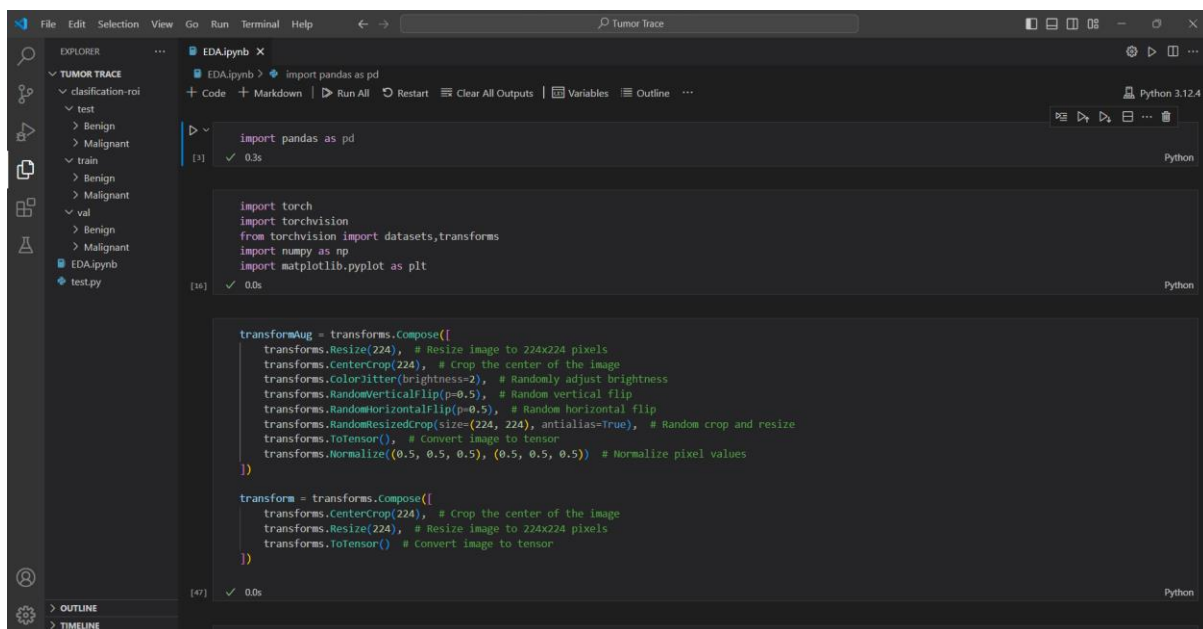


Project: TumorTrace: MRI-Based AI for Breast Cancer Detection

Name: Jaya Madhav B

Email: madhavjaya4@gmail.com

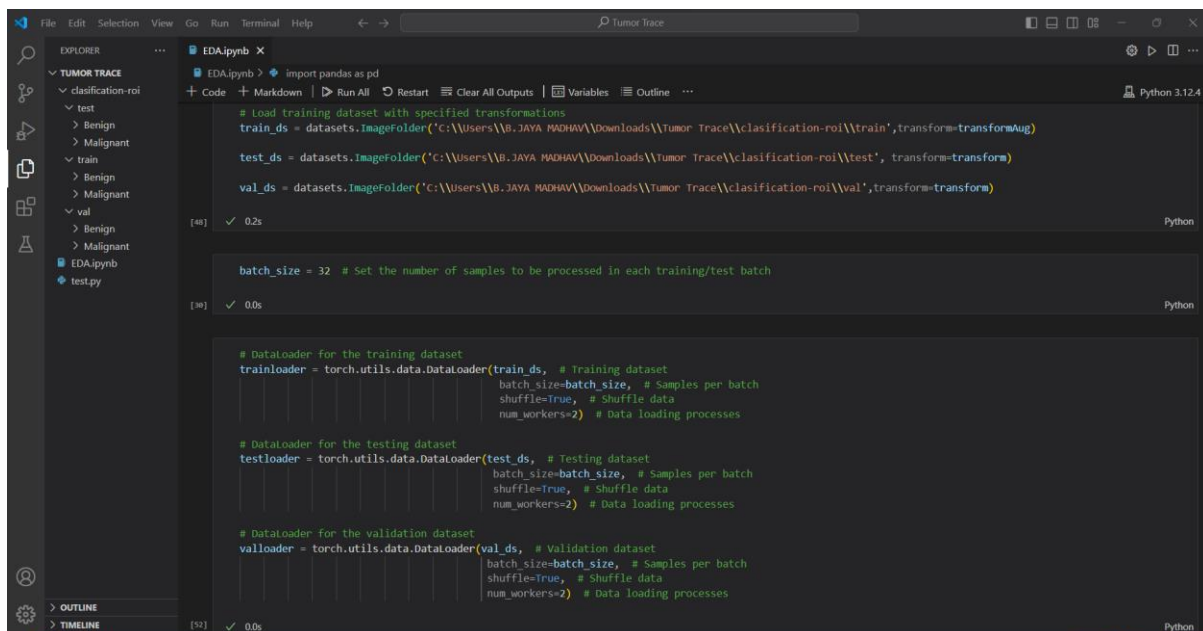


```
import pandas as pd

import torch
import torchvision
from torchvision import datasets, transforms
import numpy as np
import matplotlib.pyplot as plt

transformAug = transforms.Compose([
    transforms.Resize(224), # Resize image to 224x224 pixels
    transforms.CenterCrop(224), # Crop the center of the image
    transforms.ColorJitter(brightness=2), # Randomly adjust brightness
    transforms.RandomVerticalFlip(p=0.5), # Random vertical flip
    transforms.RandomHorizontalFlip(p=0.5), # Random horizontal flip
    transforms.RandomResizedCrop(size=(224, 224), antialias=True), # Random crop and resize
    transforms.ToTensor(), # Convert image to tensor
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)) # Normalize pixel values
])

transform = transforms.Compose([
    transforms.CenterCrop(224), # Crop the center of the image
    transforms.Resize(224), # Resize image to 224x224 pixels
    transforms.ToTensor() # Convert image to tensor
])
```



```
# Load training dataset with specified transformations
train_ds = datasets.ImageFolder('C:\\Users\\B.JAYA MADHAV\\Downloads\\Tumor Trace\\classification-roi\\train', transform=transformAug)

