8/2/2024



**Knowledge Test**

**TIDKE ASHOK TATERAO**

nsti cALICUT

**Knowledge Test**

**AIM:**

Implement various deep learning tasks using Keras and TensorFlow, including building a neural network, data augmentation, custom loss functions, and transfer learning.

**Requirements:**

* Pc/Laptop
* VS Code
* Chrome
* Python installed with required libraries (Keras, TensorFlow, numpy, matplotlib)

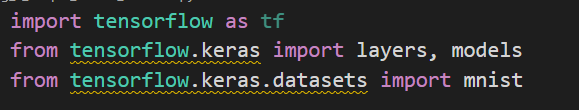
**Learning Outcome:**

Apply and understand the steps involved in building neural networks, augmenting data, implementing custom loss functions, and using transfer learning for image classification tasks.

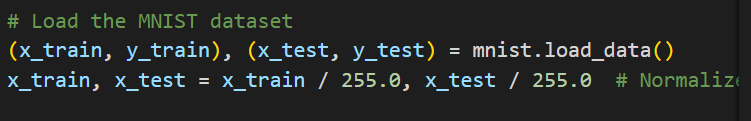
**Procedure:**

**1.** **Building a Simple Neural Network**

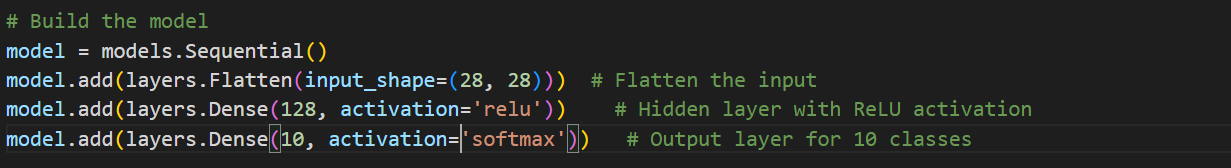
**Step-1: Import necessary packages**



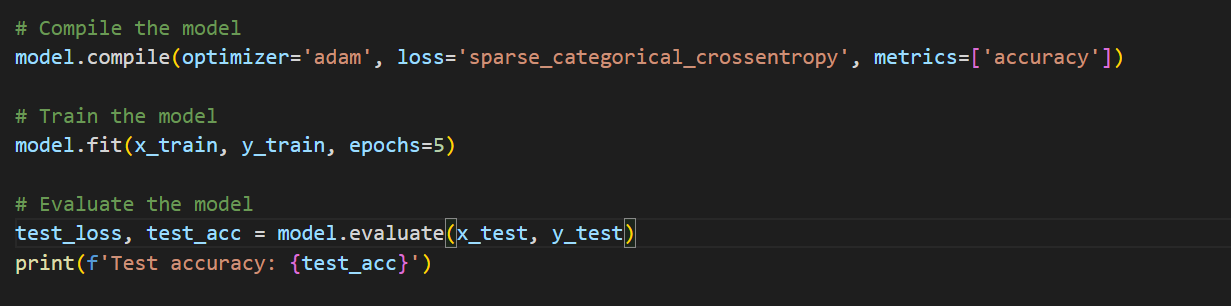
**Step-2: Load and preprocess the MNIST dataset**



**Step-3: Build the neural network model**

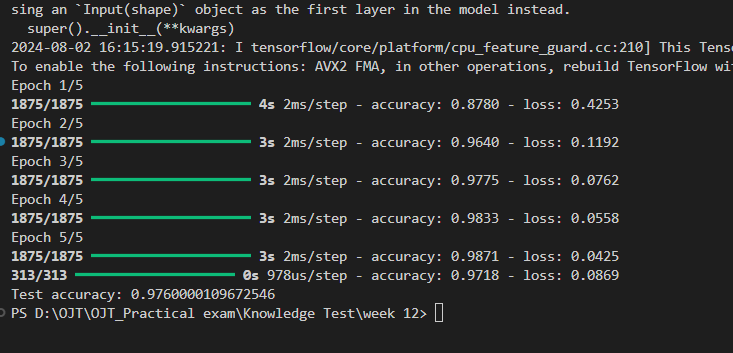


**Step-4: Compile and train the model**

****

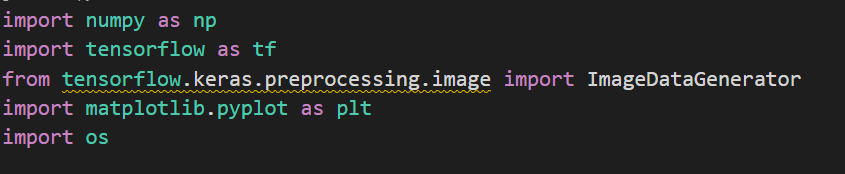
**Output:**

* Trained neural network on MNIST dataset with training and validation accuracy.

