

Various object detection algorithms and their comparison

Debani Prasad Mishra¹, Kshirod Kumar Rout¹, Sivkumar Mishra², Surender Reddy Salkuti³

¹Department of Electrical Engineering, International Institute of Information Technology Bhubaneswar, India

²Department of Electrical Engineering, Centre for Advanced Post Graduate Studies, Biju Patnaik University of Technology, Rourkela, India

³Department of Railroad and Electrical Engineering, Woosong University, Daejeon, Republic of South Korea

Article Info

Article history:

Received Jan 22, 2022

Revised Sep 13, 2022

Accepted Sep 30, 2022

Keywords:

Conventional neural network

Decision review system

Region of interest

Single shot detector

Transfer learning

ABSTRACT

This paper presents a detailed and comparative analysis of various object detection algorithms. The challenge of object detection is taken care of while studying various algorithms. Throughout the year various methods have been discovered in this field, each having its advantages and drawbacks. This paper aims at providing the systematic study of all the popular algorithms including the conventional ones. Although many methods and techniques come up each year and each of them having superiority other the previous models but at the same time even complexity increases. In this paper, some famous and basic methods of object detection and tracking are discussed. Using these developed techniques good results can be obtained and also the comparison of the efficiency of all the models can be done. Real-time applications and the outcomes are also discussed.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Surender Reddy Salkuti

Department of Railroad and Electrical Engineering, Woosong University

17-2, Jayang-Dong, Dong-Gu, Daejeon-34606, Republic of Korea

Email: surender@wsu.ac.kr

1. INTRODUCTION

Object detection is one of the most important applications of computer vision using neural networks. Object detection is a two-step process namely classification and localization. Classification is a process in which an algorithm identifies or classifies a single image or a set of images and gives them labels. Localization is a process in which an algorithm locates a particular object in the image given and also draws a bounding box around the located object. Object detection is different from image recognition in a way that image recognition only assigns a label to the image whereas object detection first draws a bounding box around an image and then assigns a label. Throughout the years several types of research have been carried out in the field of object detection and many different algorithms have been discovered namely conventional neural network (CNN), regions with conventional neural network (R-CNN), single shot detector (SSD), and transfer learning [1]. Object detection is used in many real-life situations like counting people in crowds [2] and intrusion detection in private and highly secured areas [3]. The primary target of the object detection models is to locate specific targets in an input image and further the movement of the target in the successive frames can be tracked. When a video is played it is nothing more than but sequence of images so, when tracking an object in a frame, the movement of the object in the coming successive frames is also tracked.

Object detection would help in training self-driving cars, detect intruders in highly secure areas, detect unauthorized movements [4], recognize people, and the gaming industry also finds its application as this object detection model is already implemented in gaming consoles like X-Box 360 in which games can be played by human movements. One of the earliest applications of object detection is OCR that is optical character recognition. Here an input image is given and the algorithm tries to analyze the text present in it. With much

of the evolving sophisticated models for classifying, users have started using it in reading the registration numbers of the vehicles written in their number plate [5]. An object detection technique called YOLO is popularly used in this model. Object detection finds its application in identifying objects, it has wide areas of application and the objects being tracked vary in different scenarios. It is used to track pedestrians [6] illegally crossing the roads or pedestrians walking on the side of the road [7]. Here also a CNN-based system in our model is used. Here a lane detection model is also implemented which shows the available lanes and the vehicles on those routes. The model classifies the objects in the surrounding into various categories like motorbikes, pedestrians, and crossings [8].

The object detection algorithm is also used to find images online similar to the image given input to the model [9]. In other words, it enables search by image. Many of the large e-commerce websites have already implemented this feature. Object detection techniques are used to study and classify medical data. This has a wide variety of applications like cancer detection from radiologist reports [10]. Different models are trained with different sets of input data for better prediction accuracy. Similar training models are used in X-ray images of lungs of covid-19 patients for classification of the patients. The model is trained with the X-ray data-sets of lungs of covid-19 patients [11]. It is also used in skin disease detection, which helps in the early detection of the infection and thus helping eradicate the problem before spreading out more. It uses a variance of the CNN model [12].

One of the recent breathtaking examples and applications of object detection is self-driving cars [13]. In self-driving vehicles, various parameters are calculated and are predicted for the safe movement of the vehicle and also of others on road. This involves predicting pedestrians, motorcycles, pavements, etc. Object detection is being also now used in generic cars to alert drivers when they tend to fall sleepy during drives [14], this has been already used by BMW. Another example is video surveillance through object detection [15]. It has great advantages in tracking the movement of people and vehicles in public places and also it is very useful for CCTV coverage. The other groundbreaking research in the field of Object detection is facial recognition technology [16]. It is extremely useful for attendance at schools and universities and also for safety purposes like having face unlock in smartphones and payment systems and also in banks.

