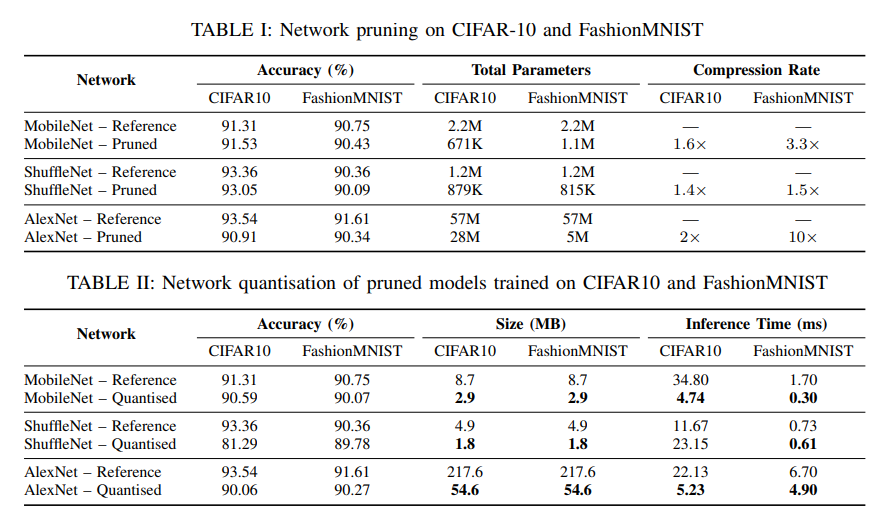
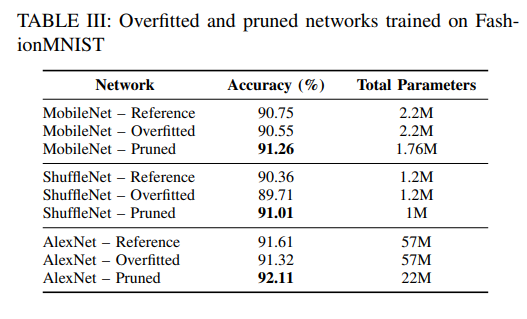
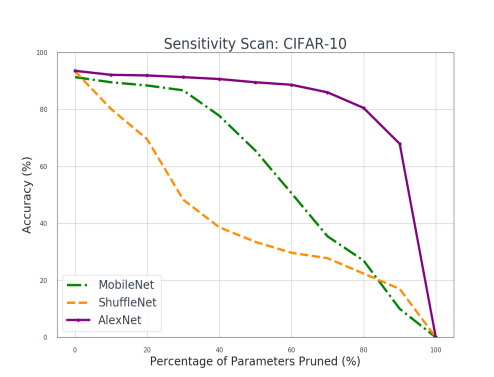
Aaron Reuven 204880082

Eitan Kerzhner 205697139

# Reproduction of Article Results

The article we chose to reproduce is “[Quantisation and Pruning for Neural Network Compression and Regularisation](https://www.arxiv.org/abs/2001.04850)” by [Kimessha Paupamah](https://arxiv.org/search/cs?searchtype=author&query=Paupamah,+K), [Steven James](https://arxiv.org/search/cs?searchtype=author&query=James,+S), [Richard Klein](https://arxiv.org/search/cs?searchtype=author&query=Klein,+R).  
The paper "Quantization and Pruning for Neural Network Compression and Regularization" addresses methods for reducing the size of neural networks through techniques like quantization and pruning. These approaches aim to make neural networks more efficient and manageable, particularly for deployment on hardware with memory constraints and for improving model regularization.  
The algorithm we’re trying to reproduce is the pruning and the quantization of the 3 Neural Networks, Alexnet, Shufflenet and Mobilenet, using two datasets FashionMNIST and CIFAR-10. The exact details of the algorithms are not specifically outlined in the paper but their code is open to the public at [this link](https://github.com/kpaupamah/compression-and-regularisation).

Original Results:  
  
  
  
  
  
A graph of a function

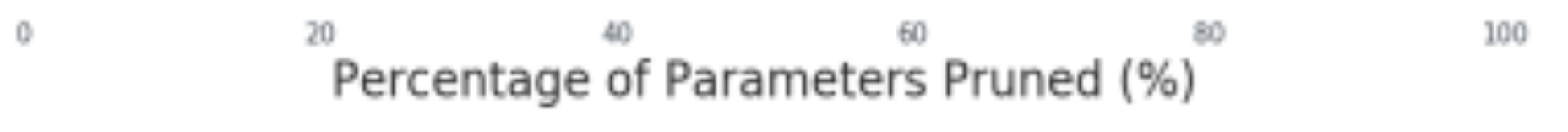
Description automatically generated with medium confidence

Our reproduction of the results:

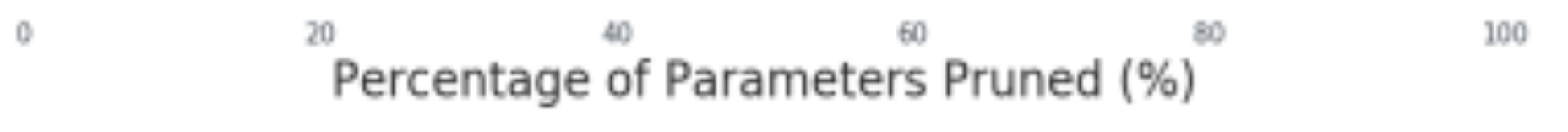
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Accuracy | | Total Parameters | | Compression Rate | |
| CIFAR10 | FashionMNIST | CIFAR10 | FashionMNIST | CIFAR10 | FashionMNIST |
| MobileNet - Reference | 88.54 | 90.23 | 2.2M | 2.2M | - | - |
| MobileNet - Pruned | 88.5 | 88.67 | 671K | 1.1M | 1.6x | 3.3x |
| ShuffleNet - Reference | 84.97 | 89.71 | 1.2M | 1.2M | - | - |
| ShuffleNet - Pruned | 87.94 | 89.81 | 879K | 815K | 1.4x | 1.5x |
| AlexNet - Reference | 85.62 | 91.81 | 57M | 57M | - | - |
| AlexNet - Pruned | 87.89 | 91.93 | 28M | 5M | 2x | 10x |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Accuracy | | Size | | Inference Time | |
| CIFAR10 | FashionMNIST | CIFAR10 | FashionMNIST | CIFAR10 | FashionMNIST |
| MobileNet - Reference | 88.54 | 90.23 | 8.7MB | 8.7MB | 34.91 | 13.03 |
| MobileNet - Quantized | 88.46 | 90.1 | 2.51MB | 2.51MB | 4.51 | 4.04 |
| ShuffleNet - Reference | 84.97 | 89.71 | 4.9MB | 4.9MB | 27.88 | 13.3 |
| ShuffleNet - Quantized | 85.19 | 89.79 | 1.4MB | 1.4MB | 7.8 | 7.51 |
| AlexNet - Reference | 85.62 | 91.81 | 217MB | 217MB | 24.53 | 10.99 |
| AlexNet - Quantized | 85.64 | 91.63 | 54.6MB | 54.6MB | 39.43 | 28.7 |

|  |  |  |
| --- | --- | --- |
| Network | Accuracy | Total Parameters |
| MobileNet - Reference | 90.23 | 2.2M |
| MobileNet - Overfitted | 90.6 | 2.2M |
| MobileNet - Pruned | 90.79 | 1.76M |
| ShuffleNet - Reference | 89.71 | 1.2M |
| ShuffleNet - Overfitted | 89.98 | 1.2M |
| ShuffleNet - Pruned | 89.95 | 1M |
| AlexNet - Reference | 91.81 | 57M |
| AlexNet - Overfitted | 92.65 | 57M |
| AlexNet – Pruned | 92.74 | 22M |

  
FashionMNIST:  
A graph with blue and orange lines

Description automatically generated  
CIFAR10:A graph with lines and numbers

Description automatically generated

There are notable differences in the results, making it difficult to reproduce the findings, especially when it comes to the graphs. We don't understand how the authors produced their graphs, and the results seem inconsistent. For example, they claim that the best-performing pruned ShuffleNet on the CIFAR dataset has 30% of parameters pruned and an accuracy of 93% in the table, but the graph shows an accuracy of around 50%.

The inference times differ, likely because our machines are either less powerful or not as optimized for the task. However, the amount of parameters pruned and the size of the neural networks remain consistent, which makes sense since the architecture of the neural networks should stay relatively similar. This consistency might vary only if the architecture changed between PyTorch versions, which likely occurred with MobileNet and ShuffleNet, causing us additional problems.

We believe that most, if not all, of the discrepancies are likely due to differences in the PyTorch versions and possibly some parameters that were not mentioned in the paper, for which we used the default values. finding a suitable PyTorch and other modules versions raised many challenges. We faced obstacles throughout the reproduction process, including the need to fix parts of the code due to a bug that couldn't be resolved with any PyTorch version (we tested every version from 1.0 to 1.8). We even speculated that they might have used an experimental version at the time, which we wouldn't have easy access to. If the authors had provided a requirements file, it would have greatly reduced the difficulties and saved us a lot of trouble. A requirements file is a simple text file that lists all the necessary software dependencies and their specific versions needed to run a project. By having this, we could have ensured that our environment matched theirs exactly. Additionally, specifying which parameters were changed for each execution of the code would have been very helpful.

Through this exercise, we've learned the importance of thorough documentation, especially when working in a team. Proper documentation is crucial for consistently reproducing results with the same package versions. Without it, outcomes from experiments can vary significantly, leading to differences in the knowledge gained from what should be the same experiment.