2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

A Review: Object Detection Models

Aakash K. Shetty
Computer Engineering Dept.
NMIMS University
Mumbai, India
aakashshetty1999@gmail.com

Ishani Saha
Computer Engineering Dept.
NMIMS University
Mumbai, India
ishani.saha@nmims.edu

Rutvik M. Sanghvi
Computer Engineering Dept.
NMIMS University
Mumbai, India
rutviksanghvi21@gmail.com

Siddhesh A. Save
Computer Engineering Dept.
NMIMS University
Mumbai, India
siddheshsave99@gmail.com

Yashkumar J. Patel
Computer Engineering Dept.
NMIMS University
Mumbai, India
pyash6291@gmail.com

2021 6th International Conference for Convergence in Technology (I2CT) | 978-1-7281-8876-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/I2CT51068.2021.9417895

***Abstract*—Object detection and recognition is an integral part of Computer Vision. It has a multitude of applications ranging from character recognition to video analysis. Object detection now play a crucial role in industries like security, video, medical, sports, and many more. With the latest research, rapid development in deep learning, and computational image understanding object detection will be more prevalent. In the future, we will be able to implement object detection in much more mundane workings of our life. Our review paper will focus on the existing techniques that are present in the community and how each technique is different from the other techniques. We will start with an introduction about object detection as a tool and technique. This will be followed by a brief background discussing the most fundamental steps involved and the basic architecture of object detection. Then, we focus on several existing techniques and perform comparative analyses of these techniques to draw meaningful conclusions. These techniques include two-stage models such as R-CNN, Fast R-CNN, and Faster R-CNN and single-stage models YOLOv1, YOLOv2, YOLOv3, YOLOv4, and SSD. Finally, we will conclude by presenting some directions that provide a scope for further analyses along that direction and hope to suggest different techniques that can be implemented in different sectors for optimal results.**

***Index Terms*—YOLO; R-CNN; Deep Learning; Object Detection; Image Processing; Convolutional Neural Networks; Single-Stage Object Detection Model; Two-Stage Object Detection Model**

I. Introduction

Object detection, as useful as it may be it also challenging to overcome its various problems. Object detection techniques can be based on a range of other methods ranging from Hough transforms [2] to, what is now a very common tool in modern object detection, deep learning techniques [5]. Our paper will be limited to the novel deep learning techniques which use CNN in promising ways to implement object detection [15]. However, even with developing techniques, there are still problems due to variations in lighting, pose, foreground shapes, and various other factors at play [9]. There are some serious efforts taken to get the optimal results in every possible input. As new techniques and developments in object detection are introduced, the methods of implementation and the results of each technique vary depending on the various factors. These factors also decide which techniques can be implemented for which applications for optimal results. Some techniques are more accurate but can take more time to process. While others give less accurate results in very less time. Depending on these factors selection of techniques may vary from application to application.

This paper's focus is on CNN based models. A convolutional neural network is a pattern recognition technique in imagery that has made object detection a commonplace technique these days. The CNN accepts input and runs it through convolutional layers containing filters to detect edges and boundaries with other features to detect objects in images [13]. Each filter creates a feature map that travels through the network from one layer to another. CNN has gained popularity in recent times due to the availability of faster GPUs and the abundance of data [4]. Most sophisticated models of object detections these days make use of CNN for better results. CNN as well plays an important role in the implementation of algorithms in object detection. Comparing CNN to traditional techniques, CNN has a clear advantage. CNN has better architecture and is much more expressive than any existing traditional techniques [28]. Supported by the vast ocean of data and learning capabilities CNN is the best option available out there. The next structure followed by CNN is a pooling layer that connects two convolutional layers. The feature maps from consecutive convolutional layer is connected to each other though a pooling layer. The Pooling layer has its own feature map which connects the two consecutive convolutional layer feature maps. A pooling layer computes the maximum of a local patch of units in one feature map to reduce the spatial representation size [10]. The fully connected layers acquire the end of convolutional and pooling layers and reach a classification decision by labeling objects in a bounding box [26].

Object detection techniques aim to localize and classify objects in each image that run through the CNN. The ease to use and better technology has made object detection ubiquitous in is used from various applications like Face detection, surveillance, video analysis, suspicious object detection, and many more [18].

The paper will describe the object detection background and

basic concepts related to object detection. Then the paper talks about the various models available. It is followed by a comparison between these techniques based on its precision, fps, and time. In conclusion, the paper deduces various conclusions based on the results from the comparison.

II. Background

Object Detection System assumes bounding frames, one for every object detected. It's normal to expect too many bounding boxes for object detection. Every box also has a confidence score that tells how often the model believes this box holds an item. As a post-processing stage, we filter out boxes whose score falls below a certain threshold (also known as non-maximum suppression). Every CNN based Image detection technique has common basic functions to perform because eventually, all these techniques have to reach a common goal that is image classification and recognition. Object Detection Technique can be classified into: The Input, The Backbone, The Neck, The Head [21].

