Problem Background

As we move into a digital age, I find myself struggling to retain knowledge from within my work books when I write in lessons. I enjoy writing but to understand what I’ve written, I need to put in extra effort, as my handwriting is quite atrocious. One could argue that it would be easier for me to just take my laptop into lesson and use that to take notes, but there are several problems with this, mainly that typing on a keyboard could act as a distraction to other students and the teacher who is running the lesson. Also, what happens if I forget my laptop? I needed a way to scan my workbooks into my computer, so that I could access my notes from wherever I was, even when I wasn’t at my computer.

Initially, I tried just taking pictures of my scribbled notes and storing them on the cloud. This didn’t make life any easier though, as I didn’t bother to document the images properly, and it was even tougher to read my notes on a smaller screen. The problem didn’t lie in my ability to read my own notes, but the size of my camera, and how far I could zoom in to see individual pieces of text. Next I attempted to use a flatbed scanner to try and improve the quality of the images I was using. However, I ended up spending a large amount of my time scanning pages of workbooks, when I could’ve been studying from those workbooks. I could save an extremely large amount of time by automating the process by which the work was scanned and documented.

Initially, I looked into current technologies which could aid me in speeding up the time to scan a document to a mobile device. There do currently exist apps that perform quite effective document scanning, but they do not rip text from the image to convert to a .txt file, and any apps that might do this will only work effectively with typed text. The issue isn’t finding an effective document scanner, it is trying to build one that can convert hand-written text to a typed text format.

Once I finally figured out what the problem really was, I set about trying to figure out how I could scan an image into a typed text image, and my initial thought was to turn to AI and Machine Learning. Upon figuring out the main problem I was trying to solve, I had to come up with a way that I solve it with relative ease and elegance.

Prospective User(s)

I will be the main user of this application, as well as any other students who wish to utilise the system. I intend for this to be a solution to help students to streamline their workflow, so other students are also prospective users.

# Proposed Solution:

My solution to the problem above lies heavily within machine learning. It should use image recognition as a main solution, so that I can pass a simple picture into the program and it will return a text file of all the text in the image. The process should start by finding the piece of paper in the image that is meant to be scanned, with the program perfecting the image so that only the piece of paper that is being scanned can be seen, without any background noise. Next I should use OCR (Optical Character Recognition) within an AI which has been trained to recognise my handwriting, and as such can figure out what any piece of text is trying to say. The information from the OCR should then be processed into a text file, however there would not be a name given to it, as I may have to pre-set one when running the command. To get around this, I could check the top line for any writing that seems larger than the rest of the writing (i.e. a title), and if it is found, then that can be assumed to be the title of the text file.

Once the program has run, it will output the text file into a folder where it can then be used by anyone who wishes to create flashcards, or turn their ramblings into a book. Either way, it removes the necessary need to worry about forgetting where a workbook has been put down, because all the data remains in one place, and unless the data becomes corrupted, it should remain reasonably safe.

Acceptable Limitations

Initially, the system will have the following limitations:

* The system will only be able to read in text data, not images or graphs/tables or mathematical formulae (This could be considered as an extension regarding more complex uses of Image Recognition)
* The system will only specify between different handwriting types in terms of the data that is put into it, different users will not and should not need to specify that they are a different user. Combined data will make the network more efficient.
* The program will only be able to determine the contours of a single sheet of paper that is a quadrilateral (most pieces of paper are quadrilaterals anyway, however), but if a user wants to run the program on multiple pieces of paper, they must run the program multiple times for the separate pieces of paper. (There could also be an extension here where the program takes more than one image, however there may be processing limitations when it comes to actually attempting to process the image.

# Objectives:

* Create a system that can recognise a piece of paper from its contours within an image.
* Build and train an AI that performs highly accurate (above 0.95 probability of correct) OCR on images.
* Check a given image for text at the very top of a piece of paper and assign the text file that title.
* Rip all text from a standard image and write it to a text file on the machine.
* Attempt to catch any mistakes that the AI makes with OCR utilising a spell checker.
* Ensure that any changes made by the spell checker are confirmed by the user, to ensure the program doesn’t mess up the file writing.

Process Diagram

Image Input

Edge Detection

Contours of paper found

Perspective Transform Applied

Text Written to .txt file

Text Returned

Image run through google tesseract