Data Mining (CS 451) Project

MobileNets: Convolutional Neural Networks for Mobile Vision

Team Members

Surajnath Sidh
Akash Agarwal
Ashwin Pilgaonkar

Introduction

MobileNets are a class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. MobileNets gives us a choice between resource and accuracy tradeoffs and shows strong performance compared to other popular models on ImageNet classification. MobileNets are capable of object detection, finegrain classification, face attributes and large scale geo-localization.

Objectives

We are deploying MobileNets on Mobile(Android) using Tensorflow Mobile for Object Detection and Classification tasks.
We also retrained MobileNets on a custom dataset of **Road** or **Not Road** Images for a classification task which runs near real time on mobile phone.

Figure 1. MobileNet models can be applied to various recognition tasks on low computation devices.

Model Checkpoint	Million MACs	Million Parameters	Top-1 Accuracy	Top-5 Accuracy
MobileNet_v1_1.0_224	569	4.24	70.7	89.5
MobileNet_v1_1.0_192	418	4.24	69.3	88.9
MobileNet_v1_1.0_160	291	4.24	67.2	87.5
MobileNet_v1_1.0_128	186	4.24	64.1	85.3
MobileNet_v1_0.75_224	317	2.59	68.4	88.2
MobileNet_v1_0.75_192	233	2.59	67.4	87.3
MobileNet_v1_0.75_160	162	2.59	65.2	86.1
MobileNet_v1_0.75_128	104	2.59	61.8	83.6
MobileNet_v1_0.50_224	150	1.34	64.0	85.4
MobileNet_v1_0.50_192	110	1.34	62.1	84.0
MobileNet_v1_0.50_160	77	1.34	59.9	82.5
MobileNet_v1_0.50_128	49	1.34	56.2	79.6
MobileNet_v1_0.25_224	41	0.47	50.6	75.0
MobileNet_v1_0.25_192	34	0.47	49.0	73.6
MobileNet_v1_0.25_160	21	0.47	46.0	70.7
MobileNet_v1_0.25_128	14	0.47	41.3	66.2

Table: Variants of MobileNet model, we can choose one to fit our latency, computation power and size budget. The size of the network in memory and on disk is proportional to the number of parameters. The latency and power usage of the network scales with the number of Multiply Accumulates (MACs) which measures the number of fused Multiplication and Addition operations. Top-1 and Top-5 accuracies are measured on the ILSVRC dataset.