*1) **The Input**:* This consists of the input that the model accepts, which is a static image or real-time video depending on the application or the purpose of the model.

*2) **The Backbone**:* This is aptly named because it plays a vital role in all object detection techniques. This is where convolutional neural networks come into the picture and help extract the feature map from our input image. These features later help in the classification of important contents of the image. Some CNN includes VGG16, RESNET-50.

*3) **The Neck**:* This is more like a subset of the backbone which facilitates the feature discrimination and robustness in the model. It is the layer that handles different spatial resolutions in the image.

*4) **The Head**:* The final component handles the prediction. The classification of the image takes place in this part of the model. This further branches into two types which are discussed in detail further.

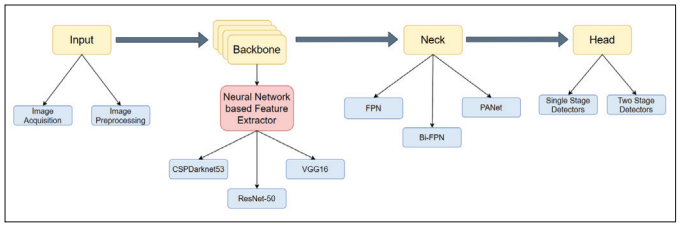

Fig. 1. Object Detection Model Framework

There are two types of region-based convolutional neural network models:

- Two-Stage object detection model
- Single-Stage object detection model

III. Two-Stage Object Detection Models

Pre-existing domain-specific detectors of image objects can typically be classified into two groups, one being a two-stage detector and the other being a one-stage detector. These models are divided into 2 stages:

- Using object proposal generation methods (To classify various regions that can contain the object). Then there is the refinement of the decision process.
- Second phase includes detecting false decisions and detections that refine boundary boxes. [24]
- Detection precision is prioritized by two-stage approaches, and examples include R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN.

In Two-Stage detectors we divide the image according to the ROI (Region of Interest) and have a high accuracy in localization and object recognition. Each stage requires special attention as complex processes are involved. Each component has to be trained individually so optimization of these models is difficult. Because of the complex computations, these models achieve higher precision but are very sluggish compared to single stage detection models.

A. R-CNN

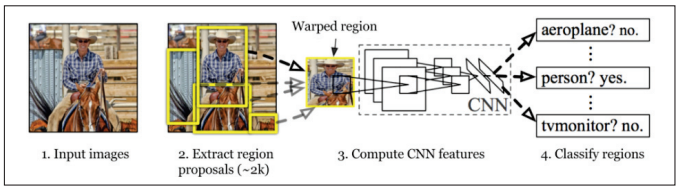

Fig. 2. Architecture of R-CNN

Regional Neural Convolutional Network, as shown in Fig. 2 [7], is model proposed by Ross Girshick. R-CNN selects regions from image with prominent features and bounds them with boxes [14]. These bounding boxes are labeled with categories. After this, CNN is used to perform computations and from each region, features are extracted. With this, categories are predicted and bounding boxes are adjusted.
R-CNN consists following parts: [24]

- To select multiple high-quality proposed regions, a selective search is carried out on input image. These regions have different sizes and shapes. Each region is categorized, labelled and is provided with boundary-box.
- A pre-trained R-CNN is used before output layer. This will help in converting the proposed regions into the dimensions required by the system. Features will be extracted from these regions as the output.
- Support Vector Machine which used for object classification is trained using features extracted from proposed regions. SVM determines which category particular feature and labels belong to.

Drawbacks of R-CNN Model:

- Selective Search algorithm is not flexible.
- It cannot be used for real-time processing due to its slow speed.
- Training requires lot of time as multiple stages are involved.

R-CNN is not commonly used in practical applications.

B. Fast R-CNN

Fast R-CNN is next version of the original R-CNN. Using selective search algorithm, several proposed regions are generated for precise object detection.

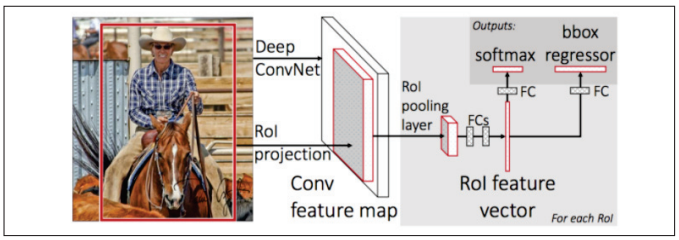

Fig. 3. Architecture of Fast R-CNN

In a Fast R-CNN architecture, as shown in Fig. 3 [8], whole image is given as input along with object proposals. Several convolutional and max-pooling layers are used to process the image and produce conv feature map as output. RoI pooling layer extracts fixed-length feature vector from feature map for each object proposal. All feature vectors are given as input to sequence of layers two types of outputs are produced, Softmax Probability for different object classes and four real valued numbers for each object class.

