**Project Description:**

"Typer", A neural network that turns your handwritten documents into typed files.

**Competitive Analysis:**

Most other tools, like the Notes Plus app, the GoodNotes app, and Google handwriting take in "hand written" text from a stylus on a tablet or a phone. My program, however, will be able to take in handwriting from a photograph and convert that into typed text. Instead of being designed to be used while taking notes or writing things down like other apps, my program is designed to be used when the user is not able to access or use the electronics that they would like to to take their notes on and are forced to take notes on paper. Once the user once again has access to their electronics, they won't have to rewrite or type their notes to get them into an easily editable format to use on their computers.

In addition to the main feature of converting handwriting to text, my program will also be able to show some interesting training and error data from its neural network and will be able to clean up and create more high contrast and legible images from images that it is given.

**Structural Plan:**

My final project is predicted to be 5 python files, 1 zip file, and 2 pickle files in a folder.

The python files will be 1 for the neural network to recognize the letters, 1 for making the training data for the neural network manageable, 1 for blob detection, 1 for making each "blob" into something that the neural network can read, and 1 final runner file for the user to run to get their resulting text.

The amount of data that I used to train the neural network was overwhelmingly large and not really necessary after it had been trained, so it will be stored as a zip file. In addition, converting the training data into usable training data and training the neural network both take a long time (1.5 hours and hours respectively) that is not useful to repeat multiple times. So, I will be saving both the usable data and the pre-trained neural network in pickle files.

**Algorithmic Plan:**

The two tricky parts of my program are creating a neural network and writing a program for blob detection. They were two areas that I didn't have any experience in and were difficult to understand and code.

The main algorithmic idea of the neural network is this: The program takes in some data and splits it into two sets, the training dataset (80%) and the testing dataset (20%). Then, it assigns each value of input (from the testing dataset) some weight and bias (kind of like m and b in y=mx+b where weight is m and bias is b). Then, it passes this info to another set of neurons that does the same thing. This happens once more and the final product is outputted. This output is a prediction. Then, loss (error) is calculated using the actual "answers" to the data from the training dataset for every layer of the network to see where the weights and biases were. Then, this loss is used to update the weights and biases in the network. Then, this process is repeated. After repeating several times, the weights and biases are fairly close to what they should be for the predictions to come relatively correctly. Then, to test this hypothesis, the testing data is sent through the network to get the testing accuracy. This number tells you how good your network is at predicting the answer when given some data.

The blob detection is something that I'm still working on, so I'm not really sure of the complete algorithmic plan for that. I believe that I will first cut up my image into small pieces and then try to detect edges on each of those pieces. If there are a lot of edges in one area, it's probably a letter. Then, once the letter is found, I will make the letter into an arrayed 1s and 0s to indicate where the writing is and where the blank page is. Then, I will make all the letters into a list and send them through my neural network. Whatever it predicts will be made into a text file that the user can now use.

**Timeline Plan:**

The neural network is trained saved. The image data is also managed and saved. I plan to do blob detection in the next week and finish up the UI in the few days after that.

**Version Control Plan:**

Versions of my project are saved on my Github (https://github.com/ShamikaD). In addition, I have been backing up my computer to a physical hard drive.

A screenshot of a social media post

Description automatically generated

**Module List:** numpy and pandas