CM3: Results

Default Network

In reference to CM1, the performance of the default is the following:

Training Accuracy (last Epoch)	Validation Accuracy (last Epoch)	Testing Accuracy	Test Loss
0.9825	0.9429	0.9406	0.1969

Table 3: Performance of the default network

The performance of the default network can be seen in Figure 13:

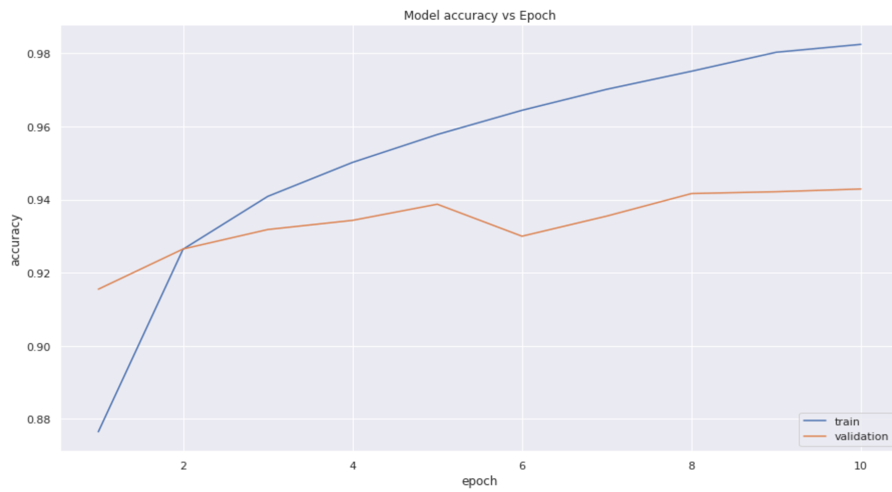

Figure 13: Model accuracy of default network vs epoch

The training model accuracy is increasing as the epoch size is increasing. Limited number of epochs were used due to limited computational resources. The validation accuracy also increases initially with epochs but begins to plateau.

Figure 13 shows that there is a significant difference between the validation and training dataset accuracy. This shows that there is overfitting that is occurring. It might be possible that if a higher epoch was set, the validation accuracy could have improved. However, due to computational limitation, higher epochs could not be used.

Furthermore, when taking the testing accuracy into account, which is similar to the validation accuracy, it is easy to conclude that there is overfitting occurring in the model.

The performance of the cross-entropy loss of the default network is shown in Figure 14:

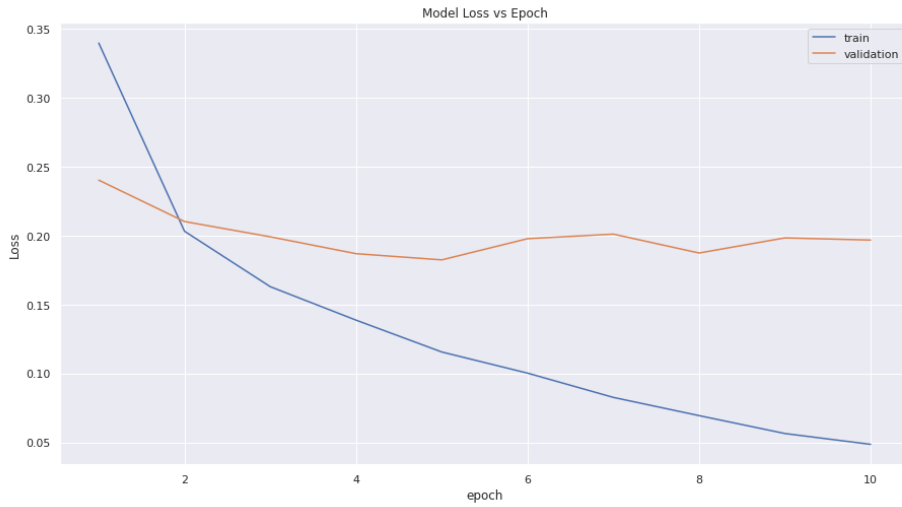

Figure 14: Model loss of the default network vs Epoch

The cross-entropy loss decreases significantly as the number of epochs increase, going down to 0.05 at the last epoch. The cross-entropy loss on the validation dataset decreases initially but begins to plateau.

Figure 14 does show that the model is not good at predicting the label of unseen data as there is a significant difference between the training and validation set. Since cross entropy loss is not bounded, some outliers can result in significantly increasing the loss. This tells us that there are some outliers that are affecting the loss of the model in the validation dataset.

To counteract the overfitting, we can do regularization to overcome this issue. We can also add a dropout layer to resolve the overfitting problem. Early stopping is not possible as we are already using a small number of epoch.

