Image Processing and Computer Vision

Thomas William Gaff

Undergraduate dissertation submitted in

partial fulfilment of the regulations

governing the award of the degree of

BSc Computer Science

at the University of Sunderland

2018

Abstract

I created a Python application for performing Optical Character Recognition (OCR) on handwritten documents.

I did it because I was trying to solve the problem that I and a lot of other people who have appalling handwriting face, which is struggling to read our own handwritten notes. This problem is a problem faced by anyone who makes handwritten notes and has bad handwriting. The target for this specific application is for students and working adults in professions such as law and medicine.

I did it twice due to a problem with Python causing me to lose the original artefact. The original artefact was done using my own custom built machine learning model, which was built with Keras, and used OpenCV to perform pre-processing and segmentation on the image. The current artefact uses PyTesseract to preform the OCR on the image, but it still uses OpenCV to preform pre-processing and segmentation.

I learnt a lot throughout this project, the new skills that I have learnt are as following, Python 3.6, Keras, machine learning, image pre-processing, OpenCV and PyTesseract. The major outcome of the project has been being able to produce 2 different programs which have been able to preform pre-processing, segmentation and OCR on handwritten documents.

The future work that has been thrown up by my project are that neural networks are great at OCR and as such due to the unfinished nature of the current prototype. The need to preform more work on it to get it to an acceptable standard for use by other people is high and needs to be done.

Acknowledgements

First and foremost, I would like to thank my girlfriend Abigail, for her unending support and love through this project.

I would also like to appreciate my supervisor Dr Sheila Garfield, and my client, Dr Chris Bowerman, for their incredible assistance and advice they have provided while working on this project.

Contents

Chapter 1: Introduction

Problem context

The problem that this project is looking to tackle is people's poor handwriting affecting their learning by not allowing them to use their notes for revision or for research.

Client Introduction

My client is Dr Chris Bowerman (chris.bowerman@sunderland.ac.uk) he is a lecturer at The University of Sunderland and specialises in the following areas Artificial Intelligence, Machine Learning, Data Mining, Natural Language, Ethical Hacking, Digital Forensics, Data Science. Dr Chris Bowerman wants a program to be built that focuses on image processing, during the client meeting with him and the project was discussed, and he was on board to support the project

Proposed solution

There are a few solutions on the market at present, but these are either too expensive for the average person, require the person to have a computer with them at all times or require 3rd party tools such as pens. This problem needs a good cheap solution which does not need the person to have a computer, or 3rd party accessories with them at the time which is why this project's solution intends to eliminate these 3 problems, by being free to use, able to upload them to a computer at your convenience, and it requires no 3rd party tools. The solution that is being proposed would allow everyone to have their notes digitised, and their spelling corrected automatically, this would allow for a greater standard of learning for everyone, whether that it is a Year 1 student or an academic. The problem statement is "Producing digitised notes based on scans of people's handwritten notes, while correcting poor spelling"

Structure of Dissertation

In Chapter 2 there will be a critical analysis of offline handwriting recognition techniques for machine learning will be discussed that could be used in the artefact.

In Chapter 3 there will be focused on analysis, specifically requirements gathering, requirements analysis and requirements specification.

In Chapter 4 the details of the project methodology as well as the design of the solution will be discussed, the design of the solution will be separated into the User Interface and system.

In Chapter 5 the development of the solution will be outlined, it will detail the rapid prototyping, as well as the constraints of the solution and the hardware and software needed to complete the solution.

In Chapter 6 there will be discussion of the testing of the solution.

In Chapter 7 there will be a critical evaluation of the project, which will look into how the literature review affected artefact development, an evaluation of the artefact as well as an evaluation of self.

In Chapter 7 there will be some concluding remarks, what the limitations of the project were and some future recommendations.

Appendix A contains the Programme route which shows the course I am on and the modules I have done.

Appendix B contains the ethics documentation for the project.

