International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Object Detection and Tracking using Faster R-CNN

Chakradhara Panda

Abstract: This paper uses a deep learning model called Faster R-CNN to detect and track objects in images. Two backbone networks such as ResNet-101 and VGG-16 are tested on a self-created dataset and PASCAL VOC dataset. Intersection over union (IoU) technique is used for the purpose of object tracking. The impacts of batch size, number of iterations and learning rate are analysed. The paper finds that ResNet-101 outperforms VGG-16 significantly by 13% on test data. This finding reinforces that deeper network is better in feature extractions and generalizations. IoU is able to track multiple objects and can identify the loss of track. The processing of frames per second is found to be 5 fps. The study has implications for many computer vision applications. For example, the deep learning based object detection and tracking can either augment the capability of LiDARs and Sensors or become an alternative to them in self-driving vehicles.

Keywords : Faster R-CNN, ResNet-101, VGG-16, Object detection and tracking.

I. INTRODUCTION

The role of deep learning techniques in computer vision problems is increasingly studied in recent time. In the 90s, traditional computer vision techniques like Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG) and other feature descriptors were used to solve image classification, object detection and face recognition problems [1], [2]. On the contrary, Tesla, for example, relied heavily on deep learning techniques to solve autonomous driving problem [3]. There were over 28000 citations since the publication of the seminal paper *ImageNet Classification with Deep Convolutional Neural Networks* [4]. In the paper, a large deep convolutional neural network was used to classify 1.2 million high resolution images. The results showed that a deep neural network was capable of achieving superior performance.

Image classification helps to identify objects in an image, however, fails to locate their positions. On the contrary, object detection performs both classification and localization (see Fig. 1). The object detection problem is approached as either a classification or a regression problem.

In the classification approach, small patches called region proposals are obtained by dividing the image. Each patch runs through a classifier to determine if there are objects in it. The patches with positive classification results are marked with bounding boxes. The classification methods have one part of their networks dedicated to providing region proposals followed by a high quality classifier to classify these proposals.

These methods are very accurate but computationally expensive [5].

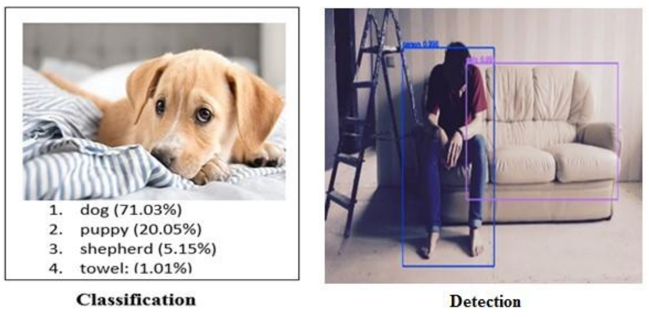

Fig. 1. Examples of object classification and detection.

On the other hand, regression methods are faster but less accurate. In the approach, the whole image runs through a convolutional neural network directly to generate one or more bounding boxes for objects. Single Shot Detector (SSD) and You Only Look Once (YOLO) are few such capable regression algorithms [6], [7]. The accuracy vs. speed trade-off is depicted in Fig. 2. As shown in the figure, SSD and Faster R-CNN fared well compared to other models [6]-[9].

In this paper, Faster R-CNN is used with two different backbone networks such as ResNet-101 and VGG-16. The remainder of the paper is organized as follows. In Section II, the related works are discussed. Section III discusses the data. The models and results are presented in Section IV. Section V summarizes findings and concludes the paper.

Manuscript published on 30 September 2019
* Correspondence Author
Chakradhara Panda*, Advanced Analytics and Cognitive Computing, IBM India Private Limited, Bengaluru, 560067, INDIA. Email: chdpanda@yahoo.com)

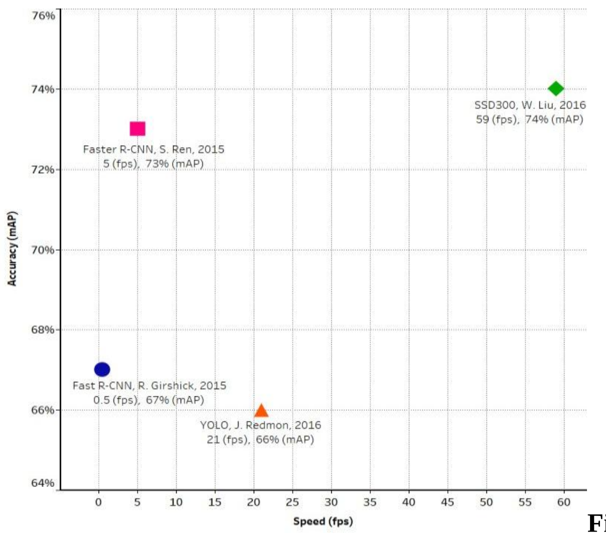

Fig. 2. The speed vs. accuracy trade-off in deep learning models.