test_ds = datasets.ImageFolder('C:\\Users\\B.JAYA MADHAV\\Downloads\\Tumor Trace\\classification-roi\\test', transform=transform)

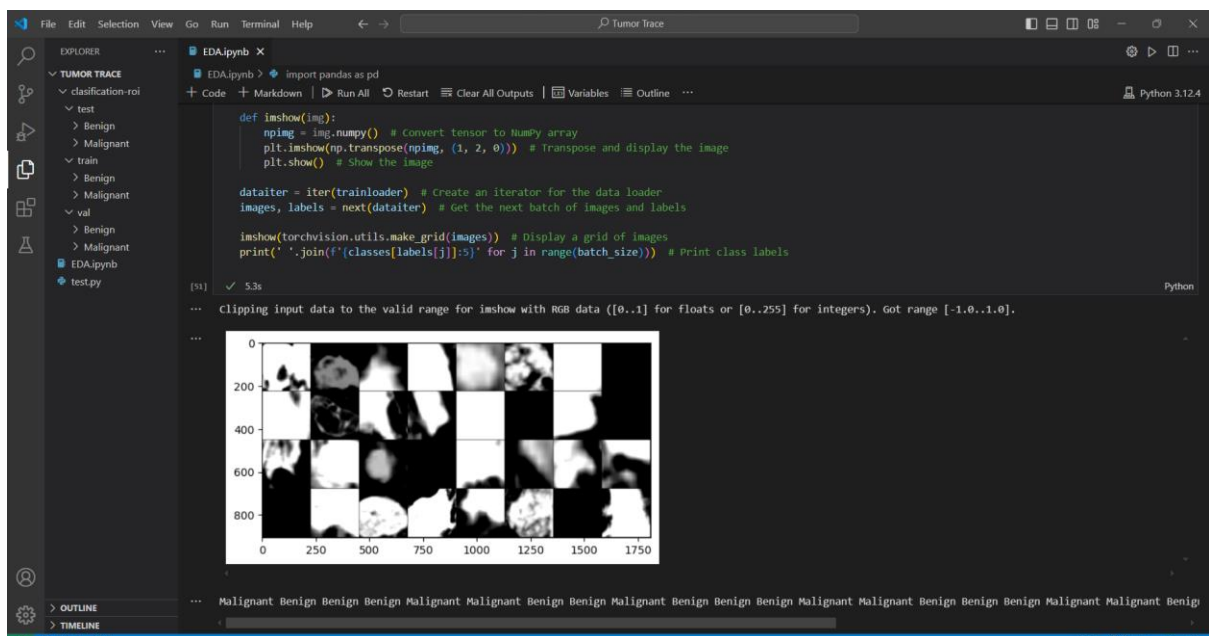
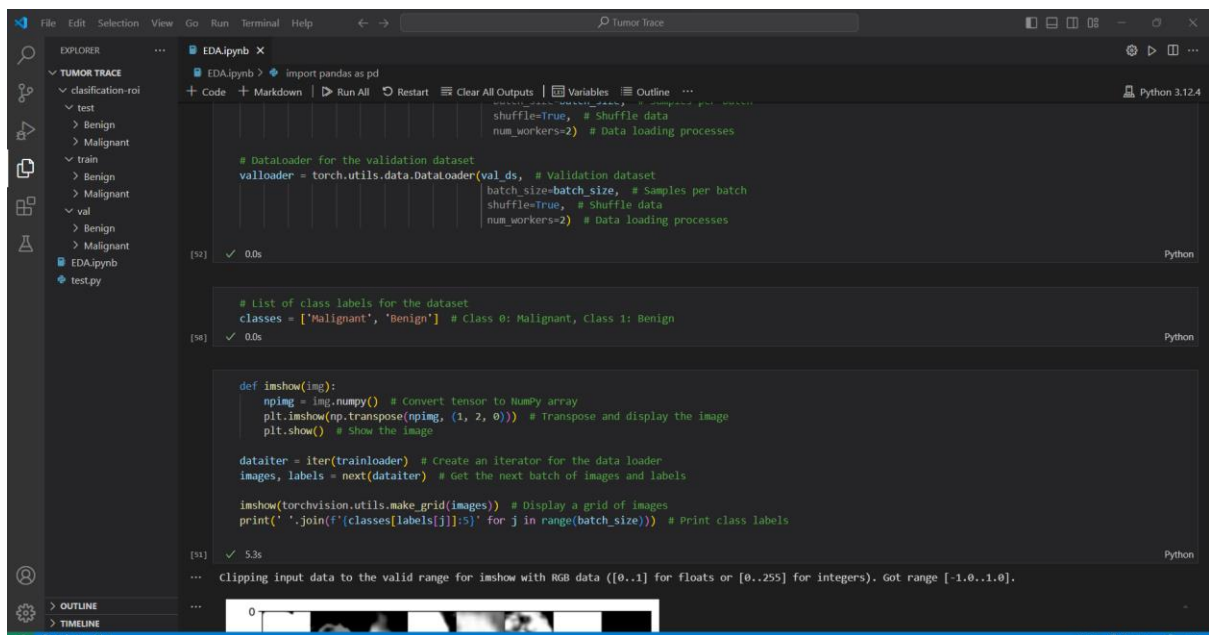
val_ds = datasets.ImageFolder('C:\\Users\\B.JAYA MADHAV\\Downloads\\Tumor Trace\\classification-roi\\val', transform=transform)

batch_size = 32 # Set the number of samples to be processed in each training/test batch

# DataLoader for the training dataset
trainloader = torch.utils.data.DataLoader(train_ds, # Training dataset
                                          batch_size=batch_size, # Samples per batch
                                          shuffle=True, # Shuffle data
                                          num_workers=2) # Data loading processes

# DataLoader for the testing dataset
testloader = torch.utils.data.DataLoader(test_ds, # Testing dataset
                                          batch_size=batch_size, # Samples per batch
                                          shuffle=True, # Shuffle data
                                          num_workers=2) # Data loading processes

# DataLoader for the validation dataset
valloader = torch.utils.data.DataLoader(val_ds, # Validation dataset
                                         batch_size=batch_size, # Samples per batch
                                         shuffle=True, # Shuffle data