Own Model

In reference to CM2, where we had to make our own network, all the 4 models were trained using the same training and validation data for standardization purposes. The results of each model at the last epoch are the following:

S. No	Model Number	Number of Filters	Training Accuracy (last Epoch)	Validation Accuracy (last Epoch)	Testing Accuracy	Testing Loss
1	0	40	0.9853	0.9422	0.9394	0.2130
2	1	48	0.9854	0.9423	0.9424	0.2121
3	2	56	0.9868	0.9448	0.9437	0.2206
4	3	64	0.9828	0.9417	0.9384	0.2232

Table 4: Performance of the different models

The performance of the different models in terms of accuracy is shown in Figure 15.

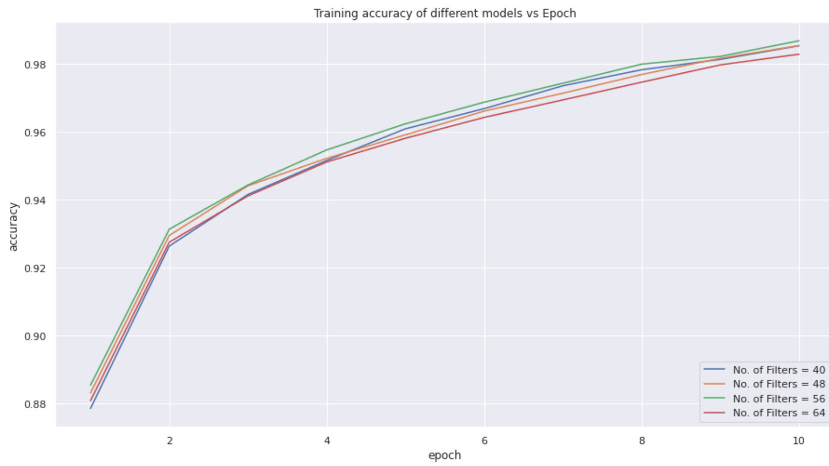

Figure 15: Training accuracy of different models vs Epochs

Based on Figure 15, the difference among the 4 models through the training epochs is narrow and similar. From the start, model 2 (56 number of filters) has a higher training accuracy than the other models. The worst performance was of model 3, which has 64 number of filters in each convolutional layer. Model 0 and Model 1 have very similar accuracy at the last epochs.

The performance of the different models in terms of validation accuracy is shown in Figure 16.

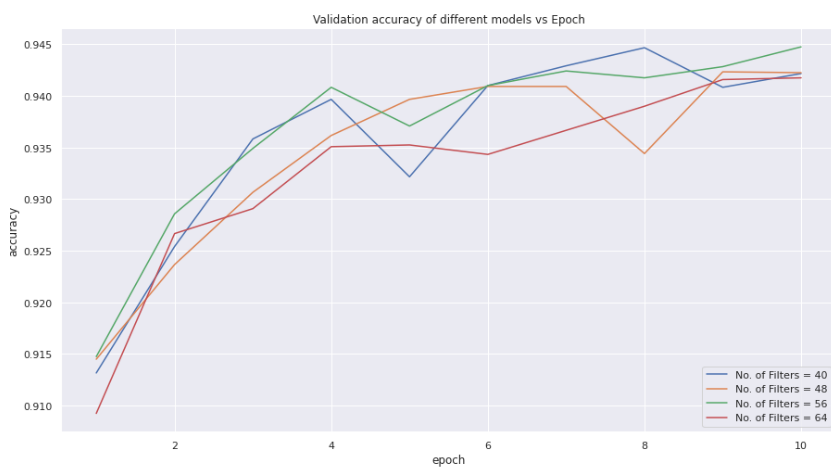

Figure 16: Validation accuracy of different models vs epochs

There is more variance in the validation accuracy when compared to training accuracy. In the start, model 2 has the highest validation accuracy, but it dips a bit and becomes second to model 0. At the last epoch, model 2 has the highest validation accuracy. The rest of the 3 models have similar validation accuracy in the end.

Based on these results, the model selected was model number 2 which has 56 filters in the convolutional 2D layers. Model 2 does not have a good cross-entropy loss compared to the other models. However, since the model 2 is better in three out of the four parameters, model 2 was selected.

