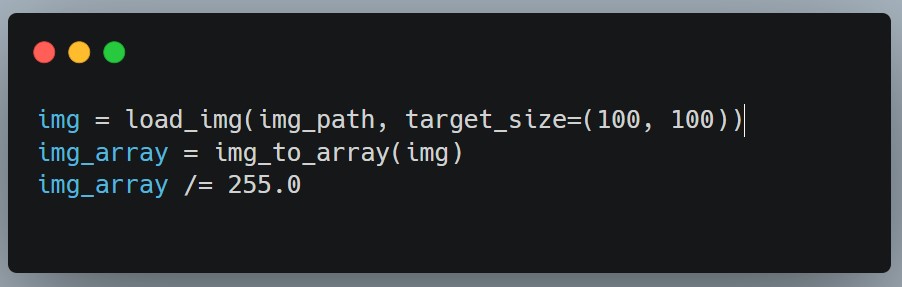
**Neural Network Project**

**“ Image Classification ”**

Data Preparation Steps :

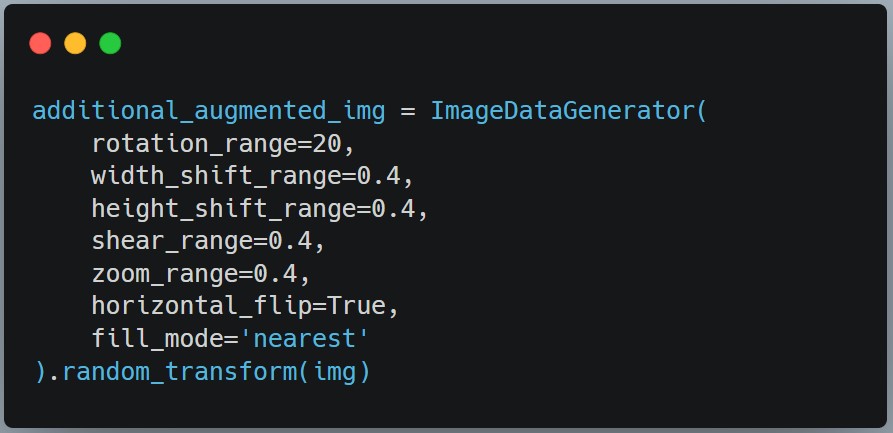
1. Loading and Normalizing Images :
   * + Images from the training dataset are loaded and resized to (100, 100) pixels
     + The pixel values are normalized to the range ]1 ,0[



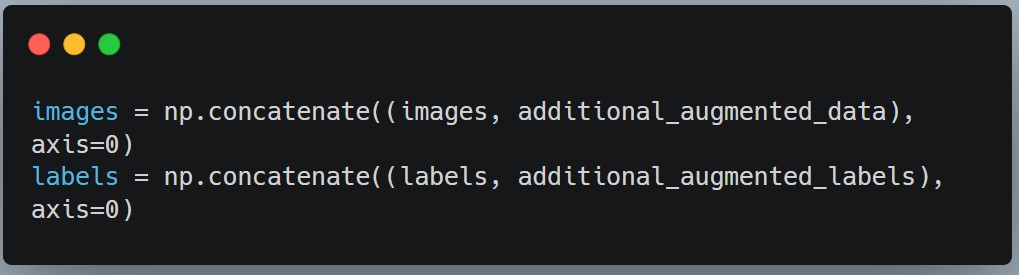
1. Labeling and Categorization :
   * + Labels are assigned to images based on their respective directories.
     + LabelEncoder is used to convert string labels into numeric labels.
     + One-hot encoding is applied to the numeric labels.



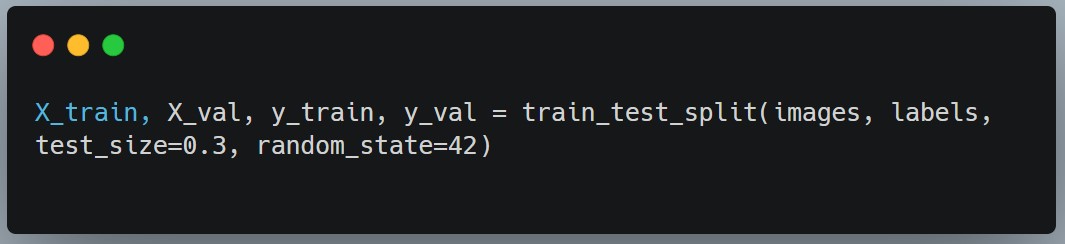
1. Data Augmentation :
   * Random transformations (rotation, shift, shear, zoom, flip) are applied to original images.
   * Augmented images are stored in separate lists.



