Milestone 1
Structural Pruning

Prathik Srinivasan
Pruned, fine-tuned, and converted mobilenet model

Drew Hardie
Ran inference, collected inference data, and made plots

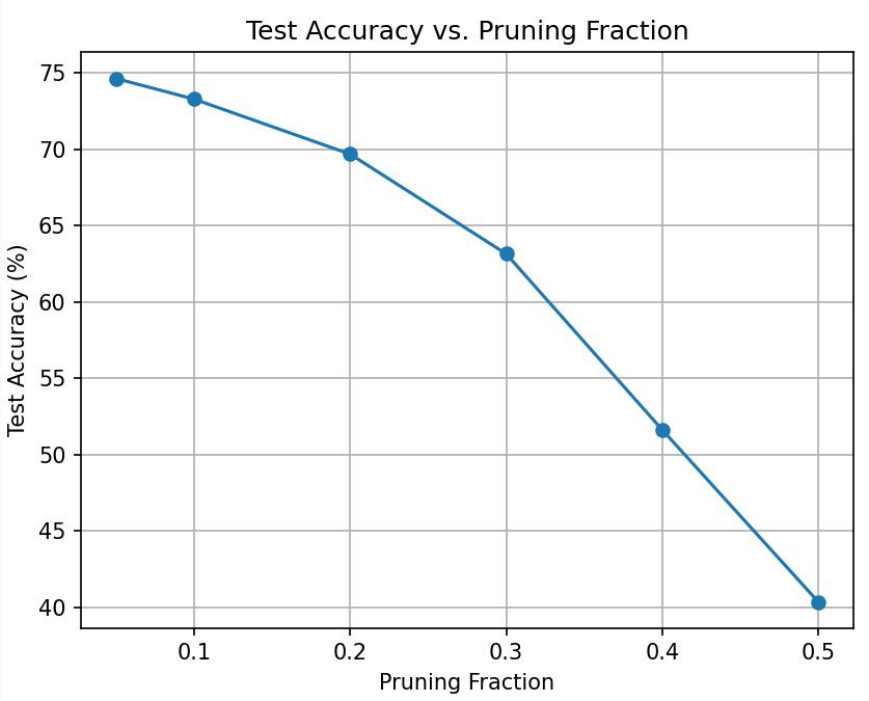
Approach

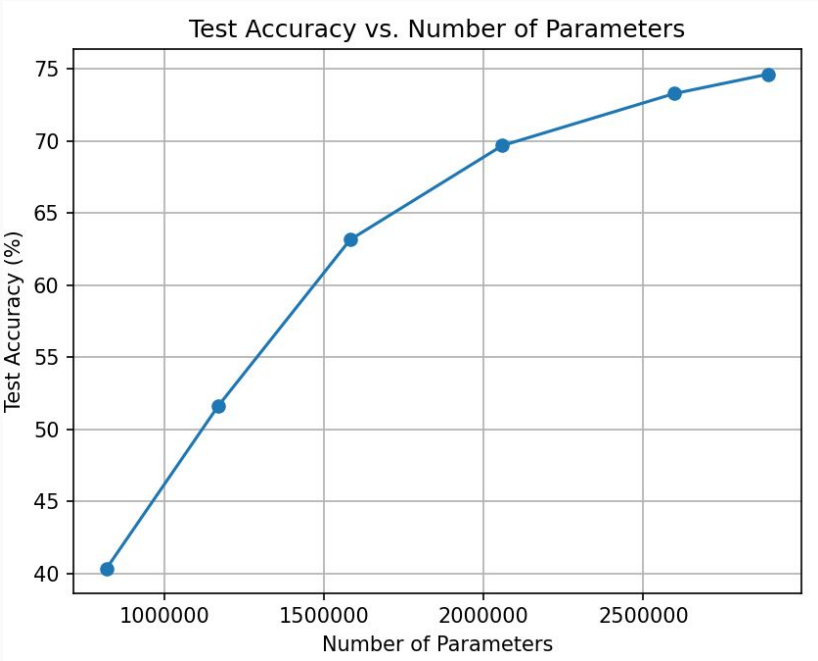
- We trained the MobileNet model on TACC GPUs
- Using the the Torch-Pruning library, we prune all of the MobileNet layers except for the final classification layer and then fine-tune the model
- We then convert and deploy the model on the Raspberry Pi 3B+
- By repeating this process with a variety of pruning fractions, we can determine the optimal value for maintaining accuracy while maximizing efficiency

Results

Pruning Fraction	Fine-tuning epochs	parameters	Maximum memory usage [MB] rpi_base=85	average latency per image [ms]	Maximum power consumption [W]	Average energy consumption per image [mJ]	Test Accuracy (base=77.68%)
0.05	5	2,891,687	45	25.71	6.68	763.57	74.63%
0.1	5	2,596,342	42	28.99	6.69	809.12	73.28%
0.2	5	2,058,145	40	21.35	6.68	616.45	69.68%
0.3	5	1,581,271	39	18.01	6.72	524.10	63.15%
0.4	5	1,168,093	35	19.90	6.68	559.53	51.63%
0.5	5	818,252	33	10.45	6.72	332.02	40.37%

Results


Test Accuracy vs. Pruning Fraction
Test Accuracy (%)
75
70
65
60
55
50
45
40
0.1
0.2
0.3
0.4
0.5
Pruning Fraction


Test Accuracy vs. Number of Parameters
Test Accuracy (%)
75
70
65
60
55
50
45
40
1000000
1500000
2000000
2500000
Number of Parameters

Conclusion

- Optimal pruning factor?
- What we can improve
- Other models to try