Object detection and object tracking form the backbone of the decision review system (DRS) used in cricket. Here we analyze the images taken in real-time to track the path being followed by the ball. It involves taking a high-quality stream of images and applying appropriate object detection models to find the ball after segregating the surrounding. After the ball is tracked, successive frames are analyzed to find the path being followed by the ball [17]. The sliding window is the widely used model for ball tracking. This ball track is also applied in other sports such as basketball where this model is utilized to track the passing of the basketball among the players. Mostly this YOLO is their underlying architecture [18]. With globalization and industrialization, low cost and efficient bulk production of the products has become the key to profit and lead the manufacturing field. They can be further used to find cracks or damages on the body of the product. So, object detection models have increased the efficiency in sorting and removing defective items [19].

With urbanization protecting state-owned land has become a difficult task. But with object detection and deep learning on satellite images, we can put a check on those illegal encroachments. Here, we take satellite images and apply different object detection and deep learning models to find any illegal constructions or encroachment lands that are demarcated as state-owned lands. This technique also helps getting an idea of the forest cover of an area, during disasters and calamities these also help in finding the affected areas [20]. At times, object detection methods are used to identify plant disease.

The transfer learning model of object detection is being used in this case [21]. Image-based plant disease classification is used to identify the diseased leaves out of the healthy leaves and it uses pre-trained weighted CNN which is fast and easy to implement. Object detection models have been used in defense since long back, one of the most important applications of object detection in defense is its application in unmanned aerial vehicles. It is primarily used in border surveillance. Unmanned aerial vehicles continuously patrol in the air space near the disputed areas or country borders. These unmanned aerial vehicles use object detection models to look for illegal movement across restricted sites or country borders and report to the station [22].

2. RESEARCH METHODS/ALGORITHMS

In this section, some of the widely used object detection algorithms, their efficiency in different scenarios, and their modeling are discussed. It is important to do a comparative study of these newly emerging object detection algorithms as there is a trade-off between accuracy and complexity, choosing a model solely depends on the requirements. Some of these implementations use variations of CNN whereas in certain cases YOLO and SSD are being used.

2.1. Convolutional neural network (CNN)

CNN is built by neurons that have their weights (w) and biases (b). Each neuron in a network receives some inputs, and after that does a dot product. The single differentiable score function is still shown by CNN. It is the neural network which is a combination of Convolutional layers followed by max pool layers and then fully connected layers shown in Figure 1.

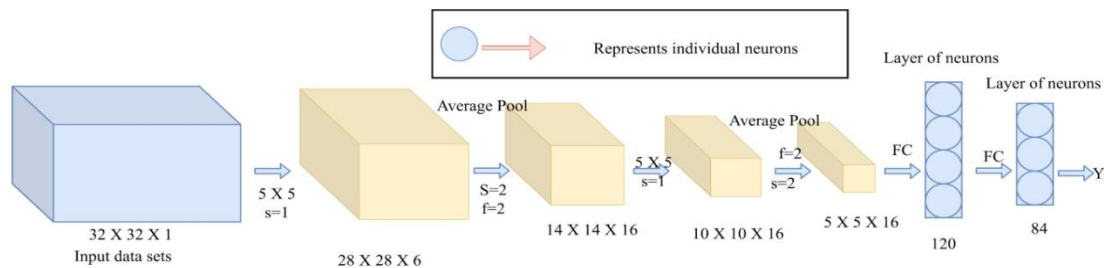


Figure 1. CNN architecture

These convolutional layers also use filters for reducing their size and these filters learn from the experiences. Each filter is of small height and width but with the same depth as that of the input image. Now we will implement convolution on a real-time image in a matrix form using filters with a pixel size of $(24 \times 24 \times 3)$. One can take the filter size $(3 \times 3 \times 3)$ or 3 can be replaced with 5 and 7 but the no of channels will be equal to the no of channels in the input image. Stride as a step in a forward pass to slide a filter through the whole input image and compute (patch) dot (weights of filters) is used. Sliding the filters through the input image, an output 2-D image is obtained and so all the number of filters is put out. Thus all filters of the image will be learned by the network. From figure 1, the architecture shows how the average pool, convolutional, strides (S), padding and filters (f), flattening are used to convert a layer into a fully connected layer without losing any important features of the image. This paper talks about the convolution layers, and every convolutional layer to change the size of the processing layer. Mathematical differentiable functions are usually used to increase or decrease the size.