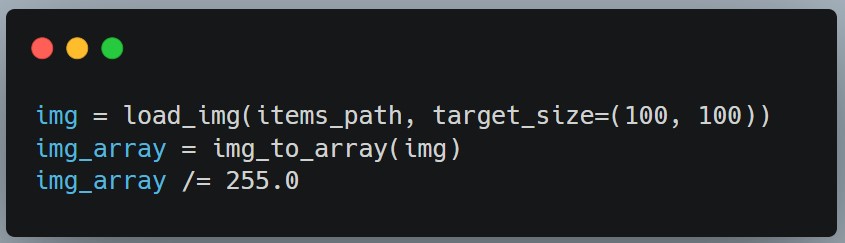
1. Extending Training Data :
   * Augmented images and their original labels are added to the training dataset.



1. Train-Validation Split :
   * The dataset is split into training and validation sets.

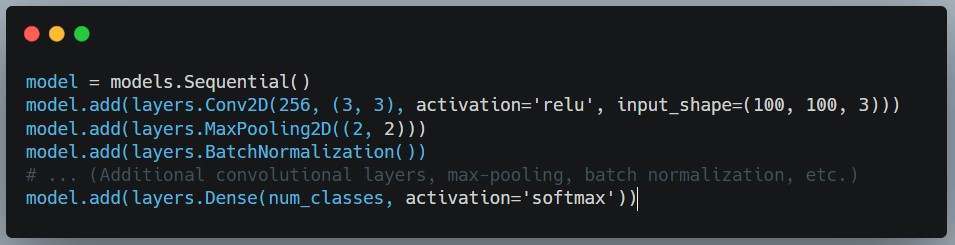


1. Loading and Preprocessing Test Data :
   * Test images are loaded and resized to (100, 100) pixels.
   * Pixel values are normalized.
   * Test labels are encoded and one-hot encoded.

