How to Cite:
Rao, G. V. V., Reddy, S. V. K., Basha, S. A., & Bharath, G. (2022). Object detection, localization and tracking using Single Shot Detector (SSD). *International Journal of Health Sciences*, *6*(S1), 7306–7319. https://doi.org/10.53730/ijhs.v6nS1.6904

Object detection, localization and tracking using Single Shot Detector (SSD)

G.Vishnu Vardhan Rao
Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India
Email: vishnuvardhan@veltechmultitech.org

S Vamsi Kumar Reddy
Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India
Email: suravamsikumarreddy@gmail.com

Shaik Ansar Basha
Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India
Email: ansarbasha7280@gmail.com

Gaddam Bharath
Department of Electronics and Communication Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India
Email: gaddambharath007@gmail.com

***Abstract*---**We are trying to Detect, Localize and classify objects in an image using neural networks. We are using Single Shot detector for performing the task. The output of Single Shot Detector is Bounding boxes. It will try to localize the objects in an image and also tell which object is present in an image. The network predict probability scores of all the classes, we will take the highest probability of all the predicted probabilities. The network also predicts bounding boxes in an image, which can be reduced using NonMax Suppression. Object Detection is most widely used task in computer vision. We can locate objects based on our interest. Object detection is most widely used in real time situations and it has many practical applications. Some of the applications include Vehicle number plate detection, face mask detection, face recognition. Applications include face detection, pedestrian detection and vehicle detection. Single Shot Detector is Deep Learning based algorithm.

***Keywords*---**Object Detection, Object Localization, Object Classification, Convolutional Neural Networks.

Introduction

There are wide variety of Object Detection techniques, in all of them SSD will give high accuracy.[1]The SSD model is trained on COCO dataset[2] which contain many classes[3].SSD is bit faster than all Object Detection algorithms with high performance and resolution. This paper shows the predictions of bounding boxes and classes. So there is improved performance of accuracies and speed of predictions. The main improvement of speed comes from neural network based predictions instead of brute force approach([4, 5]). With improving of speed and accuracies we can use this SSD based model in all our applications. We are trying to use one of the technique in Computer Vision. We summarize our contributions as follows:

1) We are using Single Shot Detector with high speed and performance used for multi class classification. This algorithm is better than all the algorithms of previous proposals. In SSD we are not using region based proposals.
2) The Single Shot Detector predicts bounding boxes and classes using specific feature maps which is exploited from neural network.
3) We are using COCO dataset to train our model and network graph architecture.

Single Shot Detector (SSD)

This SSD framework architecture predicts feature maps which are close to ground truth boxes.

