Semiconductor Optoelectronics, *Vol. 41 No. 12 (2022), 1629-1646*
https://bdtgd.cn/

COMPARATIVE STUDY OF VARIOUS DEEP LEARNING OBJECT DETECTION ALGORITHMS

Sangeeta M. Borde [1], Dr. Harsh Lohiya [2]

[1] Research Scholar, Department of Computer Science, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P., India

[2] Research Guide , Department of Computer Science, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P., India

ABSTRACT:

All visual media are interpreted by a computer as a collection of numerical values. They need image processing algorithms to inspect the contents of images as a result of this method. This article compares 4 major image processing algorithms Single Shot Detection (SSD), Histogram of Oriented Gradients (HOG), Faster Region-based Convolutional Neural Networks (Faster-RCNN), and You Only Look Once version 3 (YOLO) to find the fastest and most efficient of three.

Using the **Microsoft COCO (Common Object in Context)** dataset for this comparison study, these four algorithms' efficiency is assessed, along with their advantages, and on the basis of metrics like accuracy, precision, and F1 score, limits are examined.

According to the analysis's findings, the use cases that each algorithm applied is best suited to determine how suitable it is compared to the other three. In the same testing setting, YOLO-v3 performs better than SSD, HOG and Faster R-CNN is the most effective of the three algorithms.

Keywords: Object detection, SSD, HOG, FRCNN, YOLO-v3, COCO dataset

INTRODUCTION:

Due to cutting-edge findings in the fields of object identification, natural language processing, and image classification, deep learning technology has recently become a household term. The causes behind deep learning's popularity, are large datasets that are readily available, and strong graphics processing units that make up the two sides of learning [1].

Nowadays, Image classification and detection are the most important pillars of image detection in the research area. There is plenty of dataset available that has achieved remarkable worldwide competition such as PASCAL, VOC, COCO OR KITTY, and ILSVR [1].

We want to analyse convolutional neural network (CNN)-based deep learning approaches for object detection. Convolutional neural networks are wonderful since they don't require manually made feature extractors or filters.[1] The contents of the paper are portrayed as

follows.

Fig 1 shows the organization of the paper.

The contents of the article are portrayed as follows. Fig. 1 depicts the roadmap of the paper. Section 1 depicts the literature survey, Section 2 deals with the Evolution of CNN, Section 3 deals with existing methodologies, Section 4 discussed datasets of object detection, and section 5 discusses the experimental setup, Further, in the paper, the results and discussions with the conclusion are shown.

The popularity of deep learning increased in the late 1980s and 1990s with the development of the backpropagation algorithm proposed by Hinton et al. [2]. Deep learning's popularity started to decrease in early 2000 as a result of a lack of massive data and powerful computers as compared to other machine learning tools [2].The outcomes of evaluating the performance of several algorithms on the same dataset can provide insight into comprehending the distinctive characteristics of each algorithm, how they vary from one another, and identify the most efficient object recognition technology for the given situation.

I. Literature Survey:

In recent time periods object detection has been an eminent topic for research. With the powerful learning tool and large datasets available deeper image features can be easily detected and studied. The goal of this study is to compile data on different object-detecting technologies and methods.

This technique is utilized by several scholars to enable relevant comparisons to be made possible to form conclusions and use them in object detection. The goal of a literature review is to gain an understanding of our work.

In another paper, the author proposed their research work to introduce tiny SSD. It's a Single shot detection real-time deep convolutional neural network aimed at embedded object detection. Tiny SSD is made up of a highly optimized non-uniform Fire sub-network stack, this feeds onto a highly irregular sub-network stack efficient auxiliary convolutional feature

layers based on SSD, particularly created to reduce model size while maintaining the effectiveness of object detection [3]. In their research, Fan et al. suggested an enhanced pedestrian identification system based on the SSD model of object detection. In this piece, they included the Squeeze-and-Excitation model as a further layer to their multi-layered.
The SSD model has a layer. The enhanced model made use of self-learning that went even improved the system's accuracy for small-scale pedestrian detection. Experiments on the INRIA dataset demonstrated good precision [4].
R. Shaoqing introduces a Region proposal network (RPN) that shares full-image convolutional features with the detection network. It predicts object bounds & abjectness scores at each position. Fast Region based CNNs take advantage of GPUs [5]. The basic knowledge of R-CNN, Fast R-CNN, Faster R-CNN was discussed in this paper. The RCNN method trains CNNs end-to-end to classify proposal regions into object categories or background. R-CNN mainly plays as a classifier & it doesn't predict object bounds. Its accuracy depends on the performance of the region proposal module. Fast R-CNN enables end-to-end detector training on shared convolutional features & shows accuracy & speed. Faster R-CNN system composed of two modules. The first module is fully convolutional network that proposes regions & second is the fast R-CNN detector that uses proposed regions [5].
Loss function for an image is defined as:

$$L(\{P_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(P_i, P_i^*) + \lambda\frac{1}{N_{cls}}\sum_i P_i^* L_{reg}(t_i, t_i^*)$$

Here, i is the index of an anchor in a mini-batch & P_i is the predicted probability of anchor I being an object [5].
Another research work done by researcher Kim C. discusses CNN with background subtraction to build a framework that detects and recognizes moving objects using CCTV (Closed Circuit Television) cameras. It is based on the application of the background subtraction algorithm applied to each frame. And for the practical experiments, they constructed datasets from various real-world CCTV cameras [6].
In the research work done by Joseph Redmon acquired that Unlike sliding window and region proposal-based techniques, YOLO technique sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance.
Fast R-CNN, a top detection method, mistakes background patches in an image for objects because it can't see the larger context. As compared to Fast R-CNN, YOLO makes less than half the number of background errors [7]. YOLO can be used to rescore Fast R-CNN detections and reduce the errors from background false positives, giving a significant performance boost. Finally, it is proved in the research paper that YOLO generalizes to new domains better than other detectors on two artwork datasets.[7]

II. Evolution of CNN:

Convolutional Neural Network (CNN): Convolutional Neural Network (CNN) is one of the most successful deep architectures as manifested by its remarkable achievements in many real-world applications. CNN is mainly used to analyze images. The state-of-the-art CNN architectures such as VGGNet, ResNet, and GoogLeNet, designed by experienced researchers, exhibited performance competitive to humans. However, crafting such powerful and well-

designed networks requires extensive domain knowledge and expertise in neural network design.[8].

