, identifying trespassers under surveillance areas, detecting unmasked faces in security gates during the COVID-19 period, etc. Region-based Convolution Neural Networks(R-CNN), You only Look once (YOLO) based CNNs, etc., comes under Deep Learning approaches. In this proposed work, an improved stacked Yolov3 model is designed for the detection of objects by bounding boxes. Hyperparameters are tuned to get optimum performance. The proposed model evaluated using the COCO dataset, and the performance is better than other existing object detection models. Anchor boxes are used for overlapping objects. After removing all the predicted bounding boxes that have a low detection probability, bounding boxes with the highest detection probability are selected and eliminated all the bounding boxes whose Intersection Over Union value is higher than 0.4. Non-Maximal Suppression (NMS) is used to only keep the best bounding box. In this experimentation, we have tried with various range of values, but finally got better result at threshold 0.5.

1. INTRODUCTION

Object detection is one of the foremost demanding problems in computer vision. It is one of the areas of computer vision that is maturing very rapidly. When we check out images or videos, we will quickly locate and identify the objects of our interest within moments. Passing on of this human intelligence to computers is nothing but object detection - locating the object and identifying it. The visual system of a person is meticulous and expeditious, and it permits us to perform difficult tasks like driving. Similarly, to detect the objects automatically, systematic approaches shall be followed. Object detection is employed to detect and defines objects like humans, buildings, vehicles, animals, etc., from images and videos. Objects are recognized not only from the pictures but also from the videos [1]. There is nothing much contrast between object recognition and object detection. Both are similar techniques for recognizing objects, but they stretch in their execution. Object recognition is habitually a superset of object detection. It is a pivotal essential upshot of both machine learning and deep learning algorithms. Object Recognition technique can often be referred to as Image Recognition. Object recognition consists of identifying, recognizing, and locating objects within a picture. Image classification technique plays a vital role in digital image analysis. In image classification, class labels will be allocated to the images. It is the process of loading an input (like a picture) and yielding an output as a class label like ("cat"), or it will display the probability that the input image belongs to a class (there is a 90% probability that the input is a "cat"). So, it will assign pixels in the image to categories or classes of interest. It can often be considered as an activity of mapping numbers to symbols.

$$f(a): a \rightarrow C; a \in R^{m,} C = \{ \boldsymbol{c_1}, \boldsymbol{c_2}, \ldots \boldsymbol{c_n} \}$$

where, 'm' indicates the number of bands; 'n' indicates the number of classes. From the above equation, **f (a)** is a function assigning a pixel point '**a**' to a single class into the set of classes C. The systemization of a group of information is required to cluster into different categories or classes. The connection between the data and the classes into which they are classified must be understood. The machine trained to grasp the noesis of the classes. Training plays a principal role in obtaining better results for classification. The main intent of localization is to prognosticate the object in a picture and as its boundaries. The pinpoints to seek out the orientation of the single object inside the image. An object localization algorithm will squeeze the output which contains coordinates of the location of an object with reference to the image. In computer vision, the foremost popular way to localize an object in a picture is to represent its location with the assistance of bounding boxes. An amalgamation of both image classification and object localization is nothing yet object detection. Object detection is tougher and fuses these two tasks and draws a bounding box all-over the object of interest in the image and allocates a class label. Sample classification, localization, object detection and instance segmentation tasks are illustrated in Figure 1.

Simply, it takes input as an image and performs image classification and object localization. Then, the output will be like all the objects present in that image with a bounding box around that detected object by classifying it with a class label. An overview of all these problems depicted in the below picture. The extension to this task is Object Segmentation, which is also called "Object Instance Segmentation." It is the sole pace augmentation to object detection. In this, the

identified objects are highlighting certain pixels of the object rather than a coarse bounding box. Basic process involved in the objection segmentation is presented in Figure 2.