II. RELATED WORKS

Reference [10], contrary to SIFT and HOG algorithms, proposed a gradient based learning algorithm for character recognition. Reference [4] produced a seminal work on deep neural networks by achieving top-5 test error rate of 15.3%, which outperformed the previous best SIFT by almost 11%. A deep network called Network In Network (NIN) was used for classification tasks. The fully connected layer was replaced with a global average pooling layer and the network had *mlpconv* layers which used multilayer perceptrons to convolve the input. On CIFAR-10 and SVHN datasets, the model demonstrated state-of the-art performance [11].

Reference [12], [13] devised Inception V1, Inception V2, and Inception V3 models which finally achieved the top-5 test error rate of 3.58%. Similarly, [14] showed that deeper neural networks could achieve better image classification and localization accuracy. By making few changes like the depth of network and 3*3 filters of each convolution, the top-5 test error rate of 7.3% was achieved. Further, this error was reduced to 6.8% by using an ensemble of 2 models.

One of the issue with deep neural networks is increasing error rate due to the depth of the network. To tackle the difficulty of training a deep network, [15] promoted optimizing the residual mapping (residual network, ResNet) than to optimize the original, unreferenced mapping. The identity shortcut connections were created in the network architecture which generally skipped one or more layers. The outputs of these shortcut connections were added to the outputs of the stacked layers. A 3.57% test error was achieved on the ImageNet dataset with the help of an ensemble of residual nets.

The region proposal methods and region based convolutional neural networks (R-CNN) were proposed against computationally expensive selective search algorithm [9]. The mAPs of 73.2% and 70.4% were obtained on PASCAL VOC2007 and PASCAL VOC2012 data, respectively. The deeper model VGG-16 had a frame rate of 5 fps. A third branch was added to the Faster R-CNN network for the instance segmentation [16]. The proposed Mask R-CNN model was able to perform 3 tasks together: classification, localization and segmentation. The processing of frames at 5 fps was not impacted because the addition of third branch was just a small overhead to Faster R-CNN architecture.

Similarly, [17] used an *efficient feature fusion module* (EFFM) to reduce the number of operations and parameters for a two-stage detector. A multi-scale dilation region proposal network (RPN) was used to obtain higher accuracy than Faster R-CNN. In the same year, modified Faster R-CNN to detect small objects in optical remote sensing images [18].

On the contrary, proposal generation and subsequent pixel or feature resampling stages were eliminated by using a single deep neural network called Single Shot Detector (SSD) [6]. For a 300*300 input, SSD achieved 74.3% mAP on VOC2007 test data at 59 fps. Similarly, another single deep neural network model called You Only Look Once (YOLO) was presented by [7]. Multiple versions of YOLO were published since then. In the latest version, Yolov3 was found to be equally accurate and three times faster than SSD [19].

III. DATA

This paper uses two types of data: PASCAL VOC2007 and a self-created dataset. PASCAL VOC2007 is a well-known dataset for object detection, classification, and segmentation. The PASCAL VOC2007 dataset contains 10000 images across 20 object classes.

A self-created dataset is used with the objective of validating Faster RCNN on new data. This dataset contains 2 object classes such as *overhead* and *divider*. The choice of these classes is deliberately done to make the study relevant for any autonomous or self-driving application. The data for *overhead* and *divider* are collected from multiple public sources. They are annotated using two open source tools such as LabelBox and LabelImg (see Fig. 3). A set of 500 images is finally chosen from each class after cleaning a pool of 4000 images. For *divider* and *overhead*, there are around 1300 and 600 annotations, respectively. This data is finally merged with PASCAL VOC2007 data for training and validation purposes. In total, there are approximately 11000 images from 22 object classes.

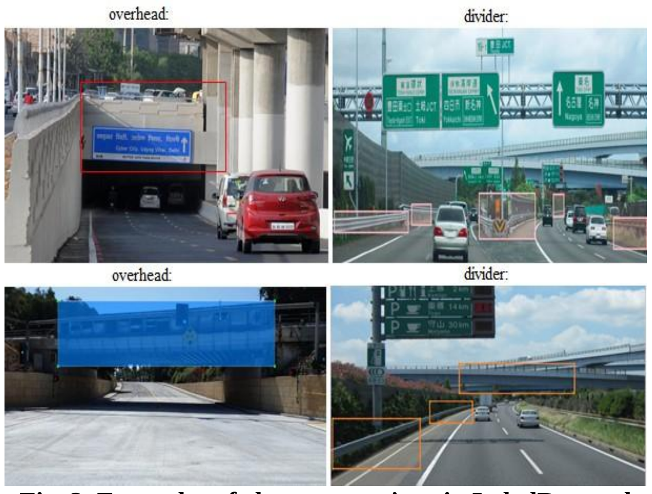

Fig. 3. Examples of class annotations in LabelBox and LabelImg.