The principal area of the brain processing visual sensory data is the visual cortex. It takes features out of images and detects structures and patterns to find items. It stands out characteristic is the presence of hidden convolutional layers. The filters are used by these layers to make patterns out of pictures. To produce the result, the filter moves across the image. Different filters identify various patterns. Filters in the initial layers can identify simple object patterns. Over time, they develop layers that make them increasingly complicated, as follows:[9]

1. **NEOCOGNITRON (1979-1980)-** It's the earliest precursor of CNNs. The concept of feature extraction, pooling layers & using convolution in a Neural Network was introduced & finally, recognition or classification at the end was proposed in the Noncognition. The process of feature extraction by S-cell & C-cells was repeated [18].
2. **LeNet-5(1988-1998)-**The name convolutional Neural Networks actually originated with the design of LeNet by Yan LeCum & team. It was developed between 1988-98 for the handwritten digit recognition task. The credit for the newer architecture of CNN's goes to ImageNet. Finally, in 2012, Alex with CNN architecture popular to these days named as AlexNet.It reduces error from 25.8% to 16.4%[18].
3. **AlexNet(2012)-** was the first winner of the ImageNet challenge and was based on a CNN and since 2012.AlexNet has 8 layers in total, trained on the ImageNet dataset. It introduces Rectified Linear Unit (ReLU) as an activation function. It has about 60 M parameters. It's about 90-95% computation but only about 5% of the parameters[18].
4. **ZFNet (2013)-** NFNet become winner of ImageNet LSRVC.Little changes are done in ZFNet irrespective of the ImageNet. Filter size was changed, and careful selection of hyperparameters. There was a significant decrease in the top 5 errors from 16.4% to 11.7%.
5. **VGGNet (2014)-** Invented by visual Geometry Group. It is a challenge for ImageNet, get a lower error rate on the ImageNet classification. Homogeneous architecture & smaller receptive fields were other key features in the design [18]
6. **GoogleNet (2014)-**It again focused on deeper networks but with the objective of greater efficiency to reduce parameter count, computation and memory usage. Inception named module was introduced. The solution to increasing the performance of the deeper model is ResNet[18]
7. **ResNet(2015)-** Kaiming He et.al from Microsoft Research came up with an idea of residual blocks which are connected to each other. A residual network is a stack of many residual blocks. Each block has 3×3 convolutional layers. ResNet won first place in all ELSVRC & COCO 2015competitions &continued a popular choice for several applications [18].

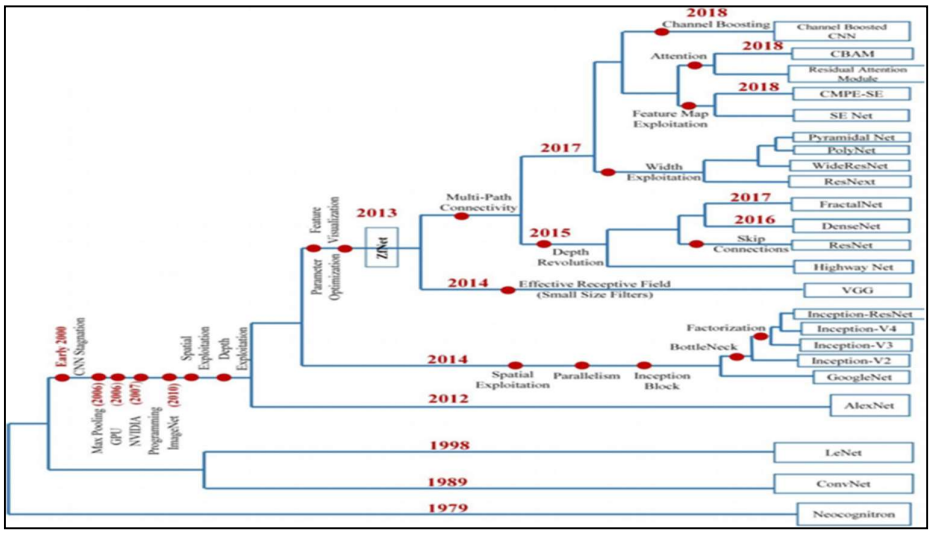


Fig 2: Evolution of CNNs from 1979 through 2018 [11]

Obtaining estimates & Analyses of the Effectiveness of the modules of CNN

In addition to the type of errors, to estimate the CNN modules & errors visually indicate the number of notches images that were fed to the input of each model.

In table1.shown the result of the considered neural networks with one model & one cut-out based on ImageNet images. [12]

Neural Network	Top-1	Top-5	No of Layers	No.of operations
AlexNet	39,7%	18,9%	8	70 M
ZFNet	37.5%	14.8%	8	70 M
VGGNet	25,60%	8,10%	19	155M
GoogleNet	29,00%	9,20%	22	10 M
Inception-v3	21,20%	5,60%	101	35 M
ResNet-152	18,38%	4,49%	152	65 M

Table-1: Result of the considered Neural Networks on ImageNet [12]

Datasets:

The most commonly used datasets used for image classification and detection are Microsoft COCO and PASCAL VOC. For review analysis, COCO is used as an evaluation metric and dataset. They applied different behaviours of analysis that leads to better precision but also for improving speed, performance, and accuracy [26].

For object detection tasks the use of computationally costly architectures and algorithms such as RCNN, SPP-NET (Spatial Pyramid Pooling Network) the use of smart data sets; datasets having varied objects, and images that have again various objects. That objects have become necessary dimensions. In the case of live video feed monitoring, the cost of image detection becomes too high. Recently more developments occur in the COCO data sets for training and classification [2]. The COCO dataset is a more popular and widely used dataset as per some research papers [2]. The classes used namely pattern Analysis, Modelling and Computational Learning Visual object classes, ImageNet, and SUN (Scene Understanding). The above-mentioned data sets vary hugely based on size, categories, and types. ImageNet was made to target a wider category where the number of different categories is powered. In one of the

modular approach SUN where the region of interest was based on the frequency of them occurring in the data set. Microsoft Common Object in Context is made for the detection and classification of the object in their classic nature [2].

I. Experimental setup

Hardware

The hardware comprised 8 GB DDR5 Random Access Memory, 1 TB Hard Disk Drive, 256 GB Solid State Drive, and Intel Core processor i5 8th Generation which clocks at a speed 1.8Ghz.

