A Review on MobileNet, ResNet and SqueezeNet for iOS & iPadOS for on Device Training and Prediction using CoreML

Anirudh Tulasi[1], S Malarselvi[2], Mansa Devi Pappu[3]

[1]Student, B.Tech, Dept. of Computer Science & Engineering, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India

[2,3]Assistant Professor, B.Tech, Dept. of Computer Science & Engineering, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India

---***---

Abstract - *Image classification plays a vital role in the current world. Given the advancements in technology and AI, this plays a major role as it basically gives the computer to process live image feed and to classify them accordingly. Our project aims to streamline this process and perform on device training and predictions on the mobile devices. This would help us in building more and more applications which uses the core processing power available on mobile devices while also increasing the efficiency of the model. We would be using the MobileNetV2 Model for this project which was developed by Google to run on devices with low computing power like mobile phones and with the advancements in the technologies these projects can be improved further by adding techniques like NLP and text processing to process the text from the live feed and provide audio output to people with inabilities.*

Key Words: **iOS, MobileNetV2, iPadOS, CoreML, image-classification, training,**

1.CoreML

CoreML is a machine learning framework built by Apple to run various models on their devices running macOS, iOS, tvOS and watchOS. It can be used to perform on device predictions at a very fast rate which gives real time results.

But for now, we only limit ourselves to iOS as MobileNetV2 is designed to run on mobile devices. MobileNet can be run on computers but it was designed to run on low processing mobile devices hence we would only consider iOS for now which covers most of the iPhones released to date.

2. MobileNetV2

MobileNetV2 is basically a inverted residual structure with a thin bottleneck layers which are used for Inputs and outputs. This uses a novel framework called SSDLite.

It contains residual block with stride 1 while another block is present with stride 2 which is used for downsizing. Each block consists of three types of layers.[1]

Table -1: Layers in residual block

Layers in residual blocks		
Input to Layer	Layer	Output
h x W x k	1x1 conv2D, ReLU6	h x w x (tk)
h x W x tk	3x3 Depthwise, ReLU6	h/s x w/s x (tk)
h/s X w/s X tk	Linear 1x1 conv2D	h/s x w/s x k′

We can see that the layer one is 2D Convolutional Layer with a ReLU6 function. Second layer is a Depthwise convolutional layer with ReLU6 function and the final layer is a convolution without any non-linearity.

Here an expansion factor t is used where t=6 for all the principal experiments.

2.1 Comparison

Since we have three models which can perform image classification on iOS devices using CoreML we should clearly find the efficiency and drawbacks of each of the model after which we could arrive on a conclusion.

The three models are MobileNet, SqueezeNet and ResNet. We will now have a comparison among these using the inference time, total time and Frames per second (FPS)

2.1.1 Comparison using Inference Time

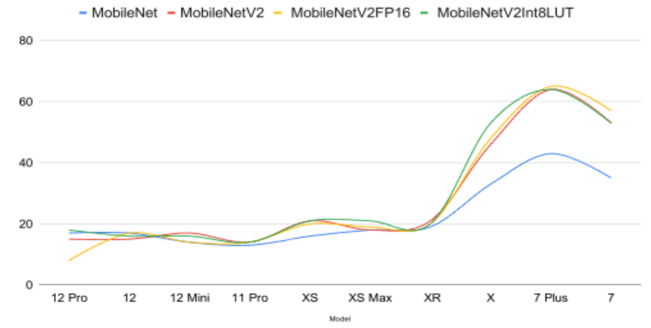

Chart -1: Inference time comparison on various iPhones for MobileNet

In the above chart we can clearly see the performance of all the four MobileNet models(MobileNet, MobileNetV2, MobileNetV2FP16 and MobileNetV2Int8LUT) on various iPhones. Considering the four generations of devices we can see that MobileNetV2 is a better option here as it has less inference time than that of the others.

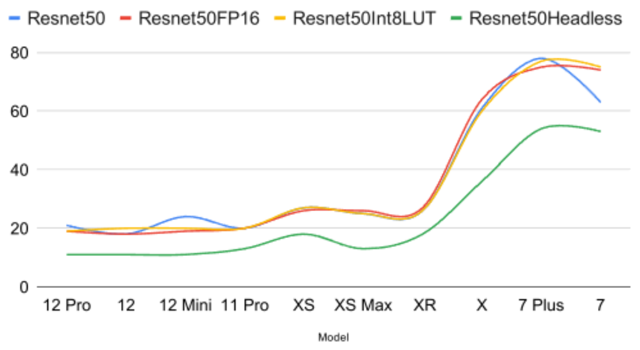

Chart -2: Inference time comparison on various iPhones for ResNet

In the above chart we can clearly see the performance of all the four ResNet models [2] (Resnet50, Resnet50FP16, Resnet50Int8LUT and Resnet50 Headless) on various iPhones. We can also clearly observe that on older models inference time almost touches 80ms which is not very good for classification in live feed. Hence this model won't be a good fit for running the applications on the older devices.

Coming to SqueezeNet we can see that we have SqueezeNet, SqueezeNetFP16 and SqueezeNetInt8LUT[3]. These have very less inference time compare to others and can be used for real time image classification as the maximum inference time on the older devices is just 35ms which is much less than the previous models. But we will also compare them with other factors like total time and FPS[4] as they also play vital role.