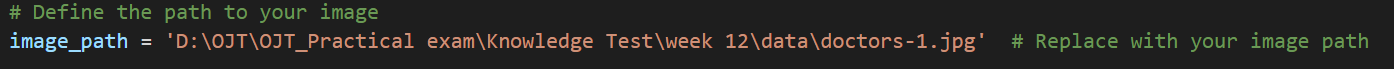
****

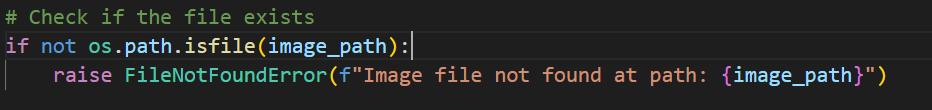
**2.** **Data Augmentation**

**Step-1: Import necessary packages for augmentation**

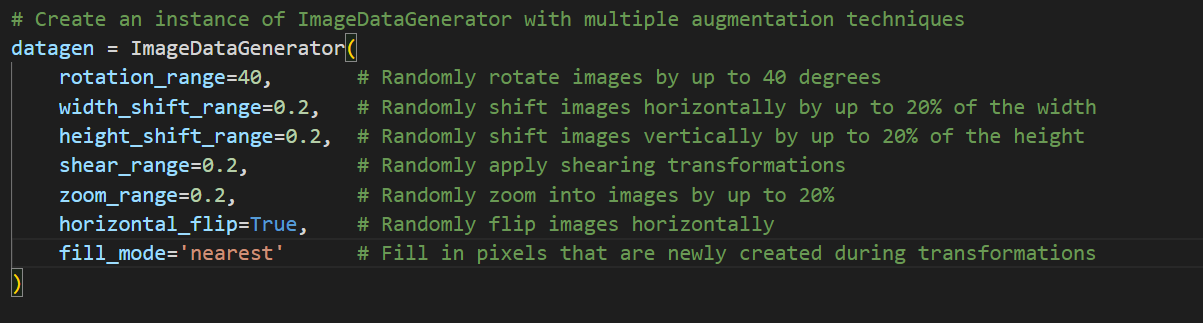
****

**Step-2:** **Define the path to your image**

****

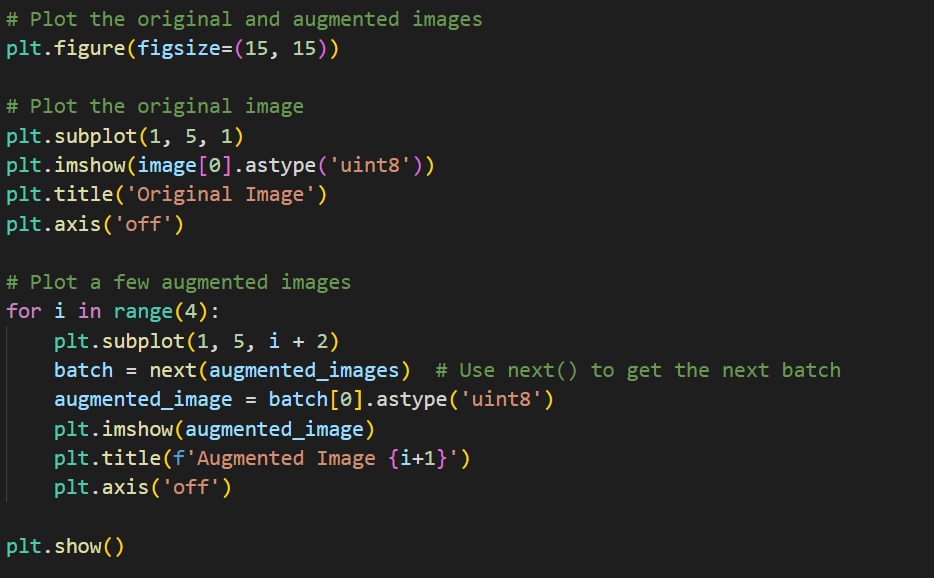
****

**Step-3:** **Create an instance of ImageDataGenerator with multiple augmentation techniques**

****

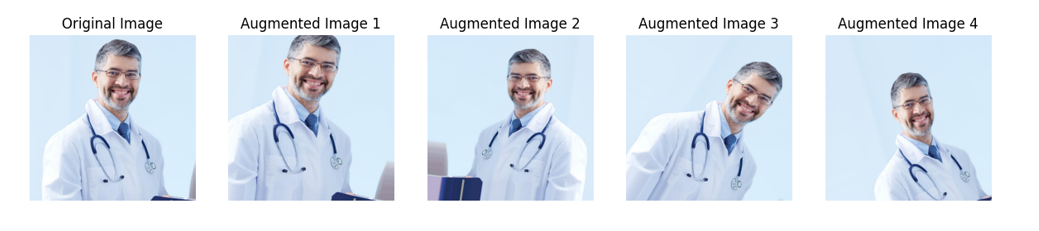
****

**Step-4:plot for image**

****

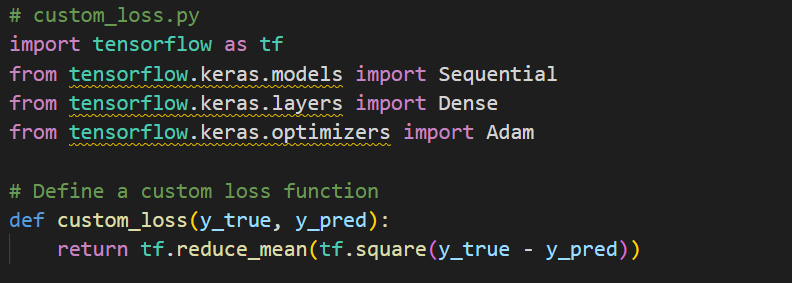
