**Progress Report**

Yehia Fahmy

May 22, 2020

**Problem:**

Classifying pictures of non-ceramic insulators as either Hydrophobic or Hydrophilic using a deep-learning implementation.

**My Approach:**

I wrote two python files to solve this problem. The first being a program that processes the images and creates the training data set. The second program then builds and trains the neural network. We also use 10% of the data to test the data.

I also wrote a .txt file that will hold the required packages and their install commands.

**Requirements.txt:**

This folder holds the packages required for the two programs to run, note that you will need Anaconda 3 installed on your device to be able to create and manipulate conda environments. The following are the packages needed.

* Pip to install the other packages
* Keras for building the model
* NumPy for array operations
* TensorFlow for building and training the model
* Matplotlib for visualizing the data
* Cv2 for reading and formatting the data

**Processing.py:**

The processing program has the following functions

* Load data
* Crop data
* Flip data
* Rotate data
* Shuffle data
* Split data
* Save data

Load data will iterate through all the pictures we have and add them to a list with their classification as either a 0 or 1. We start off with 317 training examples.

Crop data will go through this list and split each image (480x480) into 9 images (160x160) and add them with their classifications to another list. We now have 2853 training examples.

Flip data will add each image to a new list as well as the image flipped vertically, horizontally, and flipped both ways. We now have 11412 training examples.

Rotate data will add the data to a list rotated in all 4 orientations. We now have 45648 training examples.

Shuffle data uses a built-in algorithm to shuffle the training data.

Split data will create a list of X values (images as NumPy arrays) and a list of y values (classification as either 1 or 0)

Save data saves both the X and y lists to the directory using the pickle format for easy access by the next program.

**Training.py**

In the training file there are several functions to keep the process organized below is a list of them.

* Load data
* Reshape data
* Build network
* Train model

Load data will bring in the data that was saved in the previous file using the pickle format. We will store the features as an array of images called X. We will store the labels as an array of 1s and 0s called y.

Reshape data will ensure that both the lists we pass are converted into NumPy arrays. We then will normalize X so that all the pixel values are between 0 and 1 (rather than 0 and 255).

Build network will take our number of layers and number of nodes (global variables) and will build a sequential model. We will use the rectified linear function as the activation function. We will then add layers of the appropriate size until the model is built. Then we compile the model.

Train model will attempt to fit the model previously built to the data we provide it.

We will print the time taken at the end of the program.

**Results:**

For these programs I only classify the pictures as either hydrophobic or hydrophilic, so the program is not very specific in its classification (i.e. we expect the program to be very accurate). I found that the model that works the best for this problem is a neural network with two layers and 64 nodes in each layer. I also train the data through three epochs with a batch size of 32. The validation accuracy of this model is 99.778%.

**Running the Programs:**

1. Create a new virtual environment or using an existing environment install the required packages
2. Activate the virtual environment where you have installed the packages
3. Navigate to the directory with the python files and using the command line run the Processing.py file to process and save the data
4. Use the command line to run the Training.py file to train the model