The performance of the model 2 with respect to cross-entropy loss is shown in Figure 17:

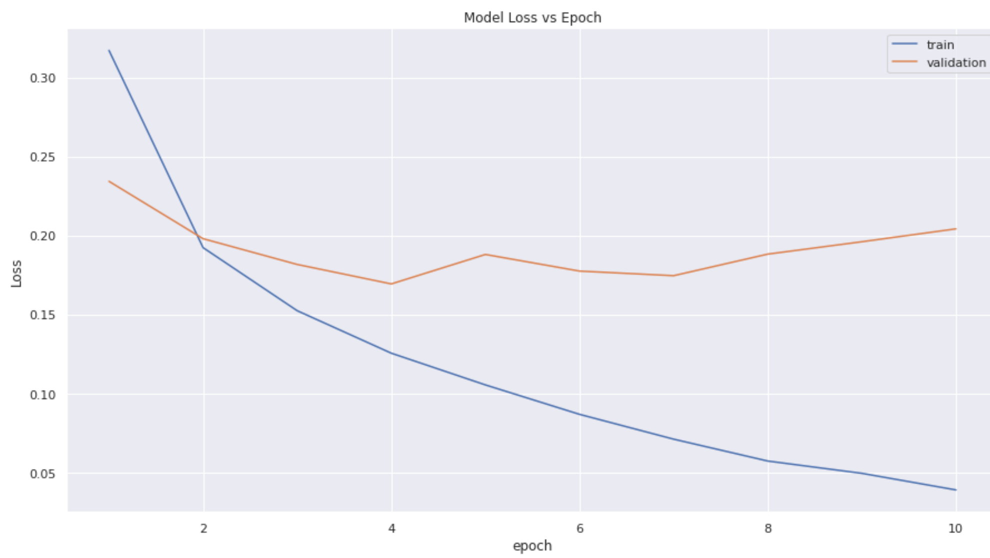

Figure 17: Cross entropy loss of model number 2 vs epoch

In a similar manner to the default neural network, the training loss decreases significantly as the epochs increase. The training loss reaches below 0.05. The validation loss decreases at a slower rate compared to that of the training loss. However, it plateaus at around 4 epochs and start to increase slightly. Due to limited computational resources, higher epochs could not be used, otherwise, validation loss might have also decreased significantly.

Figure 17 shows that the probability of predicting the correct label is better for the training set, which shows that there is a bit overfitting that is occurring as the model has not seen the validation set before and it does not seem to perform that well on it. There may be outliers as well that may be affecting the performance on the validation set.

The graph of accuracy for model 2 is shown in Figure 18.

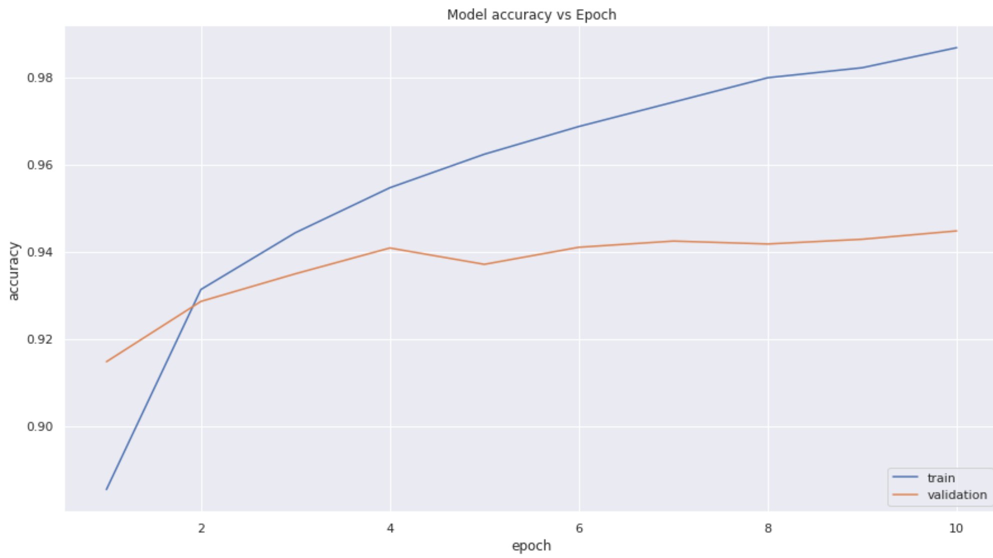

Figure 18: Accuracy graph of model number 2 vs epoch

The training accuracy increases at relatively higher rate and reaches values greater than 0.98. The validation accuracy increases in the begin of the epochs but slow down and plateaus around 0.94. If higher epochs were used, the validation accuracy might have improved.

Figure 18 does seem to show that there is a bit of overfitting by the neural network as it is performing above 0.98 with the training set, but only around 0.94 with the validation dataset. When taking the testing accuracy into, overfitting is likely the cause of the difference.

To counteract the overfitting, we can do regularization to overcome this issue. We can also add a dropout layer to resolve the overfitting problem. Early stopping is not possible as we are already using a small number of epochs.