2.1.1. Common type of layers

An example of a large image $(32 \times 32 \times 3)$ is explained next:

- Input layer: This layer has the $(32 \times 32 \times 3)$ matrix size, where 3 is an input image channel like R, G, B.
- Convolution layer: gives the dot product of input image patches and filters if an input image is of size $(32 \times 32 \times 3)$ and 8 filters, then the size of the output image will be $(32 \times 32 \times 3)$.
- Activation function layer: Any activation function like RELU, Sigmoid, etc can be used on every part of the input image matrix. In RELU the maximum value of the function is obtained in the x-direction, and the size of the entire image remains constant even after applying the activation function. Tanh, Leaky RELU, Sigmoid, etc. The image size remains the same and the output image size is $(32 \times 32 \times 12)$. Here Sigmoid function [23] is,

$$S(x) = 1/(1 + e^{-x}) \quad (1)$$

- Pool layer: It reduces the size of the input image with a huge difference. Every CNN layer with an example is explained next:
 - Convolutional layer: this is the latest convolutional version of the sliding window. Now only convolutional and max-pooling layers for further implementation are used. As an example, an input image of matrix $(28 \times 28 \times 3)$ sizes can be taken. Using the sliding window approach, this image is pass to the above CNN more than 4 times, where the sliding window does cropping the part of the input image matrix of size $(26 \times 26 \times 3)$ and again passes it across the CNN. But without using this, the full image (with shape $(28 \times 28 \times 3)$) directly into the ConvNet is used. These methods can give an output matrix of shape $(4 \times 4 \times 4)$. Now output matrix with the maximum cropping can be obtained [24].
 - Pool layer: The pooling layer is of two types max pool and average pool. But mostly max pooling is used because it focuses on the most effective features. Max pooling, the kernel of size $(n \times n)$ can be used. And for every position of the output matrix, the maximum value of the input image is taken after using the max pool kernel for example (2×2) , and then fill the value in the output matrix. Along with the filter we

also use stride and padding stride denotes stepping to move in the network layer. It is one if we don't specify any value of stride. Mostly after applying all these things we get that the size of the input is larger than the input. For equalizing the output and input, padding comes into the picture.

- Fully connected layer: feed-forward neural network Layer is nothing but it is simply a fully connected layer that comes in the last in the network. It takes the input Pooling or Convolutional Layer to the fully connected layer, it is done flattened then it is useful for the feed-forward network.

CNN is used as a base in the R-CNN model because of some feed-forward or fully convolutional layers which take an input image with the size of 224 by 224 pixels spatial pyramid in which is always made on the peak of the region of interest (ROI). In this, the region is divided into four cells with two by two grids. The region is divided into 16 cells on four-by-four grid. Average pooling is applied to each cell. So, feature vectors for all cells are concatenated and then passed as input to the fully convolutional layer. So at this time, we can apply the convolutional layers as an input image but per one on base CNN. For each sliding window, apply the best pooling layer to extract the feature from fully convolutional layers and likewise, we compute convolutional features for each of the 3,000 object proposals. And this takes us to the exponential increase in the processing speed [25]. In fast R-CNN, we use the ROI pooling layer or ROI layer. It's a simplified form of the SPP layer, here we have only one pyramid layer. In fast R-CNN, we have Region of Interest Pooling and two more modifications. First, we use a softmax classifier in place of an SVM classifier. And then we use multi-task training to train classifiers and bounding box regressors simultaneously. Fast R-CNN walks us forward. From the convolutional feature map, we can extract the feature vector using the ROI pooling layer. Detector precision can be improved by using multi-task learning. One can also get examples from different images. But we cannot do it in a fast R-CNN because computations are really expensive. In this Region of Interest (ROI) because of large pooling we get features from the whole image [26].

Faster R-CNN is the upgraded version of R-CNN with the ROI. In faster R-CNN, all the problems of R-CNN have been solved. We have removed the dependency from the external hypothesis generation from R-CNN. Now we can detect objects in a single pass with a single neural network (SNN). It is a simple fully convolutional network. It works as a proposal generator like a Fast R-CNN. We can extract the information of finer localization from localization objects with RPN. It is a small network. This small network of RPN takes its input and then usually N windows equals 3. RPN simultaneously classifies the corresponding region as an unknown object. We can get the position of the sliding window which provides localization information with reference to the image. Now the speed is significantly improved because we have removed the dependency on the external proposal generation method, so Faster R-CNN, now we know that each RPN will take a different convolution layer so the receptive field will not be of the same size. This faster R-CNN method will improve the detection of any size of the object. From Figure 2, it can be concluded that faster R-CNN is faster than R-CNN and Fast R-CNN.

