Multiple Real-time object identification using Single shot Multi-Box detection

Kanimozhi S
Department of Information Science and Technology
Anna University,CEG Campus,
Chennai, India,
kanimozhi@auist.net

Gayathri G
Department of Information Science and Technology
Anna University,CEG Campus,
Chennai, India,
demand4great@gmail.com

Mala T
Department of Information Science and Technology
Anna University,CEG Campus,
Chennai, India,
mala@auist.net

***Abstract*— Real time object detection is one of the challenging task as it need faster computation power in identifying the object at that time. However the data generated by any real time system are unlabelled data which often need large set of labeled data for effective training purpose. This paper proposed a faster detection method for real time object detection based on convolution neural network model called as Single Shot Multi-Box Detection(SSD).This work eliminates the feature resampling stage and combined all calculated results as a single component. Still there is a need of a light weight network model for the places which lacks in computational power like mobile devices(eg: laptop, mobile phones, etc). Thus a light weight network model which use depth-wise separable convolution called MobileNet is used in this proposed work. Experimental result reveal that use of MobileNet along with SSD model increase the accuracy level in identifying the real time household objects.**

Keywords—Object Detection, TensorFlow object detection API, SSD with MobileNet

I. Introduction

Object detection is a general term for computer vision techniques for locating and labelling objects in the frame of a video sequence. Detecting moving objects or targets, and tracking them on real-time video is a very important and challenging task. As many methodologies have been followed so far in detection mechanism still level of accuracy is not up to the mark. So the neural network method came into existence for detecting objects present in a video sequence. One among them is deep neural network which elaborate the hidden layers so that the level of accuracy in detecting the object present in a video can be highly improved. Among which R-CNN in 2014 is based on deep convolution neural network, was initially used for the detection mechanism. After that some other improved methods like Spp-net , fast R-CNN , faster RCNN and R-FCN came in the field. Due to their complex networks structure it cannot be applied for identification of multiple real-time object in a single frame.

So there comes a need for having single network and also faster performance. Thus Single Shot Multi-Box Detector is based on VGG and has additional layers as feature extraction layers was designed especially for real time object detection which eliminates the need of region proposal network and speeds up the process. To recover the drop in accuracy SSD applies some changes in multi-scale features and default boxes concept. The SSD object detection composed of 2 parts: i) Extract feature maps and ii) Apply convolution filters to detect objects.

As SSD uses convolution filters for detecting the objects depth it lose its accuracy if the frame is of low resolution. So we use the MobileNet technique as it use depth wise separable convolution which significantly reduces the parameters size when compare with normal convolution filters of same depth. Thus we can get the light weight deep neural network as a result. On whole SSD with MobileNet is nothing but a model in which meta architecture is SSD and the feature extraction type is MobileNet.

A. Problems Statement

The main aim is to make a real time object discovery system which can run as a lightweight application on system with i7 processor. Initially we have used our system web camera (with resolution of 640 x 480) to test how successfully the moving object was caught lively(draw a bounding box and label of object).

The main challenge of this project is in the aspect of accurateness. Evaluation metrics used in this paper were: Detection Speed (3-4 fps) and size of the model used.

B. Related Works

Most of the CNN based detection methods for example R-CNN[4], starts by recommending different locations and scales present in a test image as a input to the classifiers of objects, at the point of training and return the classifiers of resultant proposed region to detect an object. After classification, post-processing is carried out to rectify the bounding boxes along with re scoring the boxes on the basis of other objects in that frame.

978-1-5386-9471-8/19/$31.00 ©2019 IEEE

Then comes some of the improvised version of R-CNN, like Fast- RCNN[3] and Faster-RCNN[10], which utilize much policies in order to reduce manipulation of region proposal and reach the detection speed of about 5 FPS on a K40 GPU device. Faster RCNN works well on detecting of objects over traffic dataset which is KITTI[2] dataset. But this method also fails to provide a new score level in detection speed for real-time data, which clearly explains that still there is a need of lot more work in improving the inference speed for real life data.

However the problem in improving inference speed for real life data was overcome in YOLO[9] system, through the method of combining the region proposal with that of classification to a unique regression problem directly from the image pixel to bounding box coordinates also with class probabilities and assess the entire image in a single run. As a entire detection pipeline is a unique network, it can enhance direct end-to-end detection performance well.