Appendix C contains the evidence for project management and control of the project which is the project schedule, supervision records and the client meetings with Dr Chris Bowerman

Appendix D will contain analysis documentation such as the PACT.

Appendix E will contain design documentation such as the program diagrams.

Appendix F will contain development documentation such as the SCRUMs.

Appendix G will contain testing documentation such as the test outputs and notes.

Appendix H will contain evaluation documentation such as client feedback.

Appendix I will contain the code and screen cast.

Chapter 2: A critical analysis of offline handwriting recognition techniques for machine learning

Introduction

This chapter is focused on the research on the field of offline handwriting recognition. This chapter will go through each step that is taken in offline handwriting recognition, outline methods about how they are currently accomplished and discuss how these methods are relevant to the program being built and undertake a critical analysis of each method and step.

As described by Sahu and Kubde (2013) offline handwriting recognition has 5 main steps; image acquisition; pre-processing; segmentation; feature extraction and classification & recognition.

Image Acquisition

In this step as the title says it is about acquiring the image. For this a high-resolution scan of an image is needed to make sure the output is as accurate as possible although low-resolution images can be used they can cause problems such as similar characters being confused for the other such as a u and a, which drastically reduces the accuracy. As well as that it does not really matter if the image is grayscale or coloured although greyscale is preferred due to it making it obvious which areas of the page are empty space and which parts are text. As people tend to write using either blue or black ink, it is likely that coloured text will be found on a regular basis, although if the document has blue ink or any other coloured text this can cause another step to be added later to render the image greyscale, so the computer has an easier time of processing the text which can cause the processing to be slightly longer than that of a greyscale image which is why the program will have a notification to the user that the documents they submit should be in greyscale but it will also have the necessary functions to render coloured images greyscale.

Pre-processing

Pre-processing “is a series of operations performed on the scanned input image. It essentially enhances the image rendering it suitable for segmentation.” (Pradeep, Srinivasan and Himavathi, 2010) It has 6 sub sections within it; Noise reduction; binarization; edge detection; thresholding; skew detection and slant estimation & normalisation. (Sahu and Kubde, 2013)

Noise reduction is the first sub step of pre-processing. This is only relevant for poor-quality scanned images and poor-quality pictures, and while it is only necessary for these images and pictures it is still useful for even high-quality scans and pictures as even those can still have bits of noise that need to

be removed as a result of this the program will need to include this sub step as it is imperative to have the highest quality input possible. These images may contain noise which distorts the quality and can make the accuracy of the text reignition and word prediction suffer, as it can make letters look like nothing or different letters entirely, it can also cause a few other problems such as causing the line segments to become disconnected and gaps, sometimes rather large to appear between the lines (Sahu and Kubde, 2013). There are 2 ways to do this. The first is to use filters, these filters aim to reduce or outright remove noise and spurious points that occur due to an uneven writing surface (Sahu and Kubde, 2013). The other is to use morphological operations, the reason for their use is because they can be used to successfully remove noise on images due to low quality ink or paper and erratic hand movement (Sahu and Kubde, 2013) It is important that both of these methods are applied in the program as they each tackle different problems that can occur in the input and as stated previously the better the input the better the recognition. Although morphological operations are definitely the most important of the two to be applied, as generally most handwritten documents are done on smooth surfaces designed for writing and especially in MEDCs (More Economically Developed Countries) where most people have a good camera and access to a good scanner poor sampling rates are not much of a problem, although in LEDCs (Less Economically Developed Countries) both of those may prove to be bigger issues.

The second sub step is binarization. Binarization is the process of converting a grayscale image into a binary image. (Sahu and Kubde, 2013) Binarization is a very important step in the pre- processing of an image as it enables segmentation and the recognition of the characters (Jang and Hong, 1999). So, it is incredibly important for this to be in the program, as discussed in (Jang and Hong, 1999) there are a couple of ways of doing binarization, using the fact that the text in images will be darker than the paper or using the fact that characters are made up of lines usually the same width and using that. As the paper goes on to show, using algorithms which use the latter of those 2 ways have a much better success rate and are in fact faster than some which have a significantly worse success rate, as a result the latter of those 2 ways will be used in the program.

