Object Detection Algorithms Compression CNN, YOLO and SSD

Shreyas Pagare
PHD research scholar RGTU, Chameli Devi Group of Institutions, Madhya Pradesh

Rakesh Kumar, PhD
Professor in CSE Department, Rabindranath Tagore University - Bhopal, Madhya Pradesh

ABSTRACT
Object detection is a crucial component of computer vision, and since 2015, several studies have expanded with the use of convolution neural networks and their changed structures. There are techniques for detecting representative objects, including YOLO and convolutional neural networks and also use SSD. This study introduces three exemplary CNN and YOLO-based and SSD algorithm series that address the CNN bounding box issue. We examine the accuracy, speed, and cost of many algorithmic series. All model of YOLO provides an excellent balance between speed and accuracy when compared to the most recent advanced solution.

Keywords
CNN, YOLO, SSD

1. INTRODUCTION
Object detection in the areas of computer vision and image processing is still photos and videos is crucial techniques used [1]. However, in the natural environment Images are distinguished by various shapes and colors along with textures. Because it has real world adaptability, it faces a very difficult problem. Finding an item in a given image is the basic goal of object detection. To categorize, conduct accurate searches and understand what the observed object implies [2].

The human eye can effortlessly conduct these operations. That is possible, but only via the use of certain algorithms can be distinguished. The object detection algorithm includes the following steps: To categorize, identify a key object and center the bounding box around that object (Bounding Box), and the fields of application include various Volume detection/tracking (Vehicle Detection), surveillance system (Surveillance), and so on. There is [3]. Recently, face detection (Face Detection), image classification (Image Classification) [25, 29], Video Recognition, Sound Various applications such as audio recognition [28] As a major machine learning tool used in convolutional neural networks algorithm is active used [4, 5, 30]. SIFT was an early object detection technology (Scale Invariant Feature) Speeded-Up Robust Features [9], similar to the Histogram of Oriented Gradients [14] method.

To increase the recognition rate, SVM (Support Vector Machine) and the approach of creating and identifying features using [26] were used. There were further research [22, 30] that used the same machine learning approach. Alex in his research the deep learning via a convolutional neural network is used in the paper [10]. Object identification using (Deep Learning) Suggestions for improving performance. Object identification technique based on convolutional neural network is presented in this study. The CNN algorithm series and the difficulty of CNN candidate region detection Investigate the YOLO family of algorithms for solving problems. In terms of see, compare and contrast the performance of typical algorithms. The future of YOLO and the Faster R-CNN algorithm are discussed in this context.

2. CNN RELATED ALGORITHM ANALYSIS
2.1Convolutional Neural Network
A Convolutional Neural Network is a multi-layer perception that has been required in conducting to operate using visual data and specializes in vision estimation and forecasting. It performs with seeds (called filters) that are over the portrait and simply create a saliency map (which personifies how well a particular component is visible at a precise point of the image or not), and so it actually creates a comparatively tiny number of parameters initially. However, as we proceed with greater depth into the intranet, this same number of nodes develops as well as the magnitude of graphs gets smaller without having to lose pertinent data utilizing accumulating processes.

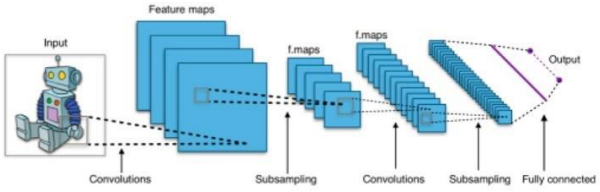


Figure 1: CNN

The layers of a ConvNet learn features of increasing complexity, such as detecting shapes during the first layer and gaining a prominent place in diverse postures in the last layer.

