# Artificial Intelligence

# Faster R-CNN vs SSD: Object Detection using Deep learning

Shivani Shah
sds510@uregina.ca

Simranjeet Randhawa
ssr779@uregina.ca

Sagarkumar Patel
scp846@uregina.ca

Department of Computer Science

Professor

Malek Mouhoub

Affiliation

University of Regina

Regina, Saskatchewan

## Abstract

The project investigates the working and functioning of various CNN algorithms for performing object detection. The analysis is performed on the keynote which factor to the performance variations in two different object detection algorithms utilized in the field of Artificial Intelligence, and implementation of models like Faster R-CNN and SSD. Priority is given to the algorithms most commonly used for object detection and incorporate the use of deep learning aspects to increase the accuracy of the result as opposed to the machine learning approach. The algorithms under analysis here are Faster R-CNN and SDD. Single implementations of both models are done to verify the accuracy and speed difference between them.

**Keywords:**
Faster R-CNN, Single Shot MultiBox Detector (SDD), Deep Learning, Object Detection, Artificial Intelligence

## 1. Introduction:

This focal point of the report is based on understanding the quality of the Convolutional Neural Network (CNN) algorithm as a base and analyze the difference between two major algorithms in performing object detection using the deep learning framework, namely Faster R-CNN and Single Shot MultiBox Detector (SSD). The paper also discussed the applications for these approaches along with an implementation of the Faster R-CNN. The basis of the CNN approach is discussed in the following section, along with the shortcomings of other approaches to object detection which make use of machine learning fundamentals.

Object Detection is a major requirement for utilizing machine learning and artificial intelligence capabilities in relation to computer vision and image retrieval. It is a technique which uses the knowledge of image processing to detect instances of objects under many classes like a car, chair, human, animal, and so on. These instances can be detected over not only images but videos as well. One of the most explored field for object detection is facial detection for surveillance systems and object detection for automobile manufacturers focused on advanced safety features in vehicles.

All the algorithms and approaches discussed here utilize the Deep Neural Network (DNN) learning models. DNN has emerged and proved to be a powerful machine learning model due to its capabilities to understand compound representations of data irrespective of the amount of hand design features. [1] [2]

There have been many approaches developed and defined in handling the object detection problem. Many of them use DNN's deep architectural capabilities to work with complex models in a more proficient way. Another part of the object detection problem is the identification of multiple instances of different objects from the same image. This is effectively tackled with the use of DNN-based regression to generate a bounding box structure around the objects. [1]

Moreover, the purpose of object detection is to define a technique which can not only detect an object in an image but also understand its position, size, class and other features which create the object. Therefore, the detection model can be distinguished into three parts: Informative region selection which handles object at any position and of any size within the image;

Classification techniques to classify the object into detailed semantic, hierarchical or informative groups for visual recognition; and Feature Extraction for obtaining detailed features of the object.

The current most used models for object detection include the Faster R-CNN and SSD models. Both are derived from the conventional CNN structure but make use of various other techniques to produce higher accuracy and speed for the given problem [3] [4]. These models perform not only object detection but also incorporate the classification and localization of the object in the provided an image. The difference between these is the major analysis goal of the project. More information regarding both is presented into the following sections, with the comparative analysis at the end.

## 2. Convolutional Neural Network (CNN):

In deep learning, CNN is one of the most ideal models. A neural network basically comprises of multiple layers which include three layers as 1) Input Layer, 2) Hidden Layers and 3) Output Layer. Here, the visible layers are input and output, handle the input and the output decisional data of the model. Whereas the major work and computation are performed in the hidden layers. These can range from one to thousands in number and handle the major tasks of the model. The hidden layer/s, in such configurations, is the convolutional layer/s which filter the inputs and transform them using specified patterns before passing it on to the next layer.

CNN architecture shares similarity to the general design principles for applying convolutional input layers and feature maps in an increasing number of dimensions. Therefore, CNN networks follow two kinds of network designs: Classical and Modern.

The networks such as LeNet-5, VGG16, and AlexxNet are examples of classical networks and implement a simple stack structure of comprehended convolutional layers. These convolutional layers consume a subset of the previous layers data channels at each filter to reduce computational cost and forcefully disrupt the network symmetry.