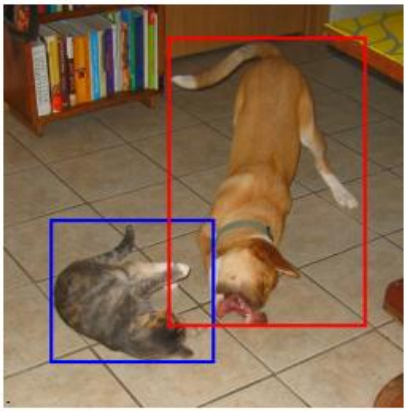

Fig1: Ground truth boxes

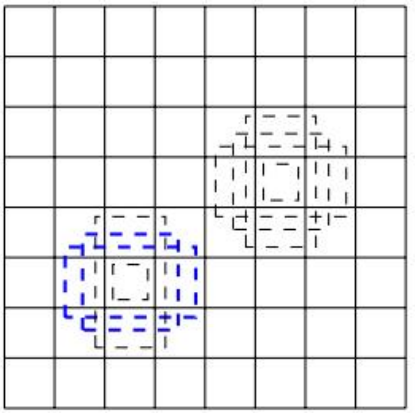

Fig2: 8 × 8 feature map

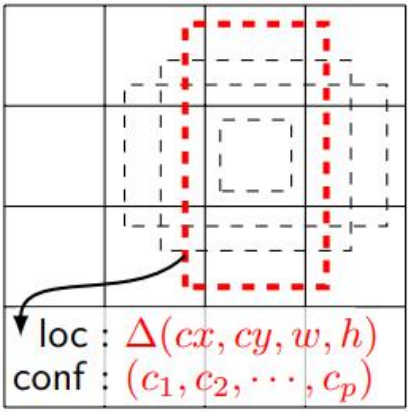

Fig3: 4 × 4 feature map

SSD framework

Fig1SSD network takes input RGB image with 3 channels and corresponding ground truth bound boxes for train the network. From Convolution operation we generate different sizes of feature maps to exploit all the features as shown in Fig2which is 8 x 8 and Fig3 which is 4 x 4. For each feature map we predict the object coordinates and confidence scores of the object present in the image (c1, c2. . .Cp).We compare the prediction boxes with ground truth boxes. In our case there are two objects in an image we compare the true boxes with predicted two

boxes, we compute the loss(example: mean squared regression error) and then we back propagate the total loss to the network to adjust weights finally we run through the soft max layer to predict the output.

We add convolutional network layers which decreases size as we move forward for prediction of output layers. The network outputs are different for differentlayers of variable feature maps (Over feat [4] which will operate on a single scale feature map).

[5] The SSD algorithm proposed in this paper will predict multiple classes and their corresponding bounding boxes. This algorithm is based on neural network and not on region proposal approach. For different sizes of target detection, the traditional approach is to first convert the images into different sizes (image pyramids), then detect them separately, and finally combine the results.

Prior Box

The indentation introduces the Prior Box in the SSD. The Prior boxes are similar to anchor boxes. Prior box will select boxes in advance for some outputs, and then the real time bounding box coordinates and target label is obtained from Soft max classification along with bounding box regression. The SSD generates a priority box according to the following rules:
It will consider the midpoint of each feature map as a center (offset = 0.5), and then produces a series of the concentric priority boxes (the coordinates of the center gets multiplied, this will map the feature map coordinates to original coordinates)