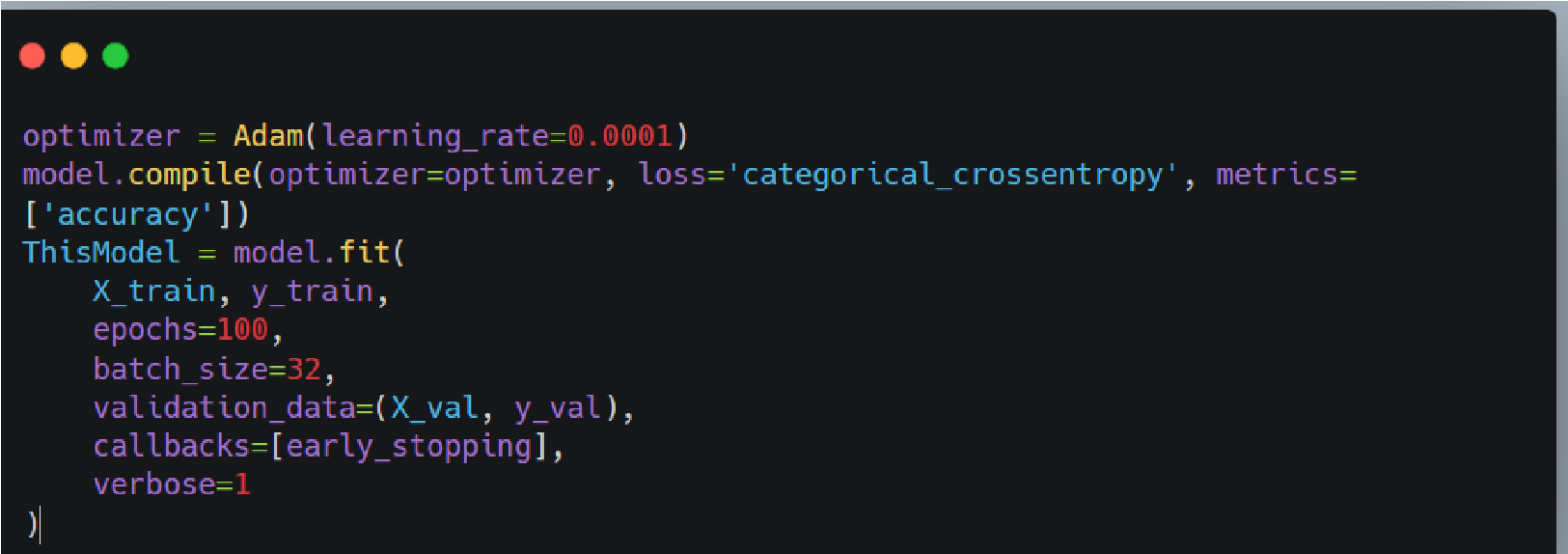


Models Architecture :

1. Convolutional Neural Network (CNN) :
   * Sequential model with convolutional layers, max-pooling layers, batch normalization, dense layers, and dropout layers.
   * The architecture consists of four convolutional layers with decreasing filter sizes and max-pooling after each convolution.
   * Batch normalization is applied after each max-pooling layer.
   * The flattened output is connected to two dense layers with ReLU activation and dropout for regularization.
   * The final output layer has softmax activation for multi-class classification.

