Deep Lite: Bringing Tensor flow model to Edge Device for offline-Inferencing

Rohith Kumar Nagulapati[1], Nageswara Rao Nandigam[2]
School of Computing and Engineering,
University of Missouri - Kansas City
Kansas City, USA

Abstract— The gap between existing heavy weight neural network models to work on mobile platform made us to bring this paper. In this paper, we proposed a Tensor flow architecture that brings pre-trained model to user's device for custom training and inferencing taking advantage of tflite version of the existing model. The 1001 categories dataset is used to build the mobile-net model on the server and then brought to user's device using established connection through socket. An Android Application is developed to load the model to user's device for situation learning

I. INTRODUCTION

Tensor flow was designed to be a good deep learning solution for mobile platforms. Currently there are two solutions for deploying deep learning applications on mobile and embedded devices they are Tensor flow for Mobile and Tensor flow Lite. Tensor flow Lite is an evolution of Tensor flow Mobile enabling on-device machine learning inference more importantly with low latency and a small binary size.

Tensor flow Lite supports a set of core operators, both quantized and float, which have been tuned for mobile platforms. They incorporate pre-fused activations and biases to further enhance performance and quantized accuracy supporting custom operations in models. It defines a new model file format, based on Flat Buffers that does not need parsing/unpacking step for secondary representation. Also, the code footprint of Flat Buffers is smaller than protocol buffers.

Tensor flow Lite has a new mobile-optimized interpreter, which has the key goals of keeping apps lean and fast. The interpreter uses a static graph ordering and a custom (less-dynamic) memory allocator to ensure minimal load, initialization, and execution latency. Java API: A convenience wrapper around the C++ API on Android.

II. RELATED WORK

Single-machine frameworks:

Many machine learning researchers carry out their work on a single—often GPU equipped—computer [1, 2], and several single-machine frameworks support this scenario. Caffe [3] is a high-performance framework for training declaratively specified neural networks on multicore CPUs and GPUs. It is easy to compose models from existing layers, but relatively difficult to add new layers or optimizers. Theano [4]] allows programmers to express a model as a dataflow graph of primitive operators, and generates efficient compiled code for training that model. Its programming model is closest to Tensor flow, and it provides much of the same flexibility in a single machine. Unlike Caffe, Theano, and Tensor flow, Torch [17] offers a powerful imperative programming model for scientific computation and machine learning. It allows fine-grained control over the execution order and memory utilization, which enables power users to optimize the performance of their programs. While this flexibility is useful for research, Torch lacks the advantages of a dataflow graph as a portable representation across small-scale experimentation, production training, and deployment. From the aspect of on situation deep learning, our project can come in handy.

III. PROPOSED WORK

We focused on training and testing 1000 categories PASCAL dataset. The deep learning model is built on a server machine. All the built models can be downloaded to Android APP. The download model button is for downloading the model from the server. This downloaded model gets converted to lite version of Tensor flow to make less resource utilization of end devices. This lite version may be used for inferencing or context learning and continuous learning on end devices offline.