The only framework which can provide 45 FPS (on GPU) and mAP value of about 63.4% on VOC2007 (real-time data) was YOLO. Still it faces problems in identifying of smaller objects in that frame. This problem was rectified using SSD [8] which follows the policy of combining anchor box proposal system of faster-RCNN and uses muti-scale features for performing detection layer. The mAP value on VOC2007 was increased to 73.9% by preserving the detection speed same as that of YOLO.

SSD shows improvised result if it goes with the models like SSD300 and SSD512[13] over the ship detection. Both the models gives less false rate in detection and accuracy rate of about 0.95 .

MobileNet [1] is based on depthwise convolution model which implies single input to each filter . Design model for MobileNet can be either thinner or shallower. To make MobileNet lightweight, the 5 layers of separable filters with feature size 14×14 × 512 has to be there. Thus MobileNet model should be thinner which gives 3% better performance than shallow model.

II. TECHNICAL APPROACH

This section consists of our proposed method SSD with MobileNet which we used for recognizing real-time object in the form of an application in detail. Section A describe about the way through which API created in Tensorflow for Object Detection is used in detecting objects. Section B contains the faster working procedure SSD in real time objects detection. Section C talks about how the desktop application is changed to lightweight application using MobileNet

A. TensorFlow Object Detection API

The Tensorflow API for object detection is a framework built over TensorFlow which make it simple for training and deploying of various object models.

It is an interface for object Detection by expressing machine learning and implementing algorithms. To do this object detection and tracking, we have used this TensorFlow API for object detection. Developing a learning model which can localize and correctly detect multiple object in a single frame is still a challenging task of computer vision. A calculation communicated utilizing TensorFlow can be executed with almost no change on a wide assortment of heterogeneous frameworks, extending from cell phones and tablets up to substantial scale dispersed frameworks of several machines and a huge number of computational gadgets, for example, GPU cards. The framework is adaptable and can be utilized to express a wide assortment of algorithm, including training and algorithms for deep neural network models.

B. Depth wise separable convolution

MobileNets works based on the depth-wise separable convolution (DSC) layer and uses some inception model to reduce computational cost of layers , as in Figure. 1(a).

Standardized convolution method has been replaced by Depth-wise approach because of two reasons: 1. In depth-wise approach, spatial convolution performed independently over each channel of an input; 2. point-wise convolution (Figure-1(b)) :In which a simple layer for convolution is used for projecting the channel information from depth-wise into a new channel space. As DSC has fewer parameter then regular convolution layers they also required only less operation to compute. Hence it is cheaper and faster.

Parameter size of standard convolution layer is:

$$(K \times K \times E_1 \times E_2) \qquad (1)$$

Its computational cost is:

$$(K \times K \times D_A \times D_A \times E_1 \times E_2) \qquad (2)$$

Parameter size of Depth-wise separable:

$$(K \times K \times E_1 + 1 \times 1 \times E_1 \times E_2) \qquad (3)$$

Its computational cost:

$$(K \times K \times D_A \times D_A \times E_1 + 1 \times 1 \times E_1 \times E_2) \qquad (4)$$

The reduction of computation cost is therefore:

$$(K\times K\times D_A\times D_A\times E_1+E_1\times E_2\times D_B\times D_B) / (K\times K\times D_A\times D_A\times E_1\times E_2\times D_B) = (1/B) + (1/K^2) \qquad (5)$$

Thus the reduced parameter is given as:

$$(K\times K\times E_1+1\times 1\times E_1\times E_2) / (K\times K\times E_1\times E_2) = (1/E_2) +(1/K^2) \qquad (6)$$

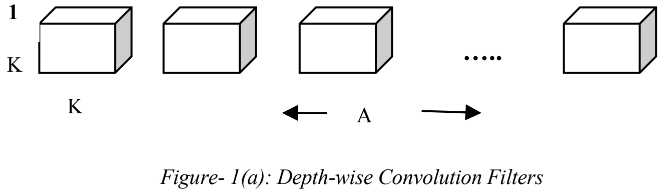
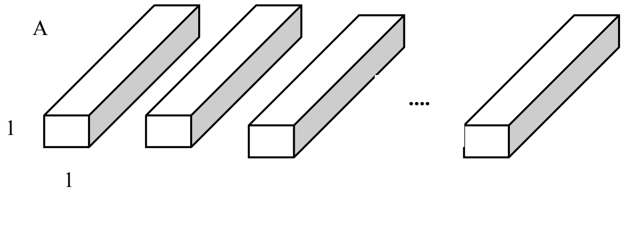


Figure- 1(a): Depth-wise Convolution Filters



978-1-5386-9471-8/19/$31.00 ©2019 IEEE

Figure- 1(b): point-wise Convolution Filters

Thus MobileNet utilizes 3×3 depthwise distinguishable convolutions which utilizes between 8 to multiple times less calculation than standard convolutions at just a little decrease in precision.

