Project Description:

This project aims to allow visually impaired individuals better access to printed and handwritten text.

The program takes in an image that could contain English language characters using a webcam and then allows a user to process it either as printed text or as handwritten text. Both modes obtain a string of characters from the image and allow the user to either use the computer’s speakers to say the text out loud or save the text as Braille in a Word Document.

Modules Used:

1. Open CV - Image Processing Module
2. PIL- Image Processing Module
3. Numpy- Companion to Open CV, used for handling image arrays
4. Pytesser – Python Wrapper for Google’s Open Source OCR Tesseract
5. Pyttsx – Module to output audio
6. Win 32 – To access Microsoft Office on Windows platform
7. Sys- to access system functions
8. Tkinter- designing UI

Project Algorithm:

Printed Text Recognition:

1. Obtain image from user
2. Resize and clean the image
3. Convert the numpy array to PIL ( pytesser handles images through PIL only)
4. Load the image into pytesser
5. Check and modify the output for non-ASCII or empty characters

Handwritten Text Recognition:

1. Obtain image from user
2. Resize, threshold and clean the image
3. Draw the contours on the image and find the bounding boxes for all objects in the image
4. Remove the boxes that are too large or too small to be letters based on algorithm
5. Segment every bounding box into 16 smaller segments
6. Find the percentage of black pixels in the segment
7. Compare that percentage to a given set of ideal values for each letter
8. Find the values that match the most.
9. Resolve ties based on more detailed count of similarity
10. Return string

Conversion to Braille:

1. Open a Word File and stream the text into it, by specifying a specific font called “Swell Braille” (downloaded externally)

User Interface

The UI is minimalist and mostly dependent on audio instructions which are repeated after a reasonable interval. A very sophisticated visual interface is unnecessary given that the target users are completely or partially visually impaired. However, each screen is a very different colour from the last to allow partially blind users to know that the mode has been switched. Also for ease of use, the user controls the input only through the arrow keys.

Background Research:

I found that handwritten text recognition is still a problem that needs a better solution for it to have a ~95% accuracy rate. Other applications that do a similar features for the blind are mostly limited to printed text.

Most data for the app and their user reviews have been taken from the App Store and this website: <http://www.applevis.com/apps/ios-apps-for-blind-and-vision-impaired>

1. KNFB iOS App: <http://in.reuters.com/article/apple-blind-app-idINKBN0HF08S20140920>

Pros: High accuracy, Great UI

Cons: No handwritten text recognition

1. SayText- iPhone app for blind users

Pros: Image Rotation

Cons: Need to account for bad lightning and accuracy, No handwritten text

1. VoiceCam for the blind

Pros: Controllable speed of audio output

Cons: UI has too many buttons, Not accurate, reads only a word at a time

1. Penultimate 4.0 by Evernote – Handwriting Recognition

Pros: Learning algorithm that improves with respect to input amounts

Cons: Text has to be input written in by user in the app, Can’t detect text from an image.

Future Plans:

1. Improve handwritten text accuracy by using more input data.
2. Reconstruct as a mobile app.
3. Allow Math symbols like square root and summation, then convert to Nemeth Braille.
4. Allow the features listed above (in Background Research)