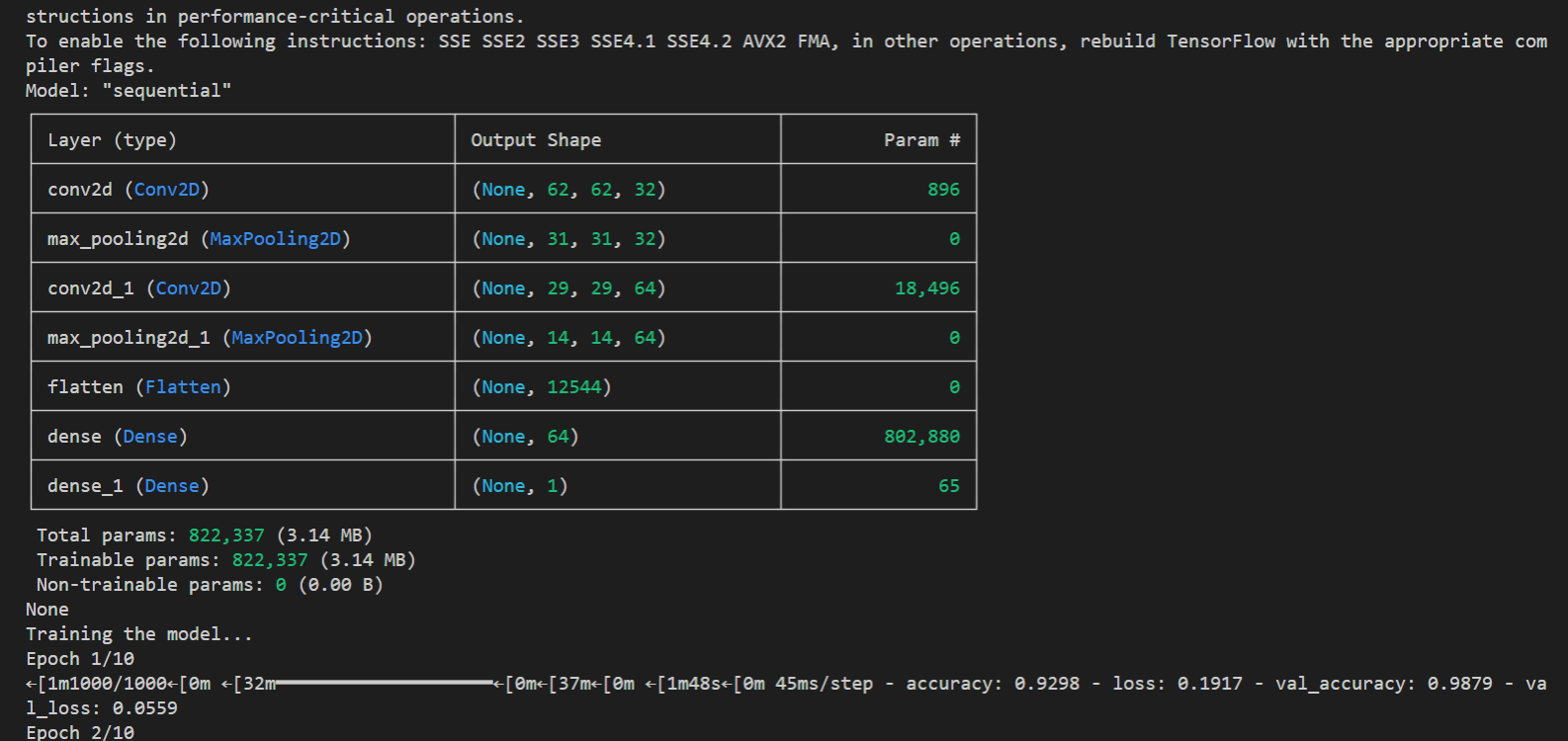
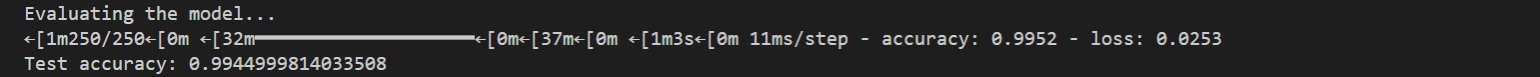
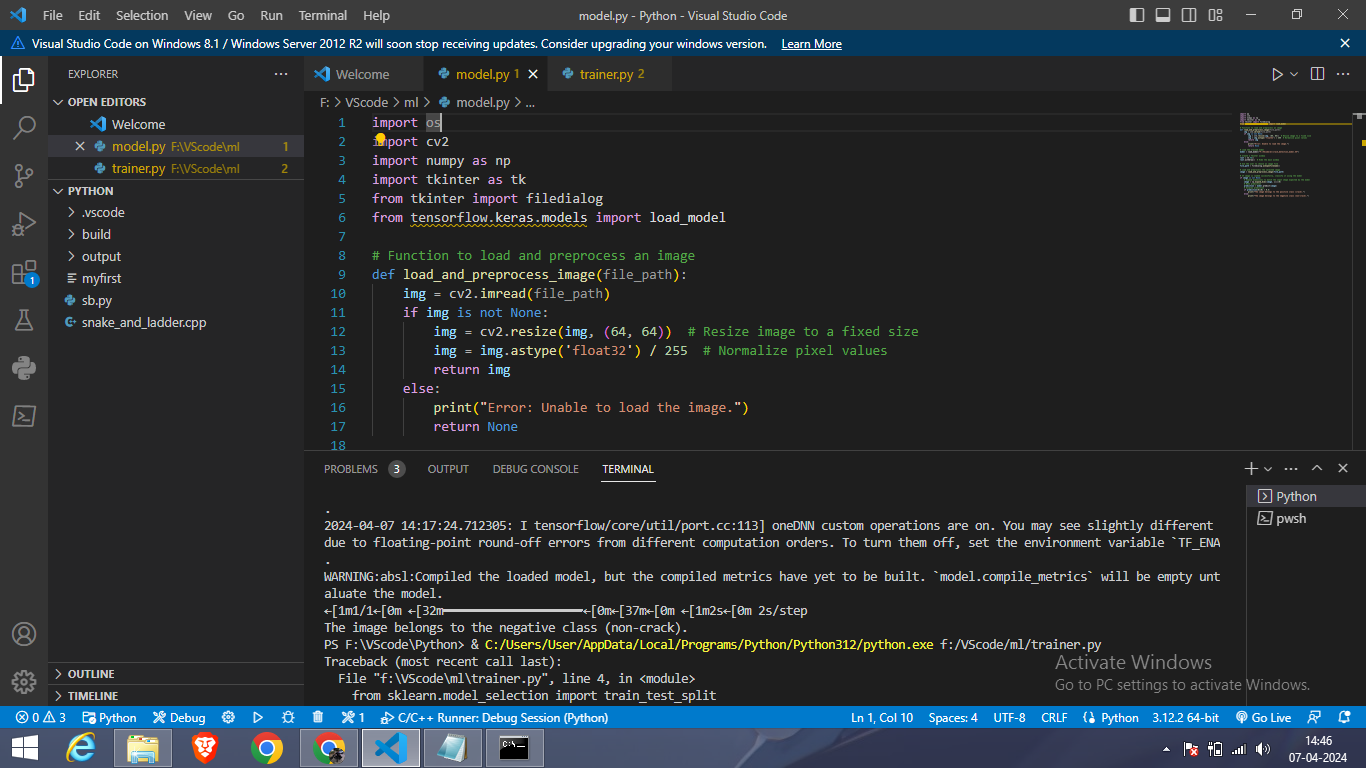
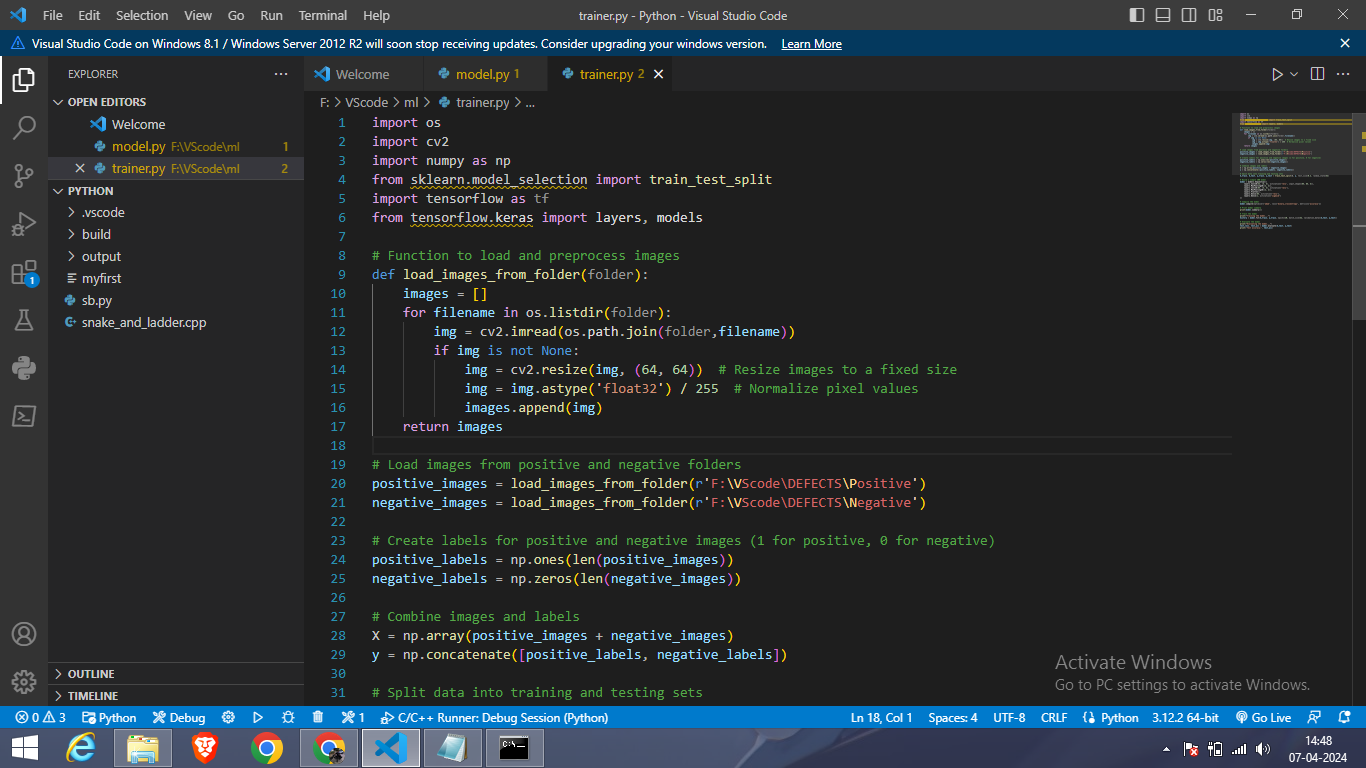
* As there were less dataset for industrial equipment so I take concrete mall defects photos.
* As a dataset I taken two folders one Positive (Defective) and second Negative (Non-Defective).
* Now I have made two codes one is trainer and second is model. Trainer trains the dataset and model give output.