On the other hand, modern architectures explore new ways of implementing the convolutional layers, for example, Inception, DenseNet, and ResNet. These architectures provide variations in the methodology of mining of features for rich computational possibilities which are vital for image classification/segmentation and object detection.

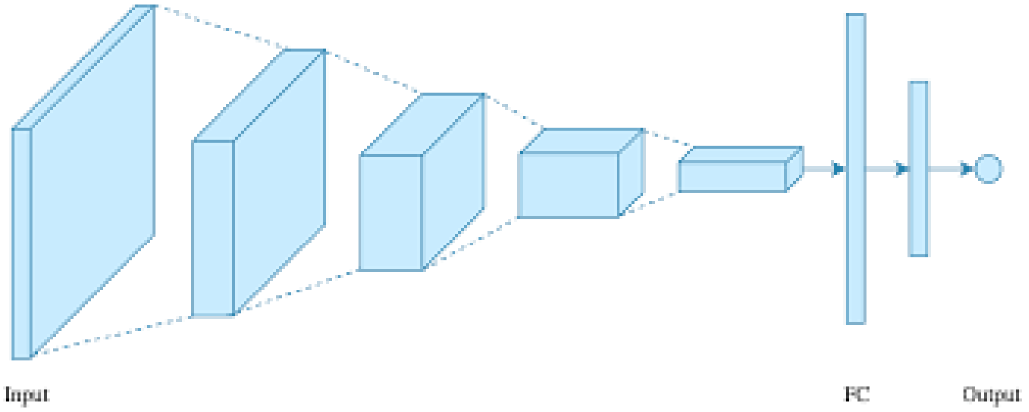

*Figure 1 Basic architecture of CNN [5]*

The CNN architecture network is the base on which the Faster R-CNN and SSD models are developed upon. These will be discussed in the following sections.

## 3. Faster R-CNN:

The Faster R-CNN model is based on the Region Convolutional Network which improves the previously defined Fast R-CNN and R-CNN approaches [3]. The R-CNN was successful in achieving exceptional accuracy in object detection using Deep Convolutional networks. However, Fast R-CNN improved on many downsides of the R-CNN model such as High expense of Training in time and space, Slow detection rate and multi-stage pipeline training. [6]

Faster R-CNN is, according to Shaoqing and group [3], a State-of-the-Art object detection network based on the region proposal network (RPN). This helps in hypothesizing the object location. Compared to other R-CNN networks Faster R-CNN employs RPN to share the complete image convolutional network with the detection networks which grants them use of the proficiency of the network to surmise the object boundary along with its score at different positions.

RPN processes an image of arbitrary size to produce rectangular object proposals set while working in the hidden layer and having the preceding layer shared with the object detection network [3].

### 3.1 Architecture of Faster R-CNN:

The simple architecture of Faster R-CNN is shown below:

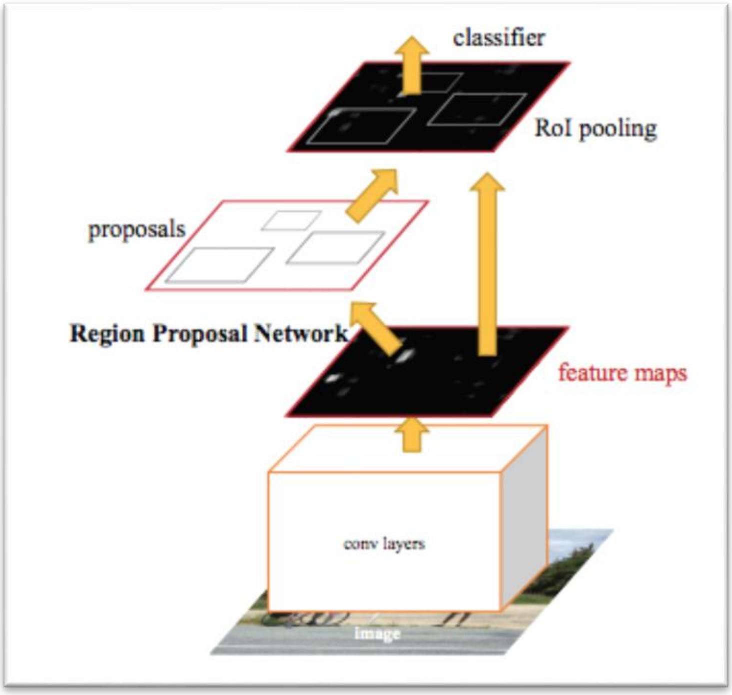

*Figure 2 Simple Faster R-CNN architecture [3]*

Thus, the above figure RPN comprises functions as a crucial part of the Faster R-CNN network. RPN generate the proposals for the main model to utilize but also contains its own structure and parts [3] as:

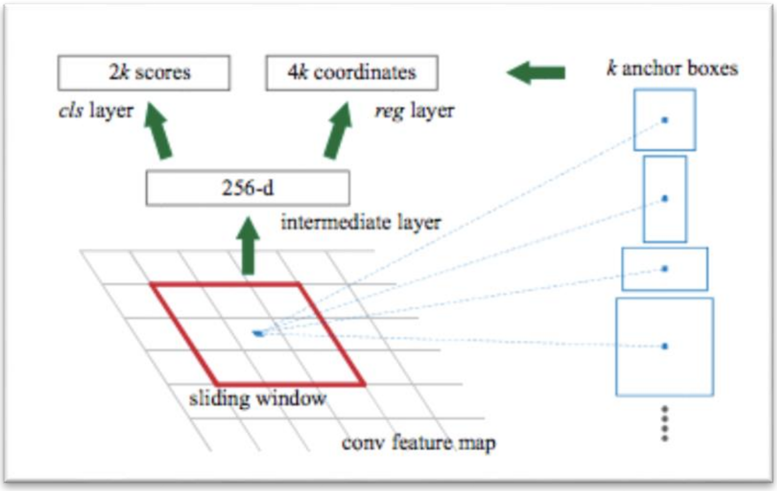

*Figure 3 RPN Architecture Structure [3]*

This architecture depicts the various parts of the RPN and shows the use of regressor and classifier layer with anchor boxes. Her, the classifier drives the probability of target object for the proposal and regression handles the regression of the coordinates of the proposal. [3]

For RPN the image is defined to have two important parameters: Aspect ratio and scale [3].

$$\text{Aspect\_ratio} = \frac{\text{the width of the image}}{\text{the height of the image}} \ \& \ \text{Scale} \ = \ \text{size of the image}$$

As a result, the developer chooses 3scales and 3 aspect ratios, which returns 9 possible proposals for each pixel. This concludes to k=9 in the results as the number of anchors. [3]

## 3.2 How RPN Works?

Being the backbone of Faster R-CNN, it has multi-scale anchors as opposed to “Pyramid of Filters”, referred to as “Pyramid of Anchors”. Consequently, to the multi-scale anchor structure, the time consumption reduces, and the process becomes more cost capable than the previously proposed algorithms. These anchors are named using two factors [3], which are:

1. Highest Intersection-over-union overlap with the turtle box.
2. Intersection-over-union overlap greater than 0.7

Therefore, RPN needs to be trained for the data and to keep the account for the loss during the process a loss function is the summation of losses of classification and bounding box regression which is specified [3] as follows:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}}$$

$$\mathcal{L}(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i)$$

*Figure 4 Loss Function for RPN in Faster R-CNN [3]*

Where i signifies anchor index, p means the probability of being an object or not, ground truth box represented by *, t represents the vector of coordinates.

$N_{cls}$ and $N_{box}$ stand for normalization. $L_{cls}$ is the log loss function over two classes and $L_1^{smooth}$ is the smooth $L_1$ loss. P* verifies if the object is identified as 'YES' then only regression will count and lastly, λ here is 10, thus both calls and reg will weigh equally [3].

Faster R-CNN is much "faster" than its predecessor, which can be observed in the following graph [7]:

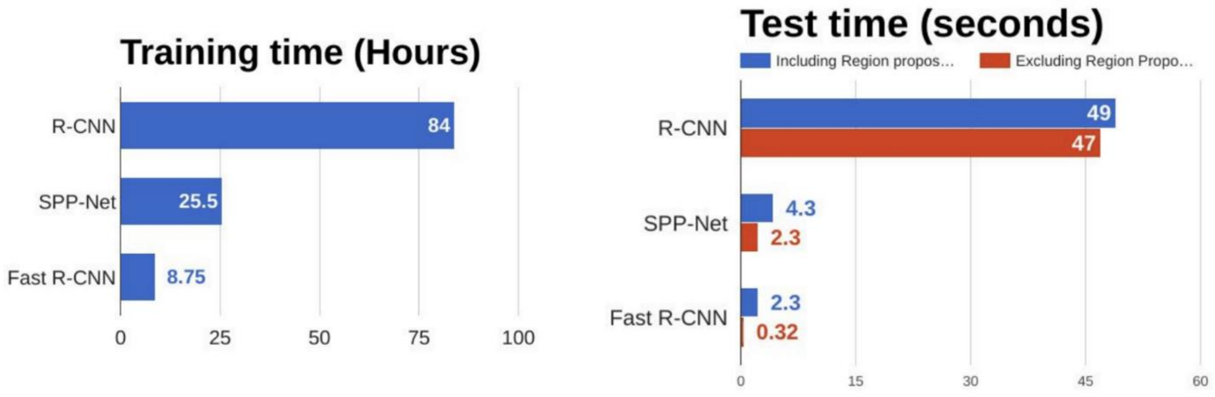

*Figure 5 Speed Comparison of various R-CNN Algorithm [7]*

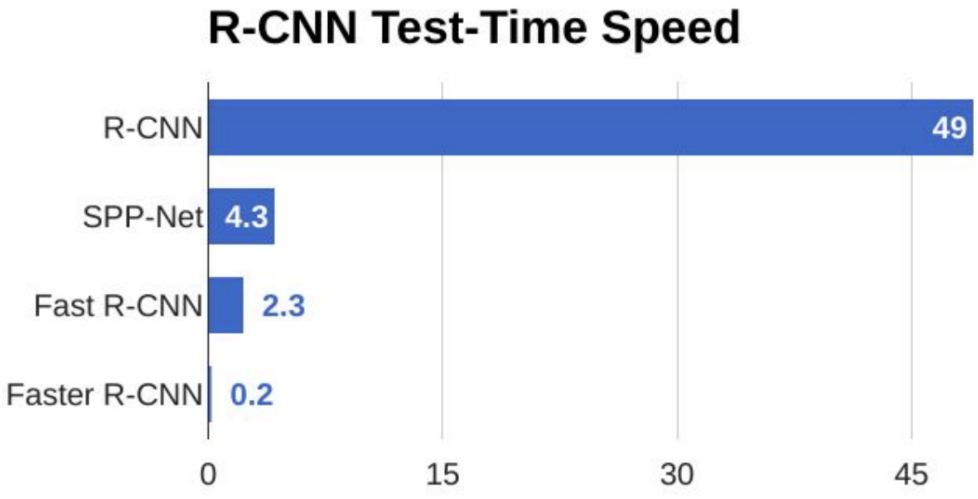

*Figure 6 R-CNN Speed Test-Time [7]*

As one can see, the training time along with the testing time is significantly reduced with the implementation of Faster R-CNN. Moreover, we can see in figure 5, that the speed of the algorithm along with the implementation of region proposal is still faster. [7]

## 4. Single Shot MultiBox Detector (SSD)

The single shot detector is a multi-scale window detector which usually depends on deep CNN networks to perform identification of the object (Classify) and locate it (Localization) [4]. Multi-scale sliding window detection, as it sounds, is a local window of varying size which slides down the whole image and scans for any objects of interest which can be identified. This justifies the "MultiBox" in the proposed name of the technique. Being multi-scale, the sliding window increases the robustness of the procedure in detecting objects of different size in an image. [4]

SDD is a brute force strategy, making it occasionally unreliable and expensive due to the requirement of a successful detection asks for a fine-grained image resolution for the sliding window to identify the vital information from the sampled image.

The single Shot detector has a good balance between speed and accuracy. The SSD also exploits a similar anchor box structure to that of Faster R-CNN at diverse aspect ratio. SSD makes predictions on bounding boxes after multiple convolutional layers. As convolutional layers work on different scales it is easy to detect the object at distinguished scale [4].

### 4.1 Architecture of SSD:

"Single Shot" refers to the process being capable of doing a forward pass in a single time by performing the object classification and localization of the network.

"MultiBox" is a technique for regression, as discussed above.

"Detector" is the network which handles the main object detection and its classification.