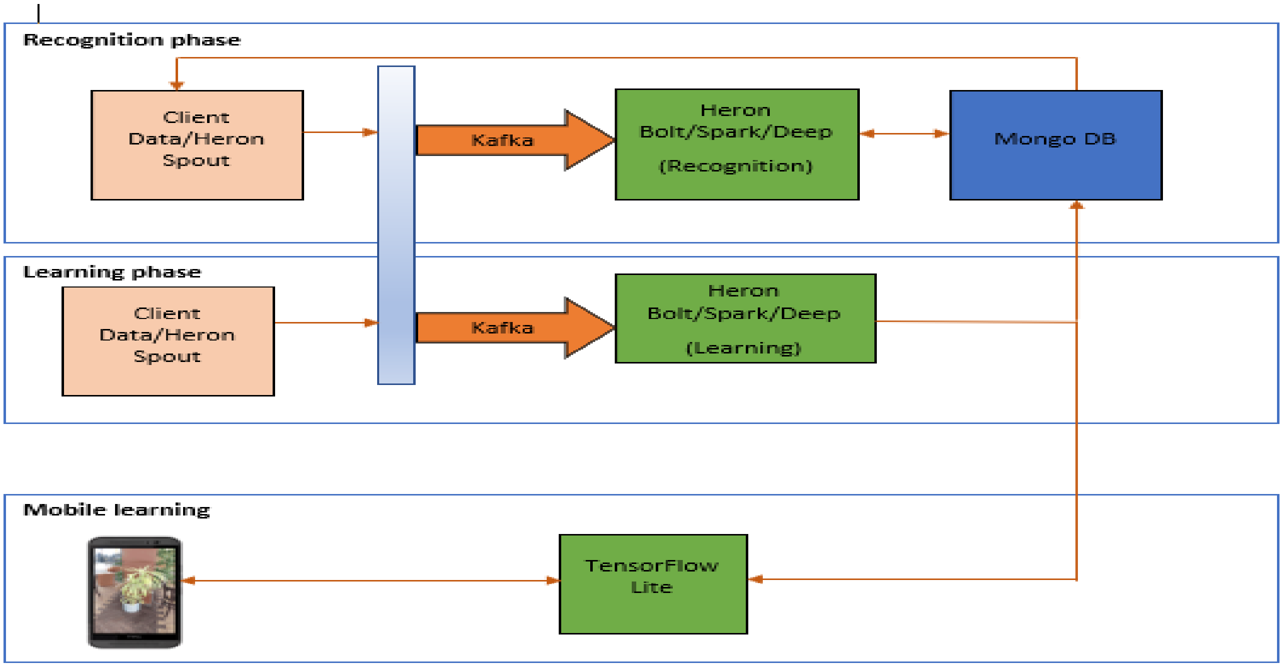


Figure 1:Architecture Diagram

Tensor flow model

We are using keras which is an open source neural network library written in Python. It is capable of running on top of Tensor flow. This functional API allow us to build model comfortably

Tensor flow lite model:

Tensor Flow Lite is a lightweight solution for mobile and embedded devices. It enables on-device machine learning inference with low latency and a small binary size. A set of core operators, both quantized and float, many of which have been tuned for mobile platforms. These can be used to create and run custom models. Developers can also write their own custom operators and use them in models.

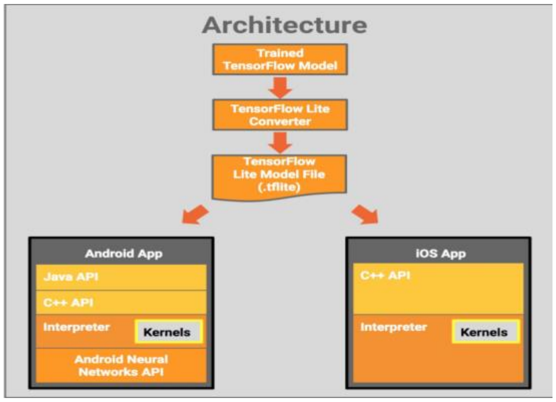


Figure 2:TFlite Architecture used for inferencing

libraries to model neural networks. So, we used convolutional neural network to find patterns in the image and finally to classify the objects in image. This part of logic runs in server and saving onto cloud repository. Below is the snippet for build model in server

IV. IMPLEMENTATION AND EVALUATION

A. System Design and Implementation

The overall workflow is in Figure 7 above. Pascal visual object detection dataset is used to train and test for three selected modules-Clarifai Spark and Tensor flow. A pre-trained model mobile net is used and its last layer is for further trained with the new inputs. The user will see the prediction of the model on the device then.

B. Evaluation and Results

1) Evaluation plan:

- Datasets:

  The dataset consists of 150,000 photographs, collected from Flickr and other search engines labeled with 1000 object categories. A random subset of 50,000 of the images with labels is used for building model and the remaining images are used for evaluation.

- System Specifications:
  We used i7-6700HQ processor, 16GB RAM CPU to build model on server machine and we inferenced the model on Nexus 7.

Implementation Details:

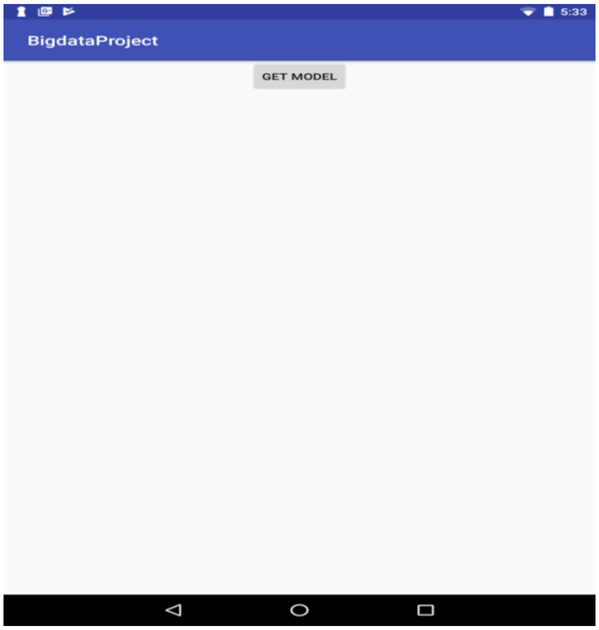
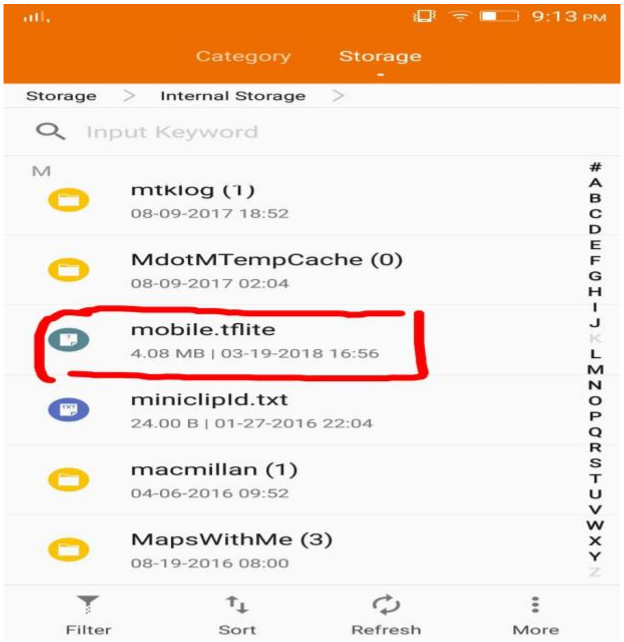


