Bridget Mansour

Roland David Oswald

Hand-Written Optical Character Recognition

Objective:

The purpose of our project is to create an algorithm that converts handwritten text into typed text. This is applicable for recognition systems that encompass the concept of neural networks. For instance, our algorithm may provide students with the ability to have their notes typed out automatically rather than manually. Additionally, poor handwriting could become easier to identify through use of our program. Finally, our program should be able to receive an input image, effectively preprocess it, detect and retrieve text for segmentation, and output the corresponding printed text back to the user.

Related Work:

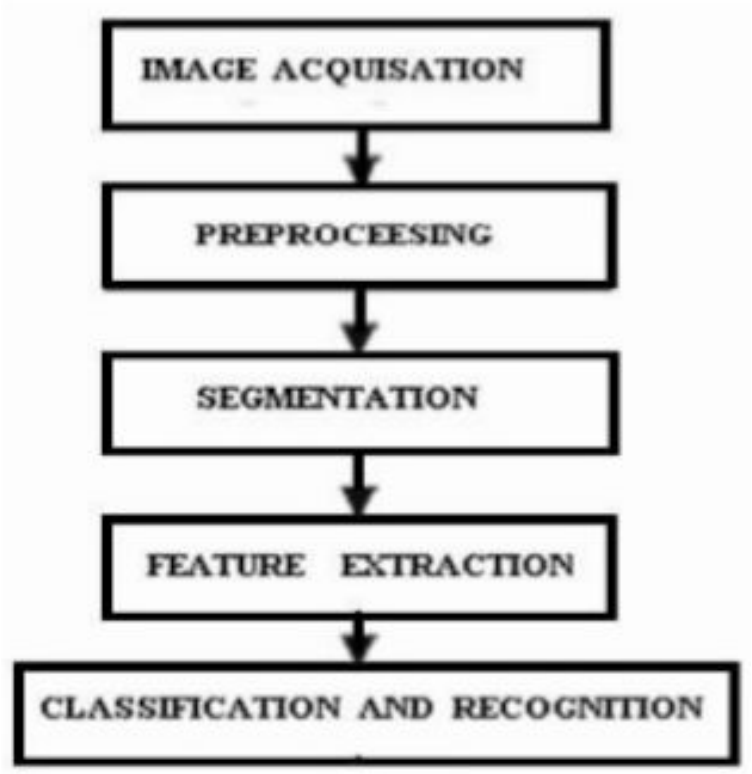
Several state of the art techniques have been used for hand-written optical character recognition in the past.

- Chirag Patel, Ripal Patel, & Palak Patel proposed using neural networks. Their paper aimed to recognize different models of neural networks that may be useful for recognizing text characters obtained from scanned images. Hidden layers, their corresponding sizes, and epochs have all been taken into account as parameters of this model.
- Ashut Aggarwal, Rajneesh Rani, & Renu Dhir proposed using gradient features. Their paper aimed to recognize the importance of feature extraction for any pattern recognition system due to its ease of use, logical simplicity, & high recognition rate. Using the sobel operator, gradients could be computed in which both the magnitude & direction of the maximum change of intensity in small neighborhood for each pixel is measured. This method had a 94% accuracy rating for detecting english characters.
- Kauleshwar Prasad, Devvrat C. Niggman, Ashmika Lakhotiya, & Deheeren Umre proposed using Matlab's neural network toolbox in a paper that stressed that there were numerous applications for recognition of handwritten text. It used Matlab's neural network toolbox and projected recognized characters onto different sized grids. This method consisted of image acquisition, noise filtering, preprocessing, segmentation, feature extraction (optional), and classification techniques. Edge detection & character extraction was critical in training their neural network to classify letters properly.

- Hanmandlu M., Murali Mohan K.R., & Kumar H.'s paper explored a fusion of both the existing ring-based method that was proposed in 1987 by W.I. Reber, and the newer sector based method. A fixed number of both concentric rings & sectors were considered. Pre-classification was obtained through extracted structural features, including endpoints & junction points.
- Dinesh Dileep proposed a geometry based feature extraction technique that was applicable to word recognition systems that used segmentation. His system used the contours of each character that could be found through the basic lines of the characters skeleton. Outputted feature vectors were then used to train a neural network based pattern recognition engine.
- For object detection, Sheng Wang reviewed gradient and edge based feature extraction techniques. His paper aimed to categorize recognition methods based off of the low level features that they used.