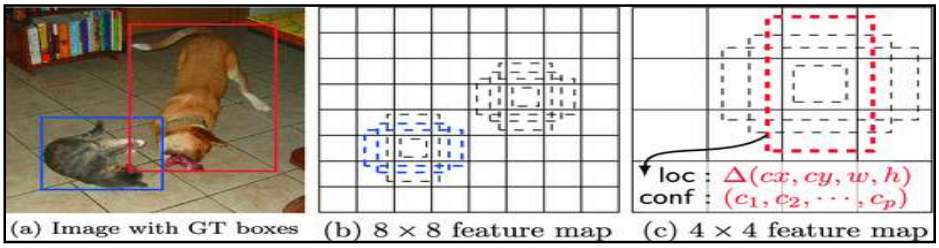
(Figs. 3, 4, 5, 6, 7, 8, 9, and,10).[36]

Software

The software configuration put to use is the Google Colab using an inbuilt engine called Python 3 Google Compute Engine Backend or Jupiter Notebook. It provides a RAM of 12.72 GB of which 3.54 was used on average. Also, it provides a disk space of 107.77 GB of which 74.41 GB was used which included the training and validation datasets. The hardware accelerator used was the synthetic GPU offered by Google Colab (Tables 2 and 3).

Existing Methodologies:

SSD: Due to the fact that SSD fully eliminates proposal generation and the subsequent pixel or feature resampling phases and incorporates all computing in a single network, it is simpler than approaches that call for object proposals.[8]. The fig 13. Bellow shows the SSD Model[32].



SSD framework. (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×88×8 and 4×44×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ((c1,c2,⋯,cp)(c1,c2,⋯,cp)). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss and confidence loss (e.g. Softmax).[8]

The SSD method uses a feed-forward convolutional network to generate a fixed-size collection of bounding boxes and scores for the existence of object class instances in those boxes. A non-maximum suppression step is then used to get the final detections [8]

In addition, compared to other ways, it is fairly simple. Because it fully eliminates feature resampling, which is a requirement for object proposals. By including all computation in a single step during the pixel and proposal creation network. Therefore, SSD can be easily integrated into systems that perform detection as one of their functions and are very simple to

train [8].

Compared to R-CNN [13] SSD has less localization error, indicating that SSD can localize objects better because it directly learns to regress the object shape and classify object categories instead of using two decoupled steps. However, SSD has more confusion with similar object categories (especially for animals), partly because we share locations for multiple categories. SSD is very sensitive to the bounding box size. In other words, it has a much worse performance on smaller objects than bigger objects. This is not surprising because those small objects may not even have any information at the very top layers [13].

Increasing the input size (e.g. from 300×300300×300 to 512×512512×512) can help improve the detection of small objects, but there is still a lot of room to improve. On the positive side, we can clearly see that SSD performs really well on large objects. And it is very robust to different object aspect ratios because we use default boxes of various aspect ratios per feature map location [13].

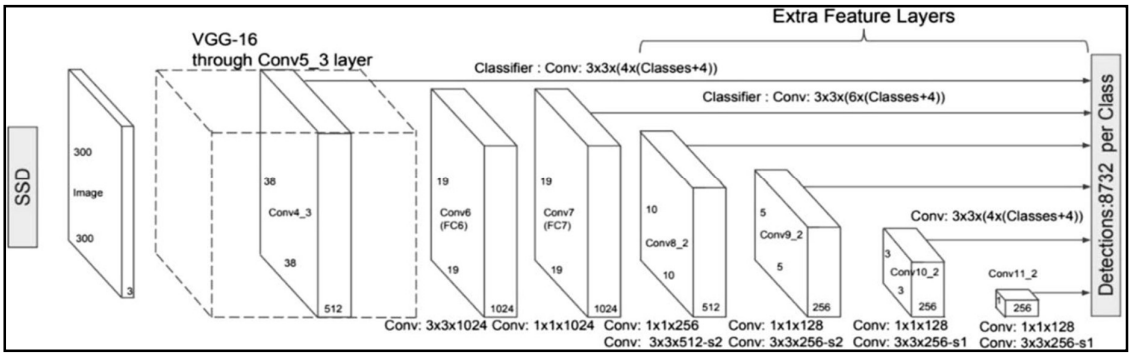


Fig.13 SSD Model[32]

Faster R-CNN:

Problem

How do you find multiple objects in an image with tight bounding boxes?

Solution

Use a (pretrained) Faster RCNN network. Faster RCNN is a neural network solution for finding bounding boxes of objects in an image.

The Fast RCNN algorithm, which was an advancement over the Fast RCNN, gave rise to the Faster RCNN algorithm. These algorithms all function similarly; A region proposer suggests potential rectangles that could have attractive images and determines what—if anything—can be seen there using an image classifier.

Faster RCNN trains the region proposal in parallel on the same feature map on which the image classification is done [19].

The object detection system, called Faster R-CNN, is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed regions.[5] The entire system is a single, unified network for object detection (Fig.12) Using the recently popular terminology of neural networks with ‘attention’ mechanisms, the RPN module tells the Fast R-CNN module where to look.

The algorithm of the original RCNN is as follows: [23]

1. Using a Selective Search Algorithm, several candidate region proposals are extracted from the input image. In this algorithm, numerous candidate regions are generated in the initial sub-segmentation. Then, regions that are similar are combined to form bigger regions using a greedy algorithm. These regions make up the final region proposals.[23]
2. The CNN component warps the proposals and vectorizes the extracted distinctive features output.
3. An SVM (Support Vector Machine) is used to recognize things of interest in the proposal using the retrieved features.