The architecture of SSD is based on the architecture of VGG-16 but castoffs the completely linked layers. It uses VGG-16 architecture as its base network because of its strong performance. But the latter set of convolutional layers added to extract the features at various levels to decrease the size of input to each subsequent layer.

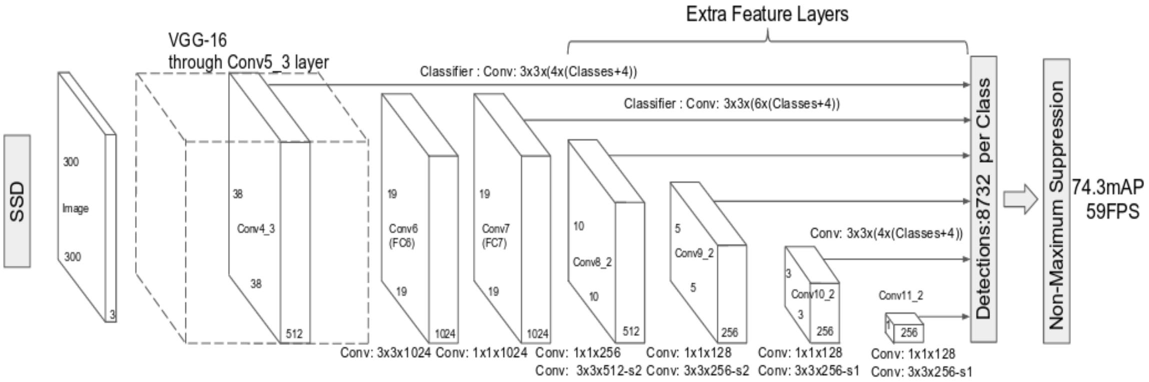

*Figure 7 Single Shot Architecture [4]*

The MultiBox technique using the bounding box regression is a source for the derivation of the fast class-agnostic bounding box coordinate proposals. Moreover, the loss function of the MultiBox is also a combination of two major factors of the SSD: confidence loss and location loss.

Confidence loss: It checks how assured the network is for the object of the computed bounding box and to compute this loss cross-entropy is used.

Location loss: It checks how far the network predicated regression is from the ground truth ones of the training set and to compute this it uses L2-norm but SSD uses smooth L1-Norm.

**Loss Equation:**

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

*Figure 8 Loss Function for SSD [4]*

where α specifies the balance of contribution between the losses and N is equal to the number of default boxes matched [4]. If N = 0, loss is set to zero.

## 5. Faster R-CNN vs SSD:

The main question here is not about the best detector, but which detector provides the best accuracy and speed under various configurations. Research has been performed in understanding the difference in performance for these algorithms using varying implementations [8].

In general, from an accuracy standpoint, Faster R-CNN is at the top whereas from a speed perspective SSD is faster. We can observe in the graph that 300 proposals applied with the Inception Resnet allow Faster R-CNN to produce the highest accuracy at 1 FPS.