IV. MODEL, RESULT AND DUSCUSSION

In this section, Faster R-CNN and intersection over Union (IoU) are discussed followed by results and analysis. The objective is to give an overview of techniques used in the paper to maintain the continuity for readers. However, interested readers may refer to [9], [20] for more discussions on these techniques.

A. Models

Faster R-CNN is a two stage network architecture or classification and localization tasks. The first stage is a region proposal network (RPN) which produces a bunch of proposals or object bounding boxes. The second stage is vital which extracts features from each

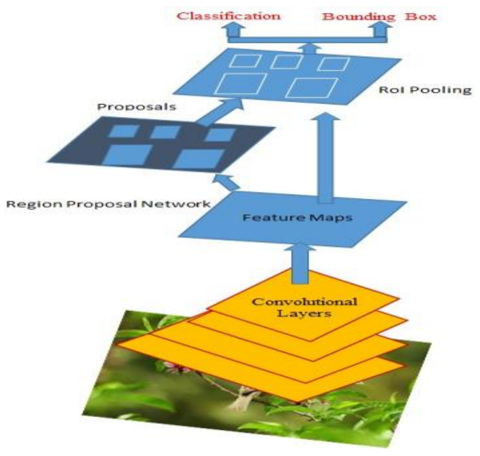

Fig. 4. Faster R-CNN is a single network which has two stages sharing the same feature map. This makes the inference faster.

proposal by using Region of Interest (RoI) pooling. RoI pooling splits the input feature map into a fixed number, say *k* roughly regions. Max pooling is then applied on each region which produces the output of size *k* regardless the size of the input. With features already extracted, it performs classification and bounding box regression. The advent of Faster R-CNN is that features extracted by CNN are shared by both stages which make the inferences faster. A Faster R-CNN architecture is shown in Fig. 4.

Object tracking is a central part of computer vision problem and has many applications like video surveillance, human computer interaction, and robot navigation. Object tracking is a process of finding the motion path of an object over time by locating its position in every frame of the video. There are multiple algorithms like point tracking, kernel tracking, and silhouette tracking used for object tracking. Each of them has own advantages and disadvantages. For example, point tracking is stable even if illumination changes, however, does not do well in the case of occlusion and multiple objects. Similarly, kernel tracking does not handle occlusion explicitly [21].

Recently, several CNN based methods have been proposed to learn tracking models [22]-[24]. For example, a real-time multiple object tracker (MOT) was proposed by combining Faster R-CNN and Generic Object Tracking Using Regression Networks (GOTURN) architecture for autonomous driving [25]. Deep reinforcement learning was also used for tracking [26].

In this paper, the bounding box based tracking is experimented by using intersection over union (IoU). It is a measure which gives the similarity between predicted bounding box and ground truth bounding box [20]. It is defined as:

$$IoU = \frac{TP}{(FP+TP+FN)} \tag{1}$$

where TP, FP, and FN are true positive, false positive, and false negative, respectively. Supposing, the detected bounding box for an object in current frame as ground truth ($B_{groundtruth}$) and in the subsequent frame as prediction ($B_{prediction}$), the IoU can be calculated as follows.

$$IoU = \frac{area(B_{prediction} \cap B_{grounndtruth})}{area(B_{prediction} \cup B_{groundtruth})} \tag{2}$$

The object can be tracked in subsequent frames depending on the IoU score and the threshold value. In order to track an object in all relevant frames of the video, the IoU must exceed a threshold value, say, 0.5.

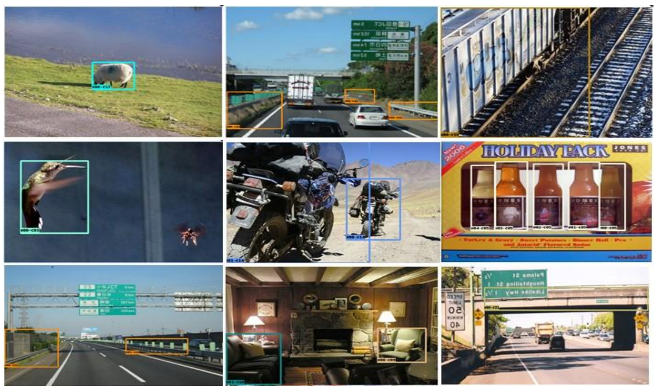

Fig. 5. Ground truth examples of training and validation sets in TensorBoard.