2.2 Recurrent Neural Network (RNN)
Recurrent neural networks identify consecutive features of the information and anticipate the next likely situation using patterns. RNNs have been used in deep learning and the creation of algorithms that replicate neuronal behavior in the nervous system. They're extremely beneficial in cases where comprehension was essential to predicting a response. They differ from those other neural networks in that they use responses to examine a set of memory that influences the accurate results. Such natural cycles and good documentation to endure the recall is a common term for this phenomenon. Prominent Recurrent neural network usage instances comprise word embeddings wherein the subsequent word in a term and the next expression in a statement are reliant on the facts that came before everything. Another innovative experiment was using a Recurrent neural network that received training on Literary plays and created Shakespeare-like prose. Computational inventiveness is a sort of RNN-based literature. This reproduction of artistic expression is facilitated by the AI's linguistic knowledge and interpretation obtained from its training phase.

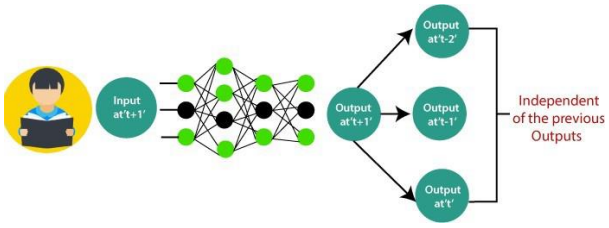


Figure 2: RNN

2.3 Region-based CNN

In the input picture, the first module is a category-independent contender. produces a region and a potential detection zone that the detector can use to identify CNN's second module consists of all contestants. Three times, extract a feature vector of the same lenth from the region. The class-specific linear SVM is employed in the second module to [11] Items inside the beam region are categorised.

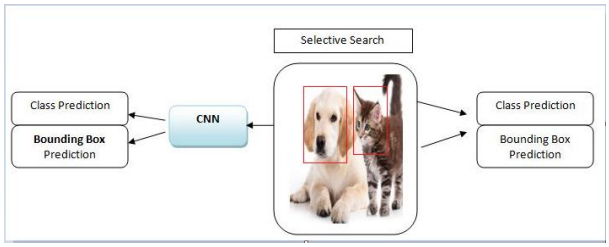


Figure 3: R-CNN

2.4 Fast R- CNN

R-CNN distinguishes between CNN, SVM, and regression learning. As a result, it has the issue of requiring a lengthy computation processing time. In addition, Fast R-CNN transforms the whole input picture into one and trains CNN on it as a candidate area. CNN has been trained. By combining one convention feature map produced by extract [24]. The primary distinction between R-CNN and Fast R-CNN is RCNN. Whereas the local detection area is supplied in pixels, candidate areas in Fast R-CNN are input in functional maps.

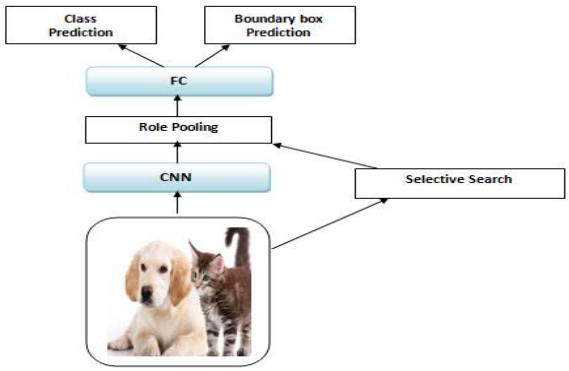


Figure 4: Fast-CNN

CNN use convolution to acquire local information near a pixel. Although it has the benefit of expressing numerous objects and detecting object position, it has a limitation. A region-based convolutional neural networks (R-CNN) Networks use deep learning regression approach to do classification. One issue was resolved [2]. R-CNN is a candidate region Proposal) that is used to train CNN. It detects an item in the unknown and is made up of three parts.