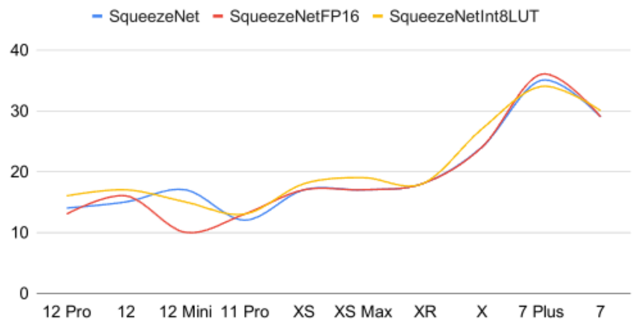

Chart -3: Inference time comparison on various iPhones for SqueezeNet

2.1.2 Comparison using Inference Time

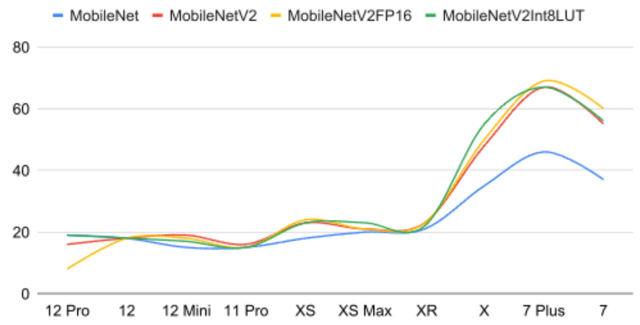

Chart 4: Total time comparison on various iPhones for MobileNet

In the above chart we can clearly see the performance of all the four MobileNet models(MobileNet, MobileNetV2, MobileNetV2FP16 and MobileNetV2Int8LUT) on various iPhones. Considering the four generations of devices we can see that MobileNetV2 is a better option here as it has less total time than that of the others.

Coming to ResNet we can clearly see the performance of all the four ResNet models (Resnet50, Resnet50FP16, Resnet50Int8LUT and Resnet50 Headless) on various iPhones is similar except for ResNet50Headless which has significantly less total time compared to other ResNet Models. We can also clearly observe that on older models total time almost touches 80ms which is not very good for classification in real time. Hence this model won't be a good fit for running the applications on the older devices.

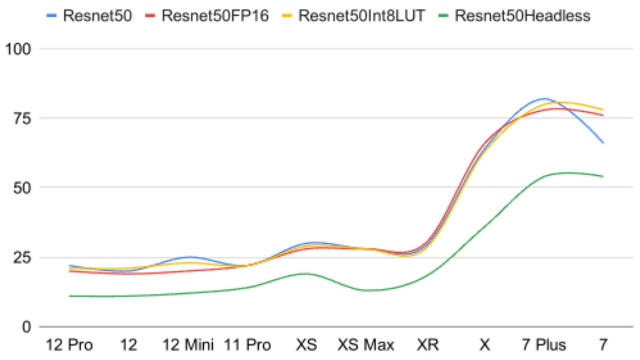

Chart -5: Total time comparison on various iPhones for ResNet

Coming to SqueezeNet we can see that we have SqueezeNet, SqueezeNetFP16 and SqueezeNetInt8LUT. These have very less total time compare to others and can be used for real time image classification as the maximum inference time on the older devices is just 35ms which is much less than the previous models. But we will also compare them with FPS as it also very important.

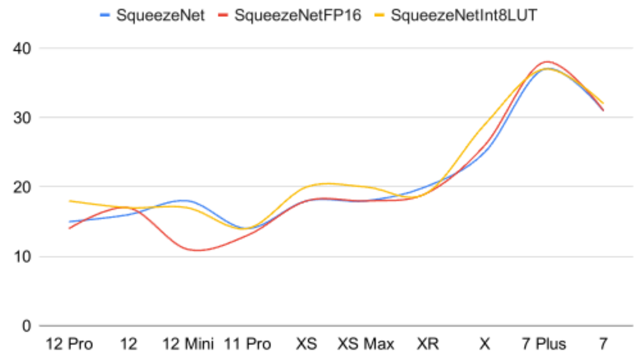

Chart -6: Inference time comparison on various iPhones for SqueezeNet

2.1.3 Comparison using Frames per second

FPS(Frames per second) is also very important for any real time application because it shows how fast the object detection model process the live feed and give the desired output.

We can clearly see FPS for all the different models in the chart. We could see that MobileNetV2 & MobileNetV2FP16 gives out good FPS compared to others.

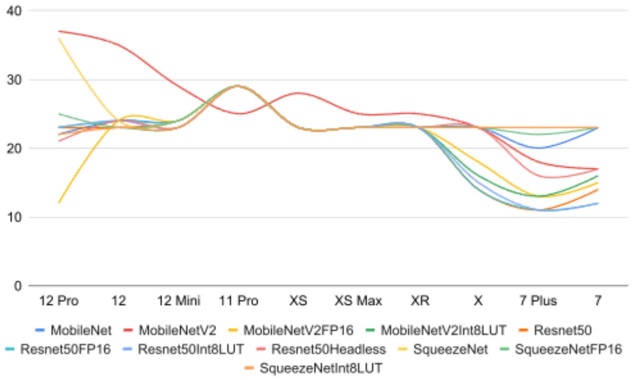

Chart -7: FPS (Frames per second) comparison on various iPhones

3. CONCLUSIONS

Upon exploring various models like MobileNet, ResNet and SqueezeNet we can conclude that MobileNetV2 performs better on most of the devices. Though we have seen that ResNet and SqueezeNet models had good inference and total time they could not achieve good FPS.

This makes MobileNetV2 significantly better than other models for Image Classification on iOS devices using CoreML.

REFERENCES

[1] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks” arXiv:1801.04381 [cs.CV]

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition” arXiv:1512.03385 [cs.CV]

[3] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size” arXiv:1602.07360 [cs.CV]

[4] Xin, M., Wang, Y. Research on image classification model based on deep convolution neural network. J Image VideoProc. **2019**, 40(2019). https://doi.org/10.1186/s13640-019-0417-8