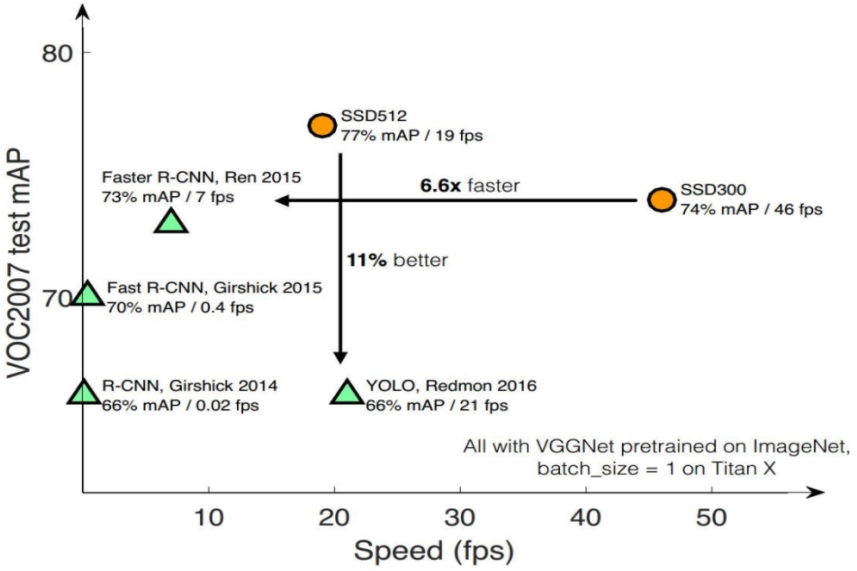

Fig 4: FPS in SSD Model

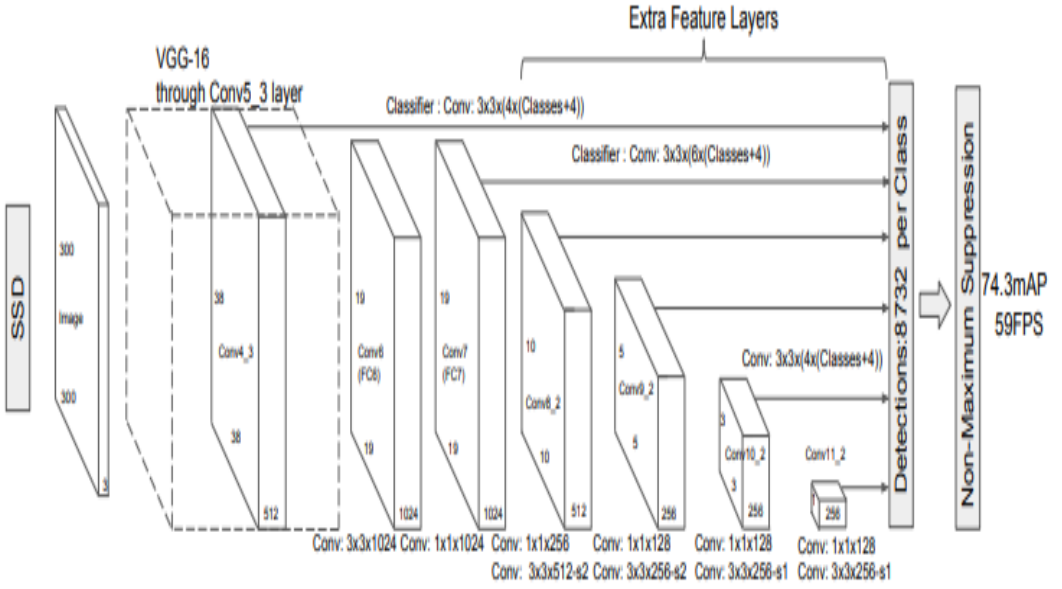

Fig 5: SSD Network structure

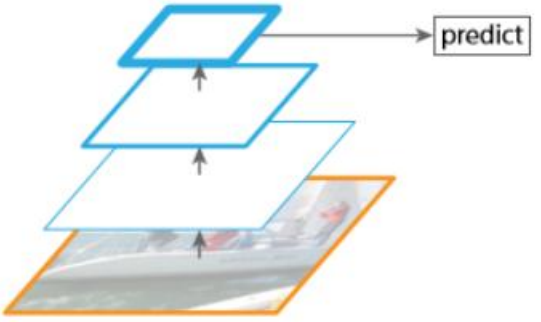

Fig6: Single Feature Map

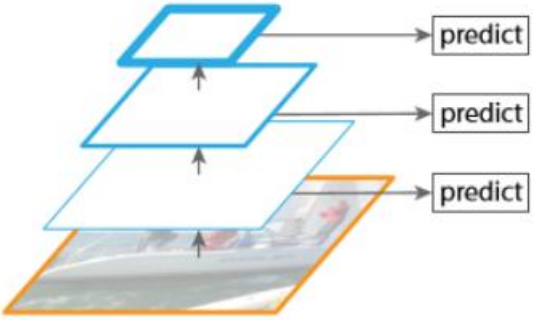

Fig7: Pyramidal Feature Hierarchy

The minimum side length of a square priority box is minimum size and the maximum sidelength of square priority box is:

$\sqrt{minsize * maxsize}$ ---------> (1)

For each aspect ratio set in pretext, 2 rectangles will be created, the dimensions of a rectangle are (length and width) are:

$\sqrt{aspectratio * minsize}$And $1/\sqrt{apectratio * minsize}$---------> (2)

The minimum size and maximum size of each feature map corresponding to the priority box are determined by the following formula, where m is the number of feature maps used in detection (m = 6 in SSD 300):

The different feature maps with respective empirical field sizes are [13].

$$Sk = Smin + \frac{Smax - Smin}{m-1}(k-1),\ k \in [1,m] \text{---------->(3)}$$

The minimum size = S1 and maximum size = S2 corresponding to the feature map in the first layer; the minimum size = S2 and maximum size = S3 in the second layer;and but the priority box setting in SSD is 300 does not correspond to the above formula.

Table 1
SSD low level feature maps and high-level feature maps

	Min_ Size	Max_ Size
Conv4_3	30	60
Fc7	60	111
Conv6_2	111	162
Conv7_2	162	213
Conv8_2	213	264

However, it can still be seen that SSD uses high level features to identify huge targets and low level features to identify small targets, which should also be an outstanding contribution of SSDs [5]. Among them, the conv4_3_norm_priorbox layer pretext of the priority box generated by SSD 300. Knowing how the priority box is generated, let's analyze how the priority box is used.