Fast R-CNN consumes less computational time and has improved accuracy of detection because:

- Entire image is given as input to CNN for extraction of features.
- For classification, Softmax is used in Fast R-CNN and not SVM like in R-CNN.
- A pooling layer named Region of Interest (RoI) makes the model faster and end-to-end trainable.

Limitations:

- The approach is complicated and lengthy.
- Calculation time is very high.
- Computation time for the model is around 2 seconds.

C. Faster R-CNN

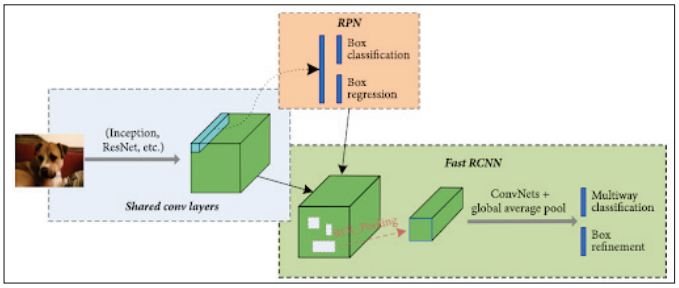

Fig. 4. Architecture of Faster R-CNN

Faster R-CNN (Regional convolutional neural network), as shown in Fig. 4 [22], is the 3rd part of R-CNN series. The disadvantage of Fast R- CNN was that it used a selective search algorithm that led to low speed. The selective search is discarded, and regional proposal network is implemented, which reduces the number of proposed regions, while ensuring accurate object recognition in the process [24]. RPN generates bounding boxes with scores. It is a fully convoluted neural network placed on ConvNets features. The bounding box scores helps to decide the degree of similarity of the object with the classes, in a way that is both very accurate and cost effective. Drawbacks: In the Object Region proposal method various operations are aligned in sequence forming a pipeline waiting for other previous operations and due to this it becomes time consuming to detect various objects.

D. Mask R-CNN

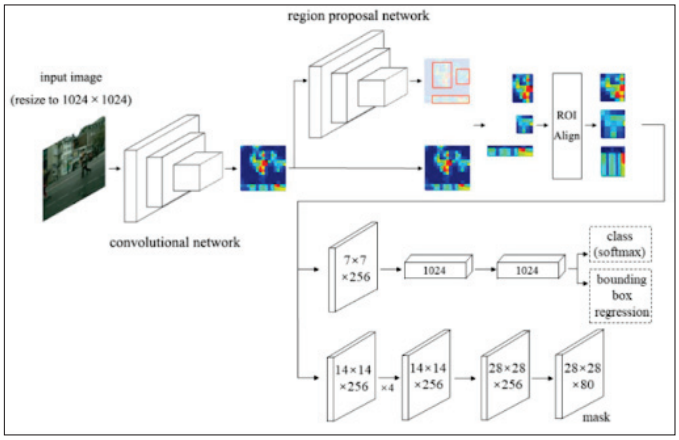

Fig. 5. Architecture of Mask R-CNN

Mask regional convolutional neural network (Mask R-CNN), as shown in Fig. 5 [27], is aimed at performing pixel level segmentation. Mask R-CNN is a modification to Faster R-CNN as it uses an additional step called Mask. It scans each pixel and predicts whether or not it is a part of the object. If the training data is labelled with pixel-level position of each object in the picture, Mash R-CNN model can improve the precision of object detection efficiently using these detailed labels. Mask R-CNN models replace the RoI pooling layer with an RoI alignment layer. The RoI alignment layer predicts the categories and bounding boxes of RoIs and allows us to use an additional fully convolutional network to predict the pixel-level positions of the object as it outputs feature maps of all RoIs of the same shape.

IV. Single-Stage Object Detection Models

Single-stage detectors require only a single pass through the neural network and predicts all the bounding boxes at once and are having feedforward type neural networks [24] [1] [19]. It is much faster as there is reduction in computational cost and hence there is improvement in performance. By introducing various changes in the two stage model and eliminating pipelines of the model we can convert them into a single stage model because of this the model can achieve high processing speed [1] [19]. The most common examples of one-stage object detectors are YOLO, SSD, SqueezeDet, and DetectNet.