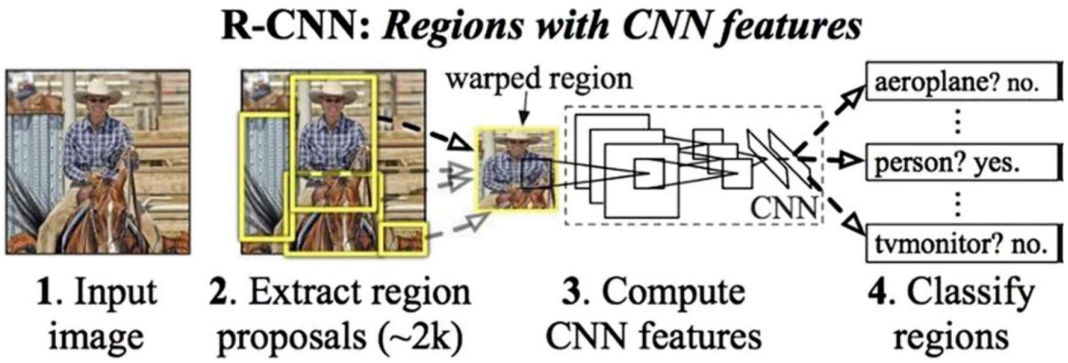


Fig. 12-1 R-CNN Model[33]

Faster R-CNN outputs numerous feature maps from a deep CNN after receiving an input image. Instead, these convolutional feature maps create region recommendations of the first raw image. Additionally, sliding windows and related techniques are replaced by a Region Proposal Network (RPN) for the development of region proposals [22]. Another one is RPN a deep fully-convolutional network with object bounding box prediction training. The objectness score (the likelihood that an object will be found in a particular location) at each simultaneously, the feature map grid's position. Fast RCNN is an algorithm used for object detection. It solves the drawback of RCNN [25].

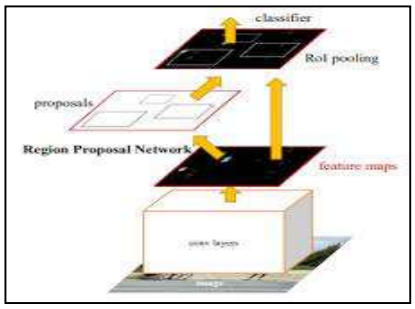


Fig.12-2Faster R-CNN [22]

There were numerous issues with this method. The training of the CNN takes a long time because it needs to categorize 2000 region ideas. This renders real-time implementation impossible because it would take about 47 seconds for each test image to complete. Additionally, because the Selective Search Algorithm is a fixed algorithm, machine learning could not be used. This can lead to the generation of candidate region ideas that are less

desirable [23].

It uses an approach similar to that of its predecessor, but as opposed to using region proposals, the CNN utilizes the image itself for creating a convolutional feature map, following which region proposals are determined and warped from it.

An RoI(Region of Interest) pooling layer is employed for reshaping the warped squares according to a predefined size for a fully connected layer to accept them. The region class is then predicted from the RoI vector with the help of a SoftMax layer [25].

Fast RCNN is faster than its predecessor it is not necessary to feed the CNN 2,000 suggestions as input to each execution. Convolutional processing is used to create only one feature map per image [25]. When compared to R-CNN, this algorithm demonstrates a significantly shorter training and testing time. However, it was noted that adding the regional proposal bottlenecks the algorithm severely, lowering its performance [5].

For determining the region Proposal Fast CNN and its predecessors both use the selective search algorithm which is a faster search algorithm for image detection [26].

Faster R-CNN did away with the requirement for its implementation because this is a very time-consuming technique and allowed the suggestions to be learned by the system. Similar to how Fast R-CNN works, a convolutional map is created from the image[26]. But a separate network replaces the Selective Search algorithm to predict proposals. Using RoI (Region of Interest) pooling these proposals are then reshaped and classified[26].

Complexity analysis:

Selective Search was used by both Region based Convolutional Neural Network (RCNN) and Fast-RCNN. One of the Greedy Algorithm is a Selective Search algorithm and Greedy algorithms don't always return the best result [34]. Also, it needs to run multiple times. However, RCNN runs selective search about 2000 times on the image. Fast-RCNN extracts all the regions first and runs selective search just once. This way it reduces time complexity by a large factor **Fig. 13** SSD model [32] **Fig. 12-1** R-CNN model [33]

Faster RCNN (FRCNN) removes the final bottleneck—Selective Search. It does so by instead using the Region Proposal Network (RPN). RPN fixes the regions as a grid of n ×n. It needs to run fewer number of times as compared to selective search [5]. As shown in the diagram above, FRCNN consists of Deep Fully Convolutional Network(DFCN), Region Proposal Network, ROI pooling, Fully Connected (FC) networks, Bounding Box Regressor and Classifier.

We will consider DFCN to be ZF-5 for consistent calculation [20]. First feature map, M of dimensions 256 × n × n is extracted from input image. [35]. Then it is fed to RPN and ROI.

RPN: There are 'k' anchors for each point on M. Hence, Total anchors = n × n × k.

Anchors are ranked according to score; 2000 anchors are obtained through non-Maximum Suppression [5]. The Complexity comes out to be O(N2/2).

ROI: Anchors get divided into H × W grid of sub-windows based on M. Output grid is obtained by max-pooling values in corresponding sub-windows. ROI is special case of spatial pyramid pooling layer used in SPP-net, with just one pyramid layer [25]. Hence, complexity becomes **O(1)**.

VOLOv3:

In morden era,YOLO(You Only Look Once) is one of the most efficient and accurate algorithms for object detection available nowadays. It is only possible because of a truly altered

& customized Darknet architecture [27].The first version of DarkNet was inspired by GoogleNet, to sample down the image & an image prediction tensor was used to get more accuracy. To decrease the no. of individual computations and make an analysis switter; the tensor is generated on the basis of similar structure & procedure which is also seen in the Region of Interest (ROI) that is pooled & compiled and that is used in faster R-CNN network. The generation utilized are architecture with just 30 convolutional layers from which only 19 layers are considered from Darknet-19 & extra 11 for detection of natural objects or objects in natural context as the COCO dataset & matrices have been used. It provides more accurate detection & with good speed. It fought with pictures of small objects & small pixels. But this is the drawback of version 1 & 2. YOLO version 3 has been the greatest & most accurate version of YOLO which has been used widely because of its high precision. This is possible because of multiple layers in the architecture.[28]

YOLOv3 makes use of the latest darknet features like 53 layers & it has undergoes & it has undergone training with one of the most reliable datasets called ImageNet. The layers used are from an architecture DarkNet-53 which is convolutional in nature. For detection, 53 layers were supplemented instead of the pre-existing 19 & this enhanced architecture was trained & instructed with PASCAL VOC. After adding so many layers (53 layers) the architecture maintains one of the best response times with accuracy. It is also very helpful in analyzing real video feeds because of its object detection techniques. This version is very useful in analyzing satellite imaging even for the defense department of some of the countries' previous versions because the previous version did not work well with the images in small pixels. The architecture works in 3 different layers which makes it more efficient but the process is slower.