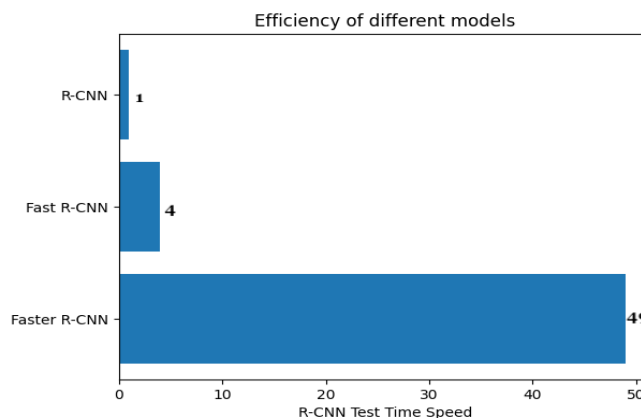


Figure 2. Comparison of R-CNN, Fast R-CNN, and Faster R-CNN

2.2. YOLO

Joseph Redmond and his team presented the earlier models of YOLO in 2015 [27]. YOLO stands for You Only Look Once. It is one of the finest object detection algorithms having applications in real-time images and videos. The algorithm first predicts multiple bounding boxes and assigns some weights to them and then combines those predicted boxes into one intelligently using the weights assigned. YOLO uses only one single neural network for the whole image. The whole image is taken into consideration at test time, so its predictions

are set and weights are assigned globally. Models before this are performed the above-mentioned tasks in several intermediate stages. These intermediate steps are usually difficult to modify as per our needs and optimize the models as per the demands. The overall architecture of the YOLO model is depicted in Figure 3. In the architecture, our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1X1 convolutional layers reduce the feature space for the preceding layer. We keep the convolutional layers on the ImageNet classification task at the half resolution (224×224 input image) and then double the resolution for detection.

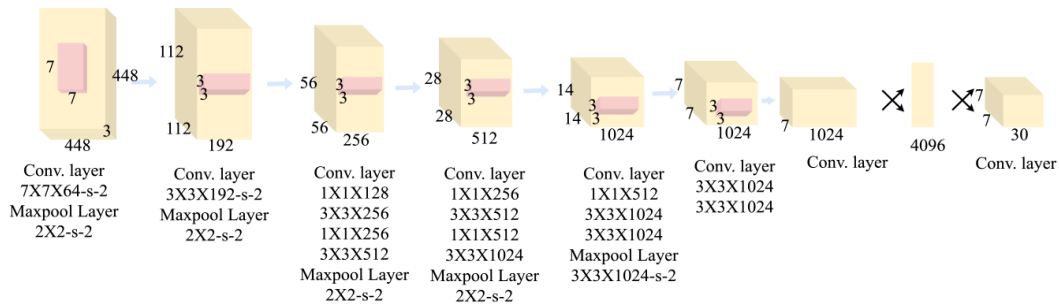


Figure 3. The architecture of the YOLO network

The training process has been described in the following ways. We have to first train the first 20 convolution layers using data-sets like ImageNet or COCO and use an image size of (224×224). After the first iteration, increase the size of the image to (448×448). Then train the entire neural network for an epoch of more than 130 and use a batch size of either 32 or 64 depending upon our hardware. Initially, for the first round of epochs, the learning rate has to be slowly raised by a factor of 10. After training for about half of the total epochs, start decreasing it. The method of data augmentation is adopted by random scaling the input and by randomly adjusting the exposure and the saturation, and it is shown in Figure 4.

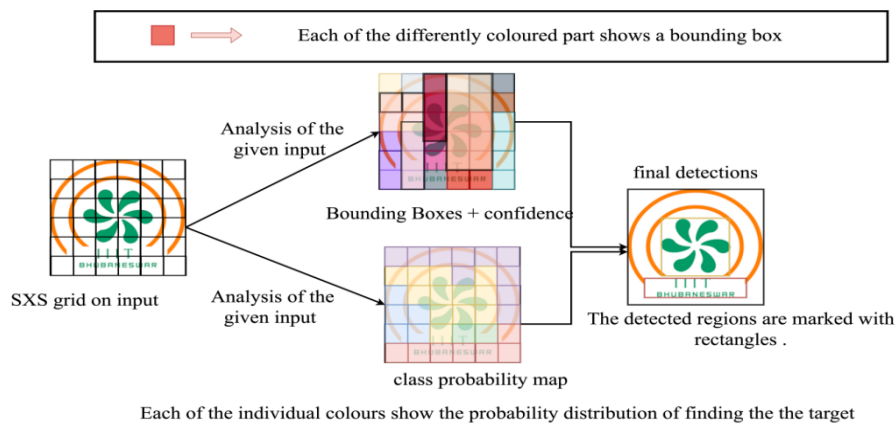


Figure 4. An illustration showing how the YOLO algorithm works

2.3. Transfer learning

Transfer learning is a type of deep learning technique in which a model (say A) is trained with a large available data set, then the knowledge gained after the model in classifying other models as illustrated in Figure 5. The working of this technique can be understood by considering the model, we have a neural network that has numerous intermediate layers, and this model is originally used for image classification by using the existing large data set [28]. Initially, after training the model for image classification the model learns a lot of things as in input to the model we have given the images vector and the classification vector for learning.

