ECE 657A Assignment 3

Submitted By :
Amandeep Kaur (21044104)
Bhupesh Dod (21046099)

Question 3 : Results Analysis

Solution:

3.1 Runtime Performance for training and testing CNNs

Default Network:

For this model, the EarlyStopping was used with monitor variable as validation loss and patience = 2 this helps in stopping the training of the model when the change in monitor variable is not significant. For default network, the **training runtime** with 100 epochs and batch size of 128 is **49.79** seconds. The **testing runtime** is **1.24** seconds. So, in nutshell, it took longer time to train the model with 813,157 learnable parameters as compared to test the model.

Own Network:

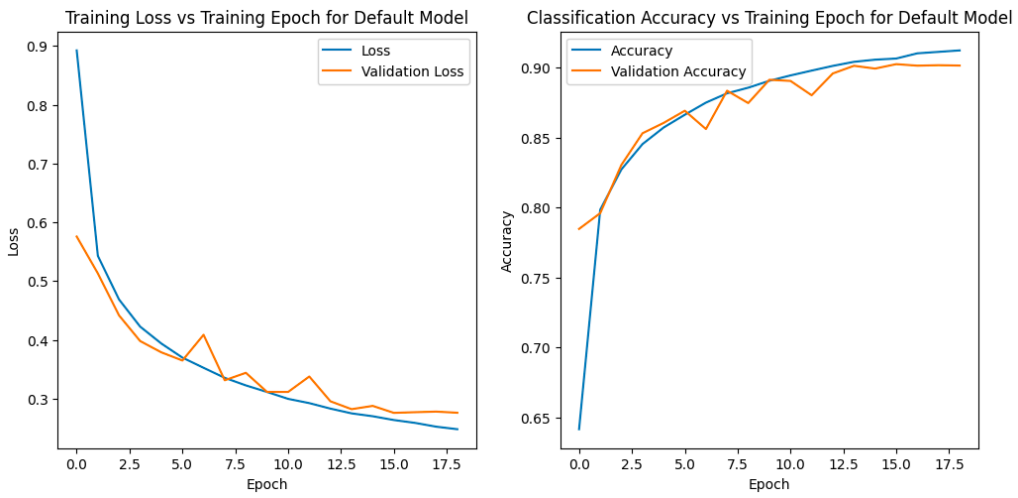
For this model, the EarlyStopping was used with monitor variable as validation loss and patience = 2 this helps in stopping the training of the model when the change in monitor variable is not significant. The **training runtime** with 100 epochs and batch size of 128 is **49.588** seconds, which is a considerable improvement from the default network. The **testing runtime** is **1.369** seconds which is more or less the same as default. To conclude, it took longer time to train the model with 138,277 learnable parameters than it took to test it.

3.2 Comparison of the different hyper-parameter and designs tried

Default Model:

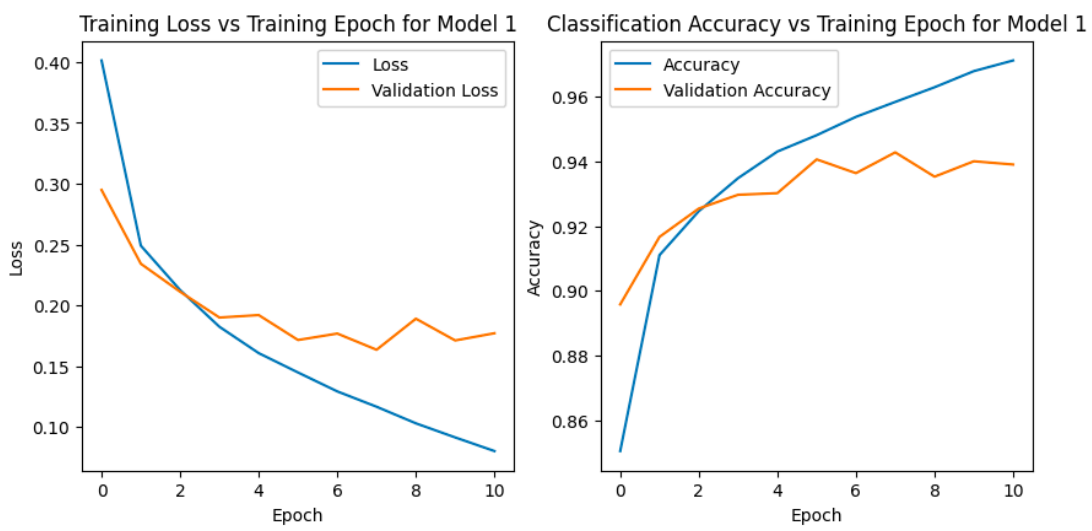
We first started with a basic model of 2 CNN layers with 32 kernels, 3*3 filter, 1 Stride and same padding, a Maxpool layer of pool_size 2*2, a flatten function and 2 fully connected layers. The hidden layers have activation function Relu and the output layer has softmax as the activation function and optimizer as Stochastic Gradient Descent.

