© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

Development of an Object Detection in Live Video Using Tensorflow

JIMOH BABATUNDE OLAWALE

Abstract- ***This is an android-based system for object detection. It provides object detection in the near area for video challenged users. Object detection is the process of detecting and defining objects of a certain known class in an image. This system helps visually impaired people to identify any objects in his or her environment like a cars, house, cat, dog, chair, table, phone, laptop etc. Here, we have explored the possibility of implementing object detector on an ubiquities mobile devices powdered by Android capable of maintaining real time frame rate while keeping high precision using Tensorflow Object detection API and pretrained ssd mobilenet v2 model. This system proven to be more efficient for object detecting challenged people.***

I. INTRODUCTION

Over the past decade years, computer vision is excelling in the field of segmentation, feature extraction and object detection from image data. It has attracted more research attention and is gaining more immense interest from a different application such as healthcare, traffic monitoring, surveillance, robotics etc. [1]. The ability to detect the object more precisely is an essential factor due to its application in sensitive domain. However, with the development in information technology, optimum computer processing capabilities and affordable hardware's (camera, data storage, etc.) attracts the researchers to emerge in the field of analyzing high dimensional images and videos.

The deep learning boosted the growth of computer vision and produced state-of-the-art results in image recognition, feature extraction, and object detection. The deep learning requires a significant amount of dataset to train and a high computation power. The graphical processing units fulfill the requirement of computer vision to process efficiently. In the field of computer vision, object detection is an important task. The object detection is a process in which the instances of the objects are detected for a particular class in an image. The object detection is trending due to its applicability in a broader area [8].

Object detection is the process of detecting and defining objects of a certain known class in an image. Only a few years back, this was seen as a hard problem to solve. Before Krizhevsky presented the CNN AlexNet [2] at the ImageNet Large Scale Visual Recognition Competition in 2012, researchers were struggling to find a solution to image classification with very low error rate. Since then, many object detecting methods applying CNN has been presented showing great performance and efficiency. However, much computing power is still needed to run visual tasks effectively and on a device with limited hardware resources, running an object detecting system can be challenging.

Here, we have explored the possibility of implementing object detector on an ubiquities mobile devices powdered by Android capable of maintaining real time frame rate while keeping high precision using Tensorflow Object detection API and pretrained ssd mobilenet v2 model.

II. PROBLEM STATEMENT

Real-time object detection requires a lot of processing power and on a system with limited performance, achieving a speed that can be considered as real-time is a challenging task. There are many different methods that can be used to detect objects. One of the most popular method is called SSD. This method is implemented on an Android device to see if it is suitable to run on such low performance hardware. An implemented object detector is considered suitable if it achieves high enough accuracy and frame rate to be useful in practical applications. The evaluation is done by running a few tests on the detector, measuring how it perform in detection accuracy, inference time and frame rate.

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

III. LITERATURE REVIEW

(Huang et. al. 2017): Research about speed and accuracy trade-offs for modern object detectors where they discussed some of the main aspects that influence the speed and accuracy of object detectors. Some new techniques were identified for improving speed without sacrificing much accuracy. They discovered that limiting the number of region proposals in Faster R-CNN could reduce computation significantly without affecting the precision score too much. They also observed how they could decrease input resolution to reduce inference time and still get high performance on large objects. However, the results were significantly worse on small objects.

(Velasco-Montero et al. 2018): They investigated the performance of real time DNN inference on a Raspberry Pi. They compared four different frameworks and four popular deep neural network models used for image classification. The authors demonstrated that it was possible to achieve real time inference speed on a Raspberry Pi Model B. This was however done with image classification. Object detection, which is the focus of this paper is a more computationally demanding task.

(Chandan G. et. al. 2019): Developed a Real time object detection and tracking using Deep Learning and OpenCV. They combined Single shot detector SSD and Mobile Net Algorithm to build an algorithm that can detect and track object. The authors demonstrated the algorithm by developing an application using OpenCV running in Ubuntu IDE. They trained the total of 21 objects and they obtained result from scanning, detection and tracking of video sequence provided by camera.

(Zhong-Qiu Z. et al. 2019): Review object detection in deep learning. The authors used several algorithms like R-CNN, SPP-Net, Fast RCNN, Faster R-CNN, R-FCN, FPN, YOLO, SSD and YOLOv2 using different framework (Caffe, Caffe, Caffe, Caffe, Caffe, Tensorflow/Keras, Tensorflow, Darknet, Caffe and Darknet) and programing language (Mathlab, Mathlab, Python, Python/Mathlab, Mathlab, Python, Python, C, C++, C) respectively. Finally, after their review on several algorithms, they provided valuable insights and guideline for future progress.