B. Result and Discussion

The Faster R-CNN model is experimented with two backbone network architectures such as ResNet-101 and VGG-16. The network is trained on GPU NVIDIA Tesla M60. The monitoring of network performance during the training is achieved by using TensorBoard visualization. A snapshot of TensorBoard visualization of ground truth examples during a training event is shown in Fig. 5.

The model sensitivities to key parameters such as batch size, learning rate and number of iterations during the training event are measured. The results are presented from Fig. 6-7. For example, in Fig. 6 (a), the activation function is forcing the activation to 0. The maximum zero activation is around 85% in the bottleneck layer which indicates a better model generalization.

The impacts of batch size and learning rate on the model performance are evaluated. The combination of high batch size and large learning rate is found to be better if we compare activation in Fig. 6. The batch size 256 in combination with a high learning rate of 0.005 provide model improvements (see Fig. 7). In the figure, the cross entropy and total loss on both training and validation sets converge well.

The training time is also varied and the results suggest no significant model improvement after 70,000 iterations. The VGG-16 exhibits weaker training performance as compared to ResNet-101. The results are not presented here to save the space.

The average precision on testing dataset is presented in Table I for all object classes. The mean average precision(mAP) is found to be 71% for ResNet101 which is 13% higher than VGG-16.

This supports the findings of [16] that deeper neural networks are better in object detection. If we compare object classes, the highest precision is obtained to detect *dog* which is followed by *horse* and *bus*. The *overhead* also performs well with the precision of 70%. The performance variation among classes can be explained.

One reason is the lack of diversity in data. For example, *divider* (AP=20%) and *potted plant* (AP=46%) are such cases where the data lack diversity or variety.

The calculation of average precision depends on the number of true positives in a certain class. The number of true positives further depends on the overlap threshold between the ground truth and prediction. The results discussed in Table I correspond to the overlap threshold of 0.5. The impact of this factor on test accuracy is measured for which results are presented in Table II. As it can be seen in the table, the test accuracy increases as the threshold decreases. For example, the mAP declines from 78% to 6% as the threshold increases from 0.3 to 0.9. The processing of frames is found to be 5 fps for Faster R-CNN with ResNet-101.

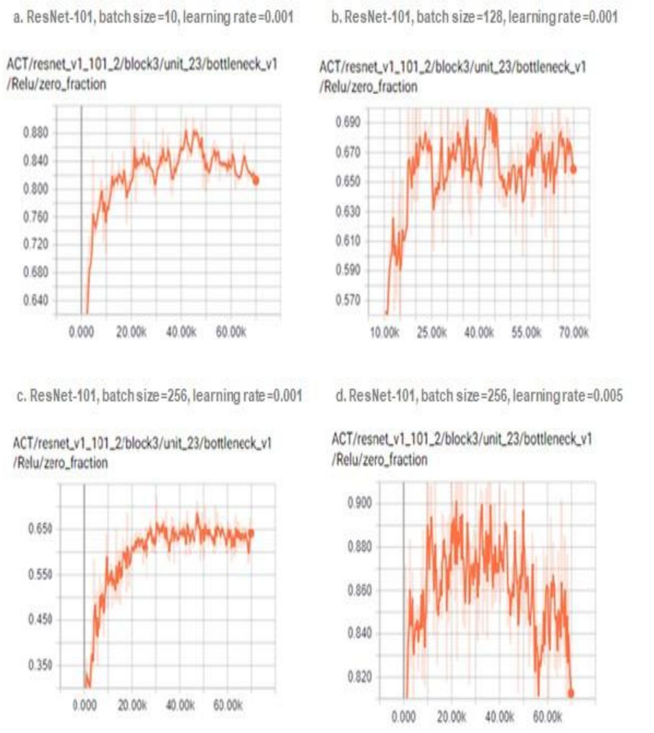

Fig. 6. Model zero activations during various trainings at different batch sizes and learning.

TABLE-I: Average precision (AP) on testing data set.

Object Classes	ResNet-101	VGG-16	Object Classes	ResNet-101	VGG-16
aeroplane	77%	62%	dog	87%	76%
bicycle	80%	70%	horse	85%	69%
bird	73%	61%	motorbike	77%	67%
boat	60%	42%	person	78%	69%
bottle	59%	46%	potted plant	46%	35%
bus	84%	57%	sheep	76%	66%
car	78%	62%	sofa	75%	54%
cat	83%	66%	train	78%	63%
chair	56%	44%	tv-monitor	75%	65%
cow	83%	75%	overhead	70%	56%
dining table	71%	63%	divider	20%	19%
			mAP	71%	58%

As it can be seen in the figure, the object *car* is detected in frame 5 and tracked subsequently in all frames. In frames 386 and 387, the object *car* is not detected due to occlusion. Hence the tracking is lost in these frames. In the subsequent frame, the object *car* is detected and identified as the re-entry of lost object. The tracking thereafter continues in following frames. However, the IoU tracker is not able to track multiple instances of the same object.

