CNN Classification

1. Describe the CNN model you have defined.

→

The CNN model we have defined is a variant of the VGG architecture, specifically VGG13.

1. Input Layer:
   - Accepts 3-channel images (assuming RGB images with channels for red, green, and blue).

2. Feature Extraction (Convolutional Layers):
   1 - Convolutional layer with 64 filters, each with a kernel size of 3x3 and padding of 1.
      - ReLU activation function applied element-wise after each convolution.
   2 - Convolutional layer with 64 filters, same kernel size, and padding.
      - ReLU activation.
      - Max-pooling layer with a kernel size of 2x2 and a stride of 2 (reducing spatial dimensions).

   3 - 1st Convolutional layer with 128 filters.
      - ReLU activation.
   4 - 2nd Convolutional layer with 128 filters.
      - ReLU activation.
      - Max-pooling layer.

   5 - Convolutional layer with 256 filters.
      - ReLU activation.
   6 - Convolutional layer with 256 filters.
      - ReLU activation.
      - Max-pooling layer.

   7 - Convolutional layer with 512 filters.
      - ReLU activation.
   8 - Convolutional layer with 512 filters.
      - ReLU activation.
      - Max-pooling layer.

   9 - Convolutional layer with 512 filters.
      - ReLU activation.
   10 - Convolutional layer with 512 filters.
      - ReLU activation.

- Max-pooling layer.

3. Fully Connected Layers (Classifier):
   - Flattening the output from the convolutional layers to a 1D tensor.
   11 - 1st Fully connected layer with 4096 neurons.
      - ReLU activation.
      - Dropout layer for regularization.
   12 - 2nd Fully connected layer with 4096 neurons.
      - ReLU activation.
      - Dropout layer.
   13 - 3rd Fully connected layer with 1000 neurons (output layer)

4. Output:
   - The final output size is determined by the last fully connected layer, which has 1000 neurons for ImageNet classification. We can adapt this model for a different number of classes by modifying the `num_classes` parameter in the constructor and adjusting the last fully connected layer accordingly.

The above VGG13 model consists of a stack of convolutional layers with ReLU activation followed by max-pooling layers for feature extraction, and it concludes with fully connected layers for classification.

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

→

1. Base Model had 76.1% Test Accuracy:
   - This represents the performance of the initial VGG13 model without any additional regularization techniques, dropout, or data augmentation.

2. Base Model + L1 Regularization + Dropout Layer had (82.29% Test Accuracy):
   - L1 Regularization: This technique involves adding a penalty term to the loss function based on the absolute values of the weights. It helps prevent overfitting by encouraging sparsity in the learned weights.
   - Dropout Layer: Dropout randomly drops some neurons during training, which acts as a form of regularization by preventing co-adaptation of hidden units. This technique introduces noise during training and helps in creating a more robust model.

The model with L1 regularization and dropout layer achieved an improvement in test accuracy from 76.1% (base model) to 82.29%. This indicates that L1 regularization helped in reducing overfitting and the dropout layer has helped the model generalize better to unseen data.

3. Base Model + L1 Regularization + Dropout Layer + Image Augmentation + Early Stopping had 83.27% Test Accuracy:
   - Image Augmentation: This technique involves applying random transformations to the input images during training, such as rotation, flipping, and zooming which helps in creating a more diverse training dataset and improves the model's ability to generalize to different variations of the input data.
   - Early stopping involves monitoring the model's performance on a validation set and stopping training once performance stops improving.

The test accuracy increased from 82.29% to 83.27% with early stopping and image augmentation. This suggests that image augmentation has enhanced the model's robustness and generalization capabilities and Early stopping likely prevented the model from overfitting to the training data, allowing it to halt training at an optimal point and avoid degradation in generalization performance.

3. Discuss the results and provide relevant graphs:

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

→

For the best model, below were the statistics -

1 - Training accuracy = 82.10%

2 - Training loss = 0.4709

3 - Validation accuracy = 83.02%

4 - Validation loss = 0.4416

5 - Testing accuracy = 83.18% (3rd Epoch)

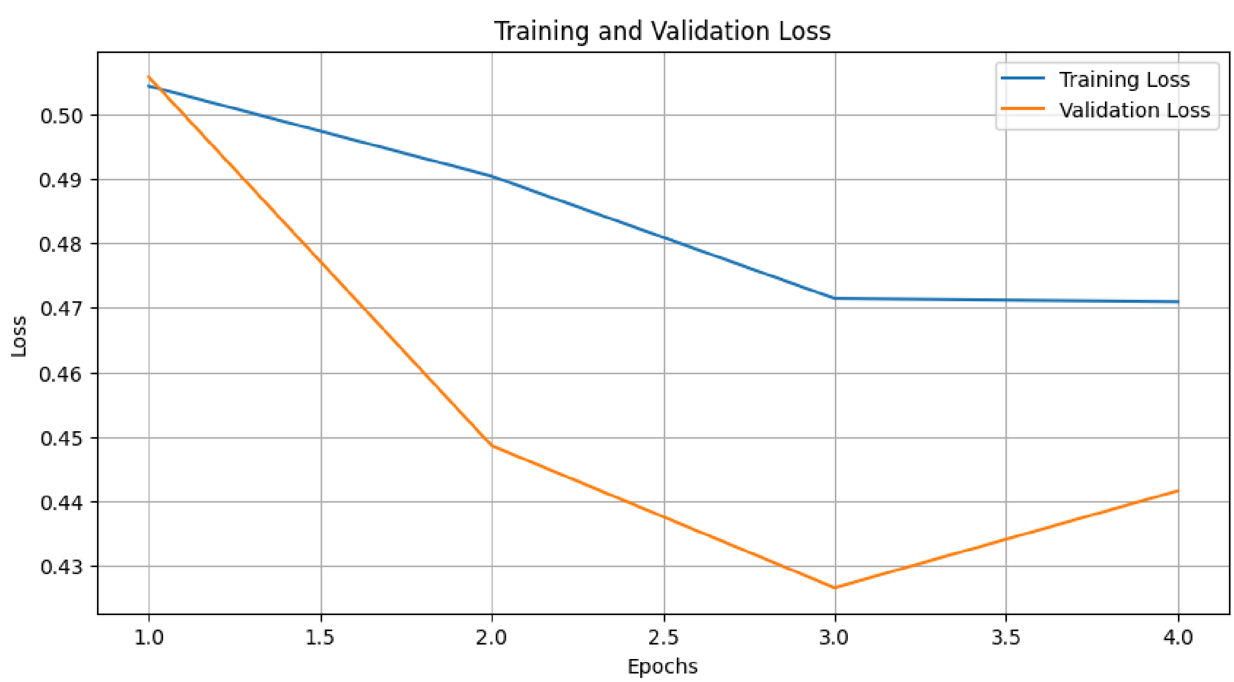
6 - Testing loss = 0.4338 (3rd Epoch)

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

→

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

→



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

→

Confusion Matrix:

```
[[1282  157   53]
 [ 209 1252   17]
 [ 218  121 1191]]
```

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

→

Precision: 0.8386

Recall: 0.8278

F1 Score: 0.8292