Real time Object Detection And Localization: Autonomous Vehicles

Dr. Nalini C Iyer
nalinic@bvb.edu

Akshay Gudiyawar
akshaymg99@gmail.com

Shruti Maralappanavar
shruti_m@bvb.edu

Anoop Revadi
anooprevadi@gmail.com

***Abstract*—** Obstacle detection plays a major role in autonomous vehicles which has to be performed at high speed with good accuracy with constraints in computational resources. To solve this we choose MobileNet_SSD neural network and performed experimentations to increase its performance. We initially carried out testing of MobileNet_SSD with standard convolution networks such as Fast R-CNN, Faster R-CNN and YOLO. In our approach, we present the optimal values of MobileNet_SSD parameters, image input resolutions, aspect ratio to increase its performance in terms of speed and accuracy. We carried out tests on different versions of MobileNet (v1 & v2) and SSD on COCO and ImageNet Datasets. We also compared MobileNet_SSD in different scenarios. Further, we demonstrate the factors responsible for best trade-off between speed and accuracy. To validate the SSD framework, we trained SSD300 and SSD512 architectures on PASCAL VOC Dataset. We successfully performed object detection on testcases such as in-motion frames, multiple-objects. Compared to other single stage methods, SSD has much better accuracy even with a smaller input image size.

Keywords — MobileNet, SSD, Faster R-CNN, COCO

Introduction

Autonomous vehicles have the potential to significantly reduce road-accidents, making them a safer transportation mode. Object detection is a key component of Autonomous Vehicles, they need it to see the surrounding and navigate through it. To help with this cause, object-detection has to be light-weight and low on system resources, so that there won't be any lag in computation and the obstacles are detected instantaneously.

According to *Wei Liu, et. al* [2] MobileNets are built primarily from depthwise separable convolutions initially introduced by *L. Sifre et. al* [13],and subsequently used in Inception models [14] to reduce the computation in the first few layers. Related papers on small networks focus only on accuracy, but do not consider speed. Even the fastest high-accuracy detector, Faster R-CNN, operates at only 7 frames per second (FPS).

On the other hand, *Andrew G. Howard, et. al* [1] proposed Convolutional neural networks which are present everywhere in computer vision ever since AlexNet [15] popularized deep convolutional neural networks by winning the ImageNet Challenge: ILSVRC 2012 [16]. The trend has been to make deeper and more complicated networks in order to achieve higher accuracy. However, these advances to improve accuracy are not necessarily making networks more efficient with respect to size and speed. In many real world applications such as robotics, self-driving car and augmented reality, the recognition tasks need to be carried out in a timely fashion on a computationally limited platform.

Compared to image classification, object detection is a more challenging task that requires more complex methods to solve. Recently, *Ross Girshick, et. al* [3] came up with deep ConvNets [15] which have significantly improved image classification and object detection accuracy. Due to this complexity, current approaches (e.g., [17, 18, 19]) train models in multi-stage pipelines that are slow and inelegant.

Based on the above observations, the mentioned methodologies focus only on accuracy but not speed. To address this gap, we present techniques for tweaking MobileNet_SSD and finding optimal values for its parameters to increase its object detection performance. We performed experiments on MobileNet [1] and SSD [2] to obtain better trade-off between speed and accuracy. Furthermore, by improving the speed of object detection we can have positive effect on many computer vision applications.

The organization of the paper is as follows. In Section I, we explain the Methodology and validation techniques used in tweaking the MobileNet_SSD to increase the performance in-terms of speed and accuracy. Section II details Experimental results on speed vs. accuracy trade-off, varying input image size evaluated on PASCAL VOC, COCO and are compared to range of standard Convolution Networks. We conclude with summary and conclusion in Section III.

I. METHODOLOGY

For the purpose of object-detection for Autonomous Vehicle, we needed a light-weight detection and classification platform. We explored various neural-network Architectures like Fast R-CNN[6], Faster R-

CNN[7], Yolo[8] and SSD[1-2]. As our vehicle had very less system resources to allocate for the object-detection task, we choose MobileNet since it has lower footprint on the computational requirement.