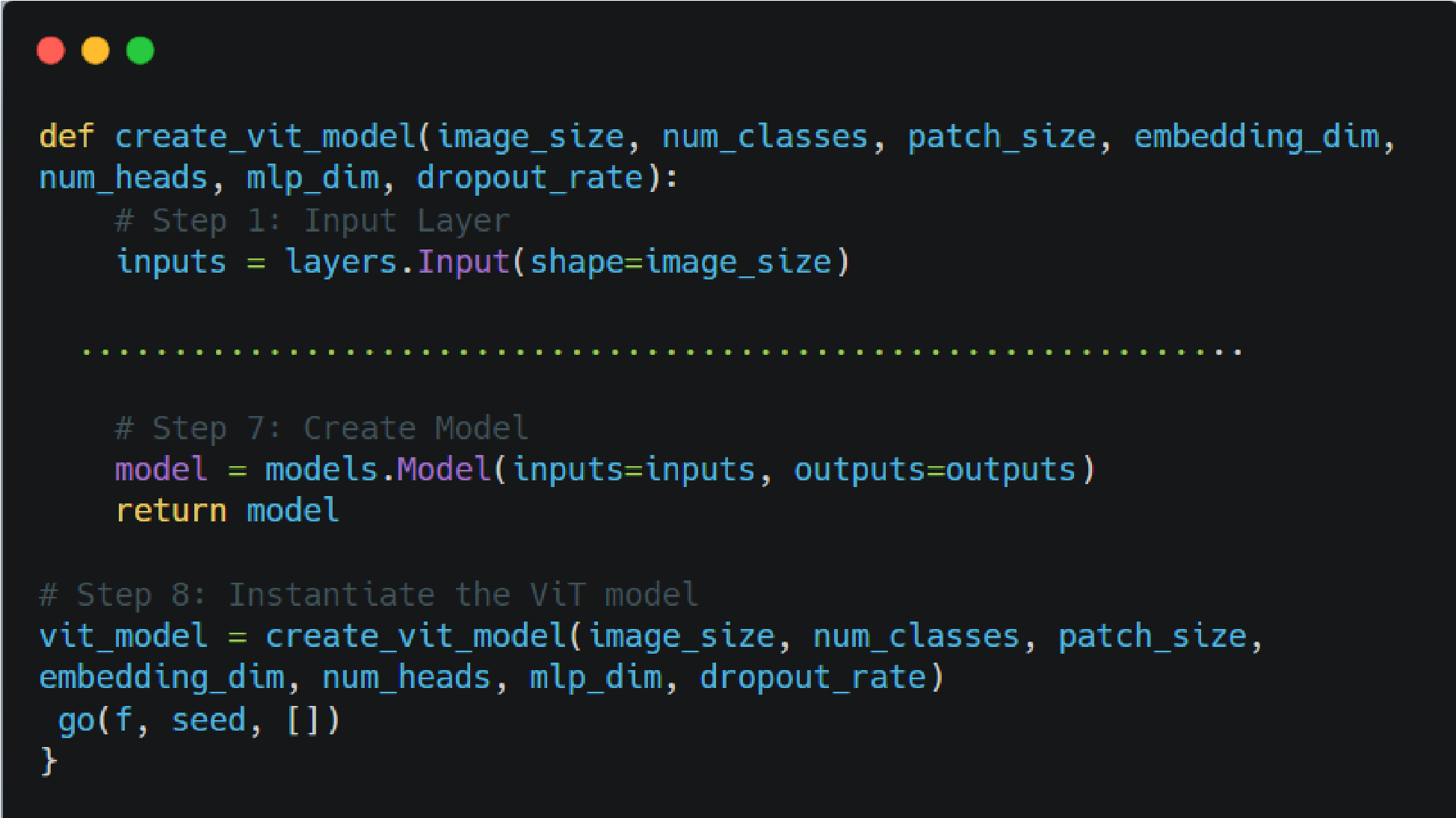


1. Model Compilation & Training :
   * Adam optimizer with a learning rate of 0.0001 is used.
   * Categorical crossentropy is chosen as the loss function.
   * Accuracy is chosen as the metric for evaluation.
   * The model is trained using the training dataset with early stopping to prevent overfitting.
   * Training is performed for 100 epochs with a batch size of 32.



Vision Transformer Model ( ViT ) :

1. Model Architecture :
   * Patching Layer: Applies convolution to split the input image into patches of size patch\_size x patch\_size with a stride of patch\_size
   * Positional Encoding: Embeds positional information into the patches using an embedding layer.
   * Multi-Head Attention: focus on different parts of the input.
   * Residual Connection: Adds the original input to the attention output
   * Layer Normalization: Normalizes the output to stabilize training
   * Feedforward Network: Applies a feedforward neural network to capture complex patterns
   * Positional Encoding Update: Updates the positional\_encoded\_patches for the next iteration.
   * Global Average Pooling: Aggregates the information from all patches into a fixed-size vector by taking the average
   * Classifier: A dense layer with softmax activation for classification. Create Model: Instantiates the ViT model using the defined inputs and outputs

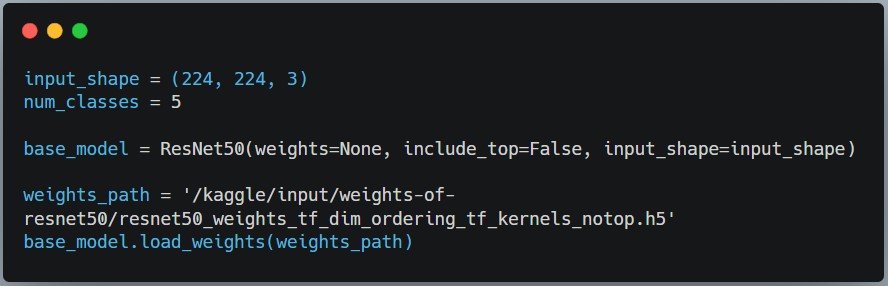


1. Model Compilation & Training :
   * compiles the model, specifying the optimizer, loss function, and metrics
   * Preprocesses the data and trains the Vision Transformer model
   * prints the details of the model, including layer information and parameter counts
   * Evaluates the trained model on the test set and prints the accuracy
   * Makes predictions on the test set and converts one-hot encoded predictions to class labels



Another Model ( ResNet50 ) :

1. Base ResNet50 Model:
   * ResNet50 is used as a base model for feature extraction.
   * The pre-trained weights are loaded .



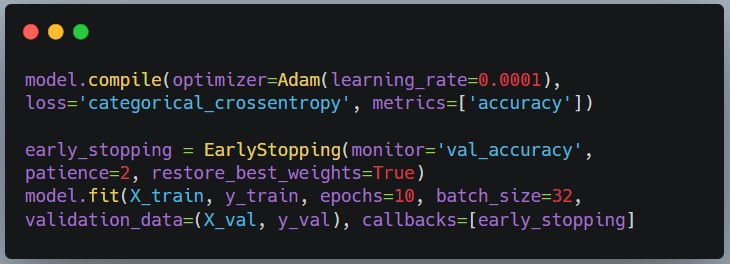
1. Custom Model Head:
   * A Sequential model is constructed by stacking the base ResNet50 model with additional layers.
   * The base model is followed by a Flatten layer, a Dense layer with ReLU activation, a Dropout layer, and a final Dense layer with softmax activation for classification.