(Asim Suhail et. al. 2020): Reviewed a Convolutional Neural Network Object Detection. They reviewed benchmark datasets that are used and the different directions of the object detection i.e. Salient object detection, Objectness object detection, and Category-Specific object detection. The salient detection is based on the distinguished objects which are highlighted and detected. The objectness goes with all the possible chances of the object in the image from the feature map, whereas the category-specific based detection is based on the classification, it discriminates the image into classes through the pre-trained functions.

(Baohua Qiang et. al 2020) proposed an object detection algorithm combined with semantic segmentation for images. Their algorithm used the hourglass structure to regenerate multi-scale features, and the feature mapping used to predict small targets also has rich abstract features. Finally, multi-scale feature was used to predict the location and category of the object in real time.

IV. OBJECT DETECTION IN TENSORFLOW

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings or cars) in digital images and videos. Typically, we have three steps involve in object detection algorithm: Training data, Classification and Testing Data.

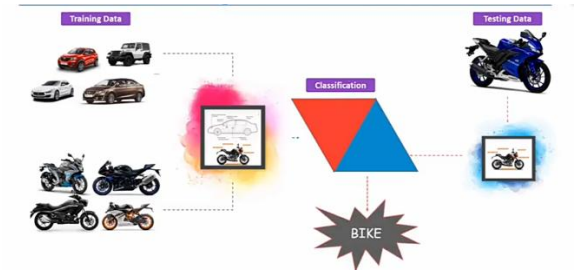


Fig 1. Workflow of Object Detection Algorithm

Here, Object detection is done using tensorflow, we provide our input which is a set of images and using tensoflow we train our model using deep learning. The aim of training the model is to extract the features, which are visual features based on edge detection, the half features, the facial recognition and many more. Now when these features are extracted and the model

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

is created. To test the model, we provide our test data which is again a set of images that are converted to numpy array in the tensorflow object detection so the image computation will be easy. Then, we generate the TFRecord of the images that will be used to train the model.

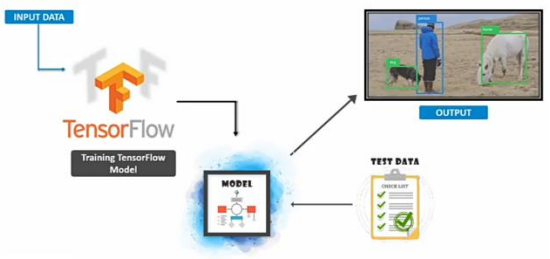


Fig 2 Workflow of Object Detection in TensorFlow

The TFRecord which is the tensorflow record that contain the record of the image along with the tags. As we can see the person tag, dog tag and horse tag from the output above.

V. METHODOLOGY

Here, we perform object detection using Tensorflow and deployed it on android devices for testing and evaluation. The following steps are taken.

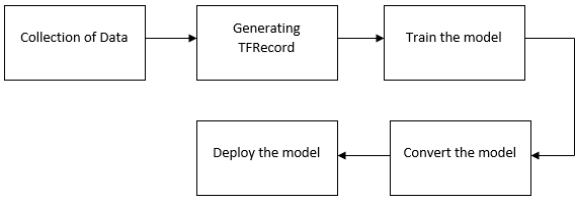


Fig 3 Flow Diagram of Objection detection using Tensorflow on Android

Collection of Data: To train a model we need a dataset which are set of images to be trained. Here COCO Dataset is employed. COCO is a large-scale object detection and captioning dataset with over 330K images, labelling over 200K images with 80 Object categories [5].

Generating TFRecord: Since Tensorflow object detection API is used for training the model we can't feed png or jpg images to the training script. First we have to convert them to TFRecord. TFRecord is the TensorFlow's own data storage format. But fortunately enough. The COCO dataset used is already in that format. For a custom dataset, we run the *generate_tfrecord.py* python scripts.

Train the model: To train the model we will use pretrained model as our initial checkpoint. This way we are not training our model from scratch and it will take less time for our new model to get trained. But here, we used the pretrained model from Tensorflow object detection API which is (ssd_mobilenet_v2_quantized_300x300_coco_2019_01_03.tar.gz). [10] This is model is trained based on Single shot MultiBox Detector algorithm [9].

Convert the model: Since the model will be deployed on android devices, it needed to be converted to Tensorflow lite using Tensorflow lite converter. Tensorflow Lite is designed to execute models efficiently on mobile and other embedded devices with limited compute and memory resources [11]. Some of this efficiency comes from the use of a special format for storing models. Tensorflow models must be converted into this format before they can be used by Tensorflow Lite. Converting models reduces their file size and introduces optimizations that do not affect accuracy. The Tensorflow Lite converter provides options that allow you to further reduce file size and increase speed of execution, with some trade-offs.

Deploy the model: Here the pretrained model above is deployed in android device by developing a camera like app in android studio where the two files obtained from the above step *detect.tflite* and *labelmap.txt* are imported to the access folder of the application. The *detect.tflite* is the converted model from the above while the *labemap.txt* is the classification of the object trained.