Fit the model using EarlyStopping and with 100 epochs and 128 batch size. The accuracy of this model came out to be 89.68% and loss 29.20%.

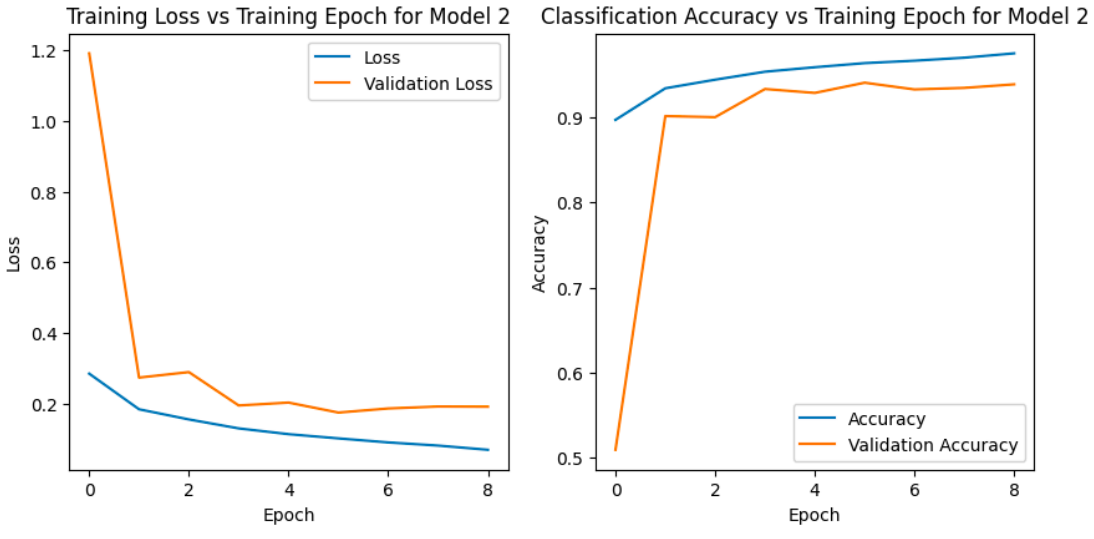


Model 1:

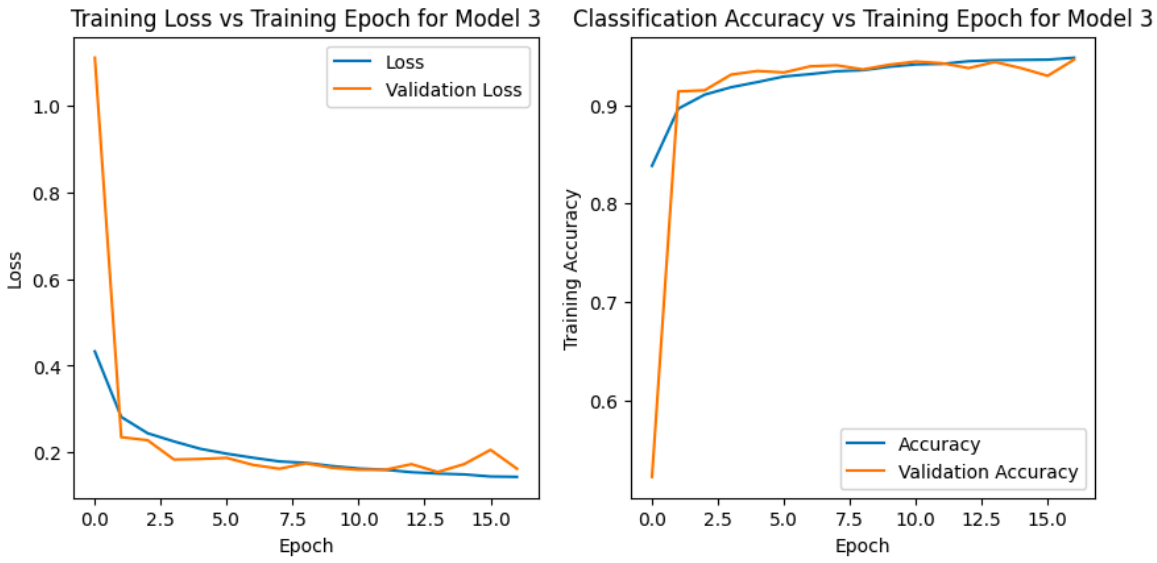
For this model, we changed the optimizer Adam as it is a widely used optimizer in CNNs, rest all the parameters are same as Default Model. The accuracy of this model increased to 93.70% and loss has decreased to 19.51%. Although the performance of this model has improved but the model is overfitting as can be inferred from learning loss curve (validation loss > training loss), so to avoid overfitting we add batch normalization and max pooling layer after second convolution layer.