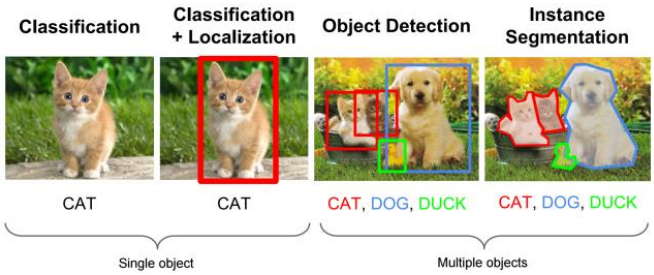

Figure 1. The example of image classification, localization and object detection

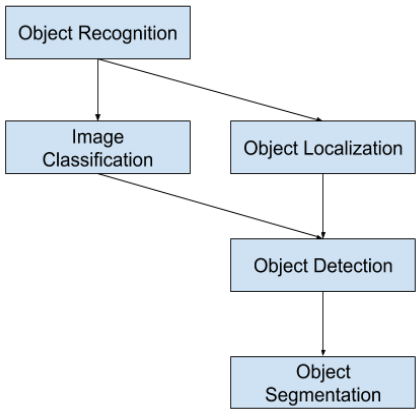

Figure 2. Overview of object recognition tasks

Together all these problems are mentioned as object recognition. Object detection has found its application in varied domains such as video surveillance, image retrieval systems, autonomous driving vehicles, and many more. Profuse methods are used for object detection, which comes under either "Machine Learning" based stratagems or "Deep Learning" based approaches. In machine learning applications [2], features are extracted using one of the methods and then classified using a classifier such as "Support Vector Machines" (SVM). Some Machine Learning approaches are Viola-Jones object detection framework based on Haar features, Scale-invariant feature transform (SIFT) [2], and Histogram of oriented gradients (HOG) features. On the contrary, deep learning techniques [3, 4] are adept at doing end-to-end object detection without notably defining features, and are typically hinge on Convolutional Neural Networks (CNN) [5, 6], Region based-CNN [7], You Only Look Once (YOLO [8], YOLOV2 [9], YOLOV3 [10]), etc.

A sliding window was used in conventional computer vision techniques to seek out objects at various locations and scales. Because this was an expensive action, the aspect ratio of the object was usually assumed to be fixed. R-CNN and Fast R-CNN [3] methods use selective search to decrease the number of Bounding Boxes in the test image. The over-feat method scans the image at various scales by using sliding window-like mechanisms. The RPN also used the design features extracted for identified objects for proposing potential bounding boxes, and thus, it saves a lot of computation. In this work, an improved YOLOv3 (You Only Look Once Version 3) model is proposed. YOLOv3 model [10] consists of a single neural network which is trained end-to-end. It takes photographs as input and speculates the bounding boxes and class labels for one and all bounding boxes directly [11]. It forwards the whole image only once through a deep learning network, whereas multiple scans required for other algorithms. This technique limits with low predictive accuracy, although it operates at 45 frames per second (FPS) up to 155 frames per second (FPS). This model computational complexity is low when compared to existing algorithms like R-CNN and Single Shot MultiBox (SSD) [12].

The main objective of the proposed work is to detect the objects in a still picture and from the video using OpenCV, Python, using the improved YOLOv3 algorithm in this work. Amidst the cutting-edge techniques for knowledgeable learning object location (Faster R-CNN, SSD, YOLO, etc.), Yolov3 pop up because of its implausible harmony among speed and exactness. It will detect objects quickly with high exactitude and has been successfully appealed in many detection problems [13-15].

For every cell in the network, some bounding boxes forecast and fabricated at the same time with class probabilities/scores for anticipating the objects related to that lattice cell. Every score returns how sure the model which contains a class of item. The main shortcoming of the Yolo network is that usually, it can't accomplish high precision when working with the tiny-size object detection in high-resolution images. Another fruitful region proposal extraction method for the Yolo network is to inaugurate a whole detection structure named ACF-PR-YOLO [16].