The third sub step is to perform edge detection on the image. The edges of paper or another writing surface set the boundaries of the object and are therefore important for the segmentation, registration, and identification of the handwriting. Using edge detection on an image significantly reduces the amount of information in the image and therefore reduces not only the image size but also gets rid of any unnecessary bits of information. There are two major categories of edge detection techniques, gradient and Laplacian “The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.” (Sahu and Kubde, 2013) and the “Laplacian method searches for zero crossings in the second derivative of the image to find edges”

(Sahu and Kubde, 2013). Out of the two methods Laplacian is the better method for image recognition so therefore the best method for the program. Laplacian has a specific subset known as Sobel, as discussed in (Shrivakshan and Chandrasekar, 2012) the Sobel method is excellent although it has problems with noise, however the 2 previous sub steps should have eliminated the noise as much as possible, meaning it is the best one for the program.

Thresholding is the fourth sub step. Thresholding is the needed to make sure that the datasets and the files which are needed to be processed do not take up too much storage space on the storage device and that when it is being processed it does not take too long either. There are two categories of thresholding, global and adaptive. Global is picking one threshold for the entire image. Adaptive thresholding is used when an image might need different levels throughout (Sahu and Kubde, 2013). In (Trier and Jain, 1995) a number of different thresholding techniques were evaluated by comparing the character recognition rates of each of them, the most successful being Wayne Niblack's method, which is what will be used in the program as it is the best method to do thresholding as it provides the best character recognition rate out of all of those which were tested.

The fifth sub step is skew detection. The first step in this process is to detect if there is a skew to the image. Skew is most commonly detected by using the connected parts and finding the average angels that connect the "detecting connected components and finding the average angles connecting their centre (Sahu and Kubde, 2013). If a skewness is detected the skew angle is used to correct it to make the image perfectly horizontal (Pratap and Arya, 2012). As discussed in (Farooq, Govindaraju and Perrone, 2005) the best method of doing this is take the start of a line and work out the height difference between the start and end of the line will tell you the skew angle, which you can then use to correct the skew. This will be the method used in the program as it is a good method and is easy to implement as the program will have all this data regardless, and it will help later down the line with the later steps, other methods are less precise and take a longer time to complete, such as working doing the difference between the average of the centre of each word.

The final sub step is to do slant estimation & normalisation. This is necessary as most people when they write even on lined paper have a slight slant to their handwriting. Characters which are slanted slope one of two ways, right to left or left to right. These two different variations may appear not only within the text but within each individual word (Sahu and Kubde, 2013). As outlined in (Farooq, Govindaraju and Perrone, 2005) a way to do slant estimation and normalisation is to first calculate the angle of skew to do that you have to find a baseline word, the best one to use for this is the first word. Then you rotate each word on its centre until it is aligned with the first word. Other methods proposed such as the algorithm in (Kavallieratou, Fakotakis and Kokkinakis, 2001) is done by slanting

each word left and right by a maximum of 45°, then the vertical histogram of each image is calculated, after that the Wigner-Ville distribution is calculate for the histograms, followed by extracting the curves of intensity of the Wigner-Ville distribution, then the curve of maximum intensity with the greatest peak, corresponding to the histogram with the most intense alternations, is chosen, finally that image is selected as the baseline and all other words are altered to meet it. The best way out of these two for my program is the algorithm as it easier to implement into the program and it is a lot better at doing slant estimation and normalisation as the former of the two options outlined, does not work properly all of the time as the first word may be the one that is the worst off and skewing everything else to meet it can have unexpected results, the algorithm on the other hand finds the least slanted word in the document making it the best way to do it, and it requires less computational power than the former as well.