A. SSD

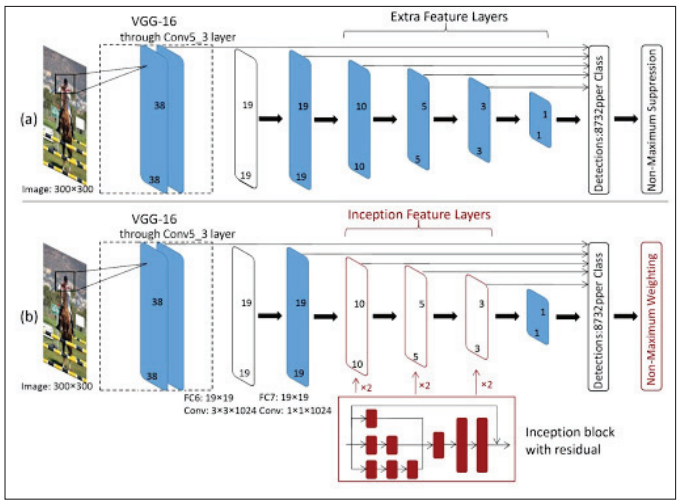

Fig. 6. Architecture of SSD

The Single-Shot Detector (SSD) model, as shown in Fig. 6 [17], was published in 2015. SSD is a single stage object detection model involving only one pipeline for performing object detection. To improve accuracy, SSD introduces:

- Filters that can accommodate different aspect ratios.
- The use of small filters is incorporated for prediction into object classes.
- The model can estimate the predictions generated from other feature maps along with changing ratio and resolution.

The SSD model is flexible and can be used in augmentation with systems requiring object detectors [24]. The higher resolution layers in the architecture of SSD are responsible for detecting small objects but such layers have some insignificant features which are not useful and are less informative for object detection. SSD is faster because of its single staged design and because of the architecture being single staged the whole model can be trained end to end. SSD makes more number of predictions as compared to YOLOv1 and other two stage models and can incorporate various aspect ratios. The accuracy of this model is a bit less when it is compared with the two staged models.

B. YOLOv1

YOLOv1 was a new approach to object detection. Previous research on object detection repurposes classifiers to perform detection. Instead of this object detection is performed as a regression problem to spatially separated bounding boxes and associated class probabilities [19] [24]. A single neural network predicts bounding boxes and class probabilities directly from the complete image in one evaluation. YOLOv1 is simple since it is a unified model or a single-stage detector. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes; there is no requirement for a complex pipeline involving different stages for different functions [19]. YOLOv1 detection mechanism involves three stages [24]:

- Converting the given image into required resolution.
- Passing the whole image into the neural network, all at once.
- Determining the probabilities of the objects detected.

YOLOv1 takes the whole image for detection and after scaling the model divides the input image into N × N grids of equal size. Each cell or grid is responsible for determining the object for that the centre of the object has to be placed into a particular grid [1]. The bounding box for any detected object consists of 5 values which involves the coordinate location of the bounding box along with the height and width for the bounding box and the most important value is the probability of the object belonging to the detected class. Probability determines the likeliness of the detected object belonging to a particular class. The class of the detected object is stored in the form of the number. The output of the detected bounding box is stored in an array involving the 4 values of location along with detected class number and class probability [19]. Thus YOLO prediction has the total shape of N * N * no of boxes * (5+Probability) [1].

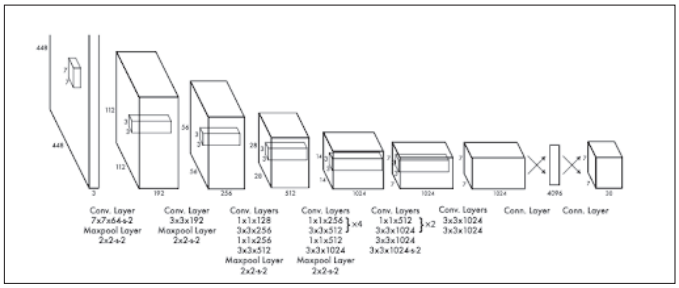

Fig. 7. Architecture of YOLOv1

YOLOv1 architecture, as shown in Fig. 7 [16], has 24 convolutional networks along with 2 fully connected layers. The feature space of the data is reduced the by the alternating 1*1 convolutional network. The training is done for 224*224 resolution and detection is done on 448*448 resolution of the image [19].

Limitations:

- Because of the low resolution classifier of YOLOv1 the model could not determine the low resolution objects present in the image.
- Due to absence of anchor boxes the model cannot predict more than one box for a particular region.
- Also due to congestion of some objects it can miss out some objects and can also miss some details.
- The model is less flexible and cannot incorporate testing with various resolution, and finds difficulty in localizing [1].