An accepted application of object detection is face detection, that's utilized in almost all mobile cameras. A lot of generalized (multi-class) application is being used in autonomous driving wherever a range of objects have to be compelled to be detected. Object detection plays an indispensable title role in surveillance systems. It may also avail oneself of tracking the objects and consequently can be employed in robotics and medical applications.

2. RELATED WORK

Object detection is breaking into the big choice of industries, with use cases fluctuating from personal security to productivity within the workplace. Face detection is one of the dominant applications of object detection. During this section, some current works are discussed about the various methods used for object detection. Nowadays, object detection employed in diverse applications. A survey on the deep learning for generic object detection [4] outlines the recent achievements within the object detection field using deep learning techniques. Object detection tries to locate the object instances from the given pre-established categories in natural images. About 300 research contributions presented that cover many aspects of the generic object detection framework like object proposal generation, object feature representation, training strategies, context modeling, popular datasets, and evaluation metrics. CNN [5] used to build mobile robots, which perform certain tasks like surveillance, navigation, and explosive ordnance disposal (EOD). Using vision systems within the robots makes aware that what sort of environment it had been and what sort of objects are there in the environment. The results have shown that SSD (single shot multi-box detector) has fast detection capability in real-time applications, and Faster R-CNN can detect the objects with high accuracy. In recent years, object detection has attracted

much attention towards research because it's a close relationship with image understanding and video analysis. R-CNN [6] proposed can handle some sub-problems like occlusion, low resolution, and clutter with various modifications on. Recently, deep neural networks (DNN) [3] have manifested an impressive performance on the image classification tasks. A regression problem is applied to get bounding boxes. A multi-scale inference procedure proposed to get the object detections at low cost with high resolutions. The performance is evaluated using the PASCAL-VOC dataset and yields better results.

As deep neural networks (DNN) are harder to train, so deep residual learning frameworks were proposed by He et al. [17] that reduce the hassle within the training of networks. The evidence provided has shown that the network depth is far important. While training the model with more network depth, then the deep networks may be able to converge such a degradation problem. Accuracy gets saturated at an equivalent time, and it degrades rapidly. There is an existed constructed solution to present the degradation problem, which says that it should not produce higher training error. The authors experiments shown that they are unable to seek out an honest or better solution than then existed constructed solution on ImageNet and COCO dataset. CNNs have accomplished better results on visual recognition tasks [2] for the past two years. When compared to HOG and SIFT, we barely acknowledge about the complexion of features learned by large CNNs. Several experiments on CNN feature learning carried out on two datasets, one for detection and another one for classification to perform pre-training and investigated the fine-tuning behavior. By performing, pre-training for an extended period, the author has exhibited that it prompts better performance. Inside-Outside Net (ION) [18] proposed to obtain accurate visual recognitions contextual and multi-scale representations, which is well known as an object detector. This object detector exploits information from both the interior and exterior regions of interest. Spatial recurrent neural networks to integrate the contextual information which is present outside of the region. In the inside of the region, a skip pooling method used to extract the knowledge at multiple scales and pitches of abstraction. Experiments are carried out on PASCAL VOC 2012 dataset and results are improved from 73.9% to 76.4% map. Similarly, the results on the COCO dataset are from 19% to 33.1% map.

Recently, various researchers have an interest in Autonomous Underwater Vehicle (AUV), and there are innumerable projects on the design of AUV [12]. CNN, with three optimization techniques, is proposed for fish detection. Those three optimizations are training process speedup, data augmentation, and network simplification. The data augmentation is employed to supply furthermore samples of data, and it is also used to keep pace with the training operations to be more systematic by furnishing enough datasets. To unravel the overfitting problem, Drop Out algorithm was selected. To refurbish the parameters amidst the network, the author has put in the loss function. At last, the author has illustrated that the suggested model is believable to extend underwater objects. For all smart vehicles, the predominant mission is to detect the environment perception. All these vehicles require subsequent steps to securely detect the roads, other vehicles, cyclists, and pedestrians. Every deep learning model which supports the object detection can't predict the certainty in their predictions. During this paper, the author presented some approaches which successively estimate the uncertainty in a one-stage object detector [19]. The automotive pedestrian dataset is employed on an outsized scale to reinforce the performance of the detection of base earn approach. Yolov3 is employed in tensor flow such that it supports training from scratch. Sensible uncertainty elimination means that the predictions should have higher uncertainty than the precise ones.

