Joshua Jowers- Undergraduate Research

Using CNNs for Autonomous Drone Navigation

2/23/2023

The original goal was to continue using the code that Wint wrote that used TensorFlow. However, we decided to used Pytorch instead, so this stage of the process is converting what Wint had in TensorFlow to work in Pytorch. It is a convolutional neural network using the Xception framework that will be used to determine locations of images when trained on other sets of images from the same geographic area. The current code is divided into four files: RealGeographicCustomDataset.py, CroppingMethods.py, RealXceptionTesting.py, and XceptionModel.py. Below are descriptions of each of these files.

RealGeographicCustomDataset.py: This file contains a custom dataset class that I created that takes in a csv file and a directory of images. It converts the csv file into a pandas data frame that is uses to store the image information. Using the MinMaxScaler, the initialization function converts the input values for the x and y coordinates in meters to scaled values between 0 and 1. Within the get item method, each image is cropped when it is accessed, and the x and y coordinates associated with it are adjusted. Ultimately, the get item method of this class returns a randomly cropped image as a Pytorch tensor along with the scaled and adjusted x and y labels.

CroppingMethods.py: This file contains two a single\_image\_crop method that takes in an image tensor along with the desired output size. This method outputs a randomly cropped image along with the amount that the x and y of the top left of the image were moved in order to get the cropped image. The second method calls this method and adjusts the x and y labels on the image based on where the image was cropped. This method returns the adjusted x and y labels with the cropped image. This method is called in the get item method in the custom dataset class to crop the image.

XceptionModel.py: This file contains the Xception class. Xception is a 71-layer deep CNN which is used as the basis for this model. The code included in this file defines all of the layers that are in the model that is being trained on the images.

RealXceptionTesting.py: This file is the main running file for the code. It reads in the data from a csv file and the image folder. Then, it randomly splits the images into training, validation, and testing datasets. These datasets are then used to define data loaders which are used to put run the data through the CNN. Then, a few parameters are specified including batch size and learning rate. These numbers are currently arbitrary and just for testing to make sure that the code runs how it is supposed to. Then, there is a call to the Xception class to define a model object. Then, there are loops through whatever specified number of epochs to run the training and validation of the model.