Assignment 6: Fit a CNN to a HuggingFace Dataset

Dataset and Preprocessing

The dataset used for training was the Fruit Detection Dataset from the Hugging Face Datasets Library. The dataset consists of images of different fruit types labeled into three categories: 0 = Apple, 1 = Banana, 2 = Orange. The following preprocessing steps were applied:

- Images were resized to 224x224 pixels.
- Images were normalized using torchvision.transforms.ToTensor().
- The dataset was split into 80% training, 10% validation, and 10% test sets.

CNN Model Architecture

The model was built using **PyTorch** and consists of:

- Three convolutional layers with increasing filter sizes (12, 36, and 64 filters) and kernel sizes of 5x5, 3x3, and 3x3, respectively.
- Max pooling layers after each convolution to reduce spatial dimensions.
- Batch normalization and ReLU activations to improve learning stability.
- A fully connected output layer that maps features to three class labels.
- The model was trained using the Adam optimizer with a learning rate of 0.01 and the CrossEntropyLoss function.

Training and Evaluation

The model was trained for 30 epochs, and performance was evaluated using accuracy, precision, recall, and F1-score. The following observations were made:

- Loss curve: The loss decreased significantly in the initial epochs and stabilized after ~10 epochs, indicating effective learning.
- Accuracy curve: Training accuracy on the validation set showed steady improvement, reaching ~85%.
- Classification Report: The final test accuracy was 89.3%, with precision, recall, and F1-scores close to 0.89-0.90.
- Confusion Matrix: The model had minor misclassifications but performed well across all classes.

Conclusion and Insights

The CNN achieved a strong classification performance on the fruit dataset, demonstrating the effectiveness of convolutional layers for feature extraction. Minor misclassifications suggest potential improvements using data augmentation, learning rate tuning, or transfer learning with pre-trained models. Overall, the model effectively learned distinguishing features and generalized well to unseen test data.