1. Model Compilation and Training:

* + The model is compiled with a lower learning rate using the Adam optimizer.

* + Early stopping is applied during training to prevent overfitting.



Different Trials & Results :

1. CNN Model , First Trial :
   * The model was trained for 100 epochs with early stopping.
   * The model's accuracy on the validation set improved gradually, reaching

68.33% at the end.

However, the final test accuracy was 36.00%

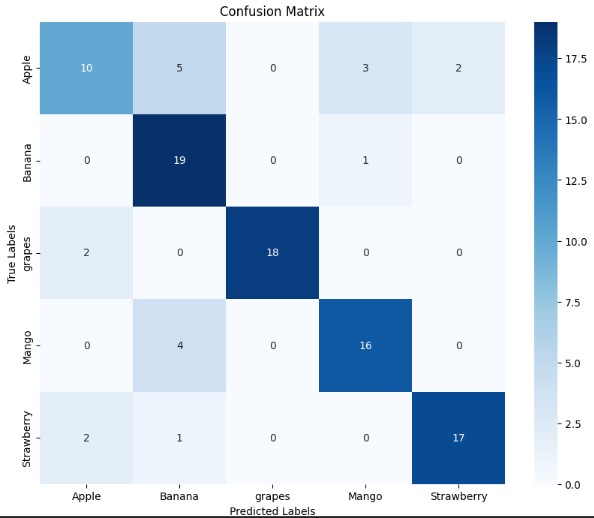
1. CNN Model , Second Trial :
   * The model was trained for 100 epochs with early stopping.
   * The model's accuracy on the validation set improved gradually, reaching

76.65% at the end.

However, the final test accuracy was 69.00%

2. CNN Model , Third Trial :

* The model was trained for 100 epochs with early stopping.
* The model's accuracy on the validation set improved gradually, reaching

92.29% at the end.

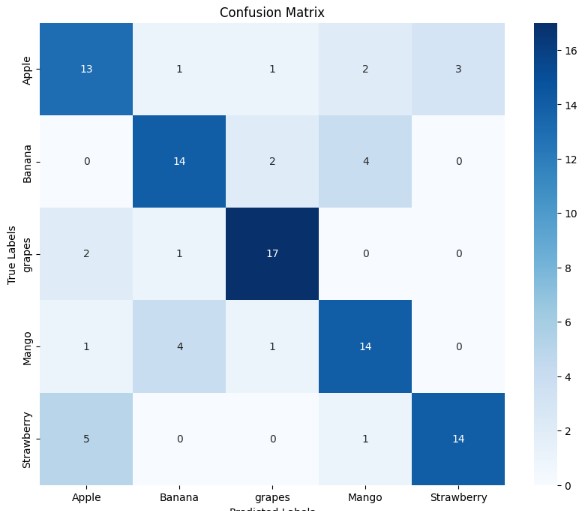
However, the final test accuracy was 80.00%



( Every Trial we changed the values of parameters , number of Layers , etc)

1. ViT Model:
   * The ResNet50-based model was trained for 100 epochs with early stopping.
   * The Training accuracy gradually increases over the epochs and reaches a peak of approximately 87.08 %

However, the final test accuracy was 72.00 %



1. ResNet50 Model:
2. ResNet50 Model , First Trial :
   * The model was trained for 10 epochs
   * The model's accuracy on the validation set improved gradually, reaching