Benefits:

- The model is suitable for real time object detection having predictions made in one pass from a single neural network.
- Region Proposal Network accesses the whole image in one pass by sliding window and the bounding boxes are limited to particular regions from the image. And in turn generates fewer false positives for the less specific regions of the image.

C. YOLOv2

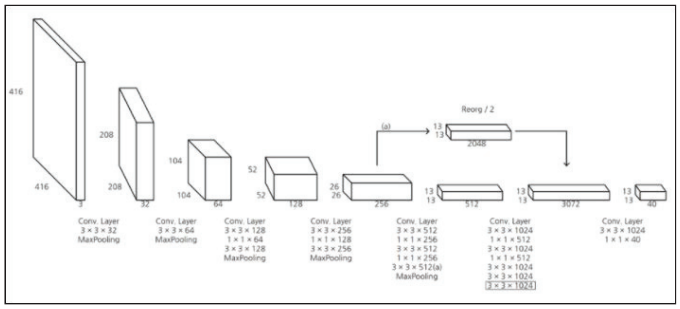

Fig. 8. Architecture of YOLOv2

YOLOv2, Fig. 8 [23], succeeded YOLOv1 [1]. Several improvements have been made in this model to make it faster and better. It is based on Darknet-19 [1] [20], which consists of 19 convolutional layers and 5 max-pooling layers, and all the fully convolutional layers of YOLO v1 have been removed [20]. YOLO v2 is trained on the combination of the COCO dataset and the ImageNet dataset to predict detections for object classes that do not have classified detection data [1].

Various enhancements have been introduced to make it simpler, quicker, and stronger. Batch normalization has been introduced which leads to more than 2% improvement in mAP [20] [12]. It regularizes the pattern and deletion of dropout can take place from the sample without overfitting it. High resolution classifier has been added which increases resolution to 448x448 [20] [12] [6]. There is more than 4% enhancement in mAP due to High Resolution classification [20] [12]. Anchor Boxes have been added for multi-class identification. There is reduction in accuracy due to usage to Anchor boxes. YOLOv1 predicts 98 boxes for each picture while YOLOv2 model predicts more than 1,000 boxes with anchor boxes [1] [14]. Dimension clustering is achieved using k-means clustering for clustering grids for artifacts and constructing bounding boxes. Fine grained surfaces help to distinguish small artifacts by reshaping layers using a revamped technique called moving through [20] [12]. Through fine graining, the image is separated into 13x13 grids in which small objects can also be observed. This precision, along with the pace, is accomplished. To make it identify more objects a hierarchical architecture has been added. By this accuracy is achieved along with speed. For making it detect more objects or making models stronger Hierarchical architecture was introduced [20] [1]. In this architecture, confidence score associated with each box helps in determining whether box has an object and if yes, then object class. YOLOv2 traverses the tree until certain threshold is reached and moves along the path with highest confidence score [20].

D. YOLOv3

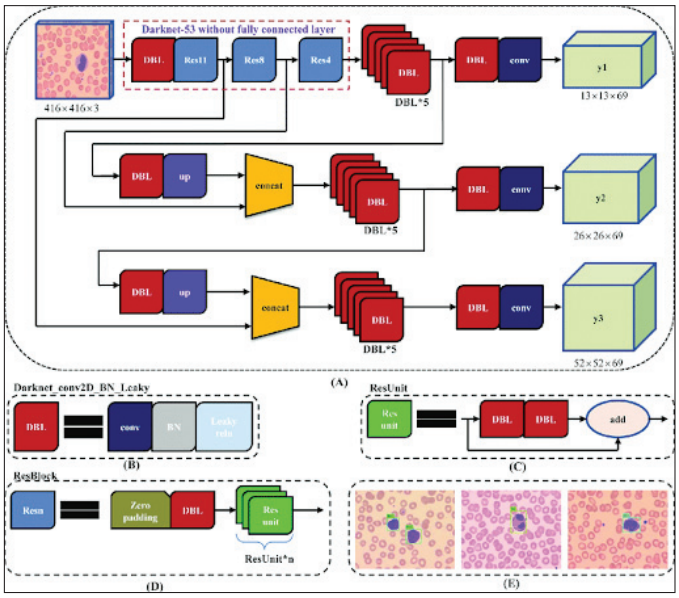

Fig. 9. Architecture of YOLOv3

YOLOv3, Fig. 9 [25], is an incremental form of YOLOv2. YOLOv3 calculates the objectness value for each bounding box via logistic regression [1] [21]. YOLOv3 modifies the way the cost function is measured. Unless the bounding box anchors cross the ground reality object rather than the others, the resulting objectness value will be 1 [21]. Other anchors with overlappings greater than the predefined threshold (default 0.5) do not incur any costs. Each ground truth object is associated with one boundary box prior only. If a bounding box anchor is not assigned, it incurs no classification and localization lost, just confidence loss on objectness [21]. We use tx and ty (instead of bx and by) to compute the loss.