One of the classical application of transfer learning is that we can use that data-sets of animals and train that to classify them between cats and dogs [29]. The usual method adopted in transfer learning is that we

train the model using Image net data sets and then we fine-tune the data as per our demand of the scenario. At the implementation level usually, the last layer of the neurons is replaced and a new layer of neurons is introduced for changing the output. If needed by the designer then the intermediate layers can also be changed and modified. This parenting model is often considered as a pre-training model, it is way better than initializing the training model with random values as we do in most cases. In some models, we can even add more than one layer of neurons towards the last. In this model, the intermediate layers act as feature extracting layers, this layer in fine-tuning our input data-sets. The further training of a model is usually referred to as fine-tuning of the model. This model of transfer learning should be used when the dataset for the second classification is less but we have a large amount of data-set for the primary model training.

One of the most popular models of transfer learning residual networks (ResNet). This model also gave new hope in the research field of computer vision as its critical design helped in the deep training of the model. There are more than 150 layers in this model. This model due to architecture and large input data sets is highly useful in object detection models.

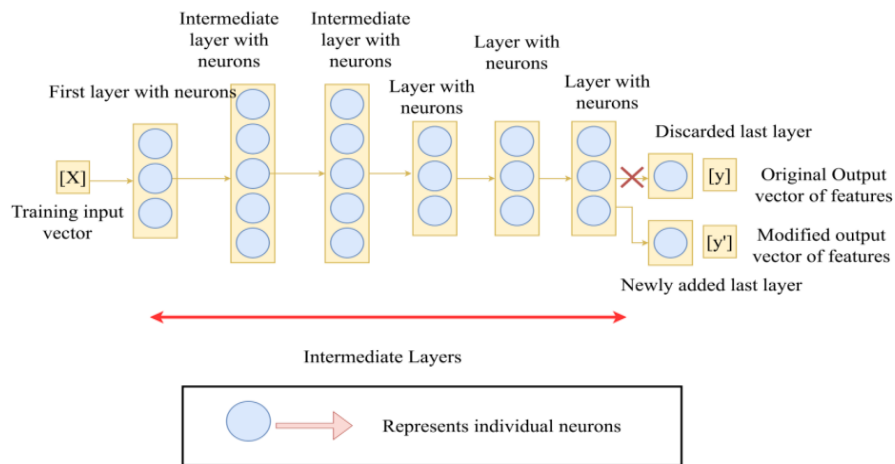


Figure 5. The architecture of a transfer learning model

2.4. Single shot detector (SSD)

SSD is used as a multi-box detector for the localization in the image [30]. From Figure 6, we can understand how can build the SSD using VGG-16 and some extra feature layers. First, we take an input image as input on that image we use the VGG-16 network and add four extra layers and fully connected layer we are now able to make SSD for multi-box detection in a single image. We have a fixed number of bounding boxes for an image and it has a fixed aspect ratio, image, and width depending on whether you are using it on a human or a car object for both the orientation will be different.

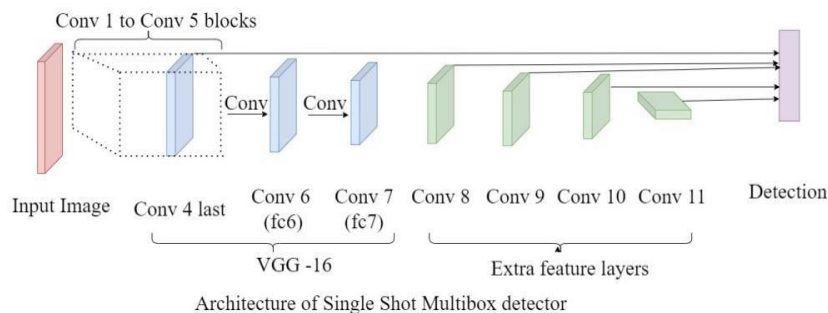


Figure 6. The architecture of SSD multi-box detector

To obtain good accuracy on the train and validation data-set we use data augmentation and Inference time. We have two types of SSD models SSD300 it means (300×300) input image with low resolution and SSD512 (512×512) input image with good resolution, and efficiency.