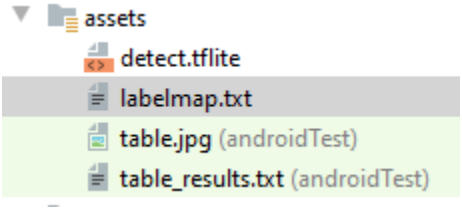


Fig 4 Assets folder in Android studio

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

Here, we have over 80 classes of object trained in the pretrained model used.

```
person
bicycle
car
motorcycle
airplane
bus
train
truck
boat
traffic light
fire hydrant
???
stop sign
parking meter
bench
bird
cat
dog
horse
sheep
```

Fig 5 LabelMap File showing clasees of Object trained.

In summary, here are the list of classes of object trained for object detection in the model.

Table 2.1: List of classes of Object trained for object detection deployed on Android

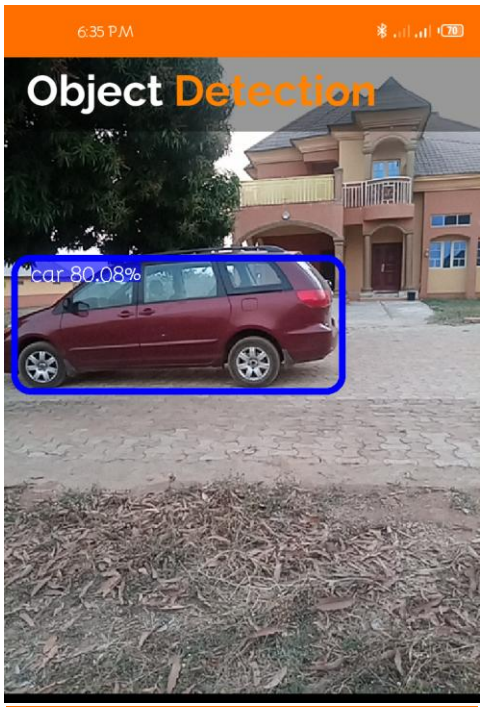
person	bicycle	car	motorcycle	airplane
bus	train	truck	boat	traffic light
fire hydrant	stop sign	parking meter	bench	bird
cat	dog	horse	sheep	cow
elephant	bear	zebra	giraffe	backpack
umbrella	handbag	tie	suitcase	frisbee
skis	snowboard	sports ball	kite	baseball bat
baseball glove	skateboard	surfboard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

VI. SIMULATION AND RESULT

Based on the algorithm used to train the model, which is the Single Shot MultiBox Detector Algorithm SSD, the deployed model on android phone after successfully training over 80 classes object and the result are obtained after successfully scanning, detection and tracking of video sequence provided by the camera in android mobile phone.

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

Fig 6 Detection of Sink and Mouse with confidence level of 63.2% and 67.9% respectively.



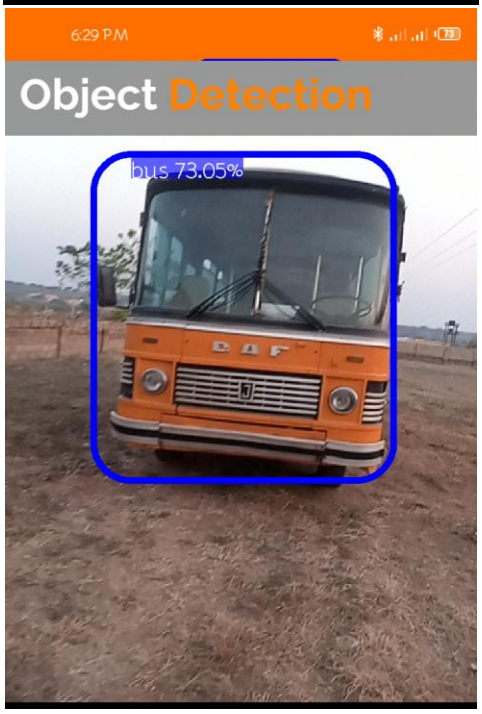


Fig 7 Detection of Car and Bus with confidence level of 80.0% and 73.0% respectively.