YOLOv3 has a better class predictor than previously discussed techniques in this paper. Since YOLOv3 is a multi-label classifier, softmax function is replaced with different logistic classifiers [1] which measure the probability that the input region belongs to a particular class [21]. Rather than using a mean square error in the estimation of the classification loss, sigmoid cross-entropy loss for each number is used. Using statistical logistic classifiers also reduces statistical complexity [21]. In this dataset, where there are several objects in it, the softmax- function would detect one class for a single object at a time, which is usually not the case in YOLOv3. A multilabel approach in YOLOv3 gives 3 predictions for every location which gives a better prediction accuracy [21]. Every prediction output consists of the objectness, a boundary box, and 80 class scores in a grid of NxN, i.e. N×N×[3×(4+1+80)]predictions. YOLOV3 has a better feature extractor. In this model, a

new feature extractor, 53-layer Darknet-53 is implemented [1] to replace the older Darknet-19. A residual network like alternate connections is created with 3 x 3 and 1 x 1 filters in Darknet-53. This residual network structure helps to utilize the GPU more efficiently, making it more robust to evaluate and making it much faster [21]. Major improvement in accuracy is because of the residual connected networks [21]. YOLOv3 has the advantage over yolov2 that certain changes in the error function are included and three-scale detection of small to considerable objects. The multi-class problem became a multi-label problem and the performance of small objects improved [1].

E. YOLOv4

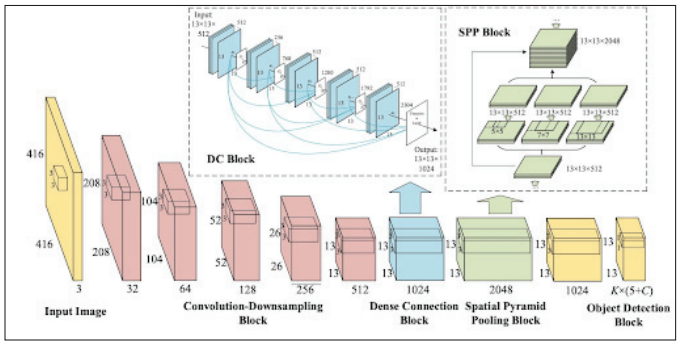

Fig. 10. Architecture of YOLOv4

YOLOv4, Fig. 10 [11], is a model which has Optimal speed and great accuracy. YOLOv4 is believed to be an object detector for production systems and is also designed for parallel computing. YOLOv4 improves YOLOv3's FPS and mAP by 12% and 10% respectively [3] and runs twice as fast as EfficientDet.

GPU plays a crucial role in training data set. To predict new and accurate models, a lot of computational power is required. To satisfy this need each GPU is assigned with mini batches for training but this leads to slow and inappropriate training. To overcome this shortcoming, a smaller mini batch is created by YOLOv4 object detector. It produces a fast and accurate target detector equipped with a 1080Ti or 2080Ti GPU [3]. YOLOv4 almost eliminates the reduction of the network operation speed by separating out the most significant context features.

CSPDarknet53 is used as a feature extractor as the backbone [3] of the architecture and it brings a significant improvement over the Darknet 53 of YOLOv3. In terms of detecting objects and classification the CSPDarknet53 was shown to be better. Spatial pyramid pooling (SPP) and Path Aggregation Network (PAN) are used as neck instead of FPN (Feature Pyramid Network) of YOLO v3 [3]. Head of YOLO v4 is the same as that of YOLO v3. It can make 3 predictions per location similar to the YOLO v3. The number of channels is 255 because of (80 classes + 1 for objectness + 4 coordinates) * 3 anchors.

There is use of data augmentation [3] which increases variability of the image which in turn increases the robustness to various environments. The training method can be optimized to achieve greater precision without increasing the cost of inference [3].

V. Comparisons and Results

A clear comparison between various detection methods is difficult. It is very difficult. Therefore, we can't give the best model a straight choice [1]. There is always a trade-off between precision and speed for real-life applications. Different models differ in speed, precision, applications and the maximum amount of FPS they provide for real time processing. We must therefore know about other features which have a major impact on performance. We have compared different models in different parameters to make the overall comparison.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

Fig. 11. Comparison of Different Models

The table, Fig.11, shows the precision and backbone of the different algorithms and provides all measurements of the accuracy. [21] It is apparent that YOLOv3 has better accuracy than all the algorithms other than RetinaNet. That still shows that YOLOv3 is an extremely strong model with great precision and high precision.