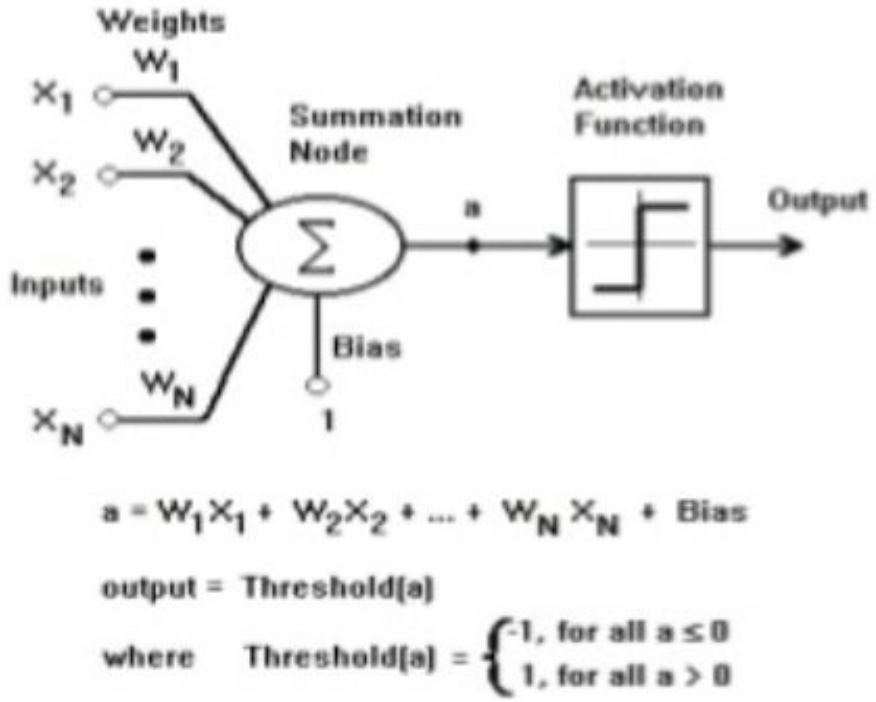
Technical approach:

Block diagram:



1. Pre-processing: This step is crucial for making an image segmentation-ready. Binarization, morphological operations, and resizing are examples of what this step may consist of.
2. Segmentation: In this step, the image's contents is separated into lines, words, and then characters. Labeling through the use of connected components is necessary to keep track of each object of the foreground order later on. Size normalization must also be done after the extraction of separate characters takes place.
3. Feature extraction: This step can be done using gradient features or geometry-based techniques. While not precisely necessary, this step is helpful for training neural networks.
4. Classification/recognition: In this step, neural networks are trained and used to classify and recognize various letters.

Typical neural network:



Technical approach:

- First the image is acquired. Since the image is usually taken with a camera, it needs to be converted from rgb to grayscale. Once the image is in grayscale, the image is resized to 750 x 600 pixels, then converted to binary images. The image needs to have white writing on black background, so the complement of the binary image is taken. bwareaopen was then used to remove unwanted small objects, then dilation is used to try to connect any letters that might have small disconnections.

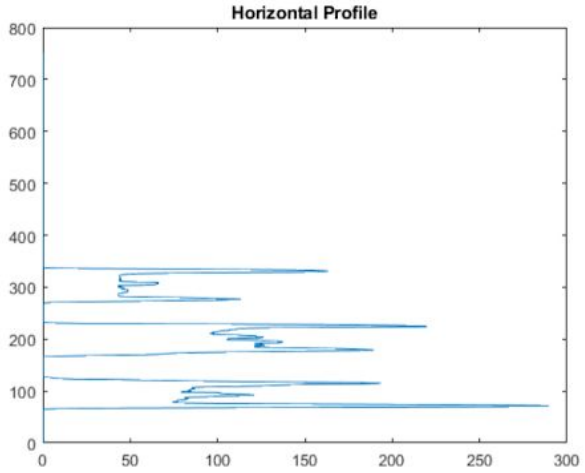
- To separate lines, the horizontal profile is used. In the horizontal profile, the valleys show where the spaces between lines are, and the peaks are where the lines of writing are. So the peaks are found using the MATLAB function, “findpeaks”. This function gives the peaks and their locations. Using the locations of the peaks, the image is cropped to just show the line of writing.

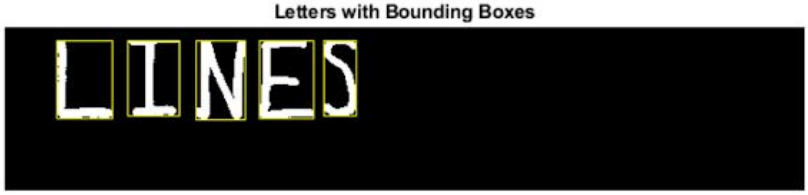
- Once we have just one line of writing, connected components are used to segment the letters to enter them in the neural network function. Using the regionprops MATLAB function bounding boxes are placed around each individual letter.

