Part I: CNN Classification

1. Describe the CNN model you have defined.

1. Input Layer:
   The model expects images with three color channels (like RGB) as input.

2. Feature Extraction:
   a. Convolutional Layers: These layers detect features in the input images. The numbers (e.g., 64, 128) represent the number of filters used in each convolutional layer.
   b. ReLU Activation: After each convolutional layer, a ReLU activation function is applied.
   c. Max Pooling: These layers reduce the spatial dimensions of the feature maps, helping to decrease computational complexity and reduce overfitting.

3. Fully Connected Layers :
   a. Flatten Layer: The feature maps from the last convolutional layer are flattened into a single vector to be fed into the fully connected layers.
   b. Linear Layer: These layers perform classification based on the features extracted earlier.
   c. ReLU Activation & Dropout: Similar to the convolutional layers, ReLU activation functions introduce non-linearity. Dropout layers randomly set a fraction of input units to zero during training.
   d. Final Linear Layer: This layer produces the final output, which in this case is the predicted probabilities for each of the 1000 classes.

4. Output:
   The model outputs a vector of probabilities, where each element represents the likelihood of the input image belonging to a particular class.

2. Describe how the techniques (regularization, dropout, early stopping) have impacted the performance of the model.

1. Base Case:
   1. The base case represents the initial performance of the model without any additional techniques applied.
   2. In this scenario, the model achieves an accuracy of 77.78% after one epoch of training.

2. Base Case + Regularization:
   1. It is used to prevent the model from focusing too much on small details that might not be important.
   2. By adding regularization, the model's performance improved slightly from the base case accuracy of 77.78% to 78.51% after one epoch.
   3. Regularization helps the model focus on important patterns and prevents it from memorizing noise in the training data.

3. Base Case + Regularization + Dropout + Early Stooping:
   1. Dropout prevents the model from becoming too dependent on specific features, leading to better generalization to new data.
   2. By stopping early, the model avoids memorizing noise in the data and focuses on learning meaningful patterns, leading to better performance on new data.
   3. With early stopping added to regularization and dropout, the model's accuracy remains at 82.00% after one epoch.

4. Base Case + Regularization + Dropout + Early Stooping + Image augmentation:
   1. This helps in exposing the model to a more diverse set of examples, making it more robust and better at recognizing patterns in new, unseen data.
   2. By incorporating image augmentation along with regularization, dropout, and early stopping, the model's performance significantly improved to 83.42% after ten epochs.

In summary, each technique (regularization, dropout, early stopping, and image augmentation) plays a crucial role in improving the model's performance by helping it focus on important patterns, prevent overfitting, and generalize better to new data.

3. Discuss the results and provide relevant graphs:

a. Report training accuracy, training loss, validation accuracy, validation loss, testing accuracy, and testing loss.

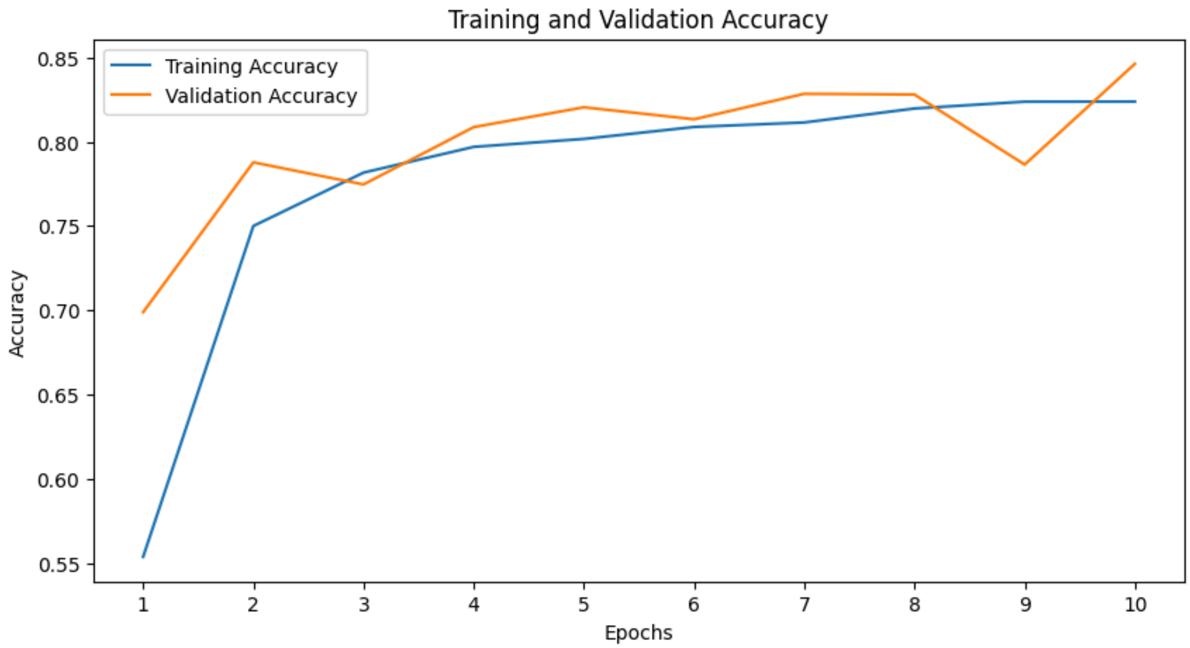
1. Training Loss and Accuracy:
   a. The training loss of 0.4597 indicates that, on average, the model's predictions are approximately 0.4597 units away from the actual values during training.
   b. The training accuracy of 82.38% means that the model correctly predicted the class labels for approximately 82.38% of the training data.

