**Task Description:**

The goal is to find the optimal set of hyperparameters of a neural network (for the dataset provided) using a reinforcement learning strategy, as explained in the paper.

The task is to be implemented in python3, and if a choice is to be made between TensorFlow and PyTorch, then TensorFlow must be used.

The implementation should be such hyperparameters for a fully connected neural network, deep convolution neural network, and a wide convolution neural network. Moreover, this implementation should be generic enough to be extended for residual networks, recurrent neural networks, and recurrent neural networks variant.

Performance of final model selected by the agent is calibrated using confusion matrix, accuracy and loss graphs (to see if it's not overfitting) and the total time it took for the agent to find an optimal solution

**Acceptance criteria:**

1. Working implementation of the code that can find optimal set of hyperparameters of a fully connected NN, CNN, and a wide CNN
2. Implementation should be generic enough so that it can be extended for other type of neural networks namely residual networks, and RNNs and it's variants
3. To make sure reinforcement learning implementation is working, the final model's performance must be checked using confusion matrix, accuracy and loss graphs using the dataset provided:
   1. neural network must be such that it fits a multi-label classification problem. There must be five classes: neutral, hard, soft, linked to hard, linked to soft (details are in the paper provided)