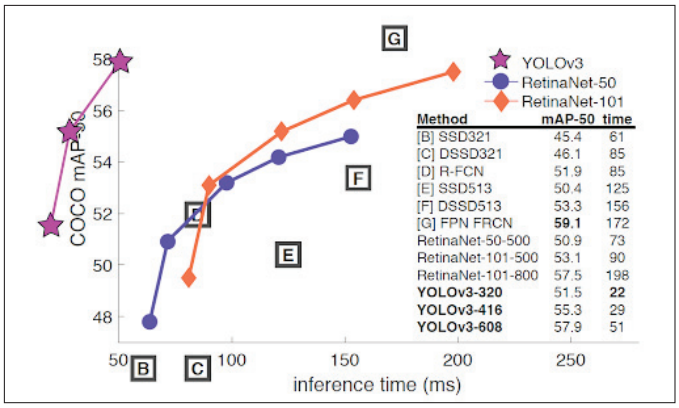

Fig. 12. Comparison of Different Models(Graph)

The graph, Fig. 12, shows speed/precision differences in the mAP at 0.5 IOU [21]. It's good to say YOLOv3, for it's very high, and far left. By these graphs and tables we can infer that YOLOv3 is good enough and by these graphs and tables we can combine the required data into a single table having the major parameters on which the algorithms are tested and we provide one stop table incorporating major object detection models with their accuracy and speed.

Models	mAP-50	FPS	Time(s)	Backbone
R-CNN	-	0.03	40-45	Pascal VOC 2007
Fast R-CNN	39.9	0.5	2	VGG-16
Faster R-CNN	42.7	7	1.1	VGG-16
Mask R-CNN	36.1	8.6	1.16	ResNet-101 FPN
YOLOv1	37.4	45	0.02	Pascal VOC 2007
SSD300	41.2	46	0.61	ResNet-101-SSD
SSD512	46.5	19	0.125	ResNet-101-SSD
YOLOv2 288x288	44.0	91	0.02	Darknet-19
YOLOv2 480x480	46.9	59	-	Darknet-19
YOLOv2 544x544	47.9	40	0.03	Darknet-19
YOLOv3 320x320	51.5	45	0.22	Darknet-53
YOLOv3 416x416	55.3	35	0.29	Darknet-53
YOLOv3 608x608	57.9	20	0.51	Darknet-53
YOLOv4 416x416	62.8	38	0.25	CSPDarknet-53
YOLOv4 512x512	64.9	31	-	CSPDarknet-53
YOLOv4 608x608	65.7	23	-	CSPDarknet-53

TABLE I
COMPARISON OF DIFFERENT MODELS(TABLE)

We have considered the following parameters for the comparison of different models:

*1) **Average Precision**:* Average Precision(AP) metric helps in measuring the accuracy of object detectors. Average precision computes the precision value over 0 to 1.
Average Precision is based on:

- Precision measures the accuracy of predictions in percentages.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- Recall value calculates the number of true positives from the relevant instances.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

- The IoU is given as intersection over union, the ratio of I (the area of the intersection), and U (the union of the bounding box and the actual ground truth bounding box).

*2) **Frames Per Second**:* Frames Per Second(FPS) in a way describes how fast a method is. It can be used with real time to measure video sequence processing speed in a video.

According to the comparisons carried out in the above table, Table 1, R-CNN techniques provide very less FPS and they take too much time to process the output. So, these techniques are not the most optimal to use while carrying out object detection. Whereas, the YOLO and SSD techniques provide a decent FPS and take less computational time so they are considered as optimal object detection techniques and are the most used techniques. Out of these techniques, the latest versions of YOLO are considered to be the best. YOLOv3 uses Darknet-53 backbone which is a modification of its predecessor YOLOv2 which uses Darknet-19. Darknet-53 is a convolutional neural network that is 53 layers deep whereas Darknet-19 is 19 layers deep. Also, Darknet-53 uses residual connections which is why it gets an upper hand over Darknet-19. Hence, YOLOv3 is considered better than its predecessor YOLOv2 as it provides a better accuracy even though FPS provided is less. YOLOv4 is the latest version of the YOLO technique which is the modification of YOLOv3. It has the same head as YOLOv3 with the only difference being the backbone which is CSPDarknet-53. CSPDarknet-53 is a convolutional neural network that uses Darknet-53 and a CSPNet strategy to partition the base layer feature map into two parts and then merge them into a cross-stage hierarchy. Using a split and merge strategy allows for more gradient flow through the network. Therefore, in industrial setting, for production systems and parallel computing YOLOv4 has proven to be highly accurate and best of all the techniques discussed in the paper.

