

BME646/ECE695DL: Homework 2

Spring 2022

Due Date: Monday, Jan. 24, 2022 (11:59pm ET)

Turn in your solutions via BrightSpace.

1 Introduction

The focus of this homework is to familiarize you with the basics of image handling in PyTorch. You will open your image files as PIL (Python imaging Library) objects and subsequently process the images with a combination of numpy and PyTorch code.

2 Goals

- Learning to use the PIL Image class for opening image files and for accessing their pixels.
- Learning the differences between the PIL, numpy, and tensor based representations of an image.
- Becoming familiar with the pixel data scaling and normalization functionality in Torchvision.

Refer to the class slides [here](#) for the background you need in order to do this homework.

3 Tasks

1. Use your cell phone to take a couple of photos (you can use selfies if you wish) from two different viewpoints and store them in your computer as JPG or PNG files.
2. Write a Python image manipulation script in which you first convert the images into neural-network-compatible tensors using the functionality provided by Torchvision. If you wish, you can first open the

image files with `Image.open()` of the PIL's `Image` class and then use Torchvision's functionality to convert the PIL objects into tensors with appropriate data scaling and normalization. You might find [this](#) useful in accomplishing this task.

3. Calculate the histograms for the R, G, and B channels of the images in their tensor representation. For each image, display the three histograms separately but in a single composite display using the `matplotlib.pyplot` library. You may also want to include the original image in the composite display for that image. You may need to use [subplots](#) to come up with the composite display.
4. Calculate the distance between the histograms on a per-channel basis.
5. Next select an affine [transformation](#) (using the functions from `Torchvision.Transforms`) and apply it to the two images and see what effect that has on the histograms and their distances.
6. Finally, select a perspective [transformation](#) (again using the functions from `Torchvision.Transform`) and apply it to the two images and see what effect that has on the histograms and their distances.

4 Submission Instructions

- Make sure to submit your code in Python 3.x and not Python 2.x.
- Compress your Python source code and pdf report(see the [submission template](#)) into a zip file, naming it as your lastname.firstname.zip and upload it onto the assignment link on BrightSpace.
- **Your code must be your own work.** We will use your source code for plagiarism detection and verification of performance. Submission of both your source code and the report (in pdf) is mandatory to receive a grade.
- You can resubmit a homework assignment as many times as you want up to the deadline. Each submission will overwrite any previous submission.