Complexity Analysis:

The YOLO network is based on a systematic division of the given image into a grid. The grids are of 3 types. These grids undergo further divisions and They serve as separate images for the algorithm. YOLO utilizes boundaries called bounding boxes. Bounding boxes are the anchors for the analysis of an image. These boxes are essentially acknowledged as results even though thousands and thousands are ignored because of the low probability scores and are treated as false positives. These boxes are the exhibition of the laborious breaking down of an image into grids of cells [29-31].

YOLO uses a K-means clustering named algorithm to clutch the boxes among the training data for determining suitable anchor box sizes, these prior boxes are the guidelines for the algorithm. After receiving the abovementioned data, the algorithm looks for objects with symmetrical shapes and sizes. YOLO uses 3 boxes as an anchor so each grid cell puts out 3 boxes. Further predictions and analysis are based on these 3 anchor boxes. Some cases and studies involve the use of 2 anchor boxes leading to 2 boxes per grid cell [30]. The use of the K-means clustering algorithm gives exponential time complexity O(kd) where k is the number of images and d is the dimension of the images. The creators have made YOLOv3 the fastest image detection algorithm among the ones mentioned in the paper and this will be after a thorough and stable optimization technique.

Result and Discussion:

Fig 3: YOLO Architecture[13]

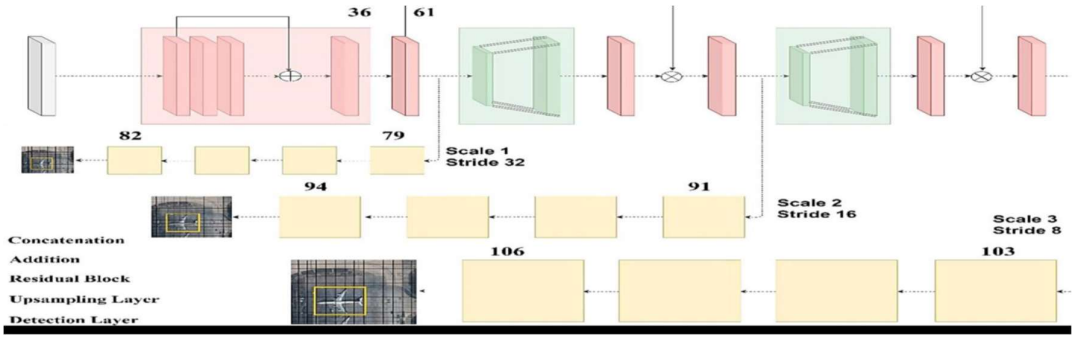


Fig: 4 YOLO Model ConvNet [14]

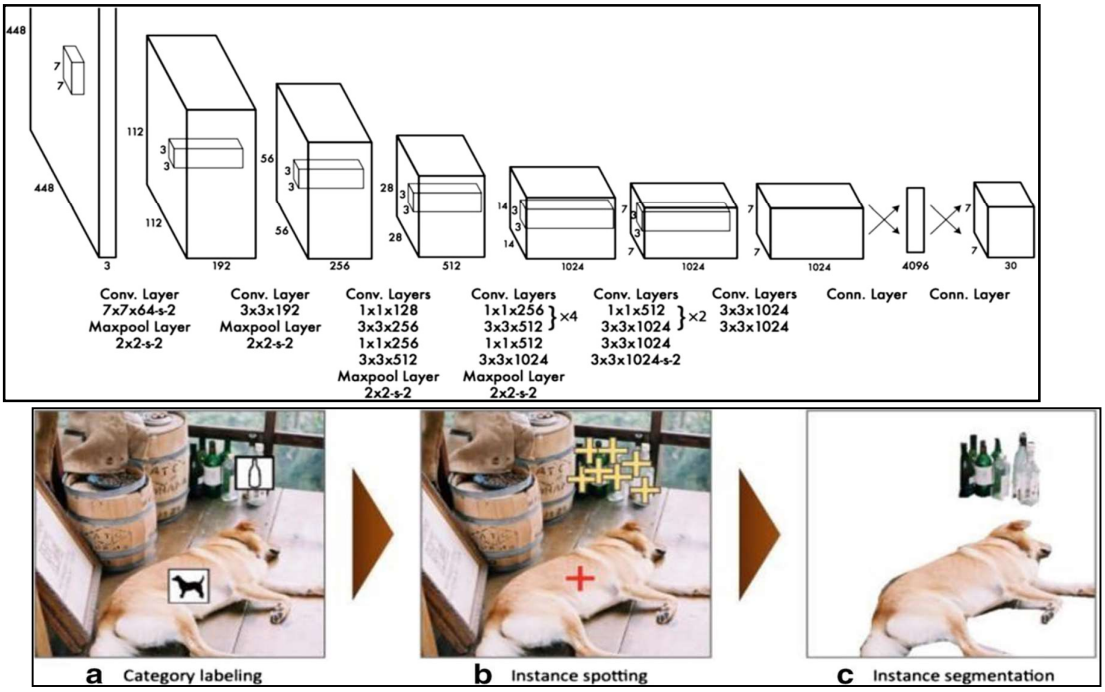


Fig 5: Annotation pipeline [15]

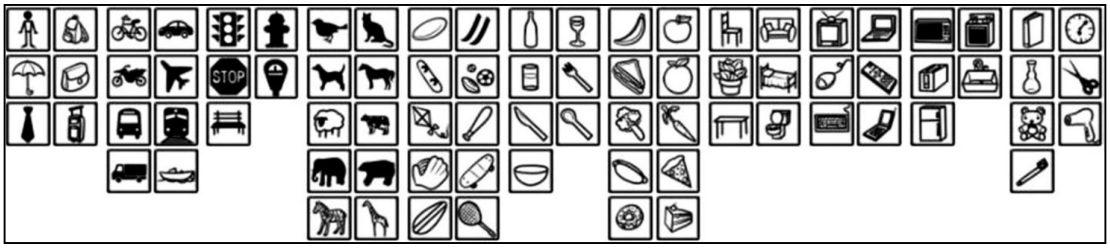


Fig:6 Categories of images[10]