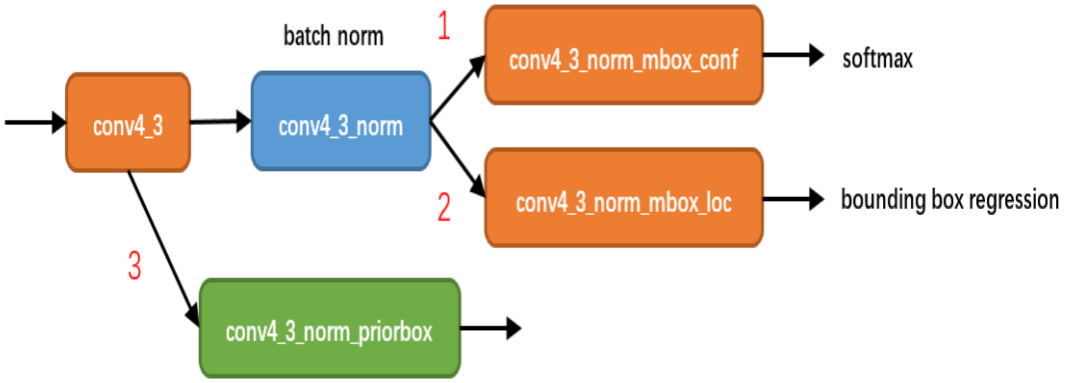

Fig 8: network pipeline

Atreus Convolution explained

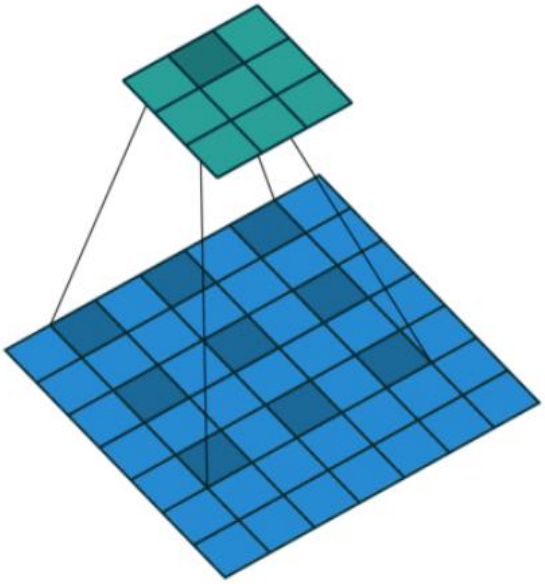

Fig9: Atreus Convolution

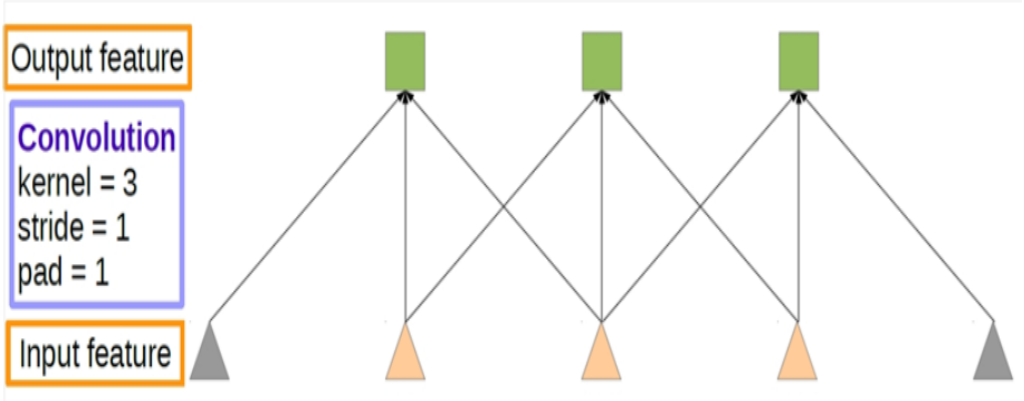

Fig 10: Sparse Feature Extraction

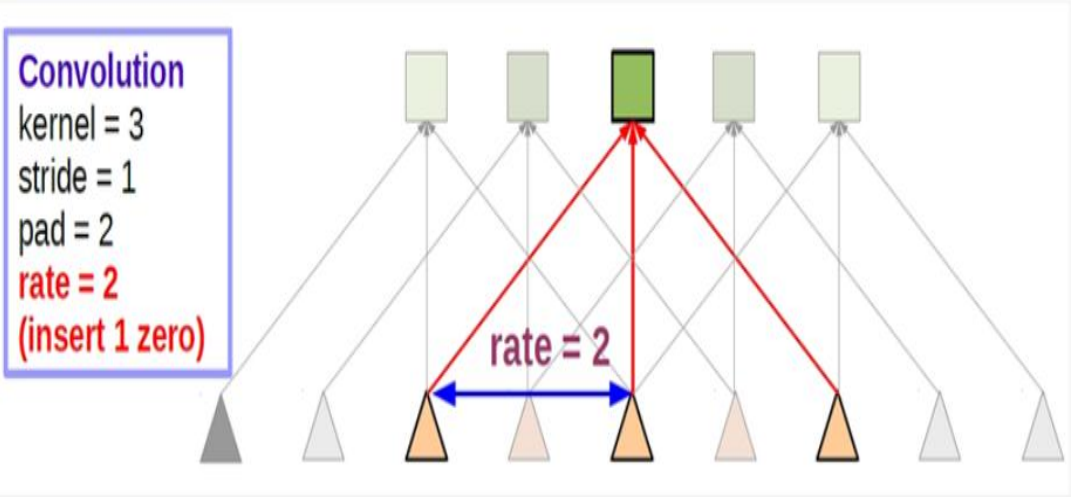

Fig 11: Dense Feature Extraction

The above figure (10) is a convolution with holes. You can jump to choose and add one every other. The third example (11) in the figure below is convolution with holes.