In deep learning approaches, the quickest object detection algorithm is YOLO (You Only Look Once) [8]. The entire neural network is pipeline such that only in one evaluation it detects the objects and outperforms well in comparison to previously used detection algorithms like DPM, SSD, R-CNN. This model pre-processes images at 45 FPS (Frames Per Second). A further extension to YOLO is YOLO9000 and YOLOv2 [9]. These models demonstrated that they outperform well in real-time detection systems on PASCAL VOC 2007 dataset with a mean average precision of 78.6 at 67 FPS. Furthermore, an extension to YOLOv2 is YOLOv3 [10]. YOLOv3 is an accurate and fast detector. This algorithm generates weights model with all images and assigns a unique class name to uniquely detected objects in that image and then generate a model. This algorithm converts each image into layers and then for each layer, extract features and add weights to the model. Whenever a new image is applied to the pre-trained weight model, it gets the best accuracy matching image label. An image with 320 x 320 size, YOLOv3 runs in 22 ms at 28.2 maps in which its accuracy is like SSD but three times faster. One of the limitations of the Yolo is the lack of high precision on small size object detection on high-resolution images. So, to overcome this, a region proposal extraction method ACF-PR-YOLO proposed by Liu et al. [16]. ACF extracts the objects from the images and merges the bounding boxes into the region proposals as an input to the YOLO net. This method is evaluated on public TDBC (Tsinghua-Daimler Cyclist Benchmark) and outperforms YOLO by 13.69% average precision (map) and SSD by 25.27% precision.

Recently vehicle detection applications through aerial images have attracted many researchers because these play a crucial role these days [14]. The three public aerial image datasets are trained through the YOLO algorithm, such that it produces a single aerial dataset. The projected model produces good test results, particularly for rotating objects, small objects, and dense objects. From aerial images, it is so tough to spot the features of the car [7]. So, Faster R-CNN and YOLOv3 are used as the fastest detection algorithms. Performance evaluation on two models is carried out on the Stanford dataset and the PSU dataset and observed the best performance from two models. A real-time detection by using the YOLOv3 algorithm with deep learning techniques is proposed by Vidyavani et al. [15]. This YOLOv3 algorithm performed better when tested on the COCO dataset, multi-label classification for the detected objects in the images. YOLOv2 used to detect the objects from the images in Sang et al.'s study [13]. To cluster the bounding boxes from the training dataset and to select six anchor boxes of varied sizes, K-means ++ clustering is used. A multilayer feature fusion strategy is used to boost feature extraction potentiality. For training the Beijing institute of technology (BIT) - vehicle validation dataset is utilized, and for testing, Compcars dataset was used. Human movements detected [11] from a video is captured by a surveillance camera using YOLO model. Liris human activities dataset is used for evaluation of the model.

3. PROPOSED METHOD

Yolov3 is the fastest algorithm for real-time object detection in comparison to the R-CNN family of algorithms. The speed and accuracy depend on the resolutions of the image dataset.

In this work, we have proposed stacked YOLOv3 architecture with the inclusion stacking of layers. YOLOv3 uses Darknet-53 architecture, which has 53 convolutional layers trained on ImageNet dataset. The proposed model is designed to spot even tiny objects from the image. The proposed model able to recognize 80 different objects in a single image. For the task of detection, 53 additional layers are stacked onto it, giving us a 106 layer fully convolutional layers. The proposed architecture is given in Figure 3.