2.5 Faster R- CNN

The candidate region creation module of Fast R-CNN is independent of CNN. Because it is conducted in a distinct module, learning and execution speed are increased. There is an issue with inefficiency in Moreover, faster RCNN object identification and posting in the same convolutional network Make a beam area. Instead of using the Selective Search methodology, a distinct area Added by utilizing Region Proposal Network (RPN) Objects are recognized as candidate areas by estimating the derived functional maps all [11]. Feature map extraction outperforms all prior CNN models. When the declaration of the feature chart is smaller than the declaration of the input picture, the procedure and contender region construction are conducted in a succession of networks

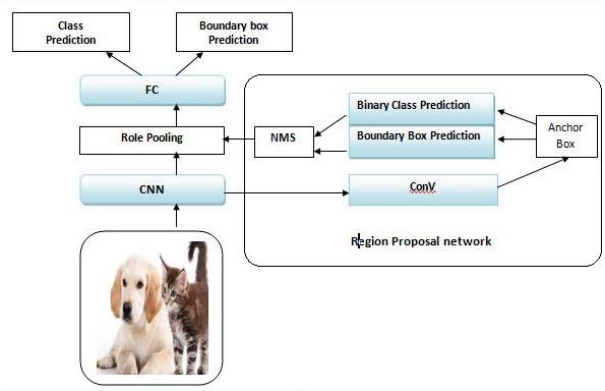


Figure 5: Faster R-CNN

CNN, R-CNN, and SppNET are all CNN-based object detection systems that were explored and residential in the sequence of Fast R-CNN and Faster R-CNN. Looking at it, what creates candidate regions and what doesn't The difference in speed was evident. After Fast R-CNN Note that the creation of beam regions has a significant impact on performance it means. Table 1 shows the realities of R-CNN, Fast R-CNN and Faster R-CNN. Compare and show line speed. CNN, R-CNN, and SppNET are all CNN-based object detection systems that were explored and developed in the sequence of Fast R-CNN and Faster R-CNN.

Table 1: Comparison of Speed for CNN [14]

	R-CNN	Fast R-CNN	Faster R-CNN
Test Time per Image	50sec	2sec	0.2sec
Speedup	1x	25x	250x
mAP (VOC2007)	66.0	66.9	66.9

3. YOLO RELATED ALGORITHM ANALYSIS

YOLO is an additional method for object discovery [15]. Items in the image and their positions are predicted by an algorithm based on a single look at the image. It has been seen Instead of identifying it as a classification item, it is approached as a regression issue using multidimensionality separation and class probabilities (Class Probability). Via CNN, the force picture is represented as a grid of tensors. Split the item bounding box and class likelihood to identify things in the vicinity according to each interval. YOLO is a contender to extract zones without using a separate network. As a result, it has demonstrated higher performance in terms of processing speed.

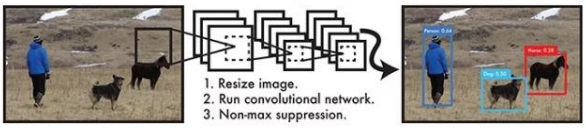


Figure 4: YOLO Architecture

3.1 YOLO v1

Split the input image into a S*S lattice, and then choose a specific item. If the center of a correspond to the midpoint of a lattice cell, then that grid cell performs the object detection job [15]. Grid cells for each grid, bounding box, self-confidence score, and class probabilities are all predicted. These predictions are calculated as a tensor S*S 5 +, where B is bounded. The cell's conditional classes are represented by the number of classes, C. The roe score predicts whether or not an item is contained by a bounding box. When determined as in Equation 1), it is a number that shows the confidence level and correctness of the model.

CS=Pr (Obj)*IOU

IOU stands for Intersection over Union. If we are not found object in that cell, the confidence score is 0. If the item found, the IOU value between the ground truth or the prediction box has been determined. Each grid cell follows conditional class likelihood, and each bounding box consists of x, y, w, h, and confidence execution class-specific confidence scores for all bounding boxes at time point. It is determined by multiplying the conditional class probability by the deity of the enclosing box (Equation 2).