Segmentation

In segmentation there are 2 different types, external and internal segmentation. According to Sahu and Kubde (2013) segmentation is essential to the process and is the most important step. Segmentation is done to make sure that there is a gap between each individual character of an image.

External segmentation is where an image of a document is broken up into logical sizes. For example, if this document was written it would be broken down into regions with text and regions without text. The regions without text are used to determine the context of what is a paragraph and what is a space etc. The regions with text are then split into paragraphs, these paragraphs would then be broken down into sentences and then words, this is where the recognition would (Bag, Bhowmick and Harit, 2011).

The other method is to use internal segmentation. Internal segmentation is the process of decompose a string of character images into sub images of each character (Sahu and Kubde, 2013). However, this process is fraught with problems especially with cursive as it is often hard to separate characters individually. This is why external segmentation is preferred over internal segmentation (Randriamasy and Vincent, 1994).

In the program external segmentation will be used as it is the most up to date method for segmentation and is the better of the two, as discussed above internal segmentation has the drawback that it is not good with cursive handwriting, not only that but external segmentation is the better of the two for this program as the way it breaks down the document would be useful for the recognition as in the program it will be built back in the opposite way, characters turned into words, words turned into sentences, sentences turned into paragraphs.

Feature Extraction

In the feature extraction step the goal of the process is to retrieve the most important data from the image (Sahu and Kubde, 2013). As part of the feature extraction stage, each individual character is represented as vector, which is what it is identified as. There are 2 ways of doing this, statistical features, structural features (Sahu and Kubde, 2013).

Statistical features are derived from the statistical distribution of points. They are high speed and quite low in terms of complexity and take into account style variations to a degree (Suen, Berthod and Mori, 1980), (Arica, 1998) a method for doing this called zoning is discussed in (Pradeep, Srinivasan and Himavathi, 2010) it involves breaking each character image into 90x60 pixels and is divides in 54 equal zones each the size of 10x10 pixels, each of these zones has 19 diagonal lines and the foreground pixels which are appear along each diagonal line are added together to get 1 sub-feature, for a total of 19 sub features, then those 19 sub features are averaged to form a single feature value.

Another method of doing it known as Characteristics Loci is discussed in (EL Melhaoui, El Hitmy and Lekhal, 2011) this method works by looking at the white points behind characters and from this vertical and horizontal lines are generated every time a line intersects a white point a vector is generated and used as a feature.

A commonly used method is crossing and distance, as discussed in (Mohamed and Gader, 1996) this is done by counting the number of transitions between background and foreground pixels along vertical and horizontal lines, it then calculates the distance of the first image pixel detected from the upper and lower boundaries of the image as well as the left and right boundaries.

Structural features allow for a high tolerance when it comes to distortions and style variations (Sahu and Kubde, 2013). “Structural features are based on topological and geometrical properties of the character, such as aspect ratio, cross points, loops, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom, etc” (Sahu and Kubde, 2013).

The method I have chosen for this step is zoning as it produces the best results, although it is slower than the others, but the wait is worth the increase accuracy.

Classification & Recognition

“The classification stage is the decision-making part of a recognition system and it uses the features extracted in the previous stage” (Sahu and Kubde, 2013).

Neural networks are used for handwriting recognition as they both fast and reliable and therefore good for use in systems where a large quantity of data needs to be processed and processed quickly such as postcodes on mail. Recurrent Neural Networks (RNNs) are the most common, as a result there are a lot of different algorithms for RNNs that exist. Some of which have achieved 99.9% recognition.

For example, putting in a dictionary and language model improves the handwriting recognition significantly, “with a dictionary and a language model this translates into a mean word error rate of 20.4%, which is a relative error reduction of 42.5% … Without the language model, the error reduction was 26.8%” (Graves et al., 2007). This shows that there are helpful bits of information which can be gleamed from online handwriting recognition. As this shows simply adding a dictionary and a language model it significantly improved the accuracy of the recognition by 15.7% which is very useful and is why such a system will be implemented later on in the program.