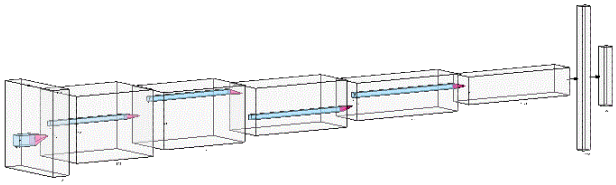

Figure 3. Architecture of stacked YOLOv3

The newer design boasts of residual skip connections and up-sampling. The foremost salient feature is that it makes detections at three completely different scales. The detection is completed by applying S x S detection kernel on feature maps of three completely different intervals within this network. The shape of the detection kernel is **S xS x (B x (5 + C)).** Here,

- B -> It predicts number of bounding boxes in a cell on the feature map
- 5 -> this is for the four bounding box attributes and one for the object confidence
- C-> It is used to predict the number of classes
- YOLOv3 trained on COCO dataset, B=3, and C=80.

So, the kernel size is 1 x 1 x 255. The first detection process is constructed by the 82nd layer. For the first 81 layers in the network, the image is down-sampled with a stride of 32. So, for an image of 416 x 416 size, the resultant feature map would be of size 13 x 13. One detection is formed here using the 1 x 1 detection kernel, giving us a detection feature map of 13 x 13 x 225. Then, the feature map from layer 79 is subjected to up samples by 2x2 to dimensions of 26 x 26. This feature map is then depth combined with the feature map from layer 61. Then, the second detection is created at the 94th layer, yielding a detection feature map of 26 x 26 x 225. Finally, 106th layer yields a feature map of size 52 x 52 x 225. The sigmoid activation function $1/(1+e^{-x})$ is used to calculate the scores. We have several activation functions like soft-max, Re-Lu, Tan hyperbolic, etc., used at different layers.

Sigmoid: $1/(1+e^{-x})$ (1)

Softmax: $e^{x}/ (\text{sum } (e^{x}))$ (2)

Re-Lu: $y= \max (0, \infty)$ (3)

Tanh: $[2/ (1+e^{-2x})]-1$ (4)

YOLO algorithm divides an image into an associate S x S grid system. Every grid on the input image is liable for the detection of an associate object. Currently, the grid cell forecasts the number of bounding boxes per object. Every bounding box contains 5 number of elements (x, y, w, h, confidence score). Where 'x' and 'y' are the coordinates of the item inside the input image, 'w' and 'h' are the width and height of the object respectively. The confidence score is the probability that the box associates with an object and the way accurate is that the bounding box.

$$\text{Confidence} = \text{probability (object)} * \text{IoU} \quad (5)$$

IoU is used to measure position accuracy. For loss function in YOLOv3, we replace the mean squared error (used in YOLOv2) with cross-entropy function. The cross-entropy loss function is as follows:

$$\sum_{c=1}^{m} \int_{\mathbf{x} \in \mathbf{c}} \mathbf{log}\ (\mathbf{P}(\mathbf{x} \in \mathbf{c})) \quad (6)$$

where, m is the number of classes and C indicates the class index, X is the observation, and log (P(X ∈ C)) is the natural logarithm, that predicts the probability of the observation X belongs to class C. After detecting the pictures using improved YOLOv3, we obtain the bounding boxes of the detected object. Then, the post-processing operations need to perform because several bounding boxes have appeared for a single object. To unravel this problem, use bounding box mapping and Non-Maximum Suppression (NMS). NMS removes all the overlapping bounding boxes and returns the right bounding box. The prediction of bounding boxes is as like in Yolov2. So, the network predicts Four coordinates for each bounding box, and they are t_w, t_h, t_x, t_y. Illustration of bounding box is presented in Figure 4.