The selected examples of object detection on test data using Faster R-CNN is shown in Fig. 8. The Faster R-CNN is found to be good both in classification and localization. The detection of an object is indicated with classification and probability. The localization of object is indicated by using bounding boxes. Some more test examples are shown in Fig. 9.

The IoU tracking results are presented in Fig. 10. Each class has an unique tracking id and the IoU score is provided along with the frame count (FC). The IoU threshold is set at 0.5. The IoU tracking is built on the top of Faster R-CNN detection module. Multiple objects are detected in a frame which are tracked in subsequent frames. Tracking depends on the IoU score. For example, if IoU < 0.5 then the tracking is considered as lost. The tracking may also be lost due to undetected objects or objects not seen in subsequent frames. The IoU tracker can handle all these situations well. Accordingly, it can flag the track loss, identify the re-entry of same objects and track them in subsequent frames.

TABLE-II: Average precision (AP) on testing set at overlap thresholds 0.3 and 0.9.

Object Classes	$AP_{0.3}$	$AP_{0.9}$	Object Classes	$AP_{0.3}$	$AP_{0.9}$
aeroplane	82%	3%	dog	89%	4%
bicycle	86%	6%	horse	85%	4%
bird	80%	6%	motor bike	84%	10%
boat	72%	1%	person	87%	7%
bottle	65%	2%	potted plant	60%	4%
bus	84%	12%	sheep	80%	4%
car	83%	13%	sofa	79%	10%
cat	89%	7%	train	87%	12%
chair	64%	2%	tv-monitor	78%	6%
cow	84%	4%	overhead	88%	5%
dining table	78%	2%	divider	29%	0%
			mAP	78%	6%

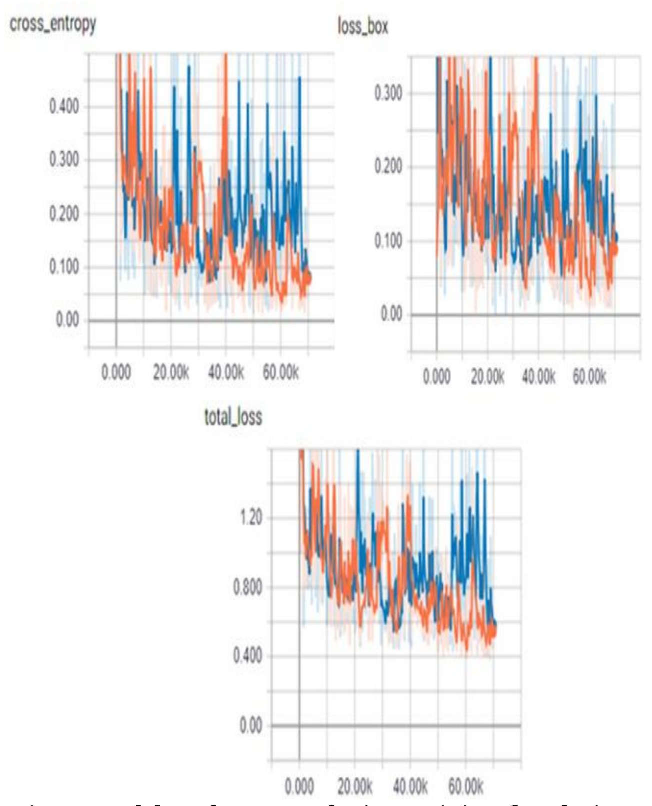

Fig. 7. Model performance during training (batch size= 256, iterations =70,000 and learning rate=0.005). Colors = Orange: Training set, Blue: Validation set.

