Question 3 Documentation

PyTorch section

In this section we selected 5 random images from the COTS dataset. Then using the PyTorch transforms.functions module we implemented the following data augmentation techniques:

- flipping (horizontal and vertical)
- saturation (0.5 and 1.5)
- brightness (0.5 and 2.0)
- translation ((10, 20) and (30, 50))
- rotation (90 degrees and 180 degrees)
- contrast (0.5 and 2.0)

Finally we saved the augmented images to an output folder that could be zipped at a later stage for submission. The function used for saving the images was generated using chat-gpt4, aswell as the list comprehension used to get all the images as a list. Generative AI was also useful in defining the translation function.

The generated images of this section were used as a baseline for comparison of our images generated in q4.

TensorFlow section

In this section, we selected the same as the PyTorch section from the COTS dataset. Then, using the TensorFlow 'ImageDataGenerator' class, we implemented the following data augmentation techniques:

- Flipping (horizontal and vertical)
- Brightness ([0.5 and 1.0] and [1.0 and 2.0])
- Rotation ([-40, 40])

For each image we generated 5 augmented versions, however, the tensorflow 'ImageDatagenerator' class does the augmentations randomly. Thus in certain cases, the flipping doesnt occur. We also noted that sometimes the object of interest is rotated out of frame. The saved images were labelled according to the transformation applied and the range asscociated with the transformation.

Generative AI was extremely helpful in this section. It was used in order to understand how to use the 'ImageDataGenerator' class aswell as the loops used to apply the transformations and save the newly generated images.