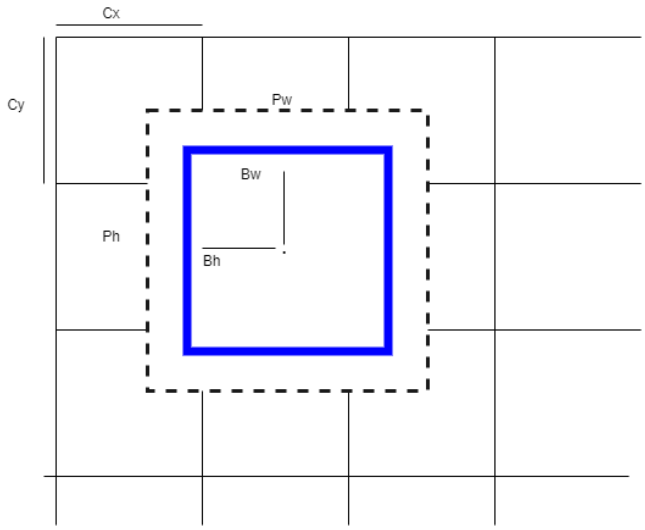

Figure 4. Prior box (Black dotted), Predicted box (Blue), Bounding box prediction

To calculate bounding box coordinates [9], use the following formulas:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w\, e^{t_w} \\ b_h &= p_h\, e^{t_h} \end{aligned} \quad (7)$$

The computation of IoU helps to calculate the mean average precision (MAP) [20]. IoU helps us to determine whether the predicted box is a false positive, true positive, or false negative. Here there is no true negative because it assumes that the bounding box may have something inside it, which in turn referred to as the bounding box is not empty. We predefine a threshold value to IoU to 0.5, which is commonly used.

- If IoU> 0.5, then we can say that it is true positive.
- If IoU< 0.5, then it is a false positive.
- If IoU> 0.5, but if any object is misclassified, then it is a false negative.

To eliminate duplicates of the similar object, YOLO practices non-maximal suppression. If we have IoU ≥ threshold between any of the predictions in the image, non-maximal suppression deletes the prediction with the lowest confidence score. YOLOv3 assigns one bounding box anchor for each ground truth object. Use of feature pyramid networks, in YOLOv3 is used to predict the boxes at 3 different scales [21].

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

One of the overriding goals of computer vision is to know the visible scenes. Scene understanding involves varied tasks such as recognizing what type of objects are present, localizing the objects in 3D and 2D, deciding the objects and scenes attributes, characterizing the relationships between objects, and providing a linguistics description of the scene. COCO (Common Objects in Context) dataset is used to evaluate the performance of the proposed model [22, 23]. COCO is a massive scale object detection, captioning, person key point's detection, and segmentation dataset. COCO has several features:

- COCO explicates 91 classes, but data uses only 80 classes.
- COCO has 91 object categories therein 82 of them have quite 5,000 labelled instances.
- The dataset has a total of 2,500,000 labelled instances respective to the 328,000 images.
- This dataset gathers the complex images of everyday scenes which contain common objects in their natural context.
- In contradistinction to the ImageNet dataset, COCO has fewer categories but more instances per each category. This will be employed in learning intimately about the object models which are capable of 2D localization.
- This dataset is much larger in the number of instances per category than the SUN and PASCAL VOC datasets.
- To get the precise localization of objects
- 1.5 million Object instances.
- 200 K pictures are labelled out of 330 K pictures.

In the following sample image, objects are recognized and the accuracy regarding the above output is shown in Figure 5.

So, here we have two types of threshold values such as threshold and other is NMS Threshold. We have evaluated the performance of the proposed stacked Yolov3 in various test scenarios [24]. Performance evaluation of the proposed approach with various threshold values is presented as follows. In the next experiment, we tested the accuracy by varying the threshold value but kept the value of the NMS threshold as 0.3. The results are shown below.

In Table 1 the accuracies are recorded by keeping the NMS threshold value constant and by altering the threshold values for the given input image. The NMS threshold value is 0.3, which is constant, and the threshold value varying from 0 to 1. We found that, when the threshold value is greater than 0.85, the accuracy is 60%, which shows that it detects only two objects out of three from the given image. When the threshold value is less than 0.85, and NMS threshold is 0.3, the accuracy is 100%, i.e., it detects the total three objects from the given input image [25]. Now we have tested the accuracy by keeping the threshold value as constant at 0.5 and by varying the NMS Threshold value. Accuracy values are tabulated in Table 2.