3. RESULTS AND DISCUSSIONS

Comparison of object detection algorithms has been presented in Figure 7. As we see in Figure 7(a) that YOLO is the fastest of all the algorithms as it executes at much higher frames per second (fps) than the rest of the algorithms. SSD comes at second with respect to speed. According to our expectation, transfer learning using ResNet should have been the fastest algorithm since it contains some pre-trained weights and uses fewer data and also we don't have to train the entire neural network but that was not the case. To achieve better accuracy in transfer learning technique we have to unfreeze the model, ie, we have to train the entire neural network so it hampers its speed of execution but it is one of the best object detection techniques. Fast R-CNN and Faster R-CNN due to their complex structure and hierarchical implementation are relatively slower than the rest of the algorithms. We usually use YOLO as a single-stage detector. We know that it is faster but lesser accurate than multi-stage detectors. It takes the image as a regression problem and learns the coordinates of the bounding box.

From Figure 7(b), we can conclude that SSD is the most accurate model followed by ResNet. Fast R-CNN and Faster R-CNN are also better than YOLO but they are relatively slow and takes much more time for the same tasks. For object detection tasks, speed and accuracy both matter. If our target object is a still photo then speed is not the factor but for real-time object detection and moving objects, speed is also an important factor. If accuracy is the goal then we should use faster R-CNN. But if speed is the goal, we should use YOLO. And if we need the model with good speed and accuracy on a complex data set then we should go with either SSD or transfer learning depending upon the nature of the data.

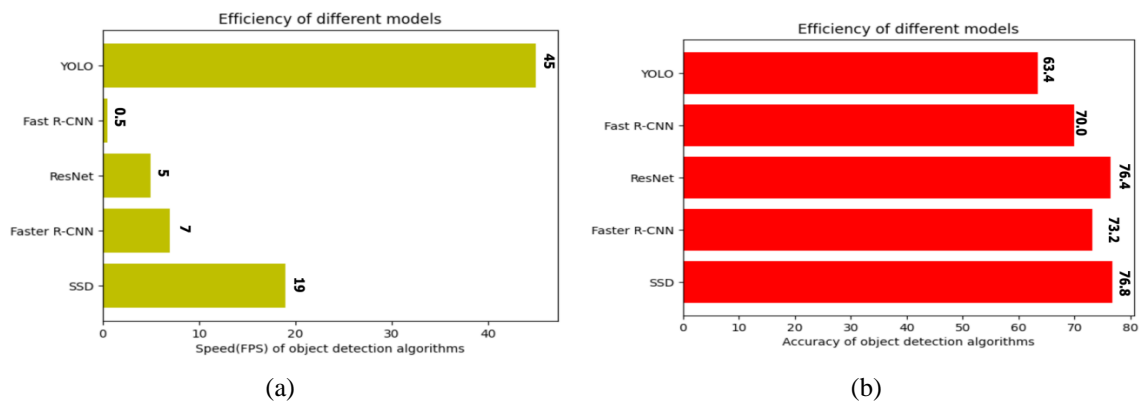


Figure 7. Comparison of object detection algorithms, (a) FPS Comparison of all object detection algorithms; (b) Accuracy of all object detection algorithms

4. CONCLUSION

In this paper a detailed and systematic study of various object detection algorithms has been carried out, each having its advantages and disadvantages. The architecture along with the detailed working procedure of each algorithm has been studied. The real-life applications of each technique were also studied and included. The shortcomings of one algorithm provided the basis for the development of a new algorithm. Many enhancements and improvements have been made in each algorithm since the time they were first proposed. For example, initially, YOLO, although was fast, its accuracy was not that good but later many different versions of YOLO were developed like YOLO9000 and YOLOv3-tiny. The latest YOLOv3 method is extremely fast, having applications in fields like a vision-based automated vending machine. Initially in transfer learning, only extra layers added were used to be trained but now we can also tweak the model according to our requirements and also we unfreeze the model to train the entire layers including the pre-existing ones. R-CNN was also improved and the latest technique using R-CNN was developed like fast R-CNN and faster R-CNN. Some algorithms use their data-sets like YOLO while having a scope to include other data-sets also. Each algorithm is perfect in its sense depending upon the task. All the algorithms have different applications depending upon their nature of work. For example, SSD and YOLO are used in aerial surveillance, YOLO is used in License Plate Recognition, R-CNN is used in image segmentation and vehicle detection and recognition. By doing the comparative study of various object detection techniques, we get an idea of when to use what algorithm and how can we tweak the model to our need, and also how can we improve an existing technique to develop a new one or to increase its accuracy or speed. The accuracy of each algorithm also depends on the data sets. The clearer is the data-sets, the more accurate is the algorithm.

