ECE361E Final Project

M1: Structural Pruning

Project Members: Uttami Godha, Jesus Lopez Neira

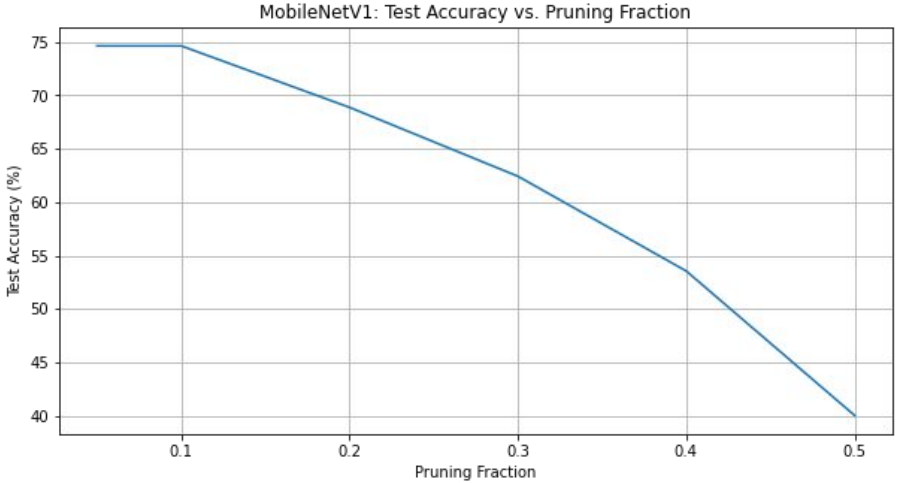
Jesus's tasks- initial training, torch pruning function, TACC training
Uttami's tasks- deployment on RPi with ONNX, Table 1 calculations, graphs

General Approach

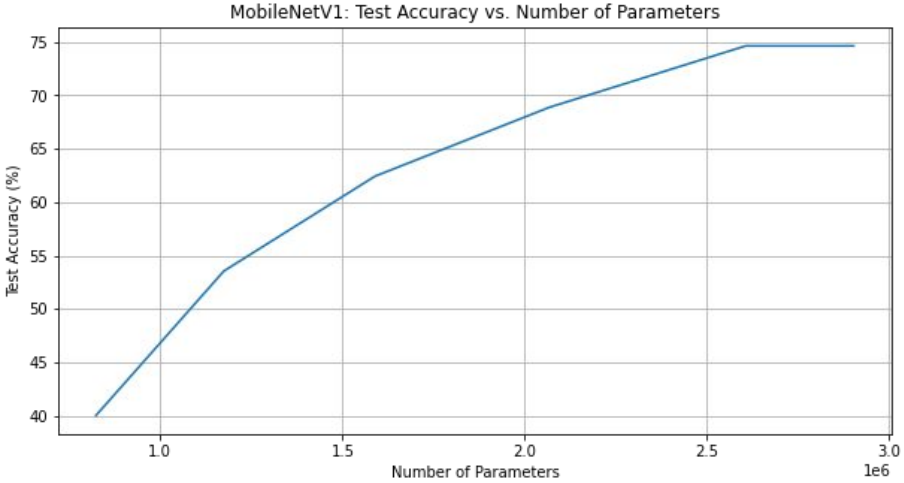
In this project we will compare the efficiency of different pruning ratios. Following the next methodology:

1. Define MobileNet in Pytorch
2. Pruning MobileNet model with different ratios
3. Retraining models to improve accuracy
4. Deploying models in Raspberry Pi using ONNX
5. Comparing models through a results and metrics discussion

Test Accuracy vs. Pruning Fraction



Test Accuracy vs. # of Parameters



M1.5. Table 1

Pruning fraction	#Fine-tuning epochs	#Params	Max memory usage [MB]	Average latency per image [ms]	Max power consumption [W]	Average energy consumption per image [mJ]
0.05	5	2902069	82	56.2127	4.37	1260.3807
0.1	5	2606173	75	57.4931	4.37	1254.4235
0.2	5	2066884	79	45.5078	4.366	1045.6234
0.3	5	1588917	79	38.0608	4.504	829.4638
0.4	5	1174647	81	30.6280	4.317	736.6639
0.5	5	823722	77	27.0487	4.313	619.0004

Conclusion and Further Plans

✓ Overall, we see that pruned models are more memory-efficient, energy-efficient, and inference faster. Pruned models are less likely to overfit data (given proper tuning), so we think that the addition of quantization to each model will provide much more accurate results.

✖ However, there is a slight tradeoff between these results and the corresponding accuracy for each model, in that we see more loss of accuracy the more we prune.

💡 The goal is to find a balance between the size of the model (ie. the level of pruning) and the performance of the network (ie. the accuracy) while considering the power usage and time delay (latency) resulting from each method. We will experiment with quantization in the next milestone, as well as the number of fine-tuning epochs with each pruning fraction.