Figure 5. Object detection and accuracy computation using stacked Yolov3

Table 1. Accuracy of the proposed method by varying threshold values

Threshold	Accuracy
0	100%
0.25	100%
0.5	100%
0.65	100%
0.75	100%
0.85	100%
0.9	60%
0.95	60%
1	60%

Table 2. Accuracy of the proposed method by varying NMS threshold values

Threshold	Accuracy
0	60%
0.05	60%
0.1	100%
0.15	100%
0.25	100%
0.3	100%
0.45	100%
0.75	100%
0.85	100%
0.95	100%
1	100%

In Table 2, the accuracies are recorded by keeping the threshold value as constant and by changing the NMS threshold values for the given input image. It is found from results that if threshold value is 0.5 and the NMS threshold value is less than 0.1, and accuracy as 60% is obtained, which means it detects only two objects out of three from the given image. When the NMS Threshold value is ≥ 0.1, and the threshold is 0.5, we got accuracy as 100%, which results that it detects the total three objects from the given input image.

Now, we will find the accuracy by changing both NMS threshold and the Threshold value. The results are tabulated in Table 3.

From Table 3, the accuracies are recorded by keeping the threshold is more than 0.5 & less than 0.85, and NMS Threshold is ≥ 0.5, then the accuracy is 100%. For every detected object, a bounding box is drawn with an assigned class label. The confidence score is also displayed above the object. When compared with existing algorithms, the proposed stacked YOLOv3 is a bit faster with better accuracy.

Table 3. Accuracy of the proposed method by varying Threshold and NMS threshold values

Threshold	NMS Threshold	Accuracy
0.75	0.5	100%
0.75	0.5	100%
0.8	0.9	100%
0.75	0.75	100%
0.95	0.75	60%
0.95	0	60%
0.95	0.01	60%
0.95	0.5	60%
0.85	1	100%
0.9	1	60%

5. CONCLUSION

We instigate YOLO, which is a unified model for object detection. We have different approaches for object detection like R-CNN, fast R-CNN, faster R-CNN, YOLO (You only look once), and SSD (Single shot detector). R-CNN is very deliberate. Comparing to R-CNN, fast R-CNN is fast, but it uses selective search, which has slowed down the detection process. Faster R-CNN uses the convolutional network called regional network instead of selective search, which makes it 10 times faster than fast R-CNN. Choosing the correct approach for object detection is based on the problem we are solving. If accuracy is not much considered, but speed is considered, then we do use YOLO for computation problems SSD is a better choice. Contrasting to classifier-based approaches, YOLO is trained on a loss function and which badly keeps in touch to the detection performance, and therefore the whole model is trained conjointly. YOLOv3 is the quickest general-purpose object detector within the literature, and YOLO plunges the state-of-the-art in real-time object detection. Stacked YOLOv3 additionally generalizes well into the new domains by creating an ideal for the application that considers quick, sturdy object detection. To realize the scaling challenges and to detect small objects, multi-scale concept will be integrated with Yolov3.

REFERENCES

[1] Shakil, S., Rajjak, A., Kureshi, A.K. (2020). Object detection and tracking using YOLO v3 framework for increased resolution video. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(6): 118-125. https://doi.org/10.35940/ijitee.E3038.049620

[2] Agrawal, P., Girshick, R., Malik, J. (2014). Analyzing the Performance of Multilayer Neural Networks for Object Recognition. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision - ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8695. Springer, Cham. https://doi.org/10.1007/978-3-319-10584-0_22

[3] Szegedy, C., Toshev, A., Erhan, D. (2013). Deep neural networks for object detection. Advances in Neural Information Processing Systems, 26: 1-9.