ACKNOWLEDGEMENTS

This research work was funded by “Woosong University’s Academic Research Funding-2022”.




REFERENCES

- [1] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
- [2] Z. Liu, Y. Chen, B. Chen, L. Zhu, D. Wu, and G. Shen, “Crowd counting method based on convolutional neural network with global density feature,” *IEEE Access*, vol. 7, pp. 88789–88798, 2019, doi: 10.1109/ACCESS.2019.2926881.
- [3] A. Boukhalfa, N. Hmina, and H. Chaoui, “Parallel processing using big data and machine learning techniques for intrusion detection,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 553–560, 2020, doi: 10.11591/ijai.v9.i3.pp553-560.
- [4] H. Ding *et al.*, “Arbitrator2.0: preventing unauthorized access on passive tags,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 835–848, 2022, doi: 10.1109/TMC.2020.3017484.
- [5] C. Henry, S. Y. Ahn, and S. W. Lee, “Multinational license plate recognition using generalized character sequence detection,” *IEEE Access*, vol. 8, pp. 35185–35199, 2020, doi: 10.1109/ACCESS.2020.2974973.
- [6] D. M. Gavrilă, “Sensor-based pedestrian protection,” *IEEE Intelligent Systems*, vol. 16, no. 6, pp. 77–81, 2001, doi: 10.1109/5254.972097.
- [7] M. A. M. Azizi, M. N. M. Noh, I. Pasya, A. I. M. Yassin, and M. S. A. M. Ali, “Pedestrian detection using doppler radar and LSTM neural network,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 394–401, 2020, doi: 10.11591/ijai.v9.i3.pp394-401.
- [8] Q. Zou *et al.*, “Robust lane detection from continuous driving scenes using deep neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.
- [9] D. P. Sudharshan and S. Raj, “Object recognition in images using convolutional neural network,” in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Jan. 2018, pp. 718–821, doi: 10.1109/ICISC.2018.8398912.
- [10] Ö. Günaydin, M. Günay, and Ö. Şengel, “Comparison of lung cancer detection algorithms,” *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*, 2019, doi: 10.1109/EBBT.2019.8741826.
- [11] A. M. Magar, H. N. Mali, S. U. Thakare, T. R. Bankar, and V. M. Kamble, “COVID-19 detection through transfer learning using multimodal imaging data,” *International Journal of Advanced Research in Science, Communication and Technology*, pp. 836–841, May 2022, doi: 10.48175/IJARST-4145.
- [12] R. B. Roslan, I. N. M. Razly, N. Sabri, and Z. Ibrahim, “Evaluation of psoriasis skin disease classification using convolutional neural network,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 2, p. 349, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp349-355.
- [13] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, “Self-driving cars,” *Computer*, vol. 50, no. 12, pp. 18–23, 2017, doi: 10.1109/MC.2017.4451204.
- [14] N. N. A. Mangshor, I. A. A. Majid, S. Ibrahim, and N. Sabri, “A real-time drowsiness and fatigue recognition using support vector machine,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 4, pp. 584–590, 2020, doi: 10.11591/ijai.v9.i4.pp584-590.
- [15] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, “A systematic review of intelligence video surveillance: trends, techniques, frameworks, and datasets,” *IEEE Access*, vol. 7, pp. 170457–170473, 2019, doi: 10.1109/ACCESS.2019.2955387.
- [16] J. Y. W. Jien, A. Baharum, S. H. A. Wahab, N. Saad, M. Omar, and N. A. M. Noor, “Age-based facial recognition using convoluted neural network deep learning algorithm,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 424–428, 2020, doi: 10.11591/ijai.v9.i3.pp424-428.
- [17] M. A. Zaveri, S. N. Merchant, and U. B. Desai, “Small and fast moving object detection and tracking in sports video sequences,” in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 2004, vol. 3, pp. 1539–1542, doi: 10.1109/ICME.2004.1394540.
- [18] Y. Yoon *et al.*, “Analyzing Basketball Movements and pass relationships using realtime object tracking techniques based on deep learning,” *IEEE Access*, vol. 7, pp. 56564–56576, 2019, doi: 10.1109/ACCESS.2019.2913953.
- [19] L. BinYan, W. YanBo, C. ZhiHong, L. JiaYu, and L. JunQin, “Object detection and robotic sorting system in complex industrial environment,” in *2017 Chinese Automation Congress (CAC)*, Oct. 2017, vol. 2017-Janua, pp. 7277–7281, doi: 10.1109/CAC.2017.8244092.
- [20] K. Lim, D. Jin, and C.-S. Kim, “Change detection in high resolution satellite images using an ensemble of convolutional neural networks,” in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov. 2018, pp. 509–515, doi: 10.23919/APSIPA.2018.8659603.
- [21] S. B. Jadhav, “Convolutional neural networks for leaf image-based plant disease classification,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, p. 328, Dec. 2019, doi: 10.11591/ijai.v8.i4.pp328-341.
- [22] W. Kurdthongmee, “A comparative study of the effectiveness of using popular DNN object detection algorithms for pith detection in cross-sectional images of parawood,” *Heliyon*, vol. 6, no. 2, p. e03480, Feb. 2020, doi: 10.1016/j.heliyon.2020.e03480.
- [23] M. T. Tommiska, “Efficient digital implementation of the sigmoid function for reprogrammable logic,” *IEE Proceedings: Computers and Digital Techniques*, vol. 150, no. 6, pp. 403–411, 2003, doi: 10.1049/ip-cdt:20030965.
- [24] L. Liu, C. Shen, and A. V. D. Hengel, “Cross-convolutional-layer pooling for image recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2305–2313, Nov. 2017, doi: 10.1109/TPAMI.2016.2637921.
- [25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-Based convolutional networks for accurate object detection and segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016, doi: 10.1109/TPAMI.2015.2437384.
- [26] M. M. Jan, N. Zainal, and S. Jamaludin, “Region of interest-based image retrieval techniques: A review,” *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 520–528, 2020, doi: 10.11591/ijai.v9.i3.pp520-528.
- [27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [28] U. Budak, A. Sengur, A. B. Dabak, and M. Cibuk, “Transfer learning based object detection and effect of majority voting on classification performance,” in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2019, pp. 1–4, doi: 10.1109/IDAP.2019.8875920.