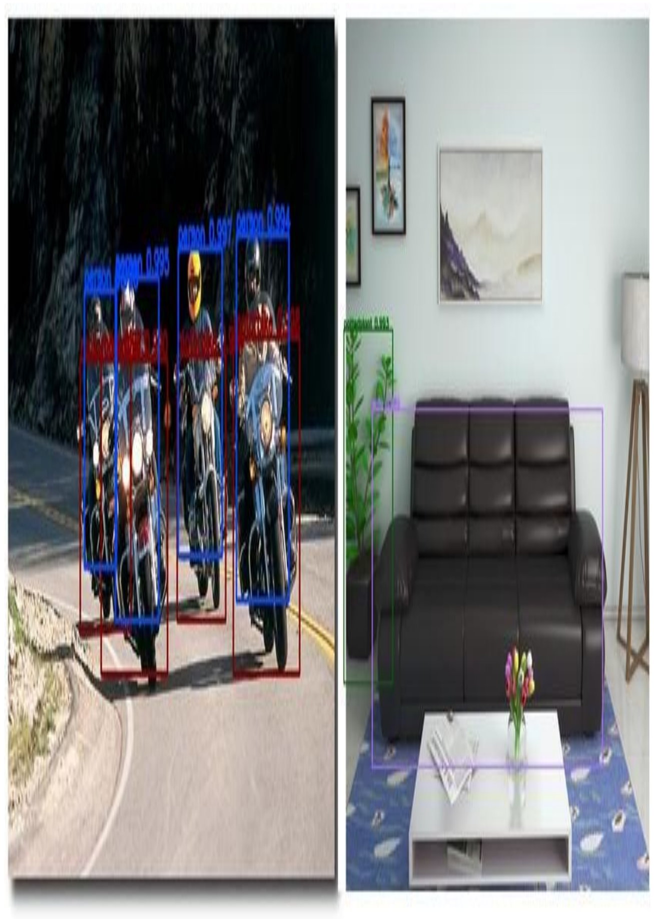

Fig. 8. The Faster R-CNN results on test data. The model classifies objects with probabilities. It also localizes objects by using bonding boxes.

V. CONCLUSION

In this paper, Faster R-CNN is used for the object detection. The PASCAL VOC2007 and self-created dataset are used for 22 object classes. The overall model accuracy is found at mAP 71% which increases to 78% at overlap threshold 0.3. The processing of frames is 5 fps. On the comparison between ResNet-101 and VGG-16, the results reveal that the deeper network is better. ResNet-101 adds 13% to the performance of VGG-16. IoU tracking performs well on multiple classes.

The results reinforce the power of deep neural networks in object detection and tracking. The study also adds new classes to the set of the existing ones. One among many critical business applications of object detection and tracking is self-driving or autonomous vehicles. The increasing effort in recent time is to use vision based decisions in autonomous driving. The new classes are added by keeping the autonomous driving application in view. The results indicate that deep networks can det

ect and track objects with significant accuracy. SSD and RetinaNet [27] are few more deep neural networks which are tested for real time applications. The findings of the current paper further strengthen the case for deep neural networks in computer vision applications. More researches are required in this direction.

ACKNOWLEDGMENT

The author thanks Satheeskaran Navaratnam, Senior Director at HAVI and Dr. Tilak Raj Singh, Lead Data Scientist at Daimler Trucks Asia-Japan for their valuable comments, suggestions and feedback.