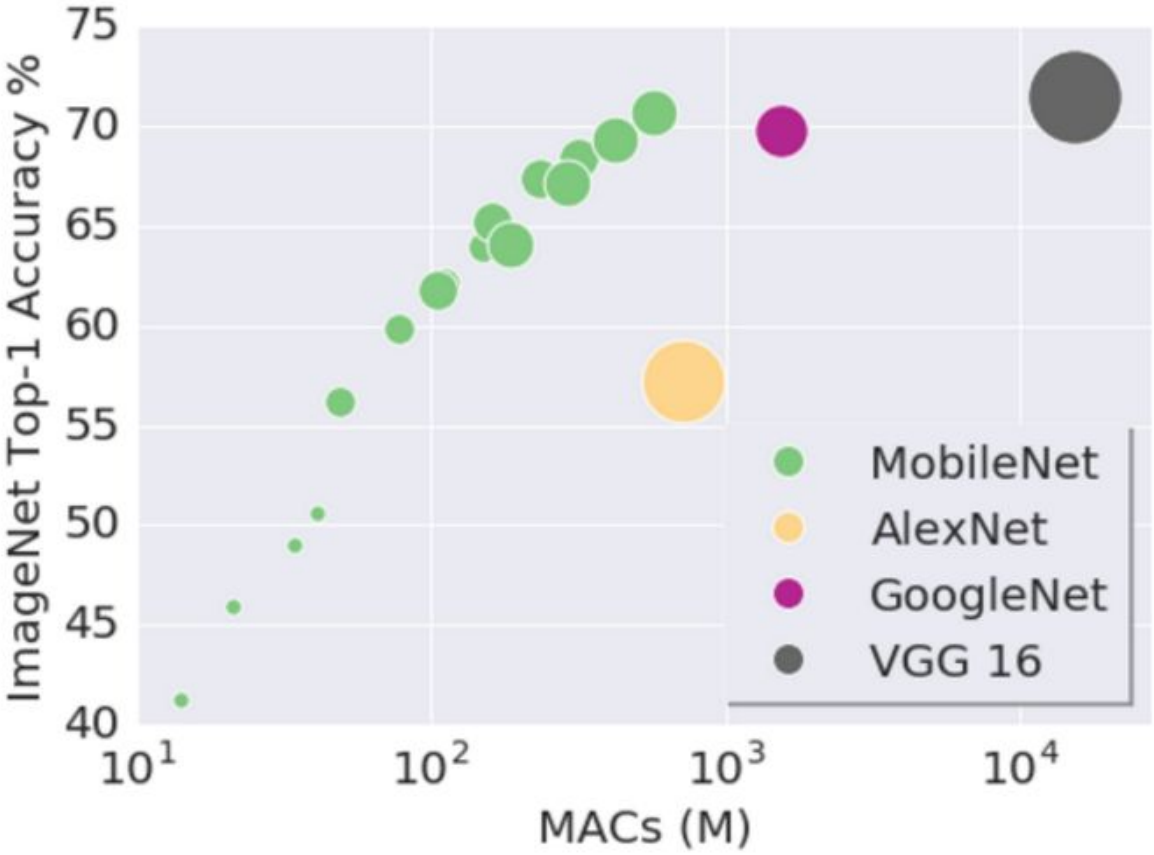

Fig: ImageNet Top-1 Accuracy(in %) vs MACs(Multiply Accumulates in Millions)

Experiments and Results

MobileNets can be run efficiently on mobile devices with TensorFlow Mobile.
They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large scale models, such as Inception, are used.

Experiment setup

Our Experiment was conducted using TensorFlow Mobile and **Xiaomi Mi A1** used as the test device.
It has following technical specifications-

Chipset: Qualcomm MSM8953 Snapdragon 625
CPU: Octa-core 2.0 GHz Cortex-A53
GPU: Adreno 506
RAM: 4GB
OS: Android 7.1.2 (Stock ROM)