Vehicles	Household	Animals	Other
Aeroplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/monitor	Sheep	
Train			

Fig: 7 The classes of objects considered in the challenge [16]

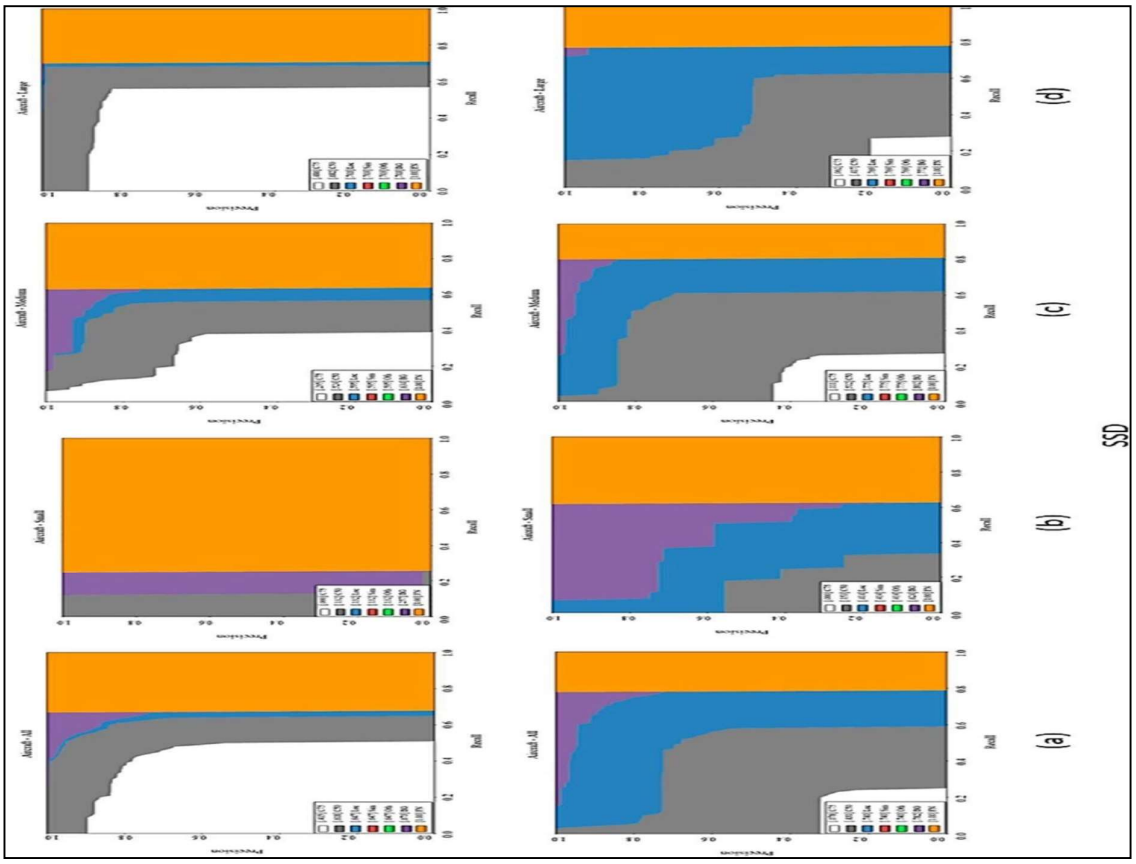


Fig 8: Graph for SSD [13]

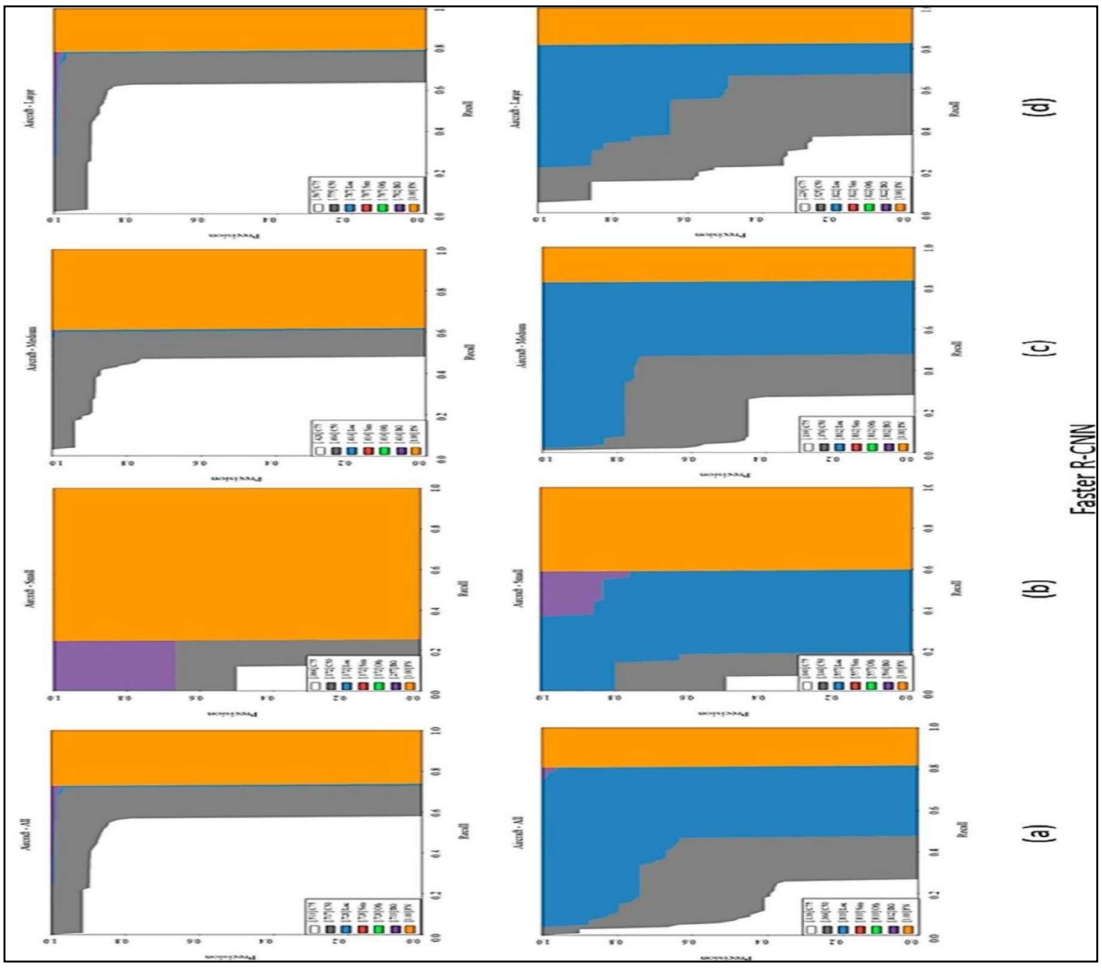


Fig:9 Graph for Faster R-CNN [13]

