Student Number: [redacted] Name: Bryan Hoang (16bch1)

1 Part 1: Image Classification using CNN (50 points)

The performance of the initial model on the training and testing datasets are 75.0 % and 75.1 %, respectively.

The following table summarizes the parameters and structure of the initial model below:

```
================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
================================================================================
ConvolutionalNeuralNetwork               --                        --
├─Conv2d: 1-1                            [32, 6, 28, 28]           456
├─MaxPool2d: 1-2                         [32, 6, 14, 14]           --
├─Conv2d: 1-3                            [32, 16, 10, 10]          2,416
├─MaxPool2d: 1-4                         [32, 16, 5, 5]            --
├─Linear: 1-5                            [32, 128]                 51,328
├─Linear: 1-6                            [32, 13]                  1,677
================================================================================
Total params: 55,877
Trainable params: 55,877
Non-trainable params: 0
Total mult-adds (M): 20.87
================================================================================
Input size (MB): 0.39
Forward/backward pass size (MB): 1.65
Params size (MB): 0.22
Estimated Total Size (MB): 2.27
================================================================================
```

Steps to replicate the experiment:

1. Setting up the notebook by importing necessary models and packages. Notably, PyTorch will be used instead of Keras.
2. Define the hyperparameters for the initial model.
3. Define a custom `FashionProductImageDataset` class to load the data.
4. Define a transformer for the images to all be the same size and normalize the values into the range of $[-1, 1]$.
5. Load the dataset using the custom class created into step 3. Note that this is the directory structure of the files:

```
.
|-- data
|   |-- img/
|   |-- test.csv
|   `-- train.csv
`-- cmpe_351_asgmt_2.ipynb
```

6. Test the flow of data through layers mimicing the initial model to validate dimensions with a trial.
7. Initialize the model and summarize it.
8. Train the model with 2 epochs.
9. Evaluate the model.

2 Part 2: Improved Image Classification (50 points)

A 3rd convolutional layer was added to the base CNN model in addition to increasing the number of epochs from 2 to 4.

The following table summarizes the parameters and structure of the improved model below:

```
================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
================================================================================
ConvolutionalNeuralNetworkV2             --                        --
├─Conv2d: 1-1                            [32, 6, 28, 28]           456
├─MaxPool2d: 1-2                         [32, 6, 15, 15]           --
├─Conv2d: 1-3                            [32, 16, 11, 11]          2,416
├─MaxPool2d: 1-4                         [32, 16, 6, 6]            --
├─Conv2d: 1-5                            [32, 32, 2, 2]            12,832
├─MaxPool2d: 1-6                         [32, 32, 2, 2]            --
├─Linear: 1-7                            [32, 32]                  4,128
├─Linear: 1-8                            [32, 13]                  429
================================================================================
Total params: 20,261
Trainable params: 20,261
Non-trainable params: 0
Total mult-adds (M): 22.58
================================================================================
Input size (MB): 0.39
Forward/backward pass size (MB): 1.74
Params size (MB): 0.08
Estimated Total Size (MB): 2.22
================================================================================
```

The performance of the improved model on the testing datasets is 81.6 % accuracy, respectively. This is approximately an 6.5 % improvement in performance over the initial model on the testing dataset.

Steps to replicate the experiment:

1. Setting up the notebook by importing necessary models and packages. Notably, PyTorch will be used instead of Keras.
2. Define the hyperparameters for the initial model.
3. Define a custom `FashionProductImageDataset` class to load the data.
4. Define a transformer for the images to all be the same size and normalize the values into the range of $[-1, 1]$.
5. Load the dataset using the custom class created into step 3.

```
.
|-- data
|   |-- img/
|   |-- test.csv
|   `-- train.csv
`-- cmpe_351_asgmt_2.ipynb
```

6. Test the flow of data through layers mimicing the initial model to validate dimensions with a trial, **now with a 3rd convolutional layer**.

Student Number: [redacted] Name: Bryan Hoang (16bch1)

7. Initialize the model and summarize it, **now with a 3rd convolutional layer**.
8. Train the model, **now with 4 epochs**.
9. Evaluate the model.