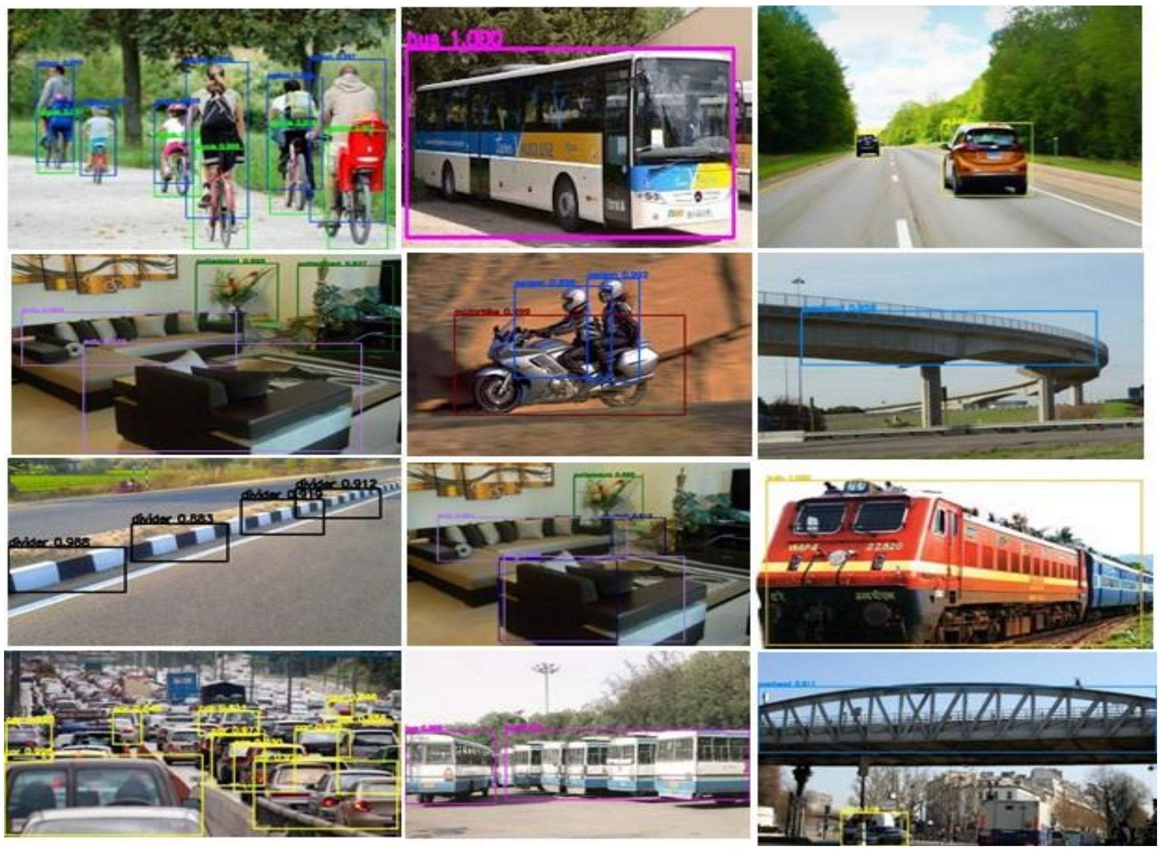

Fig. 9. More results of Faster R-CNN on test images using ResNet-101. Frames are processed at 5 fps.

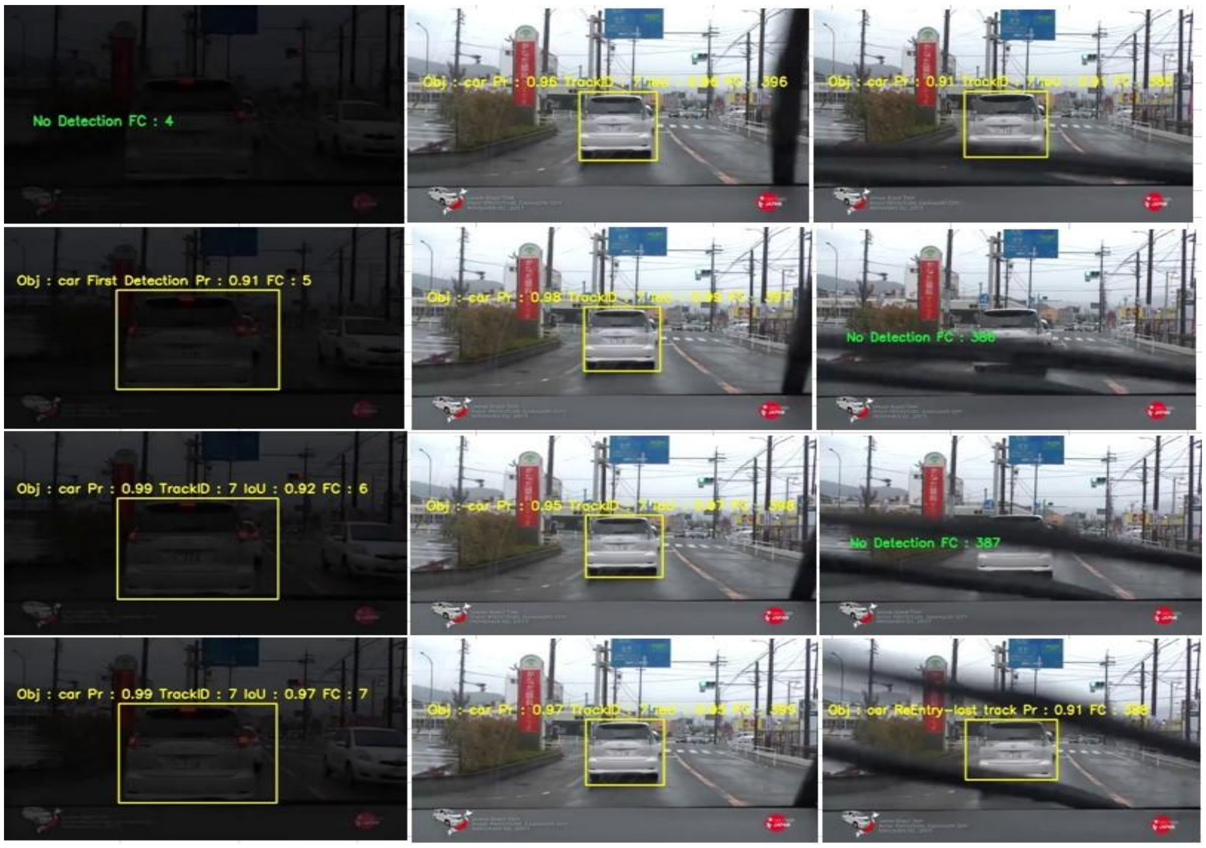
No Detection FC : 4
Obj : car First Detection Pr : 0.91 FC : 5
Obj : car Pr : 0.99 TrackID : 7 IoU : 0.92 FC : 6
Obj : car Pr : 0.99 TrackID : 7 IoU : 0.97 FC : 7
No Detection FC : 387
Obj : car ReEntry-lost track Pr : 0.91 FC : 388

Fig.10. IoU tracking of object. In frame 4 (FC:4) there is no object detected. In frame 5 (FC:5), a new object car is detected which is then tracked in subsequent frames in FC:6, FC:7 and further. One unique tracking id is assigned to each class. IoU scores are also provided.