C. Model Structure

Our network model is shown in Figure-2 which consist of various of DSC module. The layers in the DSC module are ReLU, batch normalization, depth-wise and point-wise operations. First layer in the module is standard convolution while the ending layer is an average pooling which helps in reducing the spatial resolution to 1.

In whole, the developed model is like VGG network, which evacuate the utilization of residual connections for faster computation. MobileNet spends 95% of computation time in standard layer.

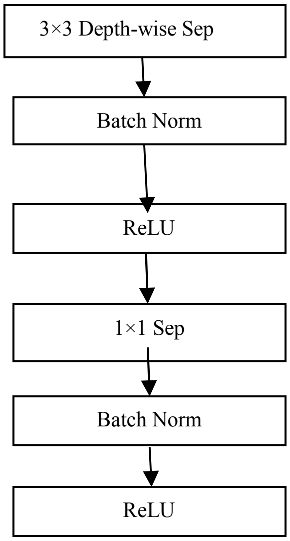


Figure-2: structure of a typical Depth-wise Separable Convolution module

III. Single Shot Multibox Detector (SSD) with MobileNet

We have implement a version of MobileNet called Mobile-Det, a jointly version of both MobileNet classifier and Single Shot MultiBox Detector (SSD) structure [13]. To check the benefits of utilizing this combined version and do a reasonable examination with other state-of- art models [VGG based SSD, YOLO] we have developed this jointly version. The aspect of SSD is huge and far extent of this project so we will just have a short prologue to how it functions in the accompanying parts.

Generally SSD uses various feature layers as classifiers, in which a set of different aspect ratio in the form of default boxes at each place a convolutional way is used to evaluate each feature map. Also every classifier predicts the class scores and shape offset score with respect to the boxes. At the time of training , the correctness in predicting the default boxes is taken into accounts only if its jaccard overlap with the ground truth box has threshold score more than 0.5. Remaining scores that are not falls under the predicted category are then computed using confident score and also localization score.

Figure-3 shows the structure of Mobile-Det which has the framework structure same as that of SSD-VGG-300. But instead of using VGG in our work we are using MobileNet as a base . Furthermore depth-wise separable convolution method is restored instead of standard convolution in our approach.

Finally it is evident that implementing SSD framework works good in training the image completely instead of depending on reference frame. Consequently the temporal information progressively precise in principle too. But main problem in this model is that it become very slow as more convolutions are included.

A. Experimental Setup :

Our image data is taken from our own customized dataset similar to the COCO dataset. We have created dataset with the images that are used in day today life like cell phones, water bottle, person etc. For each objects samples of nearly 500 images were taken. Among them 60% of data is utilized in training phase and 40% of data is utilized in testing phase with distributions of images and objects across the training/validation and test sets. Hence validation set contains 300 stilled frames and testing contain 200 stilled frames.

With the assumption that the input to the classifier contains one object so region of interest is drawn to only one targeted object in the training set. If the web camera image contain multiple objects then region of interest is drawn over the targeted image and extracted separately finally stored.

B. Experimental Results

In this figure, we have tested our project object detection and tracking using the web camera in a laptop with i7 processor. In this image, we can found that the labelled objects have been detected and tracked while the web camera is running. By using the pre-defined dataset COCO (Common Objects in COntext), we have already labelled the objects by giving its images which have been the default dataset COCO. For customized object detection a new object which was captured by the web camera are labelled and store in our own personalized dataset.

In this paper we have used mAP as a performance evaluation criteria which shows (Figure-4) of about 60.6% as a result. Precision indicates the ratio of true positive result to that of total detection results, which rely in Eq. (7), and recall indicates the correct detection to all objects, as shown in Eq. (8).

$$\textit{Precision} = (\#\ \textit{true positive}) / (\#\ \textit{Total detection}) \qquad (7)$$

$$\textit{Recall} = (\#\ \textit{true positive}) / (\#\ \textit{Total ground truth}) \qquad (8)$$

After training and testing the images result given by our approach are shown on the Table-1.