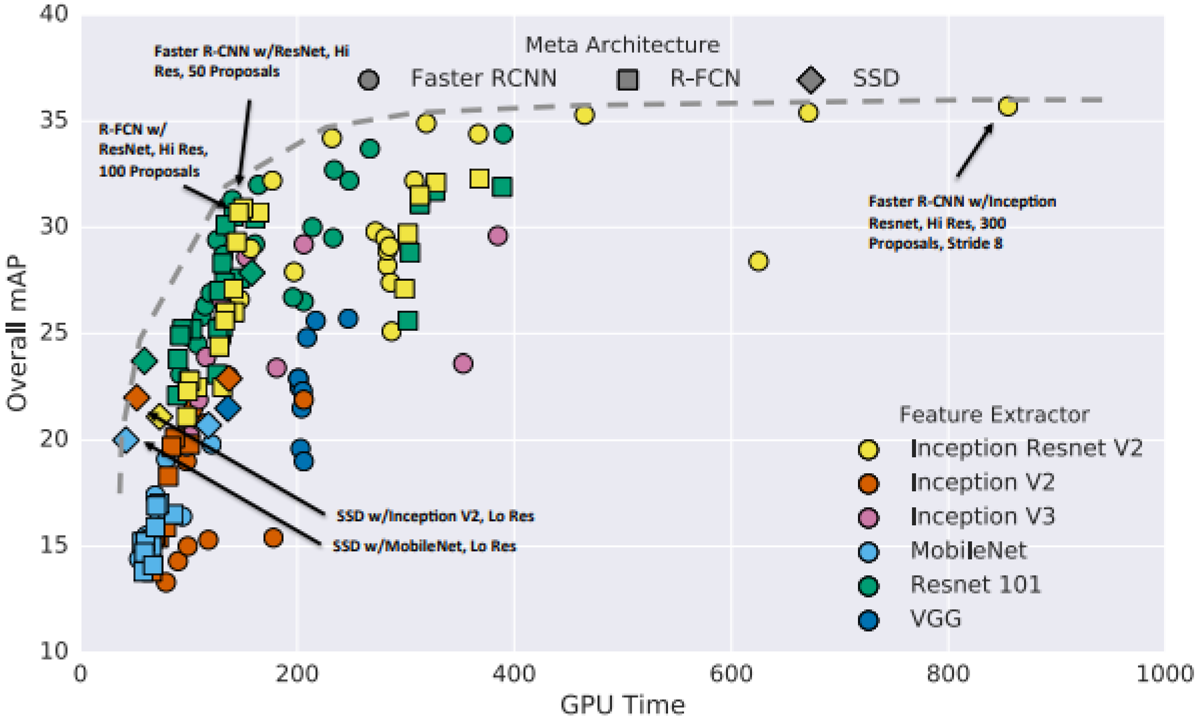

*Figure 9 Accuracy vs Time [8]*

Moreover, looking into the capabilities of feature extraction and its effect on the overall detectors accuracy performance we can observe, in the figure below, that the Faster R-CNN takes advantage of better feature extractors while in SSD the significance of these extractors is less. [8]

In addition to this, SSD handles larger objects very well and can closely match the performance of the other using better extractors, however, smaller objects are where SDD is not preferable over the others. [8]

Thus, if speed is not a priority then Faster R-CNN model is one of the best suited for the job. If speed and accuracy both are of equal or high importance then using SSD with better extractors and high resolution of images (to improve the small object shortcoming) can be utilized.