Figure 3: Event to trigger model to download


9:13 PM
Category
Storage
Storage > Internal Storage >
Input Keyword
M
# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Filter
Sort
Refresh
More

mtklog (1)
08-09-2017 18:52

MdotMTempCache (0)
08-09-2017 02:04

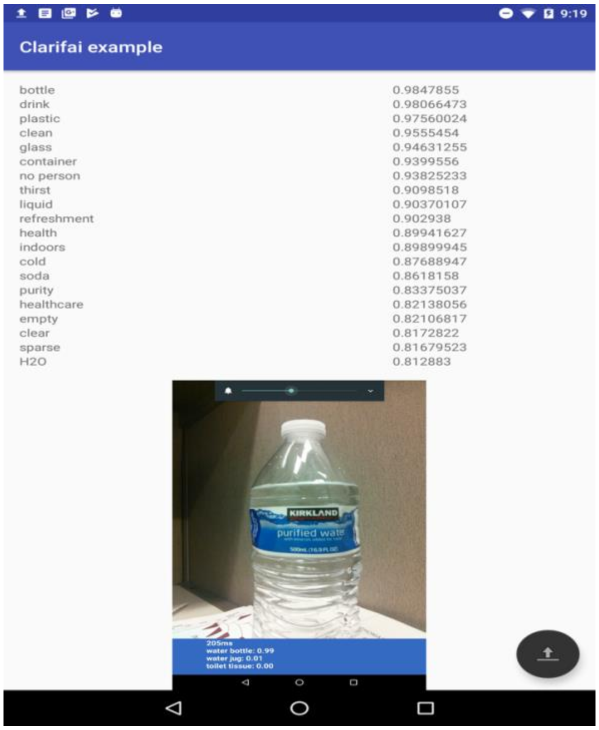
mobile.tflite
4.08 MB | 03-19-2018 16:56

miniclipId.txt
24.00 B | 01-27-2016 22:04

macmillan (1)
04-06-2016 09:52

MapsWithMe (3)
08-19-2016 08:00

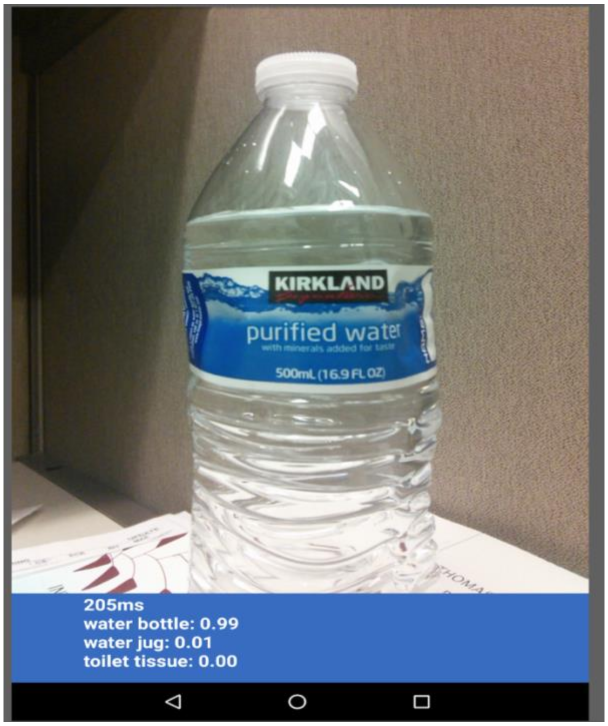
Figure 3: Downloaded Tensor flow Model on User's device



```
bottle          0.9847855
drink           0.98066473
plastic         0.97560024
clean           0.9555454
glass           0.94631255
container       0.9399556
no person       0.93825233
thirst          0.9098518
liquid          0.90370107
refreshment     0.902938
health          0.89941627
indoors         0.89899945
cold            0.87688947
soda            0.8618158
purity          0.83375037
healthcare      0.82138056
empty           0.82106817
clear           0.8172822
sparse          0.81679523
H2O             0.812883
```

```
205ms
water bottle: 0.99
water jug: 0.01
toilet tissue: 0.00
```

Figure 5: Inferencing on Mobile using Clarifai API



```
205ms
water bottle: 0.99
water jug: 0.01
toilet tissue: 0.00
```

Figure 6: Inferencing on Mobile using downloaded Tflite model

V. REFERENCES

1. Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
2. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
3. Ran, Xukan, Haoliang Chen, Zhenming Liu, and Jiasi Chen. "Delivering Deep Learning to Mobile Devices via Offloading." In Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, pp. 42-47. ACM, 2017.
4. Lane, N.D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L. and Kawsar, F., 2016, April. Deepx: A software accelerator for low-power deep learning inference on mobile devices. In Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on (pp. 1-12). IEEE