Question 3 Documentation

Two jupyter notebooks were created for data augmentation, one for using Tensorflow’s ImageDataGenerator class and the other for using Pytorch’s transforms module.

In the Tensorflow jupyter notebook, the necessary libraries including tensorflow, Numpy, and ImageDataGenerator were imported. The paths for the input images and the temporary folder to save the augmented images to were specified. The program then selects 5 images randomly from the COTS dataset.

Then the ImageDataGenerator was configured with parameters specifying the range and type of transformations to apply to the images. Each selected image was loaded into memory, converted to a NumPy array, and reshaped to match the input format required by the ImageDataGenerator class.

The data augmentation process was applied using a loop. In the loop, 3 augmented versions of each image were generated by applying random transformations specified in the ImageDataGenerator. The program then saves the augmented images to the temporary folder.

In the Pytorch jupyter notebook, the process is very similar. The libraries were imported, this time including PyTorch, and transforms from torchvision. Paths for the input images and the temporary folder were specified, and another 5 images were once again randomly selected, from the COTS dataset.

Then, transformations were defined using the transforms.Compose() function, specifying the range and type of transformations to apply to the images. The augmented images were then saved to the temporary folder.