**Output:**

* Augmented images demonstrating various augmentation techniques.

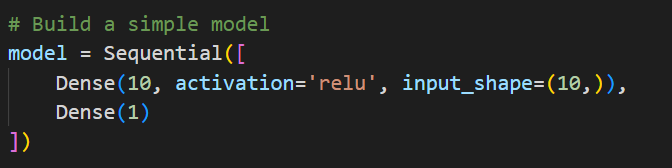
****

**3.** **Custom Loss Function**

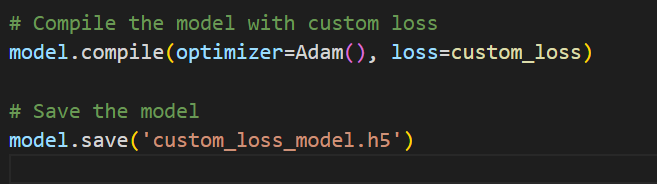
**Step-1: Define the custom loss function**

****

**Step-2: Build a simple model using the custom loss function**

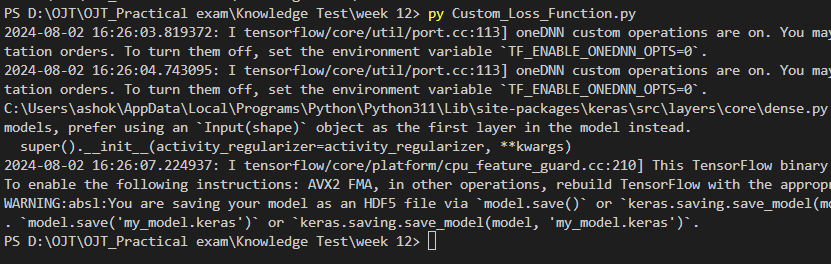
****

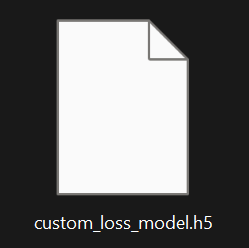
**Step-3: saved the model**

****

**Output:**

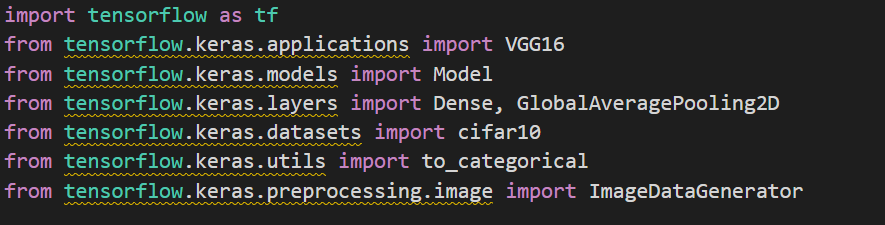
* Simple model using custom loss function with Mean Absolute Error (MAE) as a metric.