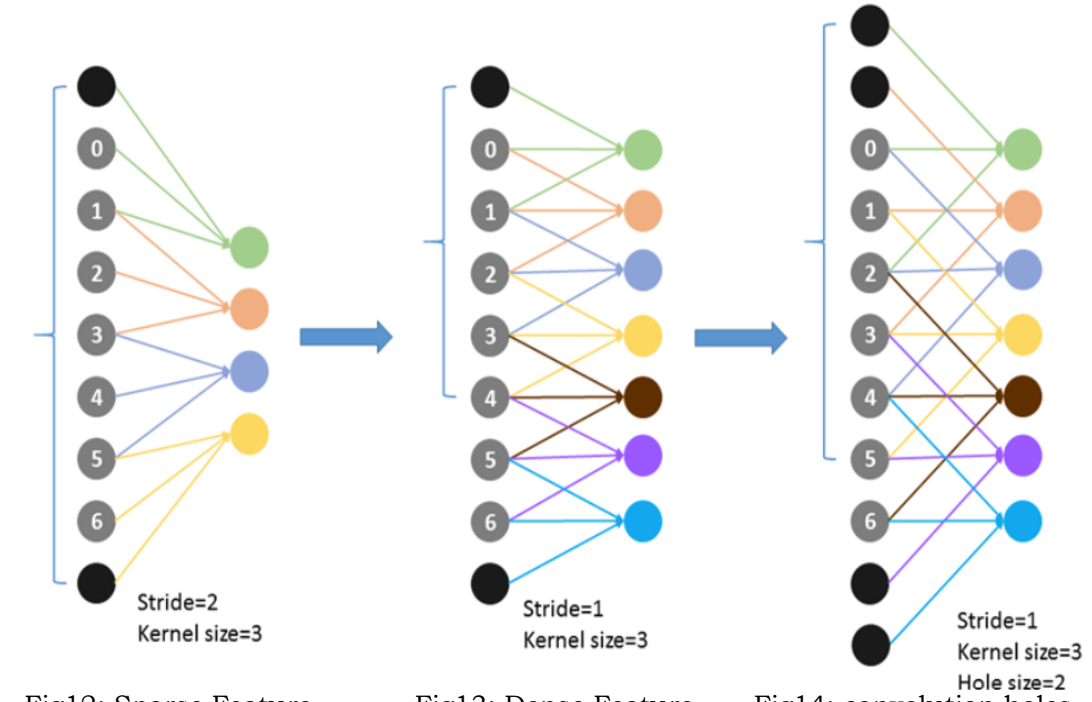

Fig12: Sparse Feature Fig13: Dense Feature Fig14: convolution holes

Convolution reduces the size of the kernel and also saves the memory. And theeffectiveness of convolution with holes assumes that closely adjacent pixel values are almost the similar, and all of them are redundant, so it is better to jump to H (hole size) and take one.