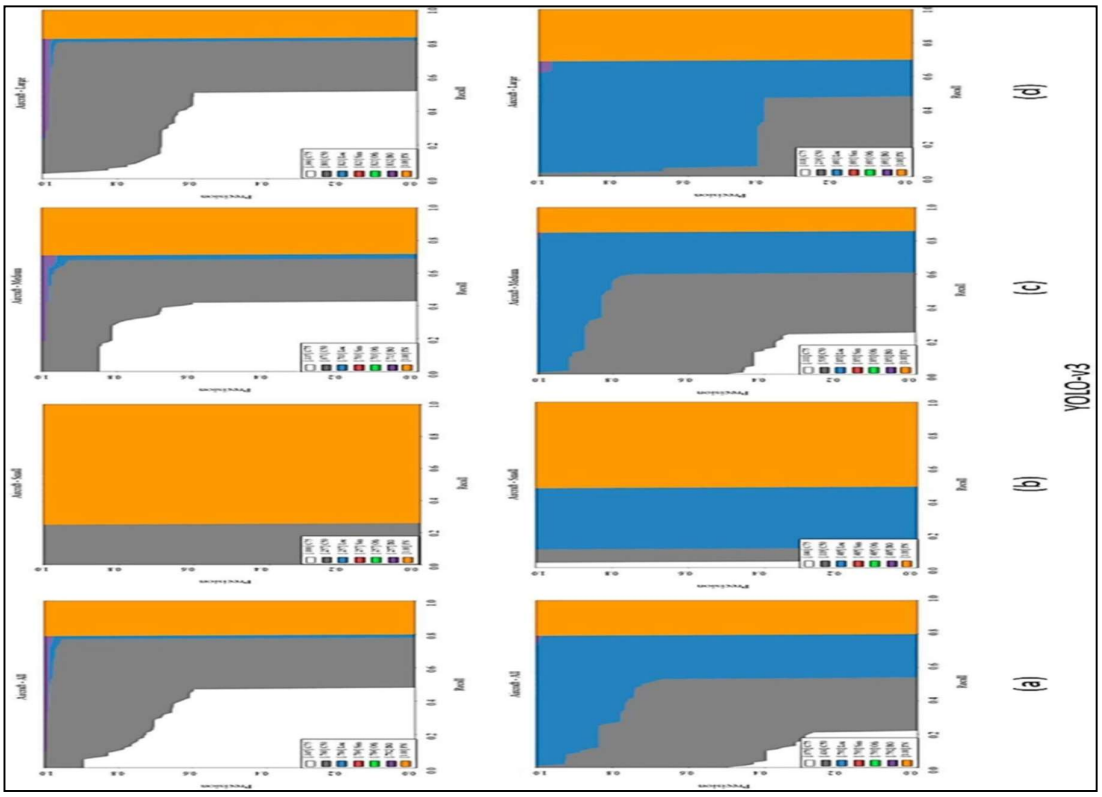


Fig.10: Graph for YOLO-V3 [13]

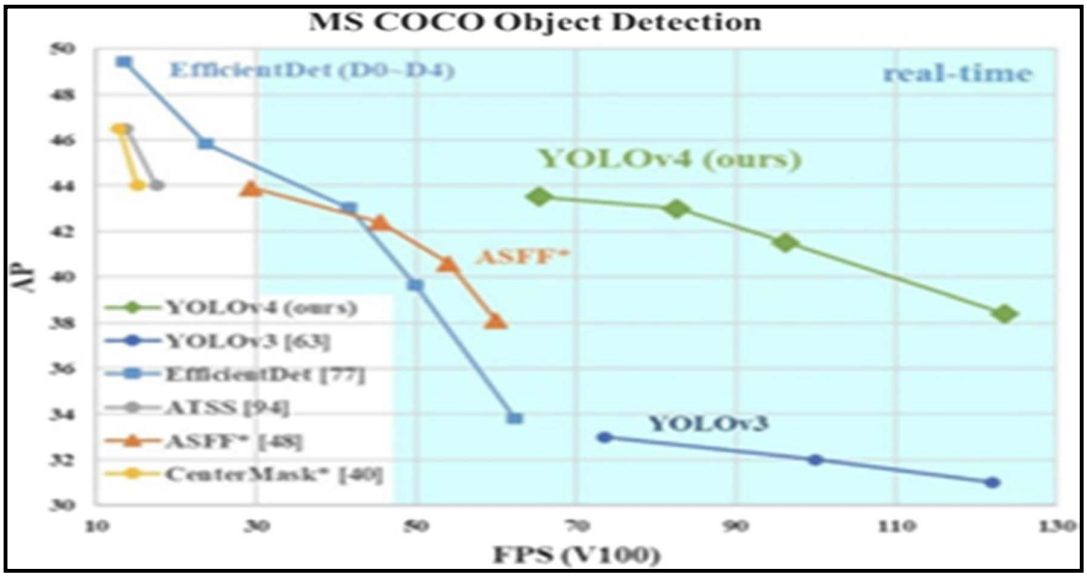


Fig.11: Compared with YOLOv3, the new version of AP (accuracy) and FPS (frame rate per second) are improved by 10% and 12%, respectively [17]

Table 2: COCO metrics [10]

```
Average Precision (AP):
   AP                  % AP at IoU=.50:.05:.95 (primary challenge metric)
   AP^IoU=.50          % AP at IoU=.50 (PASCAL VOC metric)
   AP^IoU=.75          % AP at IoU=.75 (strict metric)
AP Across Scales:
   AP^small            % AP for small objects: area < 32^2
   AP^medium           % AP for medium objects: 32^2 < area < 96^2
   AP^large            % AP for large objects: area > 96^2
Average Recall (AR):
   AR^max=1            % AR given 1 detection per image
   AR^max=10           % AR given 10 detections per image
   AR^max=100          % AR given 100 detections per image
AR Across Scales:
   AR^small            % AR for small objects: area < 32^2
   AR^medium           % AR for medium objects: 32^2 < area < 96^2
   AR^large            % AR for large objects: area > 96^2
```