VI. CONCLUSION

There are a variety of methods for identifying and finding objects, with a difference in speed and results in precision [1]. There is no perfect algorithm or model for all the applications. Each algorithm is equally good for its application, and each algorithm has its appropriate application and it should be chosen following the application model. Based on the recent research carried out, this paper gives a comparative study comparing the various object detection techniques that work on different subproblems. The architecture, advantages, and disadvantages of various object detection techniques are discussed, and the comparison was made based on accuracy, FPS provided, and computational time. Of the two types of region-based convolutional neural network models(single-stage and two-stage), under the two-stage object detection model, the R-CNN techniques are discussed. In the single-stage object detection model, various versions of the YOLO object detection model and SSD techniques are discussed. According to the experimental result, the two-staged object detection techniques were proved to be less optimal as compared to the single staged object detection techniques. Single staged object detection techniques proved to provide better FPS indicating high-performance speed along with great accuracy and hence they can be used for real-time applications. Overall YOLOv4 was proved to be the most recent and powerful object detection technique as it provides a decent FPS and comparatively higher accuracy than the other object detection techniques. The hardware requirement for YOLOv4 is higher as compared with other models. Various versions of YOLO such as YOLOv2 and YOLOv3 are equally good for their applications. The object detection system is applied in various areas such as video surveillance and face recognition. So, based on the experimental results and comparison carried out in this paper, the researchers and developers will be able to choose the most suitable object detection technique for their project depending on their requirements.

REFERENCES

[1] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 687–694. IEEE, 2020.

[2] Olga Barinova, Victor Lempitsky, and Pushmeet Kholi. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] C Dong, CL Chen, K He, et al. Learning a deep convolution network for image super-resolution [m]. computer vision-eccv 2014, 2014.

[5] Juergen Gall, Nima Razavi, and Luc Van Gool. An introduction to random forests for multi-class object detection. In *Outdoor and large-scale real-world scene analysis*, pages 243–263. Springer, 2012.

[6] Dweepna Garg, Parth Goel, Sharnil Pandya, Amit Ganatra, and Ketan Kotecha. A deep learning approach for face detection using yolo. In *2018 IEEE Punecon*, pages 1–4. IEEE, 2002.

[7] R Girshick, J Donahue, T Darrell, and UC Berkeley. Malik., j.(2014). rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[10] J Han, D Zhang, G Cheng, N Liu, and D Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. ieee signal process mag 35 (1): 84–100, 2017.

[11] Zhanchao Huang, Jianlin Wang, Xuesong Fu, Tao Yu, Yongqi Guo, and Rutong Wang. Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection. *Information Sciences*, 2020.

[12] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Ashwani Kumar, SS Sai Satyanarayana Reddy, and Vivek Kulkarni. An object detection technique for blind people in real-time using deep neural network. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pages 292–297. IEEE, 2019.

[15] Yu Liu, Hongyang Li, Junjie Yan, Fangyin Wei, Xiaogang Wang, and Xiaoou Tang. Recurrent scale approximation for object detection in cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 571–579, 2017.

[25] Qiwei Wang, Shusheng Bi, Minglei Sun, Yuliang Wang, Di Wang, and Shaobao Yang. Deep learning approach to peripheral leukocyte recognition. *Plos one*, 14(6):e0218808, 2019.

[16] Junyan Lu, Chi Ma, Li Li, Xiaoyan Xing, Yong Zhang, Zhigang Wang, and Jiuwei Xu. A vehicle detection method for aerial image based on yolo. *Journal of Computer and Communications*, 6(11):98–107, 2018.

[17] C. Ning, Huajun Zhou, Yan Song, and J. Tang. Inception single shot multibox detector for object detection. *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 549–554, 2017.

[18] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. *Procedia computer science*, 132:1706–1717, 2018.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[20] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[21] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[22] Yun Ren, Changren Zhu, and Shunping Xiao. Object detection based on fast/faster rcnn employing fully convolutional architectures. *Mathematical Problems in Engineering*, 2018, 2018.

[23] Seonkyeong Seong, Jeongheon Song, Donghyeon Yoon, Jiyoung Kim, and Jaewan Choi. Determination of vehicle trajectory through optimization of vehicle bounding boxes using a convolutional neural network. *Sensors*, 19(19):4263, 2019.

[24] Jyothi Shetty and Pawan S Jogi. Study on different region-based object detection models applied to live video stream and images using deep learning. In *International Conference on ISMAC in Computational Vision and Bio-Engineering*, pages 51–60. Springer, 2018.

[26] Xiaohan Yi, Liangcai Gao, Yuan Liao, Xiaode Zhang, Runtao Liu, and Zhuoren Jiang. Cnn based page object detection in document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 230–235. IEEE, 2017.

[27] Yiqing Zhang, Jun Chu, Lu Leng, and Jun Miao. Mask-refined r-cnn: A network for refining object details in instance segmentation. *Sensors*, 20(4):1010, 2020.

[28] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.