Lab 08 Reflective Journal

The objective of the lab is to build a neural network classifier that evaluates data input to distinguish visual images of a muffing or a chihuahua.

Introduction:

The problem is there is such visual similarities between a muffin and chihuahua images, and we want to determine whether the model in use will effectively evaluate the data inputs and accurately identify the chihuahua.

The matplotlib library was utilized in the lab which produce graphics and plot images.

Data preparation was minimal, as distinct data folders were available for both objects respectively and contained a large number of images for both objects.

PyTorch’s baseclass was used to build a basic neural network, as PyTorch offers a dynamic computational graph created by defining some transforms to convert images to tensors.

The loss function or error function compares model predictions to true labels which is important in training a neural network.

I used the SGD optimizer with a learning rate of 0.1, though I also used the Adam optimizer after reviewing the first set of results which decreased the model’s accuracy rate. In the first iteration, I used 3 epochs and increased the number to 5, once the accuracy rate decreased to determine what impact the change made. Because of the lower accuracy results from the changes made, I returned to the original values, but the results were the same. The lesson learned was to only make incremental changes to evaluate a single change versus multiple changes at once to help isolate which feature impacted the results.

The difference between image classification and object detection is that the former focuses on categorizing the entire image. In simpler terms, it answers the question: "What's in the image?" There can be only one predicted class assigned to the whole image. The latter aims to identify and localize objects within an image. It goes beyond classification by pinpointing the location (usually with bounding boxes) of multiple objects and assigning class labels to each one.

The model function as designed and the performance, initially, had a high accuracy rate, however, the improvements would exist by modifying different features like the learning rate and optimizer, though as mentioned above, implemented incrementally.