3.2 YOLO v2

YOLO v2 is intended to utilize large amounts of classification data as a method, only classification data through a joint training algorithm However, it is possible to train object detectors as well. Rain on YOLO v1 batch normalization to improve solution accuracy and speed (normalization) layer was added, and the completion of initial learning. The convolution anchor size was set for this, as well as the work was perfectly alright to improve performance result even with high-resolution input and the prediction of the bounding box is a fully connected layer. Instead, it's done in the anchor box, that shrinks the network while increasing output resolution [16].

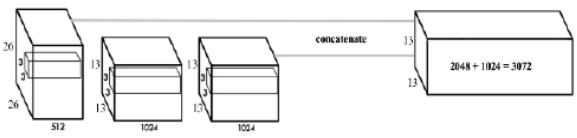


Figure 4: Pass-through Network Design of YOLO v2

3.3 YOLO v3

YOLO v3 use Logistic Regression to estimate the bounding box's abjectness score. Through this matching strategy, each bounding box has an object calculate objectivity score that calculates whether or not there is, and anchor Park the cross overlapping union (IOU) of the Swap and Ground Truth boxes is the highest matches the box to 1. Bounding box is predefined predictions can be ignored if there is not a specific amount of overlap between them. Moreover, feature detectors with improved duplication detection avoidance, class predictions, bounding box predictions, and pyramid networks (feature pyramid network) in a way similar to the concept predicting different scale boxes with extract the gong This allows processing and reporting of the combined feature maps. An additional convolutional layer is used to predict large tensors included [17].

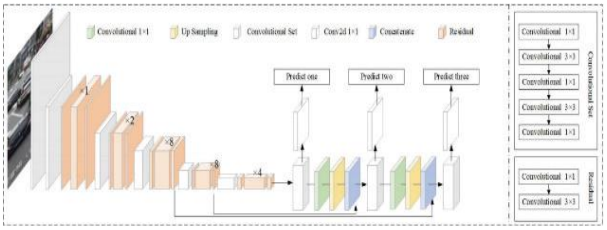


Figure 5: Structure Detail of YOLO v3 [18].

3.4 YOLO v4

YOLO v4 used to tackles the issue with creating an object detector that can be trained on a single Graphics processing unit with a reduced mini-batch size. This allows you to train a highly fast and accurate object detector with a single 1080 Ti or 2080 Ti GPU. This issue has been solved in YOLO v4 by creating an object detector that can be trained on a single GPU with a shorter mini-batch size. This enables the training of a very fast and accurate object detector on a single 1080 Ti or 2080 Ti GPU.

Although YOLO detectors are one-stage, there are two-stage detectors including such R-CNN, fast R-CNN, and faster R-CNN that are accurate but sluggish. We'll focus on the former. Consider the basic components of a contemporary one-stage object detector.

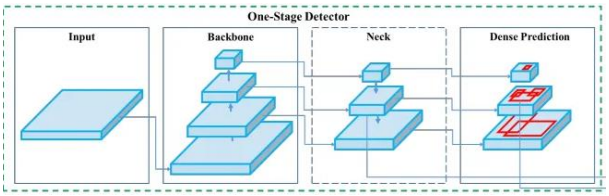


Figure 6: Object detector, [20]

3.5 YOLO v5

The same team that developed the initial YOLO algorithm issued YOLO v5 in 2020 as an open-source project, which is now maintained by Ultralytics. The success of early versions is built upon by YOLO v5, which also involves a series of updates and new features.

Based on the EfficientNet network design, YOLO v5 utilizes a more intricate architecture called EfficientDet (architecture depicted below). YOLO v5 can gain higher accuracy and better generalization to a greater range of item categories because to the use of a more robust architecture.

"Dynamic anchor boxes" is a modern technique that used YOLO v5 to create the anchor boxes. The ground truth bounding boxes are first grouped into clusters using a classification technique, and then the centroids of those organizations are used as the anchor boxes. As a result, the anchor boxes can match the size and form of the identified objects more precisely [21].