- [29] P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn, and W. Kusakunniran, "Breakthrough conventional based approach for dog breed classification using CNN with transfer learning," in *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct. 2019, pp. 1–5, doi: 10.1109/ICITEED.2019.8929955.
- [30] Y. Wang, C. Wang, and H. Zhang, "Combining single shot multibox detector with transfer learning for ship detection using Sentinel-1 images," in *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA)*, Nov. 2017, vol. 2017-Janua, pp. 1–4, doi: 10.1109/BIGSAR DATA.2017.8124924.

BIOGRAPHIES OF AUTHORS






Debani Prasad Mishra    received the B.Tech. in electrical engineering from the Biju Patnaik University of Technology, Odisha, India, in 2006 and the M.Tech. in power systems from IIT, Delhi, India in 2010. He has been awarded the Ph.D. degree in power systems from Veer Surendra Sai University of Technology, Odisha, India, in 2019. He is currently serving as Assistant Professor in the Dept of Electrical Engg, International Institute of Information Technology Bhubaneswar, Odisha. His research interests include soft Computing techniques application in power system, signal processing and power quality. He can be contacted at email: debani@iiit-bh.ac.in.






Kshirod Kumar Rout    received B Tech degree in electrical engg. from VSSUT burla in 2000, Received M Tech degree in power electronics from KIIT University in 2011 and continuing PhD degree in IIIT Bhubaneswar. He is currently serving as Assistant Professor in the Dept of Electrical Engg, International Institute of Information Technology Bhubaneswar, Odisha. He can be contacted at email: kshirod@iiit-bh.ac.in.



Sivkumar Mishra    received the B.E degree from Malaviya Reginal Engineering College, Jaipur (presently known as Malaviya National Institute of Technology, Jaipur) in 1995, M. Tech (Power System Engineering) from Indian Institute of Technology, Kharagpur and PhD (Engineering) from Jadavpur University, Kolkata. He is currently working as a Professor in Centre for Advance Post Graduate Studies, Biju Patnayak University of Technology, Odisha, Rourkela. His research interests include Intelligent Methods for Power System Planning and Smart Grid related Studies. He is a Senior Member of IEEE and IEEE Power and Energy Society. He can be contacted at email: capgs.smishra@bput.ac.in.



Surender Reddy Salkuti    received the Ph.D. degree in electrical engineering from the Indian Institute of Technology, New Delhi, India, in 2013. He was a Postdoctoral Researcher with Howard University, Washington, DC, USA, from 2013 to 2014. He is currently an Associate Professor with the Department of Railroad and Electrical Engineering, Woosong University, Daejeon, South Korea. His current research interests include market clearing, including renewable energy sources, demand response, smart grid development with integration of wind and solar PV energy sources. He can be contacted at email: surender@wsu.ac.kr.