978-1-5386-9471-8/19/$31.00 ©2019 IEEE

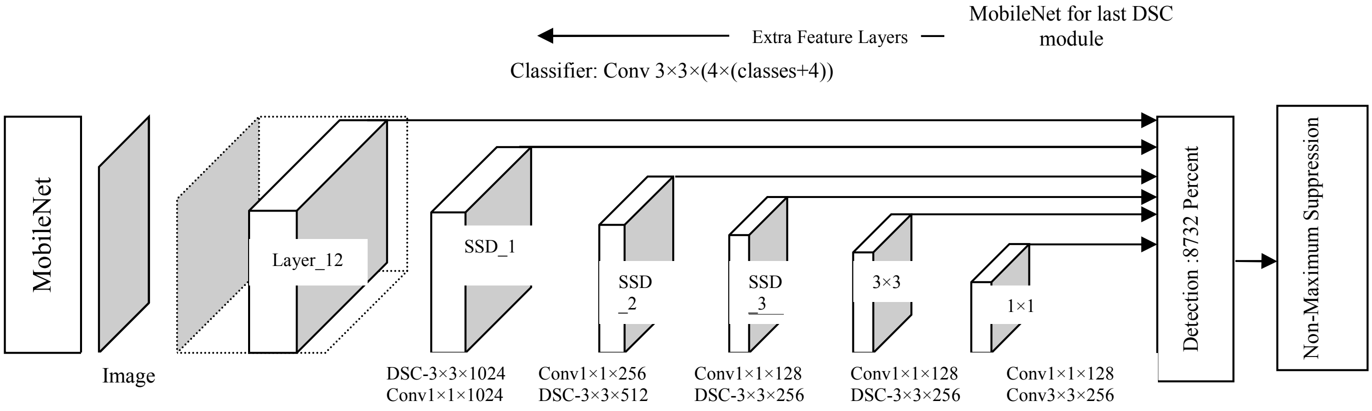


Figure - 3: SSD with MobileNet

Table-1: Experimental results on different objects

Model	**Person**	**Clock**	**Cellphone**	**Bottel**	**Chair**
Total Images	450	320	440	350	400
TP	430	300	425	342	200
TN	5	8	10	3	0
TD	445	316	430	345	210
Precision	0.966	0.949	0.988	0.99	0.95
Recall	0.95	1	0.986	0.997	0.571

From the above table it is clear that our model has correctly detected the objects like person, Clock, Cellphone, Bottle at highest detection rate except one object which is chair. Because the chair image which has handle like structure are wrongly compared with some other object like briefcase. So it is necessary to differentiate each type of chairs by uniquely represent the model type separately.

Thus the tabulation has clearly demonstrates the effectiveness of our approach over detecting objects that are used in our daily life.

C. Detection Results

In this section we have projected our result i.e. how much percntage actually detected vs correctly predicted rate for each trained and tested objects using the model. In the above graph x-axis represents the objects such as person, clock, cellphone, bottle, chair and y-axis represent the rate of detection from 0-1.

Figure-5 is the final output screen which we got after implementing MobileNet and SSD model. But for detecting one single object it takes nearly 3s. Also the objects need to be at the distance of 30meters to the web camera. If the object is away from that distance the detection rate is decreased because of the poor web camera pixel capacity is 1.3mp.

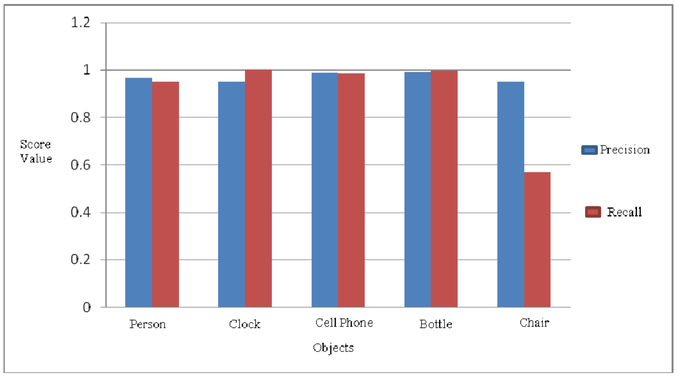
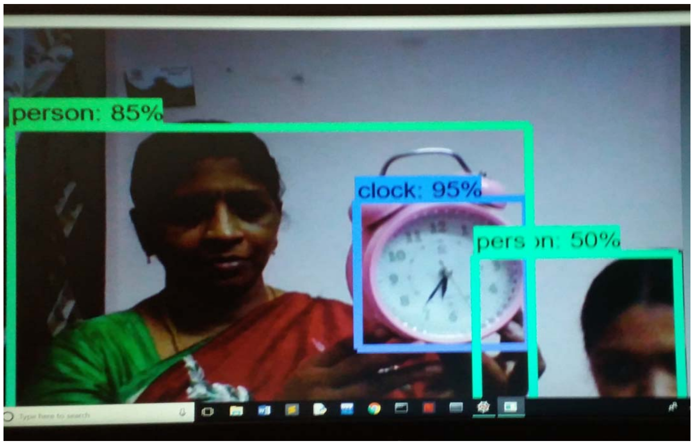


Figure- 4: Detection rate of MobileNet with SSD approach

Hence it is necessary to improve the camera quality if we want to detect all the objects inside a closed surface.