****

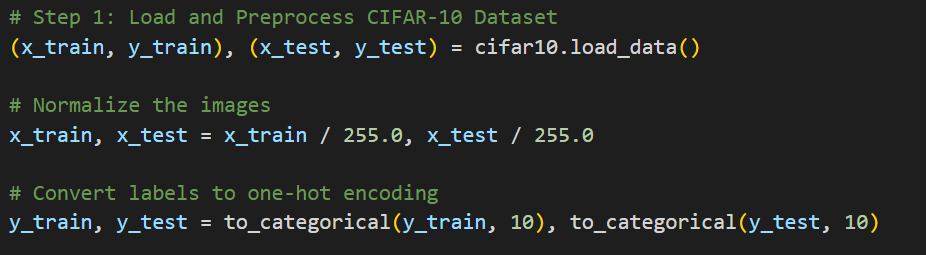
****

**4.** **Transfer Learning:**

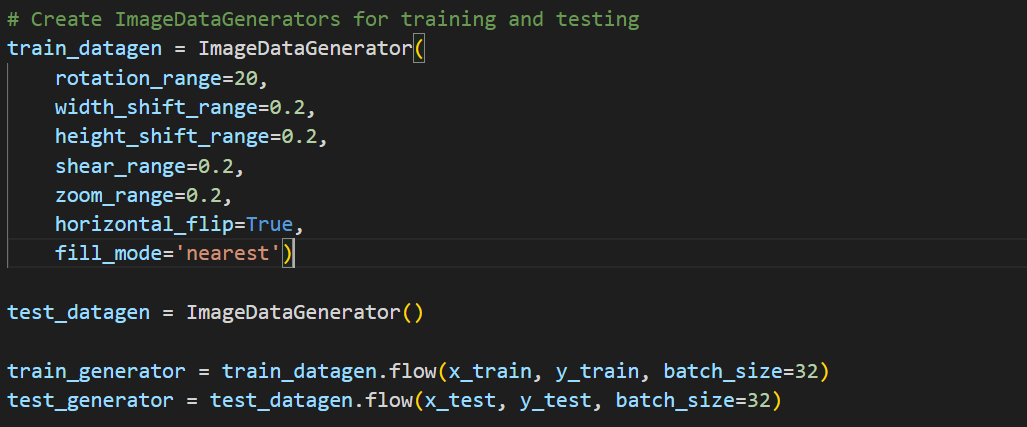
**Step-1: Imported nececary libraries**

****

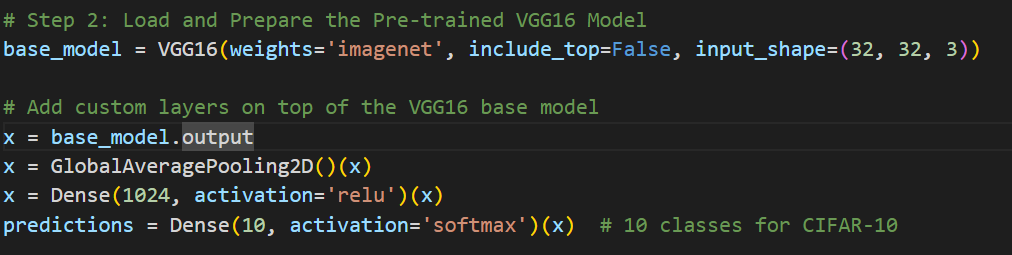
**Step-2: Load and Preprocess CIFAR-10 Dataset**

