**Com S 527X Proposal**

**Jeremy Roghair and Deepak George Thomas**

**Title**: Accelerating MobileNets using High Performance Computing

**Description**: MobileNets is an efficient convolutional neural network architecture that was developed for application in mobile cameras. The architecture consists of a type of factorized convolutions referred to as depth wise separable convolutions. These convolutions convert a standard convolution into a depth wise convolution, that subjects every channel to single filter, and then a pointwise convolution which combines the outputs by employing a 1X1 convolution on it. The authors showed that the factorization step leads to a substantial reduction in model size and computations, relative to standard convolutional neural networks. Furthermore, they arrived at an expression which demonstrated the computational efficiency of depth wise separable convolutions as compared to standard networks –

(1)

Here, denotes the number of filter channels and represents the spatial dimension of the square kernel of the standard convolutional network [1].

In addition, the authors implemented two new global hyper parameters, a width multiplier ( and a resolution multiplier (. The width multiplier is applicable for cases wherein the model size needs to be further reduced and its speed increased. It works by uniformly reducing the thickness of a network for all layers. The latter hyper parameter works by subjecting the input image to some resolution multiplier, which then reduces the internal representation at all the layers [1].

**Project Objectives**: We propose to integrate the above mentioned algorithm with high performance computing platforms, in order to accelerate training times and quickly tune the hyper parameters. We’ll set up our own implementation of this algorithm so as to determine the optimal points, wherein the code can be parallelized and sped up. The final step of our project will include comparing our optimized implementation with the benchmarks set in the paper.

**Potential Limitations**: We believe that, despite implementing high performance computing, there is a chance that our model might not be significantly faster than the author’s implementation. This is because they trained their model using TensorFlow, which has multiple in-built optimizations specifically set up for large scale numerical computing [1].

**Implementation Outline**: The algorithm will be coded using Python. We have chosen Cuda, a programming model and a platform that enables parallel computation, to express parallelism in our code [2].

**References** –

1. Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).
2. <https://developer.nvidia.com/cuda-zone>