                                         num_workers=2) # Data loading processes
```



The screenshot shows a Jupyter Notebook titled 'EDA.ipynb' in a VS Code editor. The Explorer sidebar on the left shows a project structure for 'TUMOR TRACE' with folders for 'classification-roi', 'train', 'val', and 'test', each containing 'Benign' and 'Malignant' subfolders. The Notebook has three visible code cells, each with a title bar 'EDA.ipynb X' and a toolbar with options like 'Code', 'Markdown', 'Run All', 'Restart', 'Clear All Outputs', 'Variables', and 'Outline'. The first cell, labeled [54], counts class labels in the training dataset and outputs {0: 5559, 1: 14875}. The second cell, labeled [56], counts class labels in the test dataset and outputs {0: 1938, 1: 4913}. The third cell, labeled [57], counts class labels in the validation dataset and outputs {0: 408, 1: 1581}. Each cell contains the following Python code:

```
from collections import Counter

# Count the occurrences of each class label in the training dataset
class_counts = dict(Counter(sample[i] for sample in train_ds.imgs)) # sample[i] retrieves the label for each image
print(class_counts) # Print the dictionary of class counts
```

The output for the first cell is {0: 5559, 1: 14875}. The output for the second cell is {0: 1938, 1: 4913}. The output for the third cell is {0: 408, 1: 1581}.