Conclusion:

The research paper presents the comparative study of several image processing algorithms using deep learning. As it has a large dataset; It will help the researcher to find out the more accurate result from the data. As compared to Faster CNN, YOLO V3 provides more accuracy & result. The field of Computer vision is blessed with a large amount of labeled data. In the future, we will continue looking for ways to bring different structures and sources of data together to make stronger models of the visual world.[37]

References:

1. Pathak AR, Pandey M, Rautaray S. Application of deep learning for object detection. Procedia Computer Sci.2018;132:1706–17.

2. LUBNA AZIZ, MD. SAH BIN HAJI SALAM,USMAN ULLAH SHEIKH, AND SARA AYUB Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review IEEE, September 28, 2020:170461-170495.

3. Womg A, Shafiee MJ, Li F, Chwyl B. Tiny SSD: a tiny singleshot detection deep convolutional neural network for real-time embedded object detection. In: 2018 15th conference on computer and robot vision (CRV). IEEE; 2018, p.95-101.

4. Fan D, Liu D, Chi W, Liu X, Li Y. Improved SSD-based multi-scale pedestrian detection algorithm. In: Advances in 3D image and graphics representation, analysis, computing and information technology. Springer, Singapore; 2020, p.109–118.

5. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49.

6. Kim C, Lee J, Han T, Kim YM. A hybrid framework combining background subtraction and deep neural networks forrapid person detection. J Big Data. 2018;5(1):22.

7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016, pp. 779–788.

8. Bakhshi, A., Chalup, S., Noman, N. (2020). Fast Evolution of CNN Architecture for Image Classification. In: Iba, H., Noman, N. (eds) Deep Neural Evolution. Natural Computing Series. Springer, Singapore. https://doi.org/10.1007/978-981-15-3685-4_8

9. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE.1998;86(11):2278–324.

10. COCO. [Internet]. https://cocodataset. org/#explo re. Accessed 28 Oct 2020.Table-2 fig 6

11. Khan, A., Sohail, A., Zahoora, U. et al. A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev 53, 5455–5516 (2020). https://doi.org/10.1007/s10462-020-09825-6 fig-2

12. Arsenov, A., Ruban, I., Smelyakov, K., & Chupryna, A. (2018, November). Evolution of Convolutional Neural Network Architecture in Image Classification Problems. In ITS (pp. 35-45). Table:1

13. Alganci U, Soydas M, Sertel E. Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. Remote Sensing. 2020;12(3):458.(Fig:3,8)

14. Jiang R, Lin Q, Qu S. Let blind people see: real-time visual recognition with results converted to 3D audio. Report No.218, Stanford University, Stanford, USA; 2016.(fig.4)

15. Palop JJ, Mucke L, Roberson ED. Quantifying biomarkers of cognitive dysfunction and neuronal network hyperexcitability in mouse models of Alzheimer's disease: depletion of calcium-dependent proteins and inhibitory hippocampal remodeling. In: Alzheimer's Disease and Frontotemporal Dementia. Humana Press, Totowa, NJ; 2010, p. 245–262. (fig.5)

16. Everingham M, Eslami SA, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes challenge: a retrospective. Int J Computer Vision. 2015;111(1):98–136. (Fig. 7)

17. Bochkovskiy A, Wang CY, Liao HYM. Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020. (Fig:11)

18.https://medium.com/the-pen-point/evolution-of-convolutional-neural-network-architectures-6b90d067e403

19. Deep Learning Cookbook Practical Recipes to Get Started Quickly by Douwe Osinga,O'Reilly publication p.NO 137-139 2018-05-23: First Release

20. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European conference on computervision. Cham: Springer, 2014, p. 818–33.

21. Reza Z. N. (2019). Real-time automated weld quality analysis from ultrasonic B-scan using deep learning (Doctoral dissertation, University of Windsor (Canada)).

22. Srivastava, S., Divekar, A.V., Anilkumar, C. et al. Comparative analysis of deep learning image detection algorithms. J Big Data 8, 66 (2021). https://doi.org/10.1186/s40537-021-00434-w

23. Shen X, Wu Y. A unified approach to salient object detection via low rank matrix recovery. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012, p. 853–60.

24. Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., & Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS journal of photogrammetry and remote sensing, 145, 3-22.

25. G. R. Fast r-CNN. In: Proceedings of the IEEE international conference on computer vision; 2015, p.1440–8.

26. Schulz H, Behnke S. Deep learning. KI-Künstliche Intelligenz. 2012;26(4):357–63.

27. Redmon J, Farhadi A. Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767; 2018.

28. Alganci U, Soydas M, Sertel E. Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images. Remote Sensing. 2020;12(3):458.

29.Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015, p. 1–9.

30. Zhao L, Li S. Object detection algorithm based on improved YOLOv3. Electronics. 2020;9(3):537.

31. Syed NR. A PyTorch implementation of YOLOv3 for real time object detection (Part 1). [Internet] [Updated Jun 30 2020]. https:// nrsyed. com/ 2020/ 04/ 28/a- pytor ch- imple menta tion- of- yolov3- for- real- time- object- detec tion- part-1/.Accessed 02 Feb 2021.

32. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: single shot multibox detector. In: European conference on computer vision. Cham: Springer; 2016, p. 21–37.(Fig.13)

33. Schulz H, Behnke S. Deep learning. KI-Künstliche Intelligenz. 2012;26(4):357–63.(fig.12-1)

34. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW. Selective search for object recognition. Int J Computer Vision.

35. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: European conference on computer vision. Cham: Springer; 2016, p. 630–45.

36. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014) (Figs. 3, 4, 5, 6, 7, 8, 9, 10, and 11).

37. Joseph Redmon∗×, Ali Farhadi∗†× YOLO9000:Better, Faster, Stronger The University of Washington∗, Allen Institute for AI†, XNOR.ai×2017 IEEE Conference on Computer Vision and Pattern Recognition