Design Decisions

Common to part c and d both

- Activation Function — As my activation function, I chose sigmoid. I tried training the network with ReLu, but it was getting trained a lot slower. Also, ReLu is best suited for deep neural networks—as it solves the vanishing gradient problem—but for shallow neural networks it is not that effective and rather unnecessary. So I went with sigmoid.
- Learning rate — I went with adaptive learning rate strategy, as it performs best when we get closer to the actual minima. The model was performing best at seed value of 1, for larger seed values it was not going to the minima steadily—the accuracy was not steadily increasing, it was decreasing quite frequently too. For lower value of the seed learning rate, the model was getting trained very slowly.
- Layer architecture — Using more than one hidden layer was making the model slower to train and also slower at converging towards the minima. So I went ahead with only one hidden layer with 40 nodes. Similarly, more nodes were making the training process as well as the convergence rate slower. Fewer nodes were making the training process faster, but it couldn't reach an acceptable level of accuracy.

For part d

- Feature extraction — After experimenting with various feature extraction techniques, I found canny edge detection to be the best performing one. Other techniques such as FFT and DCT were not able to increase the accuracy to an acceptable range. Edges on their own also were not adequate, so I used it combined with the original image. The results were satisfactory.