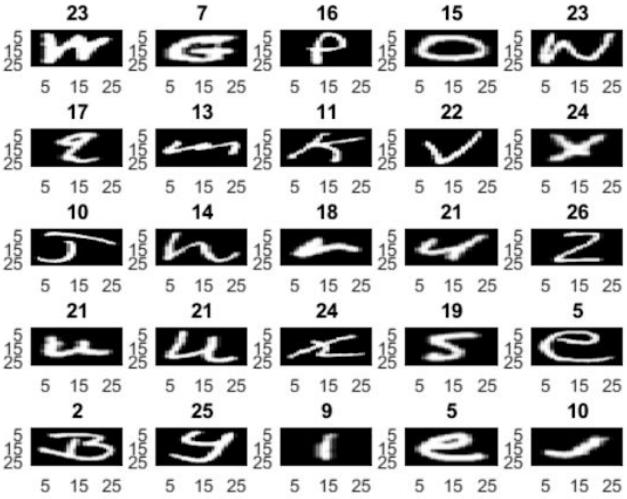
- To use the neural network function, the image is cropped to the size of the bounding boxes around the letters. Then the letters are resized to 28x28, and flattened to 784x1 vectors and then put in to the neural networks function. The function returns 26 percentages with the index of the highest percentage corresponds to the letter class of the letter entered into the function.

- The array containing the indexes of the letters are put through a function that returns the the corresponding letter and outputs that letter. Spaces are placed where the distance between bounding boxes the biggest are.

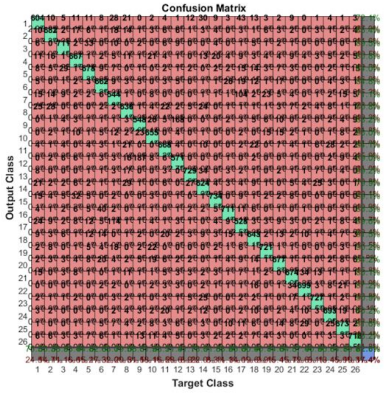
Data/Results:

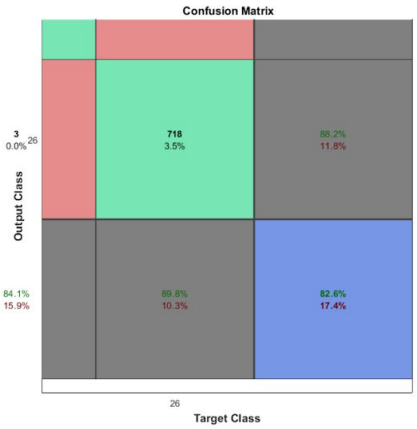


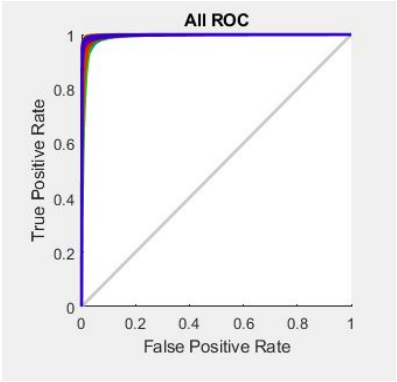


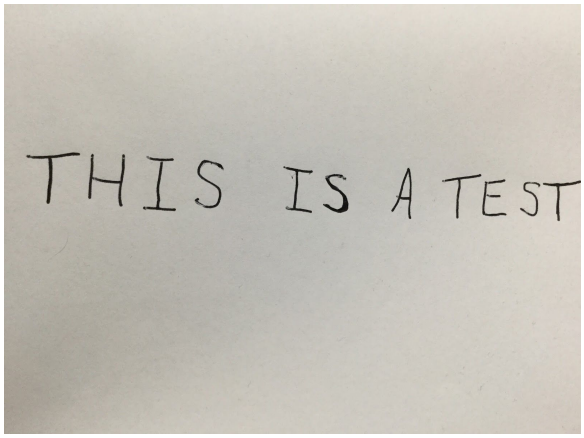


← **Dataset (NIST special database 19)**









→

```
Sent =

    'THIS IS A TEST'
```

Conclusion:

- Hand-written text can be to decipher, even with the use of extensive neural networks.
- Segmentation proved to be the most challenging aspect of our project.
- Though difficult, we were able to obtain results with roughly an 83% accuracy.

Future Work:

- Given more time, we would like to be able to account for lowercase text and digits, which would require a larger dataset.
- Going forward, cursive handwritings in which letters are not separate would be accounted for.
- Finally, punctuation and symbols would be necessary in order to fully convert an image into text.