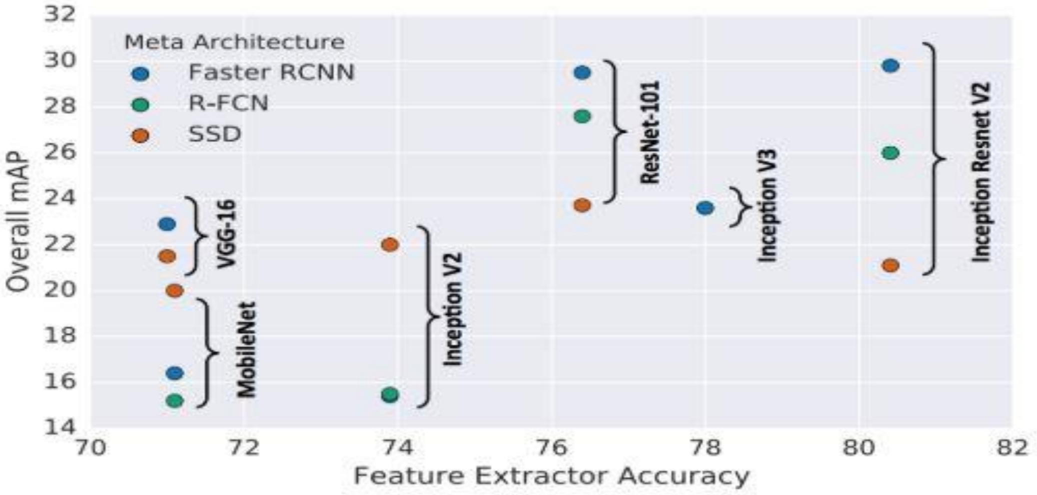

*Figure 10 Effects of Feature Extractors [8]*

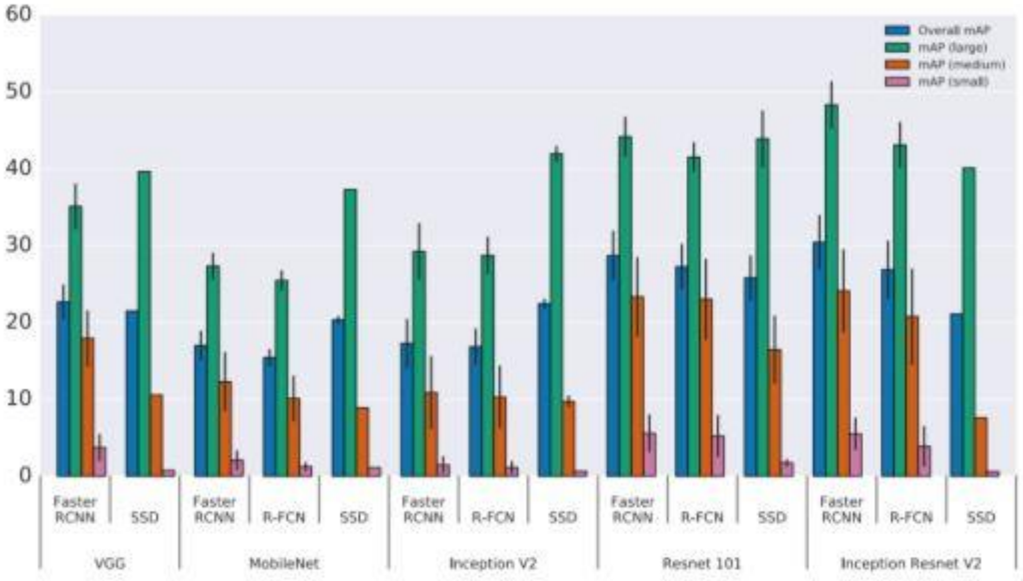

*Figure 11 Object Size Effects Comparison [8]*

## 6. Our Implementation:

For better understanding and to get a proper estimate to how the Faster R-CNN and the SSD models are implemented and also observe the difference in their performance we implemented both the models and tested these on the same data set. The results of these implementations are as shown:

Faster R-CNN was implemented with the Inception of v2 COCO. The results were as expected. The accuracy of the model was high and was able to clearly identify objects and classify them properly. SSD was implemented with the MobileNet v1 COCO and we can see that the model was not as confident in identifying the objects but based on speed was quite responsive to movements.

We can see with the same image sample the difference in the accuracy and confidence of both the models.

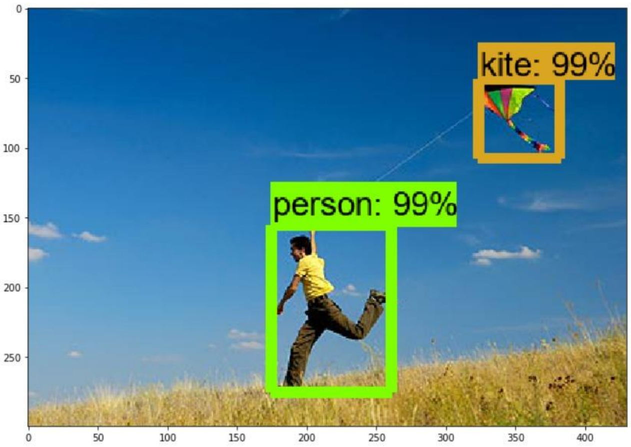

*Figure 12 Faster R-CNN*

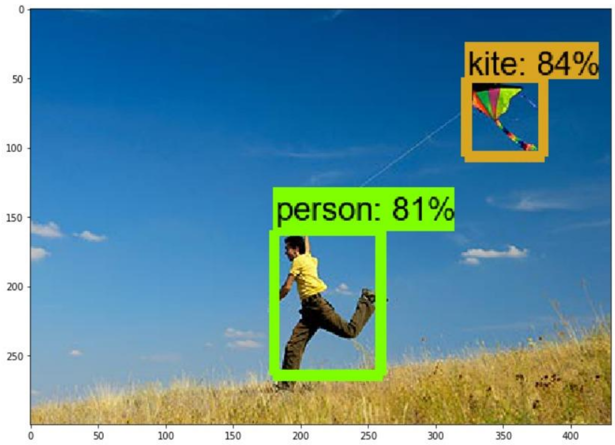

*Figure 13 SSD*

We can also see that the SSD is unable to identify some objects in the following image sample whereas the Faster R-CNN is clearly detecting and categorizing all the objects in the given image.