3.6 YOLO v6

The CNN architecture utilized in YOLO v5 and v6 is one of the primary changes between the multiple variants. EfficientNet-L2 is a variation of the EfficientNet architecture that is used by YOLO v6. It has fewer parameters and a more efficient computational model than the EfficientDet architecture used in YOLO v5. On a variety of object detection benchmarks, it can

produce state-of-the-art results.

The "dense anchor boxes" technology, which is new in YOLO v6, is also offered. YOLO v7 outperforms other object detection algorithms in terms of accuracy. Using the well-known COCO dataset, it has an average accuracy of 37.2% at an IoU threshold of 0.5, which is similar to other cutting-edge object identification methods.

3.7 YOLO v8

The release of YOLO v8, which promises extra features and enhanced performance over its predecessors, has been confirmed by Ultralytics as of the time this article was written. The new API in YOLO v8 makes training and inference on CPU and GPU devices more simpler, while the framework still covers preceding YOLO iterations. A scientific article that will provide a detailed description of the model design and performance is still being worked on by the developers.

4. SSD

Whereas regional proposal network (RPN)-based techniques like the R-CNN series require two shots—one for generating region proposals and the other for recognizing the object of each proposal—SSD (Single Shot Detector) takes just one shot to detect numerous objects inside the picture. Thus, SSD is significantly faster over two-shot RPN-based methods. Faster R-CNN (73.2% mAP at 7 FPS) and YOLOv1 (63.4% mAP at 45 FPS) are exceeded by SSD300 (74.3% mAP at 59 FPS) and SSD500 (76.9% mAP at 22 FPS).

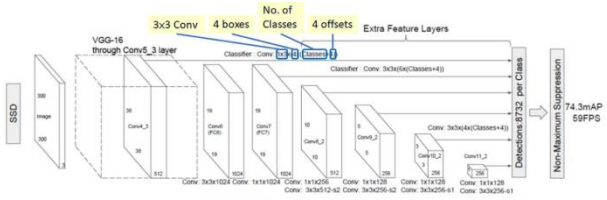
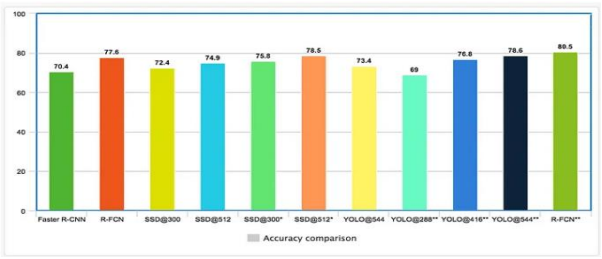
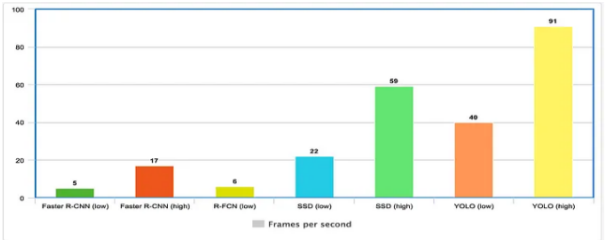


Figure: SSD Network Architecture

The model is trained using data from both PASCAL VOC 2007 and 2012 for the outcome shown below. The PASCAL VOC 2012 testing set is used to determine the mAP. Results for input photos of 300 x 300 and 512 x 512 on SSD are displayed in the graph. Results for YOLO include 288 x 288, 416 x 461 and 544 x 544 photos. For the same model, higher quality photos provide greater mAP but take longer to analyze.



Speed is affected by feature extractors and input picture resolutions. The maximum and lowest FPS recorded by the relevant publications are shown below. Nevertheless, because they were tested at various mAPs, the results below may be significantly skewed.



One of the most popular areas of computer vision is object detection, and there are many models in this area. None the less, not all models were created equally. Although each of the models we cover in this video has benefits as well as drawbacks, we are only concerned in those that are most important to us. We contrasted a Faster RCNN model from the Two Shot detector family. The SSD (Single Shot Detectors) and YOLO single shot variants were also included in the comparison. When comparing models for speed, we concentrated on how many frames each model could process in a second, or their inference speed. In terms of accuracy, we looked at which model obtained the best results as well as the reliability of those results. Finally, we considered the model's simplicity of implementation, which mainly focused on the framework (Opencv, PyTorch, TensorFlow) needed to utilize it and the smallest amount of code we needed to write to enable detections from the model.