[4] Liu, L., Ouyang, W.L., Wang, X.G., Fieguth, P., Chen, J., Liu, X.W., Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128(2): 261-318. https://doi.org/10.1007/s11263-019-01247-4

[5] Galvez, R.L., Bandala, A.A., Dadios, E.P., Vicerra, R.R.P., Maningo, J.M.Z. (2019). Object detection using convolutional neural networks. TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), pp. 2023-2027. https://doi.org/10.1109/TENCON.2018.8650517

[6] Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X. (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11): 3212-3232. https://doi.org/10.1109/TNNLS.2018.2876865

[7] Ammar, A., Koubaa, A., Ahmed, M., Saad, A. (2019). Aerial images processing for car detection using convolutional neural networks: Comparison between Faster R-CNN and YoloV3. 2019, [Online]. Available: http://arxiv.org/abs/1910.07234

[8] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. Computer Vision and Pattern Recognition (cs.CV), arXiv:1506.02640 [cs.CV].

[9] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[10] Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. Computer Vision and Pattern Recognition (cs.CV). http://arxiv.org/abs/1804.02767

[11] Shinde, S., Kothari, A., Gupta, V. (2018). YOLO based human action recognition and localization. Procedia Computer Science, 133: 831-838. https://doi.org/10.1016/j.procs.2018.07.112

[12] Cui, S., Zhou, Y., Wang, Y., Zhai, L. (2020). Fish detection using deep learning. Applied Computational Intelligence and Soft Computing, 2020: 3738108. https://doi.org/10.1155/2020/3738108

[13] Sang, J., Wu, Z.Y., Guo, P., Hu, H.B., Xiang, H., Zhang, Q., Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors (Switzerland), 18(12): 4272. https://doi.org/10.3390/s18124272

[14] Lu, J., Ma, C., Li, L., Xing, X.Y., Zhang, Y., Wang, Z.G., Xu, J.W. (2018). A vehicle detection method for aerial image based on YOLO. Journal of Computer and Communications, 6(11): 98-107. https://doi.org/10.4236/jcc.2018.611009

[15] Vidyavani, A., Dheeraj, K., Rama Mohan Reddy, M., Kumar, K.N. (2019). Object detection method based on YOLOv3 using deep learning networks. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(1): 1414-1417. https://doi.org/10.35940/ijitee.A4121.119119

[16] Liu, C., Guo, Y., Li, S., Chang, F. (2019). ACF based

region proposal extraction for YOLOV3 network towards high-performance cyclist detection in high resolution images. Sensors (Switzerland), 19(12): 2671. https://doi.org/10.3390/s19122671

[17] He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep residual learning for image recognition. Computer Vision and Pattern Recognition (cs.CV), arXiv:1512.03385 [cs.CV].

[18] Bell, S., Zitnick, C.L., Bala, K., Girshick, R. (2016). Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 2874-2883. https://doi.org/10.1109/CVPR.2016.314

[19] Kraus, F., Dietmayer, K. (2019). Uncertainty estimation in one-stage object detection. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, pp. 53-60. https://doi.org/10.1109/ITSC.2019.8917494

[20] Shreyas Dixit, K.G., Chadaga, M.G., Savalgimath, S.S., Ragavendra Rakshith, G., Naveen Kumar, M.R. (2019). Evaluation and evolution of object detection techniques YOLO and R-CNN. International Journal of Recent Technology and Engineering (IJRTE), 8(2S3): 824-829. https://doi.org/10.35940/ijrte.B1154.0782S319

[21] Venkatramaphanikumar, S., Prasad, V.K. (2013). Gabor based face recognition with dynamic time warping. In 2013 Sixth International Conference on Contemporary Computing (IC3), Noida, India, pp. 349-353. https://doi.org/10.1109/IC3.2013.6612218

[22] Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., Murphy, K. (2017). Open-images: A public dataset for large-scale multi-label and multi-class image classification. https://github.com/openimages.

[23] Lin, T.Y., Goyal, P., Girshick, R., He, K., Doll´ar, P. (2017). Focal loss for dense object detection. arXiv preprintarXiv:1708.02002.