

## L05 Chihuahua or Muffin CNN Reflective Journal

This lab is about building a convolution neural network that can be trained to tell the difference between pictures of chihuahuas and muffins. This lab is slightly different from the case using more general purpose neural networks because CNNs are specialized for image classification. This lab imports PyTorch so we can train the model with deep learning using photos of the chihuahuas and muffins as test data. A smaller set of photos is used to validate the data.

I was excited to learn about building a convolution neural network and this lab provides that experience. This lab builds on concepts that work with breaking down image pixels into matrices so they can be analyzed by the model. This is at the core of image classification because the pixel data can be grouped into patterns that the model learns from. In this case the algorithm is looking at chihuahuas and muffins. There are similarities in the patterns of these two objects in some instances. The orientation of the blueberries on the muffins tends to look like the eyes and nose of similarly-colored chihuahuas and the model must use the training data to distinguish between the two. This lab adds more epochs to the process; there were three in the NN model but this CNN model has 20 epochs.

I encountered a major issue with loading the transforms for the training data. After checking for syntax errors, I realized that I didn't clone the repository properly. Upon proper cloning, I re-ran the code and everything executed without issue.

Since this model trained in 20 epochs, there was more of a risk of overfitting. We don't want our model to perform so well on the training data that it effectively memorized the images. If the model simply memorizes the training data, there's a good chance it will perform poorly on the test data, because it didn't actually learn well. The training would be poor in this case.

In the end, this CNN model performed better than the NN model. There was only one incorrect image where as five images were incorrect in the NN model. The additional epochs provided more

training than the mere three epochs for the NN model. I adjusted the learning rate from 0.001 to 0.01 to save runtime but also to preserve training quality. I'd like to note that 10 epochs at a learning rate of 0.001 has a 100% success rate on the test data.

Image classification has many applications, and I believe it's important for human facial recognition. Modern security measures at major venues like sports arenas now use facial recognition to identify legitimate ticket holders. In my personal experience with this technology, I added my face to the registry and only a minute later I walked through the gate and a human greeter said my name and welcomed me in to an Astros game. Cameras at the entrance tracked my face in real time as I moved toward the entrance and I never had to stop walking. The speed of this process was a surprise and shows the power of what neural networks are capable of in a short amount of time.

This computer vision lab was thorough in explaining what goes on between the input and output of deep learning. Working with the PyTorch framework was great hands on experience, and using a somewhat popular case of chihuahua or muffin made this more relatable. This helped demystify tensors because that concept is new to me. It's great learning how to leverage python to do the heavy lifting on these technical math concepts. I was able to compare this specialized method to the more general purpose neural network, and the convolution methods allows for more fine tuning of the training quality.