Object Classification

Using Google Inception (5h) model to classify camera frames in near real-time

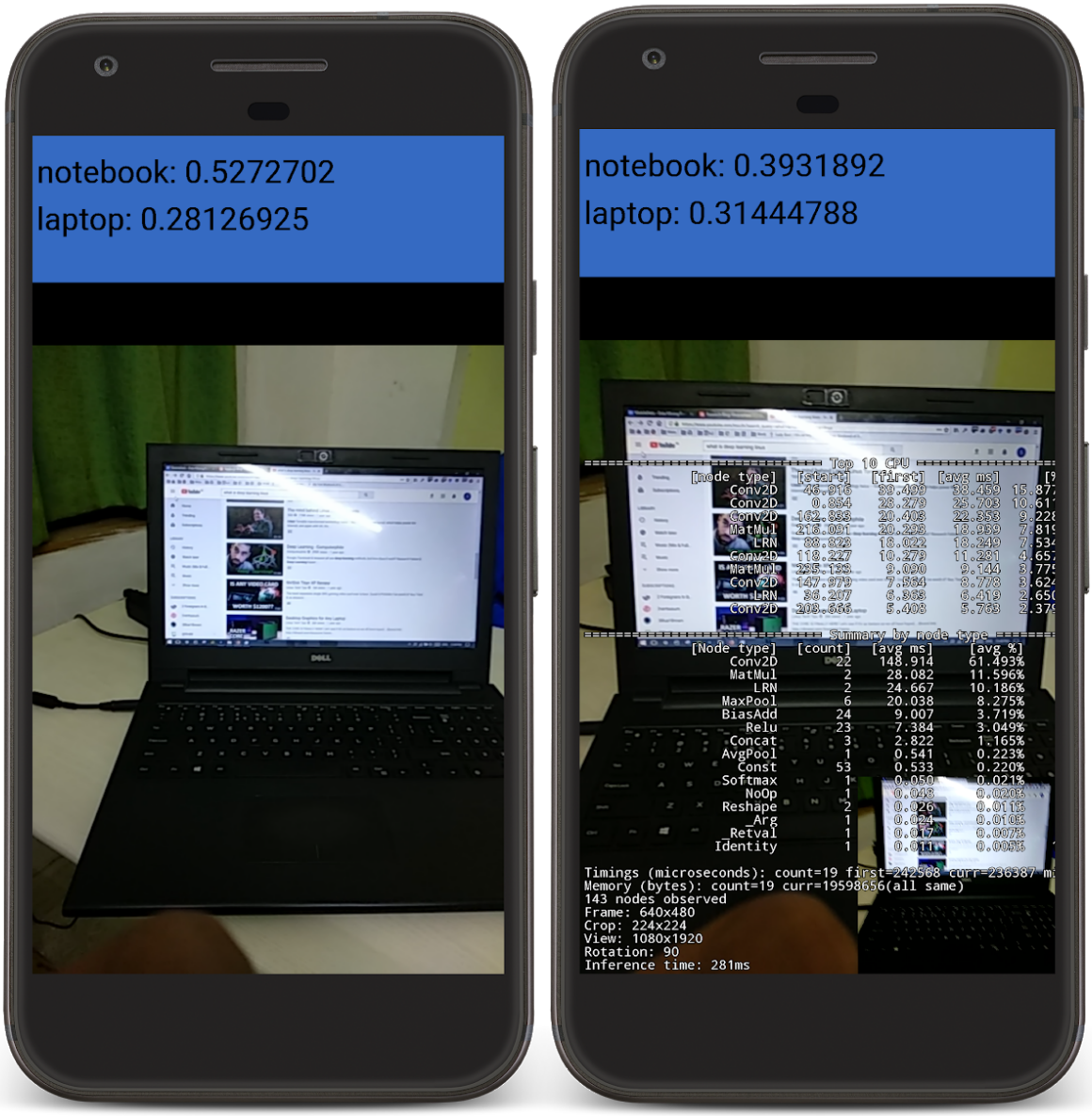

Image: Inception (5h) model in action on Android Phone, it takes around 300ms for Inference on our test device (device specifications listed above)

The image above shows Inception model is being used for Classification task. The Model was trained on ImageNet and built using Tensorflow Android (Java Framework).

It takes around 300ms for Inference on our test device, it is not in real time.
It was around 3 frames per second for inference.

Object Detection

Using SSD-Mobilenet (v1 1.0@128) model trained using the Tensorflow Object Detection API

Image: Mobilenet (v1 1.0@128) model in action on Android Phone, it takes around 700ms for Inference on our test device (device specifications listed above)

In the image shown above, MobileNet 1.0_128 model is being used for Classification task. The model was trained on ImageNet, It was built using Tensorflow Android (Java Framework).
It takes around 700ms for Inference on our test device, Which is around 1.5 frames per second for inference.
For above mentioned Object Detection and Classification, both models were bundled in a single Application and our Apk Size was 105 MB with MobileNet being 27.7 MB and Inception, 51.3 MB

These above shown experiments were performed as baseline to show that we can bring Inference and Application size down with a careful selection of model.

Road Not Road(Object Classification Task)

We have Retrained MobileNet using MobileNet 0.5 @ 128 on just 6000 images out final model size is 1.6 MB with Accuracy of 95.0% to detect Road or Not Road with final apk size 33.5 MB

Image: Object Classification for road and not road using MobileNets 0.5 @ 128, it takes around 60ms for Inference on our test device(see above for device specifications)

Conclusions

We were successfully able to reduce the apk size from 105 MB to 33.5 MB and improve the inference time from 300ms to 60ms for our custom trained model trained on Road or Not Road images.

It's performance can be made improved by compiling native TensorFlow libraries. By doing this, 30 fps can be obtained in the Road / not Road demo and the inference time can be reduced to 40ms.

When we used MobileNet for Classification instead of Inception our model size was reduced from 51.1 MB to 1.6 MB which is a 31.9375 times smaller.

We showed that we can use MobileNets on Low Resource device to perform most vision related tasks with a careful selection of model.

Our experiment was on Android Platform but this can be ported to iOS or any other mobile platform without significant effort. It can even be deployed on Raspberry PIs

References

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826).

MobileNets: Open-Source Models for Efficient On-Device Vision. (n.d.) Retrieved from https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html

TensorFlow Mobile. (n.d.). Retrieved from https://www.tensorflow.org/mobile/