MS COCO Dataset

The MS COCO dataset is a dataset used for object detection training created by Microsoft. Object Detection in Computer Vision use COCO dataset and also used for multiple vision real world projects [6].Dealing with images and videos is the concept of Computer Vision. The SSD network graph the model weights are trained by MS COCO dataset.

SSD Model Training Using Dataset

The full form of COCO is Common Objects in Context. This dataset is created of increasing efficiency of image recognition. This MS COCO dataset contains high resolution and lowresolution images used for training SSD model.

COCO dataset is used for comparing the performance of different models in Object Detection and track their performance.

Fig 15: Object Detection In COCO Data Set

And that is how we can access the bicycle images and their annotations. In conclusion we have seen how the image and annotation of the popular COCO dataset can be used for new projects, particularly in object detection.

MS COCO dataset is the constant dataset in all Object Detection and Computer Vision based projects.

Features of MS COCO dataset

There are 80 classes in MS COCO dataset which include all daily life things. (Examples: car, traffic lights, laptop, brush, person, etc.)

The 80 MS COCO dataset classes

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

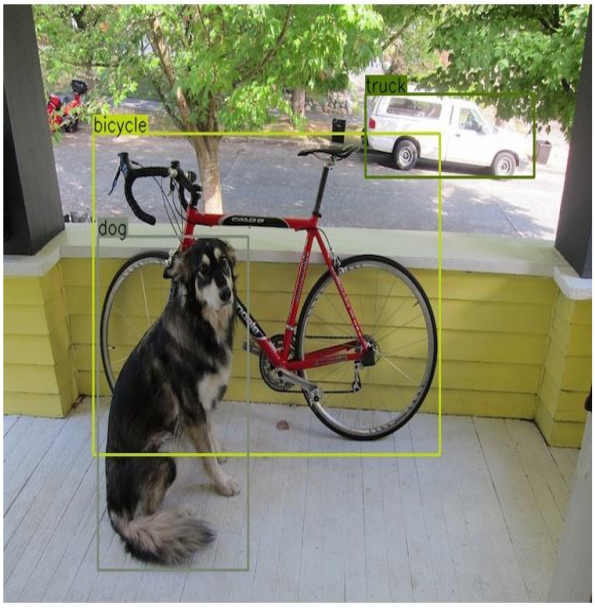

Fig 16: COCO dataset classes for object detection

Image Classification

Predicts which object is present inside the bounding box. The no of classes is equal to the number of bounding boxes in an image. The input image is mapped to class labels.

Fig 17: Image Classification

Object Localization

Draws or predict a bounding box in the location of objects. There will be one or more bounding boxes, the best bounding boxes will be filtered from by putting a threshold.

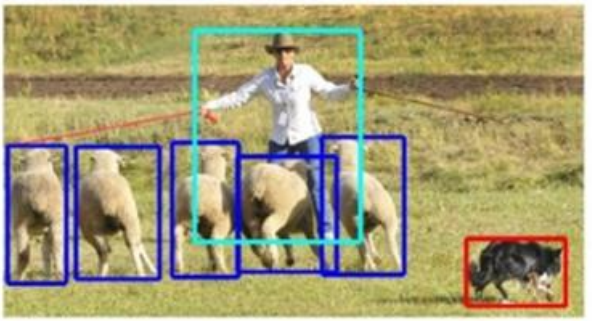

Fig 18: Object Localization

Object Detection

It is combination of both object classification and object localization. In an input image both bounding boxes and predicted class labels will be put together in a single input image.