****

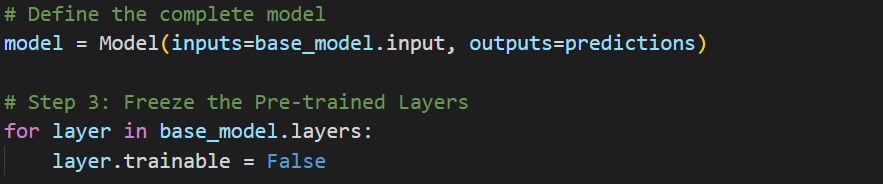
**Step-3: Create ImageDataGenerators for training and testing**

****

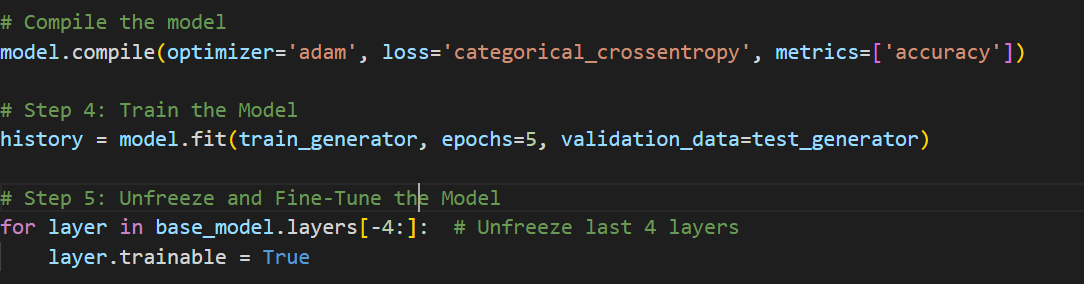
**Step-4: Load and Prepare the Pre-trained VGG16 Model**

****

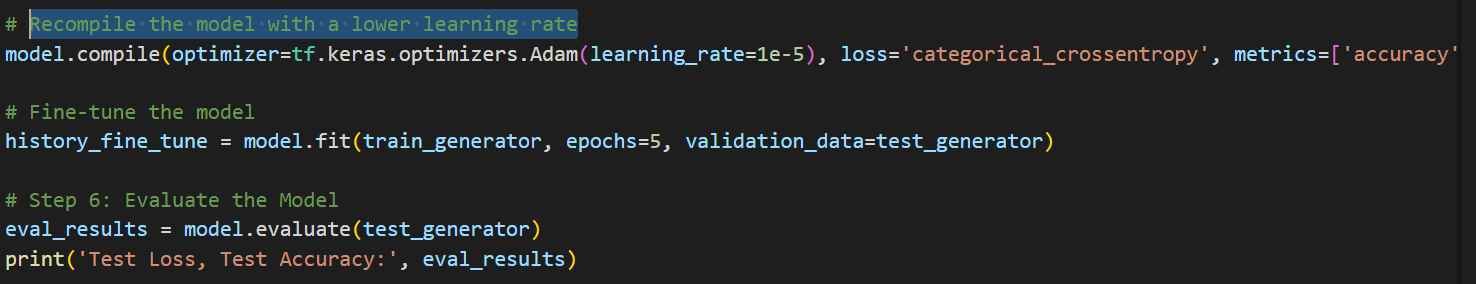
**Step-5: Define the complete model**

****

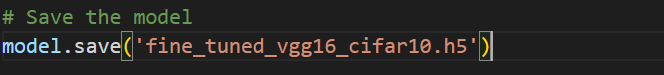
**Step-6: Compile the model**

****

**Step-6: Recompile the model with a lower learning rate**

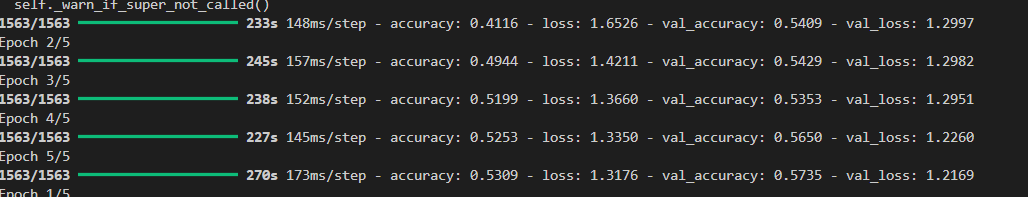


**Step-7: Save the model**



**Outputs:**

* Fine-tuned pre-trained model for a new dataset with training and validation accuracy.

****

****