Conclusion

In conclusion, this is the outline of the different methods for each of the steps of offline handwriting recognition. All of the methods that have been chosen are state of the art or as close to state of the art as possible so as to make sure that the knowledge base is as up to date as possible.

Chapter 3: Analysis

Introduction

This chapter discusses the requirements of the artefact. Ranging from getting the general system requirements from the client and how they were obtained from the client, how the requirements for the users were gathered and how they impacted the artefact requirements

Requirements Gathering

The requirements for the project were gathered by interviewing the client, Dr Chris Bowerman about what was required of this project. Ultimately after an hour-long meeting with Dr Chris Bowerman and going through different ideas for the project such as an image search engine and other projects relating to computer vision and image processing after the project was decided to be about Optical Character Recognition (OCR) of handwritten data the requirements came together into the two different categories needed and additional features. They are three requirements in each category. The first category, needed features, had the following requirements: produce a program in which someone can upload pictures of their handwritten notes to; the program will use computer vision to read the notes using neural networks and deep learning; and the user will be able to get a PDF of their notes written up by the computer. The second category, additional features, had the following requirements: create a summary of the notes; allow the uploaded images to be stored; and allow images to be searchable based on their contents. There was also discussion on the different methods of implementation that could be used and what segments of machine learning to look at and base some of the requirements off of them. As well as the client meeting with Dr Chris Bowerman, other techniques were used in order to determine what requirements for the user would be. While this was discussed a little bit with the Client, it needed more research. The decision was to look for other Optical Character Recognition websites online and analyse them and use that to inform a PACT. Four Optical Character Recognition websites were looked at and notes on their design were written underneath along with a reference this can be found in Appendix D.

Requirements Analysis

In order to analyse the requirements a PACT was used to determine them as can also be seen in Appendix D. Firstly the key users had to be looked at. Identifying the key users of the artefact helps to determine certain requirements that each user group will have. In the PACT that was undertaken two key user groups were identified based on the websites that were visited as well as logical assumptions. These are Students aged between 14-23 as well as Working Adults aged between 18-

60. These groups were chosen as they encompass the vast majority of people who make handwritten notes on a regular basis. Students especially make handwritten notes all the time while learning. Those students especially at lower levels such as Year 10 and 11 rarely have access to computers to make notes on while in class and therefore have to rely on handwritten notes. Similarly Students who are at the university level and have to do a lot of diagrams or calculations most commonly do handwritten notes as opposed to digital ones. A lot of working adults especially in skilled professions such as doctors and lawyers as well as those in other positions such as assistants make a lot of handwritten notes as well. Especially for doctors and lawyers who will generally have a lot of information that is very important will want to use an artefact that allows them to store their handwritten notes on their computer, even if it is just a backup. As with all large groups of people both Students and Working Adults have a wide range of needs so they are said to be heterogenous. As a result the artefact needs to be accommodating to a lot of different requirements. The most glaring requirement is to make sure that the artefact is usable by both of these groups. While the vast majority of students are computer literate a lot of working adults are not as they did not grow up with computer like a lot of current students have. Although a lot of them can use a computer effectively they might not be considered power users, which are users who use advanced features of computers such as use the middle mouse button to click on a link to open a new tab instead or right clicking the link. As a result the artefact needs to have both easy use for normal users as well as built in shortcuts that power users could take advantage of. As well as this there are big differences between people in terms of physical disabilities such as colour blindness or especially in older Working Adults, bad eye sight. As such the artefact will have to be accommodating to requirements such as these by using high contrast colours such as black and white so as to reduce the problem that colour blind users might face. As well as allowing UI scaling so that users with bad eyesight who user large text sizes will also be able to use the artefact. As well as this the user groups have vast range of cognitive abilities. From looking at some of the Optical Character Recognition websites their use of logical flow of steps as well as easy to understand instructions are obvious requirements for this artefact as well so as to make sure that everyone can use it easily and effectively.