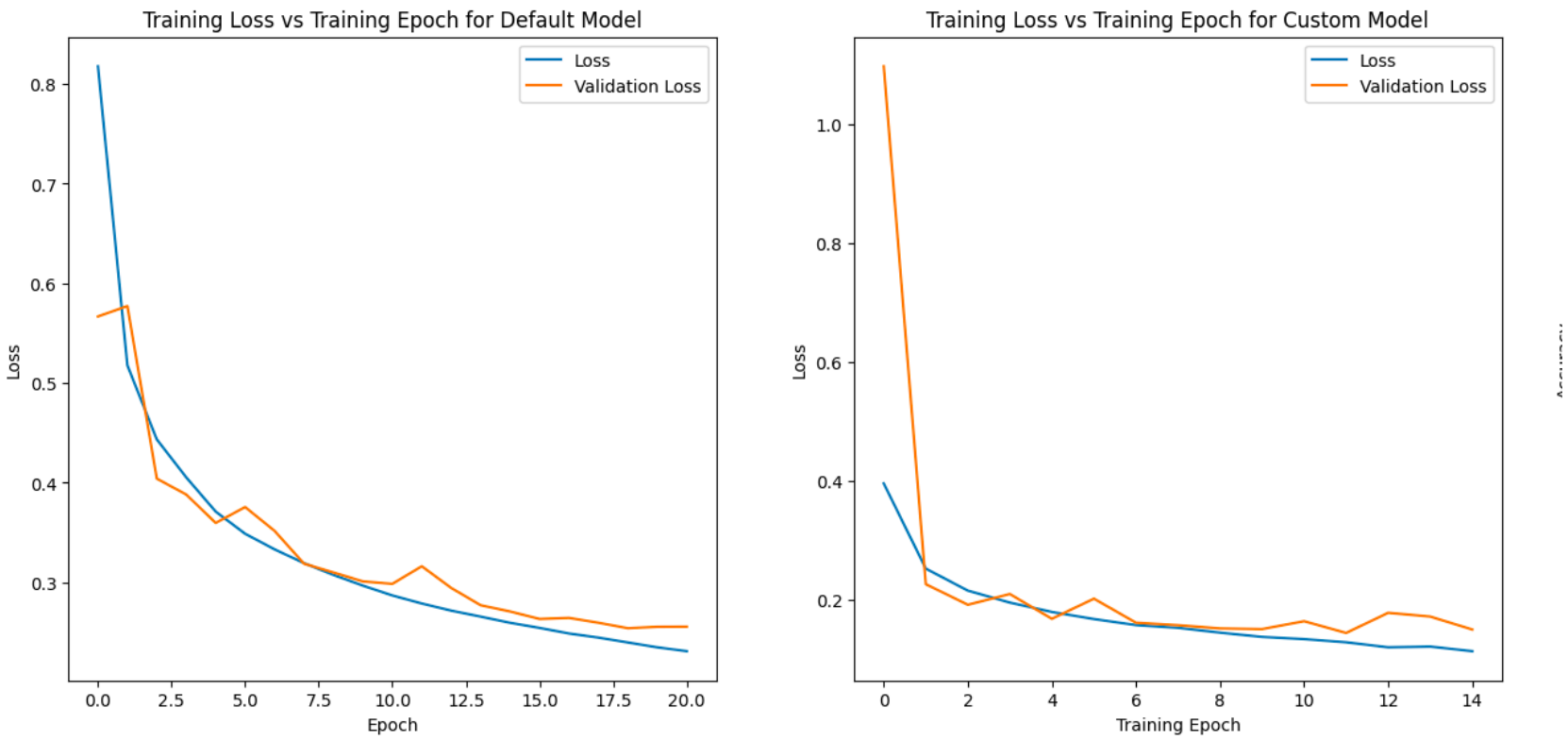
Model 2:
For this model, we add batch normalization and max pool layers to avoid overfitting of the model and reduce the validation loss. The accuracy of Model 2 is 93.52%, which is not an improvement from Model 1 and also the issue of overfitting is not resolved as it can be seen from the graphs below. So, to reduce the overfitting, we introduce the dropout layers.



Model 3:
This model adds the dropout layers on top of Model 2. By doing this, the problem of overfitting is resolved to a much extent. Also, the accuracy has improved from Model 2 and has increased to 94.02% and the loss is reduced to 17.48%.



3.3 Training loss vs. Training epoch of Default Model from Q1 and Custom Model from Q2
As the epochs increase, the loss also decrease but the gradient of decrease is also decreasing. For validation loss, the graph is spiked.



3.4 Classification accuracy vs. Training epoch of Default Model from Q1 and Custom Model from Q2

As the epochs increase, the accuracy of the model increases as well but the gradient of increase decreases gradually. There are more fluctuations in the validation accuracy for both the models as compared to the accuracy. Overall, the Custom Model performed better in terms of Accuracy. **Custom Accuracy : 94.16% Default Accuracy : 90.43%.**