Trainer: 

This code is for training a simple Convolutional Neural Network (CNN) model using TensorFlow and Keras. It loads images from positive and negative folders, preprocesses them, creates labels, splits the data into training and testing sets, builds a CNN model, compiles it, trains the model, and evaluates its performance.

Summary:

The provided code implements a CNN-based image classification model using TensorFlow and Keras. It loads images from separate folders for positive and negative classes, preprocesses them by resizing and normalizing, and creates corresponding labels. The data is split into training and testing sets using a 80-20 split. The CNN model architecture consists of convolutional and pooling layers followed by dense layers for classification. The model is compiled with the Adam optimizer and binary crossentropy loss function. It is then trained on the training data for 10 epochs with a batch size of 32. Finally, the model's performance is evaluated on the test data, providing insights into its accuracy.

Model:

This code is for loading a trained model that detects cracks in images using TensorFlow/Keras and OpenCV. It includes functions to load and preprocess an image, choose an image file using a Tkinter window, classify the image using the loaded model, and display the prediction.

Summary:

The provided code demonstrates the usage of a trained crack detection model implemented using TensorFlow/Keras and OpenCV. It includes functions to load and preprocess an image, load the pre-trained model, and classify a selected image file using the model. The Tkinter library is utilized for creating a simple GUI to choose an image file. After preprocessing the image, the model predicts whether the image contains a crack or not based on a binary classification. The result of the prediction is displayed to the user.