978-1-5386-9471-8/19/$31.00 ©2019 IEEE

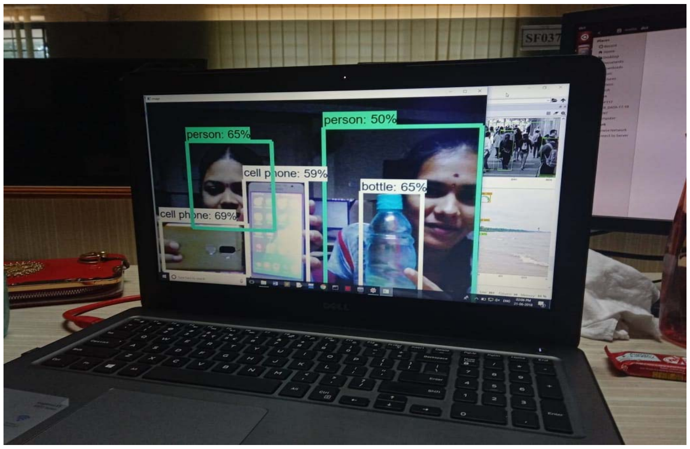


Figure- 5: Example objection detection results using MobileNet SSD

IV. CONCLUSION AND FUTURE WORK

In this paper we have tried to recognize the object which we have shown in front of a webcamera. The developed model was tested and trained using TensorFlow Object Detection API a frameworks which was created by Google. Reading a frame from web camera causes lot of issues so there is a need for good frames per second technique to reduce Input / Output issues. So we focused on threading methodology which improves frames per second a lot so that processing time for each object is greatly improved. Even though the application identify correctly each object in front of webcam it take nearly 3-5 seconds to move the object detected box over next object in that video. By using this work, we can able to detect and track the object in sports field to make the computer to learn Deeply which is none other than the application of Deep Learning.

By detecting the Ambulance Vehicle in the traffic using the Public Surveillance Camera, we can make control of the Traffic Signals. We can also detect the Crime in the Public Place by tracking the abnormal behaviour of the People. Even, we can able to apply this project in the Satellite for Preventing the Terrorist Attack by Tracking their Movement in the Border of our India. Thus, the project will be useful to detect and track the objects and make the life easier.

REFERENCES

1. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In ArXiv , 17 Apr 2017.
2. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The kitti dataset". International Journal of RoboticsResearch (IJRR), 2013.
3. R. Girshick. "Fast r-cnn", In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.
4. T. D. R. Girshick, J. Donahue and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". Computer Vision and Pattern Recognition (CVPR),2014, 2014.
5. Google. Tensorflow Graph Transform Tool. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/tools/graph_ transforms/README.md.
6. Hideaki Yanagisawa, Takuro Yamashita, Hiroshi Watanabe, "A Study on Object Detection Method from Manga Images using CNN", In IEEE, 2018.
7. Hui Eun Kim, Youngwan Lee, Hakil Kim, Xuenan Cui. "Domain-Specific Data Augmentation for On-Road Object Detection Based on a Deep Neural Network" . In IEEE Intelligent Vehicles Symposium , Pages 103-108, 2017.
8. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: "Single shot multibox detector",.In European Conference on Computer Vision, pages 21–37.Springer, 2016.
9. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016.
10. S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In Advances in neural information processing systems, pages 91–99, 2015.
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.
12. Yuanyuan Wang , Chao Wang, Hong Zhang , Cheng Zhang , and Qiaoyan Fu, "Combing Single Shot MultiBox Detector with Transfer Learning for Ship Detection Using Chinese Gaofen-3 Images", In Progress In Electromagnetics Research Symposium, Pages 712-716, November 2017.
13. Yuanyuan Wang , Chao Wang , Hong Zhang. "Combining single shot multibox detector with transfer learningfor ship detection using sentinel-1 images" . In IEEE, 2017.
14. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. "SSD: Single Shot MultiBox Detector". In ArXiv , 30 Mar 2016

978-1-5386-9471-8/19/$31.00 ©2019 IEEE