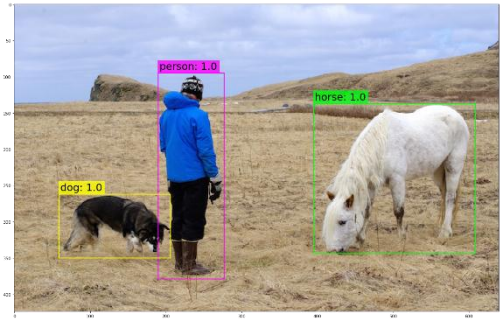

Fig 19: Object Detection

Object Recognition in an image

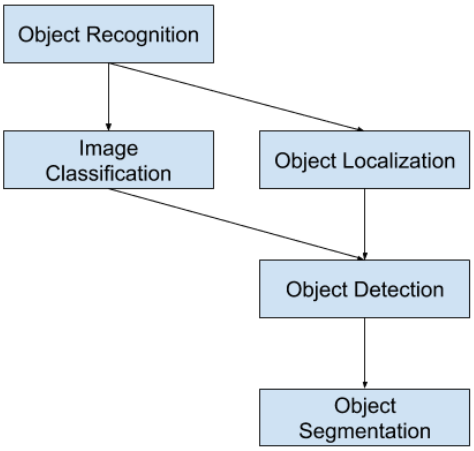

Fig 20: computer vision tasks

Results

There are two SSD models one is SSD300 and other is SSD512. In SSD300 input image is of shape 300 x 300 and in SSD512 input image is of shape 512 x 512. The results are below.

Table 2
MS COCO

Method	data	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast [6]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast [24]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster [2]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [24]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster [25]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**

Data Augmentation

To deal with different types, angles and colors of objects we the concept called Data Augmentation. With Data Augmentation we can classify all the objects with different colors, angles and sizes. Data Augmentation will increase the Mean Average Precision to large extent. And an increase of 2%-3% map is gained in multiple datasets shown below.

Table 3
COCO-TRAINVAL35K

METHOD	TRAINVAL35K	
	0.5:0.95	0.5
	0.75	
SSD 300	23.2	41.2
SSD 512	23.4	
	26.8	46.5
	27.8	
SSD 300*	25.1	43.1
SSD 512*	25.8	
	28.8	48.5
	30.3	

Table 4
Inference Time

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Analysis of the advantages and disadvantages of SSD network structure

The advantages of the indented SSD algorithm should be obvious: the running speed, and the detection accuracy is comparable to Faster RCNN. In addition, there are some trivial advantages that are not explained. Talk about the disadvantages here:

- We need to manually set the minimum size, maximum size, and aspectratio values of the priority box. The size and shapes of the priority boxescannot be learned from training and need to be set as hyperparameter. The size and shape of the priority boxes used by each layer of features in the network are exactly the same, resulting in the debugging process being very dependent on experience.
- For small objects the mean average precision is still less.

Conclusion

This paper tells about Single Shot Detector.The key task in our model SSD is to use different sizes of conv blocks with different stride and padding values to increase performance of network. The priority boxes increases the large accuracies of bounding boxes without learning from scratch. We have built the SSD model with a large number of bounding boxes for different scales of images with differentaspect ratios [5,7]. SSD model is used in real time object detection which gives high speed and accuracy.

References

1. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV (2013)
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

4. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)
5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
6. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
7. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)
8. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)
10. V. Prabhu**1**and G. Gunasekaran**2** ,Fuzzy Logic based NAM Speech Recognition for Tamil Syllables,Indian Journal of Science and Technology, *Vol 9(1),DOI: 10.17485/ijst/2016/v9i1/85763,* ISSN (Online) :0974-5645.January 2016.
11. Hariharan, B., Arbelaez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation ´ and fine-grained localization. In: CVPR. (2015)
12. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: Looking wider to see better. In: ILCR. (2016)
13. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: ICLR. (2015)
14. Prabhu.V.,Kuppusamy, P.G., Karthikeyan, A.,R. Varatharajan. (2019) Evaluation and analysis of data driven in expectation maximization segmentation through various initialization techniques in medical images, Multimedia Tools and Application, Springer , Vol 77, Issue 8, pp 10375–10390.
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: NIPS. (2015)