2. Validation Loss and Accuracy:
   a. The validation loss of 0.4134 indicates that, on average, the model's predictions are approximately 0.4134 units away from the actual values during validation.
   b. The validation accuracy of 84.62% means that the model correctly predicted the class labels for approximately 84.62% of the validation data.
   c. This suggests that the model is performing slightly better on validation set compared to the training data.

3. Test Loss and Accuracy:
   a. The test loss of 0.4281 indicates that, on average, the model's predictions are approximately 0.4281 units away from the actual values during testing.
   b. The test accuracy of 83.42% means that the model correctly predicted the class labels for approximately 83.42% of the test data.
   c. This indicates that the model is performing consistently well on the test set, with an accuracy close to that observed during validation.

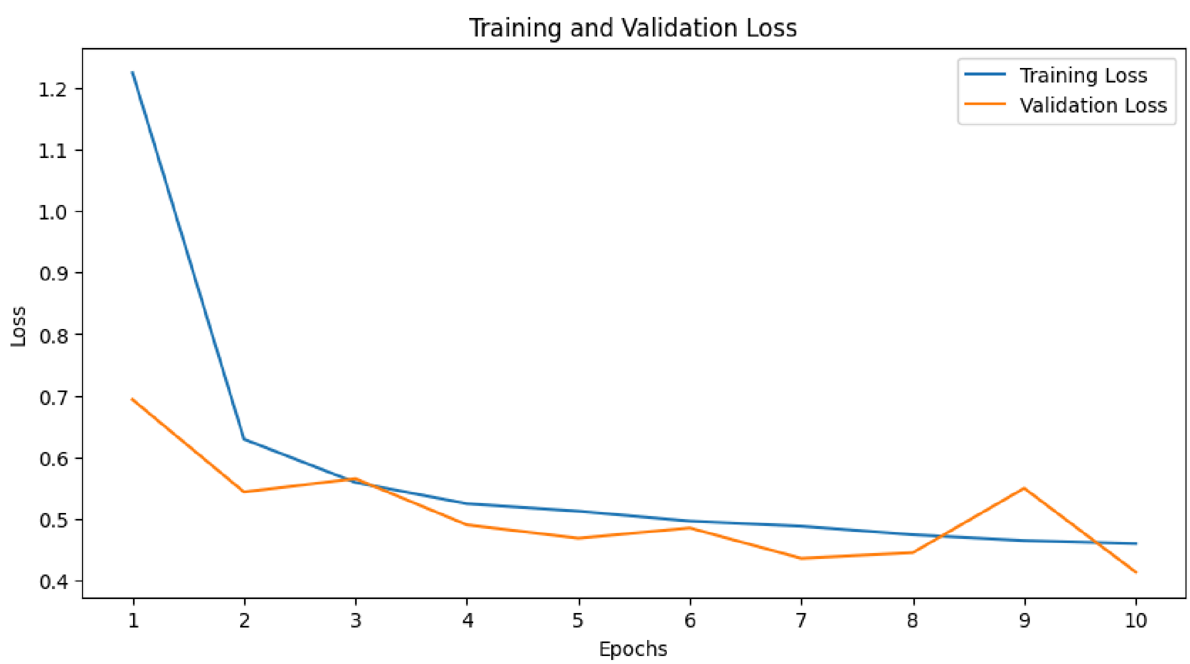
In simple terms, the model is learning well from the training data, generalizing effectively to unseen data, and making accurate predictions with consistent performance across different datasets.

b. Plot the training and validation accuracy over time (epochs).



1. The training accuracy is lower than the validation accuracy for most of the epochs. As a result, the model performs well on the validation data but poorly on training data.
2. The training accuracy starts at around 0.55 and increases to about 0.80 by the end of the training. The validation accuracy starts at around 0.70 and increases to about 0.85 by the end of the training.

c. Plot the training and validation loss over time (epochs).



1. The graph shows that the training loss is generally higher than the validation loss. This means that the model is performing better on the validation data than it is on the training data.
2. The training loss is decreasing over time, which means that the model is learning.
3. The validation loss is also decreasing over time, but at a slower rate than the training loss.

d. Generate a confusion matrix using the model's predictions on the test set.

Confusion Matrix:
[[1196 187 139]
 [180 1192   76]
 [123   55 1352]]

1. The top-left value, 1196, indicates the number of instances where the model correctly predicted the first class when it was actually the first class.
2. Moving to the diagonal, the value 1192 represents the correct predictions of the second class when the true class was the second class.
3. The bottom-right value, 1352, shows the accurate predictions of the third class when the true class was indeed the third class.

e. Report any other evaluation metrics used to analyze the model's performance on the test set.

Evaluation Metrics:
Precision: 0.8307
Recall: 0.8311
F1 Score: 0.8308

1. Precision: Precision tells us how many of the predicted positive cases are actually positive. In this case, the precision is approximately 83.07%, meaning that out of all the images predicted as positive, around 83.07% were correctly identified.
2. Recall: Recall indicates the proportion of actual positive cases that were correctly identified by the model. Here, the recall is about 83.11%, suggesting that out of all the

positive images in the dataset, approximately 83.11% were accurately detected by the model.

3. F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall. With a value of approximately 83.08%, the F1 score indicates how well the model balances precision and recall, with higher values indicating better performance overall.

This detailed analysis helps identify any potential strengths or weaknesses of the model across different classes, providing valuable insights for improvement if necessary.