We use MobileNet as the base network, which extracts high-level features along with SSD as detection network. MobileNet architecture is more suitable for mobile and embedded based vision applications where there is lack of computing power. We trained the two variants of MobileNet Architecture v1 & v2 with COCO and ImageNet Dataset having 80 classes, with 80,000 and 50,000 training images respectively

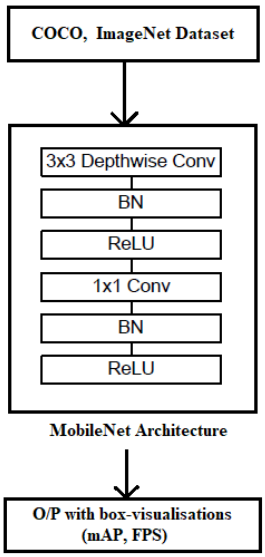

Fig 1. MobileNet Architecture

Mobile architecture uses Depthwise separable convolutions as shown in Fig 1. Batch Normalization (BN) and ReLU activation functions are applied after each convolution. Depthwise separable convolution is a spatial convolution performed independently over each channel of input as shown in Fig 2. In this, a single filter per each input channel is applied and Pointwise convolution, a simple 1×1 convolution is then used to create a linear combination of the output of the depthwise layer.

Performance of MobileNet is better than other CNNs because of the use of Depthwise Convolutions, which has the following advantages:

1. It reduces the number of parameters involved.
2. Total number of Floating point multiplications are reduced which decreases the requirement of computational power

These factors make MobileNet favorable for mobile and embedded vision applications.

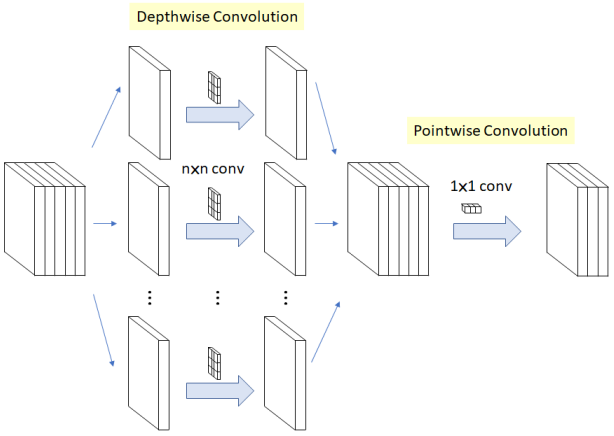

Fig 2. Depthwise Convolution Architecture

MobileNet has two hyper parameters to adapt the architecture to our needs, 'α' – the depth multiplier which changes how many channels are in each layer and 'ρ', which denotes the aspect ratio of input frames. 'α' of 1 corresponds to the default number of channels in the convolutions.

Other sensible choices are 0.75 and 0.5. Reducing the number of channels reduces the number of weights as well as the computational costs. We used mAP (mean average precision) and FPS (Frame per second) metrics to measure and compare the performance of object-detection modules. FPS is calculated by the system considering the number of frames it processes per second, while mAP calculation is based on the accuracy of predictions.

On the other hand, Input image size has significant influence on the speed and accuracy of neural networks. Henceforth, there arises a need to find an optimal value of input frame size by considering the factors of camera resolution and aspect ratio.

SSD detector requires the input images to be in the aspect ratio of 1:1, meaning they should have same height and width. We can't specify a fixed custom ratio for this architecture, but we can vary the image sizes which are fed to the network, and hence based on results obtained from experimentation, optimal values can be chosen to increase the performance.

MobileNet performs better for embedded applications because of the innovation in depthwise separable convolutions. We experimentally validate that MobileNet_SSD outperforms Faster R-CNN and R-FCN in accuracy and speed metrics.

II. EXPERIMENTAL RESULTS AND DISCUSSIONS

A). QUANTIATATIVE ANALYSIS

In this section, we investigate the performance of MobileNet_SSD relative to standard convolution networks such as Fast R-CNN, Faster R-CNN and YOLO. To understand MobileNet_SSD better, we carried out experiments to examine how each component affects performance.

We tested SSD_MobileNet and other CNNs on "07++12" dataset (union of VOC2007 trainval and test and VOC2012 trainval). Experimentation was performed on two models of SSD, which are SSD300 (default 300x300 low resolution input image, faster) and SSD512 (default 512 x 512 input image, higher resolution, more accurate version).

Taking into account the results obtained as shown in Table 1, we can infer that SSD512 performs better than the other Convolution Networks since it has the highest mAP of 74.9 among other CNNs.