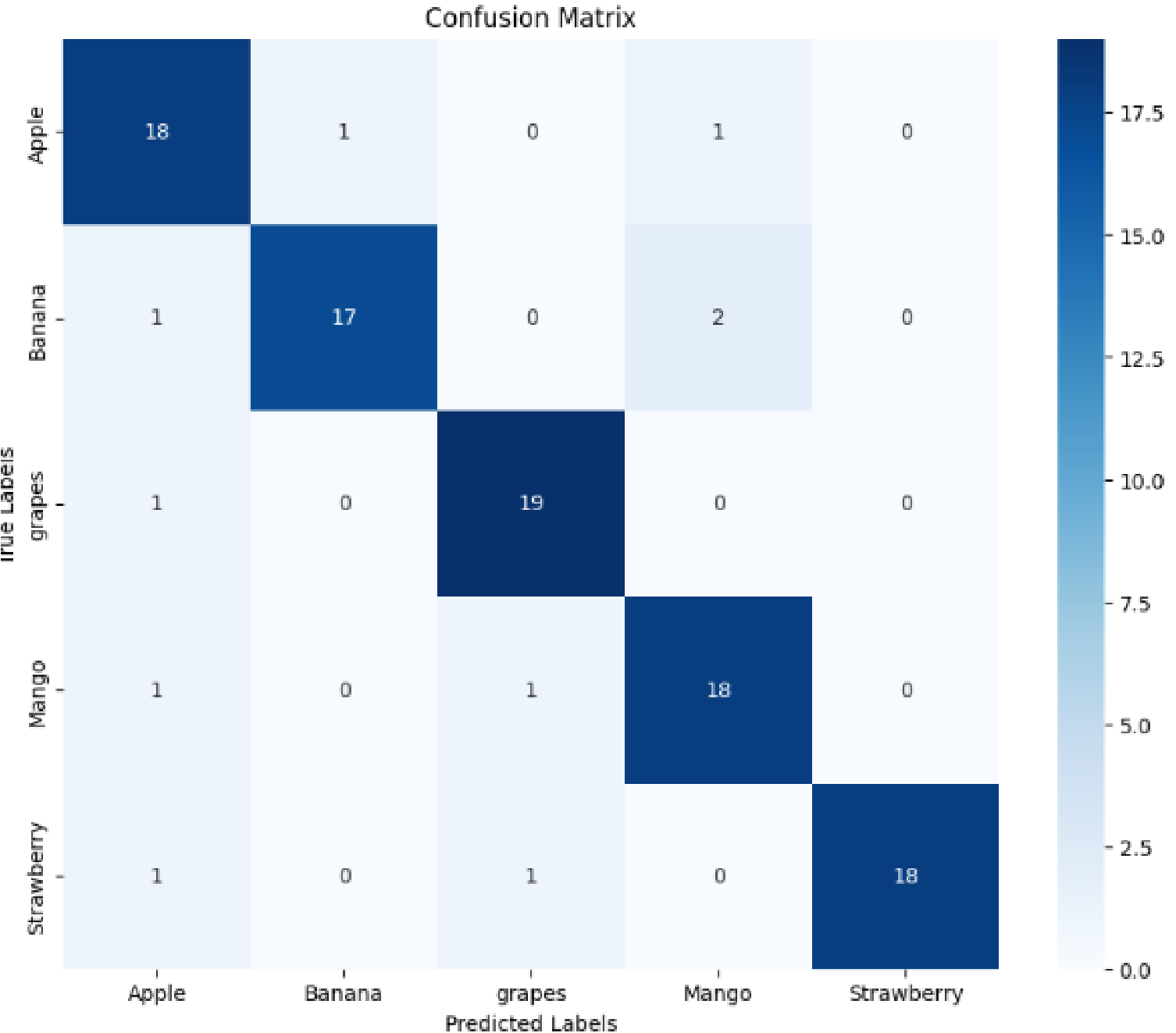
86.87% at the end.

However, the final test accuracy was 36.00%

1. ResNet50 Model , Second Trial :
   * The model was trained for 10
   * The model's accuracy on the validation set improved gradually, reaching

95.53% at the end.

However, the final test accuracy was 90.00%



1. Summary & Conclusion:

1. CNN Model:

* + Significant improvement was achieved, reaching a validation accuracy of 92.29 %, and a final test accuracy of 80.00 %

2. Vision Transformer (ViT) Model:

* + The ViT model, demonstrated consistent improvement during training, achieving a peak training accuracy of approximately 87.08% and the final test accuracy was 72.00 %.

3. ResNet50 Model:

* + Extended training (10 epochs) led to substantial improvement with a validation accuracy of 95.53 % and a higher final test accuracy of 90.00 %.

Conclusion :

* + CNN models displayed sensitivity to architecture and hyperparameter changes, showcasing a significant impact on performance.
  + ResNet50 demonstrated superior performance compared to the CNN models, especially with extended training in the second trial.
  + The Vision Transformer model, while achieving a relatively high training accuracy, showed a slight performance gap in the test set compared to ResNet50.