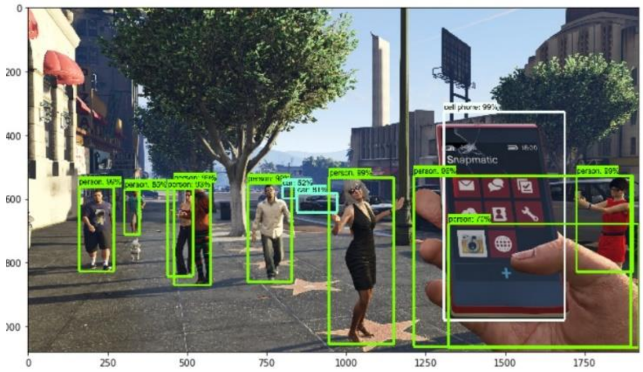

*Figure 14 Faster R-CNN sample 2*

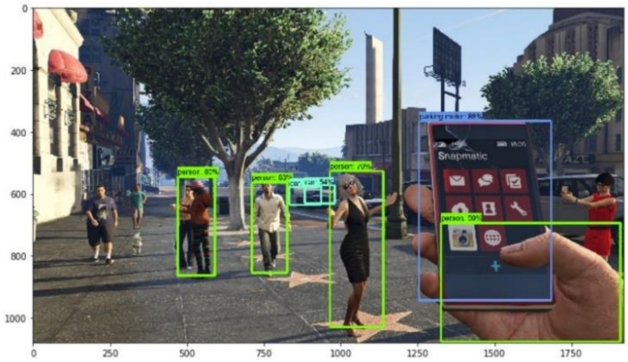

*Figure 15 SSD sample 2*

## 7. Interesting Applications:

There are so many applications which make use of the object detection technology. One of the obvious applications is to detect various kind of object in any type of image. The other few major applications are:

- **For security purpose: -**

  Such as this can be used in any video surveillance is able to detect any suspicious (explosive) objects at public places.

- **Self-driving cars: -**

  Nowadays many car manufacturer companies try to use the object detection technology to have a self-driving mode in their model.

- **Face detection: -**

  This is one of the most used applications of object detection like face recognition in the iPhone or any other cell-phones which is used by most of us in our everyday life.

- **People counting: -**

  People counting is one of the other applications which can be used at tourists places to keep an estimate to build an analysis of what number of people visit the festival or event at a specific place.

- **Anomaly detection: -**

  Detecting the variations of objects which is unusual for example in the field of medical X-ray reports and many more.

- **Vehicle detection: -**

  To detect different types of vehicles such as bike, car, motorcycle, ship, boat depending on the size of the object, shape of the object, etc.

Thus, there are numerous applications like tracking of an object, pedestrian detection technology and in the industry of manufacturing certain kinds of products and all of the above-mentioned application where the understandings of object detection are being applied in real-world problems.

## 8. Conclusion:

To conclude, we can say that object detection is solitary of the major problems that incorporate challenges of computer vision and image processing/retrieval and other computer fields. Moreover, with the development of machine learning concepts and deep learning models, we have achieved significant progress in creating structures which can provide close to perfect object detection solutions for use in various areas including but not limited to Robotics, Environment Simulation, image segmentation, and artificial intelligent systems for surveillance. We have seen the basis of two of the major deep learning models, Faster R-CNN and SSD, developed for object detection and their pros and cons.

## 9. References:

[1] C. Szegedy, A. Toshev and D. Erhan, "Deep Neural Network for Object Detection," in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 2013.

[2] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature 521,* pp. 436-444, 2015.

[3] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Network," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBoc Detector," in *ECCV 2016. Lecture Notes in Compuer Science,* 2016.

[5] A. Dertat, "Applied Deep Learning - Part 4: Convolutional Neural Networks," 08 November 2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2.

[6] R. Girshick, "Fast R-CNN".

[7] R. Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO—Object Detection Algorithms," 09 06 2018. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Gaudarrama and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017.

## Appendix:

**For further information please visit:**

https://github.com/simranjeetbazpur/Object-Detection-using-tensorflow-Deep-Learning-

**To see the running vidoes of our implementation:**

**Faster R-CNN :** https://drive.google.com/file/d/1YzLh95cSyG0ccjEJmD0-qVh7fu2d9oMS/view?usp=sharing

**SSD:** https://drive.google.com/open?id=1i_Lgl6GMBx_w0JZl1wlgopwwcRrcNOSr