Detection network	Dataset	mAP
Fast R-CNN	07++12	68.4
Faster R-CNN	07++12	70.4
YOLO	07++12	63.5
SSD300	07++12	72.4
SSD512	07++12	74.9

Table 1. mAP metric for different CNNs

To further validate the SSD framework, we trained SSD300 and SSD512 architectures on COCO dataset. It is an excellent object detection dataset with 80 classes, 80,000 training images and 40,000 validation images. Table 2 shows the results obtained for different versions of MobileNet_SSD relative to speed (ms) and mAP metrics. Here v2 scores better than v1. The inverted residual bottleneck layers in MobileNet v2 allows a particularly memory efficient implementation which is very important for mobile applications. But in particular, we can infer that SSD_512_MobileNet_v2 outperforms other tested versions.

Model Name	Speed (ms)	mAP
SSD300_Mobilenet_v1_coco	31	22
SSD300_Mobilenet_v2_coco	37	24
SSD512 MobileNet v1 coco	28	18
SSD512 MobileNet v2 coco	39	21

Table 2. Comparison of performance of V1 and V2

In SSD, we cannot specify a fixed custom aspect ratio for input frames (default value is 1:1). The general guidance is to preserve the aspect ratio of the original image while the image size can be of different value. Therefore, we varied the image sizes to find out the best possible performance in SSD network as shown in Table 3.

Module	mAP	FPS	Input resolution
Faster R-CNN (VGG16)	72.3	4.5	1024 x 1024
	73.2	7.1	1000 x 600
	73.1	12.3	480 x 480
	74.8	14.7	300 x 300
SSD300	74.3	22.3	1024 x 1024
	68.7	18.9	1000 x 600
	71.7	34.5	480 x 480
	72.8	46.1	300 x 300
SSD512	72.4	11.2	1024 x 1024
	64.5	14.3	1000 x 600
	69.8	28.7	512 x 512
	72.9	36.4	480 x 480

Table 3. Performance of CNN for different Input resolutions

We found out that the FPS varies accordingly to the input frame size, i.e. inversely proportional to the frame size. For SSD in particular, reducing the image size by half in width and height lowers the accuracy by 15.88% on average but also increases processing time by 27.4% on average.

B). QUALITATIVE ANALYSIS

The results obtained on MobileNet_SSD for custom dataset are shown in Fig 3.

Fig. 3 (a)

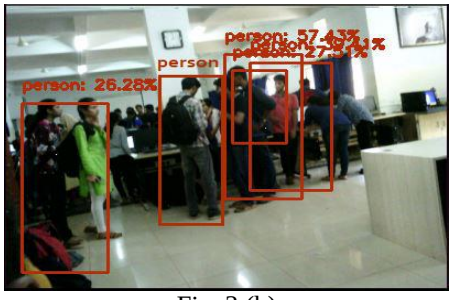

Fig. 3 (b)

MobileNet_SSD handles in-motion frames as shown in Fig.4(b) thus providing robust object detection. This turns to be a vital advantage for Autonomous Driving vehicels since it has to deal with images while in motion.

Fig. 4(a) Input in-motion frame

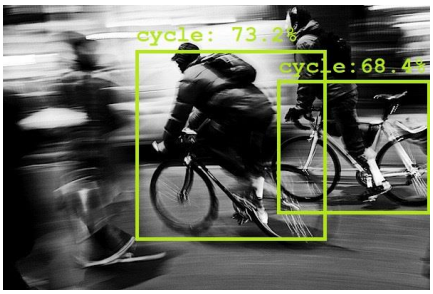

Fig. 4(b) Detection by MobileNet_SSD

In case of multiple objects in the frame, MobileNet_SSD performs better by detecting most of the objects in the frame than other CNN. Fig. 5(b) shows the result obtained on R-CNN, where it only detects larger objects. Fig. 5(c) shows the same frame processed by MobileNet_SSD where even smaller objects are detected.