REFERENCES

1. R. K. McConnell, "Method of and apparatus for pattern recognition," *U. S. Patent,* vol. 4, 1986, pp. 567-610.
2. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision*, 1999, pp. 1150-1157.
3. Y. Tian, K. Pei, S. Jana, B. Ray. (2018). DeepTest: automated testing of deep neural-network-driven autonomous cars. Available: https://arxiv.org/abs/1708.08559.v7.
4. A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks", in *Advances in Neural information Processing Systems*, 2012, pp. 1097-1105.
5. Z. Zhao, S. Xu, X. Wu. (2019). Object detection with deep learning: A review. Available: https://arxiv.org/pdf/1807.05511.pdf.
6. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg, "SSD: single shot multibox detector," in *European Conference in Computer Vision*, 2016, pp. 21-37.
7. J. Redmon, S. Divvala, R. Girshick, A, Farhadi, "You only look once: unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.
8. R. Girshick, "Fast R-CNN", in *IEEE International Conference on Computer Vision*, 2015, pp. 1440-1448.
9. S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 2015, pp. 1137-1149.
10. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86(11), 1998, pp. 2278-2324.
11. M. Lin, Q. Chen, S. Yan, "Network in network", in *International Conference on Learning Representations*, 2014, pp. 1-10.
12. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.
13. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818-2826.
14. K. Simonyan, A. Zisserman. (2015). Very deep convolutional networks for large-scale image recognition. Available: https://arxiv.org/abs/1409.1556.
15. K. He, X. Zhang, S. Ren, J. Sun. (2015). Deep residual learning for image recognition. Available: https://arxiv.org/abs/1512.03385.
16. K. He, G. Gkioxari, P. Dollar, R. Girshick, "Mask R CNN," in *International Conference on Computer Vision*, 2017, pp. 2980-2988.
17. J. Li, K. Peng, C. Chang, "An efficient object detection algorithm based on compressed networks," *Symmetry*, vol. 10(7), 2018, pp. 1-13.
18. Y. Ren, C. Zhu, S. Xiao, "Small object detection in optical remote sensing images via modified Faster R-CNN," *Applied Sciences*, vol. 8(5) 813, 2018, pp. 1-11.
19. J. Redmon, A. Farhadi. (2018). YOLOv3: an incremental improvement, Available: https://arxiv.org/abs/1804.02767.
20. M. A. Rahman, Y. Wang, "Optimizing intersection-over union in deep neural networks for image segmentation," in *International Symposium on Visual Computing*, 2016, pp. 234-244.
21. B. Y. Lee, L. H. Liew, W. S. Cheah, Y. C. Wang, "Measuring the effects of occlusion on kernel based object tracking using simulated videos," *Procedia Engineering*, vol. 41, 2012, pp. 764-770.
22. H. Li, Y. Li, F. Porikli, "Robust online visual tracking with a single convolutional neural network," in *Asian Conference on Computer Vision*, 2014, pp. 194-209.
23. H. Li, Y. Li, F. Porikli, "Deeptrack: learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25(4), 2016, pp.1834-1848.
24. N. Wang, D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809-817.
25. A. Agarwal, S. Suryavanshi. (2017). Real-time multiple object tracking (MOT) for autonomous navigation. Available: http://cs231n.stanford.edu/reports/2017/pdfs/630.pdf.
26. S. Yun, J. Choi, Y. Yoo, K. Yun, J. Y. Choi. (2017). Action-decision networks for visual tracking with deep reinforcement learning. DOI: 10.1109/CVPR.2017.148.
27. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar. (2018). Focal loss for dense object detection. Available: https://arxiv.org/abs/1708.02002.

AUTHORS PROFILE

Dr. Chakradhara Panda obtained Ph.D. in Economics (Artificial Neural Network) from University of Hyderabad, India. He worked in academics before joining industry in 2008. Currently, he is working as a senior data scientist at IBM India Pvt. Ltd. His research interest includes applied econometrics, deep learning and machine learning. He has published several research papers in refereed international journals.