Table 2: Comparison of Faster RCNN & SSD & YOLO

	Speed	**Accuracy**	**Ease of implementation**
Faster RCNN	Bad	Good	Bad
SSD	Good	Good	Bad
YOLO	Good	Good	Good

5. CONCLUSIONS

In this study, we examined the algorithm employing CNN-based object detection, including YOLO. YOLO is integrated object detection as a model, compared to CNN, the construction is simple and the whole image can be directly learned and used in real applications seems to be suitable for the Access rooms based on other classifiers Unlike the law, YOLO is a hand-me-down that directly responds to detection performance train on real functions and improve real-time objects in terms of processing time body detection is possible. We discovered the replacement issue for the re-defined module and the allocation difficulty for the dynamic label assignment during the study process. To address the issue, we suggest using the trainable bag-of-freebies technique to improve item identification precision. Application is an important process in the program, and it determines the appropriateness of this It will have to be studied together with a separate algorithm.

6. REFERENCES

[1] Asifullah Khan, Anabia Sohail, Umme Zahoora, AqsaSaeed Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", Computer Vision and Pattern Recognition, Available at https://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf [Accessed Mar. 13, 2020].

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for ccurate Object Detection and Semantic Segmentation", IEEE Conference on Computer Vision and Pattern Recognition, pp.580-587,

2013.

[3] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikainen, "Deep

[4] Learning for Generic Object Detection: A Survey", International Journal of Computer Vision, vol.128, pp.261-318 2020.

[5] Kaimin He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Spatial Pyramid Pooling in Deep Convolutional

[6] Networks for Visual Recognition", European Conference on Computer Vision, Part 3, pp.346-361,2014.

[7] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhizi Feng, Rong Qu, "A Survey of Deep

[8] Learning-based Object Detection", IEEE Access, vol.7, pp.128837-128868, , 2019.

[9] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol.60, pp.91-110, 2004.

[10] Juan Du, "Understanding of Object Detection based on CNN Family and YOLO", Journal of Physics,

[11] Conference Series, vol.1004, issue.1, 2018.

[12] Tsung-Yi Lin, Priya Goyal, Roos Girshick, Kaiming He, Piotr Dollar, "Focal Loss for Dense Object Detection", International Conference on Computer Vision, pp.2999-3007, 2017.

[13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features (SURF)", Computer Vision and Image Understanding, vol.110, issue.3, pp.346-359, 2008.

[14] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Communications of the ACM, vol.60, no.6, 2017.

[15] Yurong Yang, Huajun Gong, Xinhua Wang, Peng Sun, "Aerial Target Tracking Algorithm Based on Faster RCNN Combined with Frame Differencing", Aerospace, vol.4, no.32, 2017.

[16] Kwanghyun Kim, Sungjun Hong, Baehoon Choi and Euntai Kim, "Probabilistic Ship Detection and

[17] Classification using Deep Learning", Applied Sciences, vol.8, no.6, 2018.

[18] Rohith Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms," 2018. Available at https://towardsdatascience.com/r-cnn-fast-r-cnn-fasterr- cnn-yolo-object-detection-algorithms-36d53571365e. [Accessed: Mar. 13, 2020].

[19] N. Dalal, B. Triggs, "Histograms of Oriented Gradients

[20] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection" ,2020.

[21] Mingxing Tan Ruoming Pang Quoc V. Le Google Research, Brain Team. "EfficientDet: Scalable and Efficient Object Detection". 2020

[22] Chien-Yao Wang , Alexey Bochkovskiy, and Hong-Yuan Mark Liao, Institute of Information Science, Academia Sinica, Taiwan, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 2022.

[23] https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359