Fig. 5(a) Input frame for multiple-objects

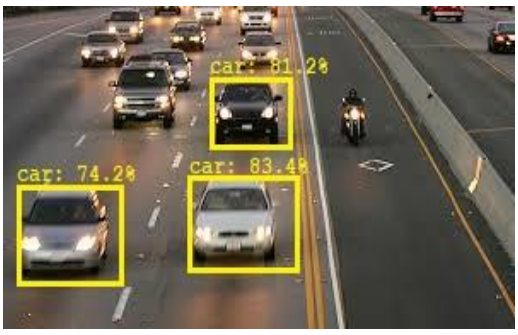

Fig. 5(b) Detection by R-CNN

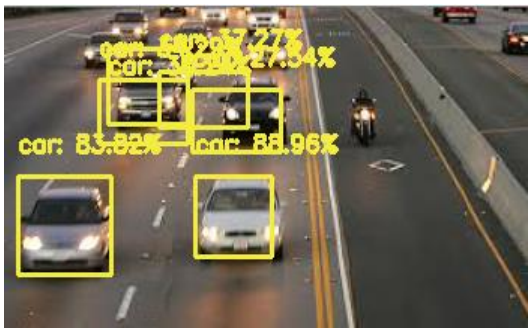

Fig. 5(c) Detection by MobileNet_SSD

Successful Detection of multiple objects in the frame would also be helpful in solving Object Counting problems of Computer vision.

III. CONCLUSION

We used MobileNet_SSD for the task of embedded object detection, and experimentally validated that it performs faster than standard CNNs like Fast RCNN, YOLO. We also found out the optimal values of input image size, SSD variants and backed them up with mAP and FPS calculations. We successfully performed object detection on testcases such as in-motion frames, multiple-objects.

We performed the tests on an Intel i5 powered host computer with NVidia 940MX GPU. These results highly depend on hardware configuration, and hence should be treated as relative speeds. Because of the innovation of depthwise separable convolutions, MobileNet does 9 times less computational work, maintaining the same accuracy compared to than other standard neural network.

SSD on MobileNet has the highest mAP among the models targeted for real-time processing. Since it requires less than 1 GB memory for processing, it has the lowest footprint on system resources. SSD with MobileNet provides the best accuracy for small objects whereas, it performs relatively bad for larger objects.

IV. REFERENCES

[1]. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[3]. Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[4] Jiang, Huaizu, and Erik Learned-Miller. "Face detection with the faster R-CNN." 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017.

[5] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[6] Girshick R. Fast r-cnn. InProceedings of the IEEE international conference on computer vision 2015 (pp. 1440-1448).

[7] Zhang, L., Lin, L., Liang, X. and He, K., 2016, October. Is faster r-cnn doing well for pedestrian detection?. In European conference on computer vision (pp. 443-457). Springer, Cham.

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

[9] Ellis, T. and Xu, M., 2001, December. Object detection and tracking in an open and dynamic world. In *Proc. of the Second IEEE International Workshop on Performance Evaluation on Tracking and Surveillance (PETS'01)*.

[10]. Benenson, R., Mathias, M., Timofte, R. and Van Gool, L., 2012, June. Pedestrian detection at 100 frames per second. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2903-2910). IEEE.

[11] Zhu J, Montemerlo MS, Urmson CP, Chatham A, inventors; Google LLC, assignee. Object detection and classification for autonomous vehicles. United States patent US 8,195,394. 2012 Jun 5.

[12] Franke U, Gavrila D, Görzig S, Lindner F, Paetzold F, Wöhler C. Autonomous driving goes downtown. IEEE Intelligent systems. 1998 Nov 1(6):40-8.

[13] L. Sifre. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014. 1, 3

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015. 1, 3, 7

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 1, 6

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252, 2015. 1

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In ICLR, 2014. 1, 3

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1, 3, 4, 8

[19] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014. 1, 2, 3, 4, 5, 6, 7

[20] Sphoorti Kunthe, Shreya Bhat, Vinuta kadiwal, Nalini C. Iyer, Shruti Maralappanavar . Kalman Filter Based Motion Estimation for ADAS Applications,presented at the S, IEEE International Conference on Advances in Computing, Communication Control and Networking.

[21] Prateek K. Gaddigoudar, Tushar R. Balihalli, Suprith S. Ijantkar, Nalini C. Iyer and Shruti Maralappanavar. (2017). Pedestrian detection and tracking using particle ltering. 110-115. 10.1109/CCAA.2017.8229782.

[22] Shruti Maralappanavar, Nalini C. Iyer, Meena M (2018) Pedestrian Detection and Tracking: A Driver Assistance System. Paper presented at the Fifth International Conference on Emerging Research in Computing, Information, Communication and Applications ERCICA, Bangalore, India, 2018.