© JAN 2023 | IRE Journals | Volume 6 Issue 7 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I7-1711294-6013

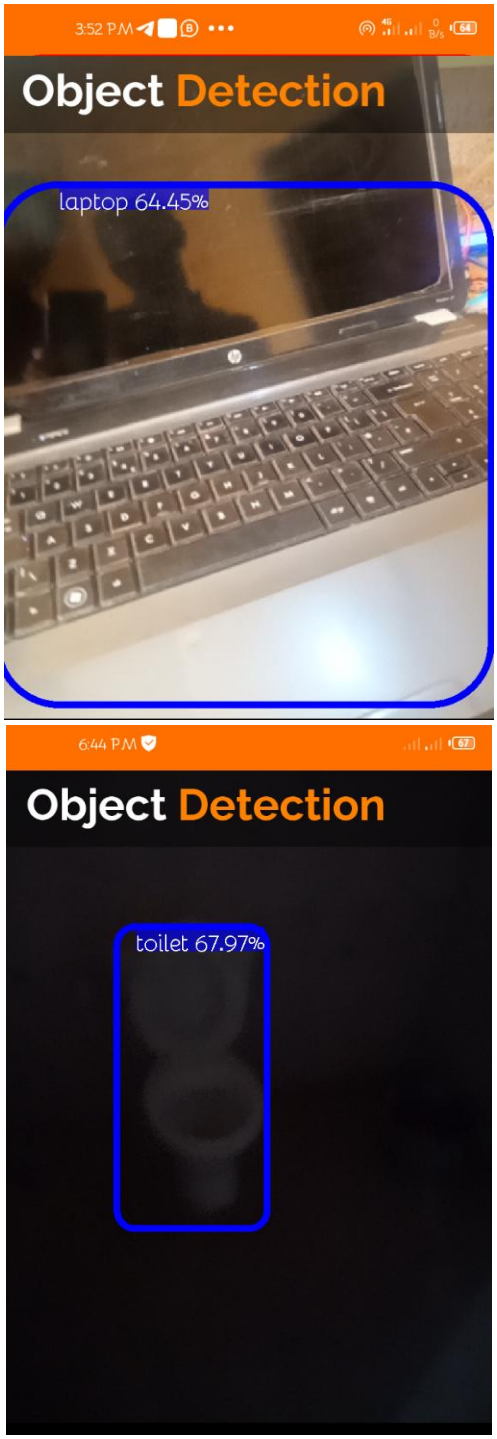


Fig 8 Detection of Laptop and Toilet with confidence level of 64.4.0% and 67.9% respectively.

The figure 6-8 shows the real time detection of some objects with good confidence level (above 60%) . The model was trained to detect over 80 classes of object listed above in the table with good percentage of accuracy.

CONCLUSION

The aim is to investigate the suitability of running a real time object detection system on a mobile phone (Android) with good confidence level, speed and accuracy. The SSD algorithm used to train the model has proven to be efficient and effective. This model showed excellent detection and tracking results on the object trained and can further utilized in specific scenarios to detect, track and respond to the particular targeted objects in the video surveillance.

REFERENCES

[1] Asim Suhail, Manoj Jayabalan, Vinesh Thiruchelvam, “Covolutional Neural Network Based Object Detection: A Review”, Journal of Critical Reviews, ISSN- 2394-5125, Vol 7, Issue 11, 2020.

[2] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, vol. 60, no. 6, pp. 84-90, 017.

[3] Baohua Qiang, Ruidong Chen, Mingliang Zhou, Yuanchao Pang, Yijie Zhai and Minghao Yang, “Convolutional Neural Networks-Based Object Detection Algorithm by Jointing Semantic Segmentation for Images” retrieved [Online] from www.mdpi.com/journal/sensors.

[4] Chandan G, Ayush Jain, Harsh Jain, Mohana, “Real Time Object Detection and Tracking Using Deep Learning and OpenCV”, International Conference on Inventive Research in Computing Applications (ICIRCA 2018) pp.1305-1308.

[5] “COCO 2017 Dataset” COCO, 2017 “[Online] Available: https://cocodataset.org/#home [Accessed: 12-Jan-2021]

[6] D. Velasco-Montero, J. Fernández-Berni, R. Carmona-Galán and Á. Rodríguez-Vázquez, "Performance analysis of real-time DNN inference on Raspberry Pi", Real-Time Image and Video Processing 2018, 2018.

[7] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[8] S. Kamate and N. Yilmazer, "Application of Object Detection and Tracking Techniques for Unmanned Aerial Vehicles," in Procedia Computer Science, 2015, vol. 61, pp. 436–441.

[9] "Single Shot MultiBox Detector" Computer Vision and Pattern Recognition, 2020. [Online] Available: https://arxiv.org/abs/1512.02325 [Accessed: 13-Jan-2021]

[10] "TensorFlow 2 Detection Model Zoo" Tensor Flow" [Online] Available https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md [Accessed: 1-Jan-2021]

[11] "TensorFlow Lite Object Detection" Tensorflow Lite" [Online] Available https://www.tensorflow.org/lite/models/object_detection/overview [Accessed: 13-Jan-2021]

[12] Zhong-Qiu Zhao, Shou-tao Xu, Xindong Wu, Object Detection with Deep Learning: A Review, College of Computer Science and Information Engineering, Hefei University of Technology, Apr - 2019, China.