There are two activities which users will want to with the artefact. They are submit an image of notes to be digitised using Optical Character Recognition as well as downloading the notes once they have been digitised. These pose two requirements, a way for the user to submit an image and to download the notes once they have been digitised.

For the context that this artefact will be use in a variety of different place, ranging from a school classroom/an office to a coffee shop, meaning that the images for Optical Character Recognition could come in a variety of different formats as well as meaning the images that get uploaded will all

have different light levels and quality levels. As a result there is a requirement that the artefact will be able to handle a wide variety of image formats as well as using image pre-processing and segmentation to make sure that the Optical Character Recognition is as accurate as possible.

The technological requirements needed to make this artefact work are an image upload field, so that the user can easily upload their image to the artefact for Optical Character Recognition, a file download button so that the user can easily download their digitised notes, a machine learning component to do the Optical Character Recognition, possibly deep learning so that the machine learning can continue as the artefact keeps getting used, finally image processing is necessary to make sure that the images that are sent for Optical Character Recognition are as clean as they can be.

Requirements Specification

Lists of Functional (behavioural) requirements and Non-functional (quality) requirements.

Functional Requirements:

- Take an image in a variety of image formats.
- Pre-process the image using industry standard techniques.
- Take the image, cut out the background leaving the text, break down to the text into paragraphs, the paragraphs into sentences, the sentences into words and the words into characters.
- Perform Optical Character Recognition on each character, put the characters back into the word, put the word back into the sentence, put the sentence back into the paragraph and the paragraph back into the overall document.
- Output the digitised text into a PDF document and allow the user to download the PDF.

Non-functional Requirements

- Advanced features for power users.
- Make sure the User Interface is accommodating for people with disabilities such as colour blindness.
- Make sure that the layout of the artefact is logical and that there are instructions for use prominently displayed.
- Make sure that the artefact is stable.

Summary

Through meetings with the Client, Dr Chris Bowerman, and doing research on similar programs that exist, an exhaustive list of functional and non-functional requirements was put together. The

requirements were analysed through the use of a PACT, which was useful at allowing the requirements to be easily identified.

Chapter 4: Design

Introduction

The artefact that that was originally being developed was a Keras based machine learning model for Optical Character Recognition (OCR), that would use OpenCV to do the image pre-processing and the Keras based machine learning model to do the optical character recognition. It was designed to be built in multiple prototypes starting simply as single character recognition and ending with full document optical character recognition. The artefact that was originally being developed also had a basic web-based user interface for the user to interact with the python scripts. Due to the original artefact being corrupted a new artefact was created to take its place. The current artefact is a python based Optical Character Recognition system, OpenCv for the pre-processing just like the previous artefact as well as using PyTesseract for the optical character recognition. PyTesseract is a 3rd party library based on Google's Tesseract optical character recognition deep learning model, and that is what is used to perform the optical character recognition.

Original Artefact Work

Methodology and Procedure

Due to the nature of the project, the need to be adaptable to problems and to be able to work on other things while the training of the machine learning models were important factors while selecting a development approach. As a result, the project will be using the SCRUM methodology. The SCRUM will be broken down into four sprints, each sprint will be focused on one step of development, for example Sprint 1 will be for prototype 1, Sprint 2 will be for prototype 2 etc. Each prototype will focus on a different aspect of the application. Prototype 1 will focus on training a Keras based neural network to recognise characters, the second prototype will focus on word recognition using the Keras based machine learning model. Prototype 3 will focus on sentence and paragraph recognition using the Keras based machine learning model followed by the final program which will use OpenCV to segment the file down into manageable chunks which the Keras based machine learning model will use as its basis to perform optical character recognition to digitise the text.

Interface Design

The user interface will be very simplistic.There will be a very simple file upload section which once the file is uploaded will then disappear to present a waiting screen/progress bar followed by the output which will be easy to download for the user. The reason it will be a simple user interface is

because the majority of the application will be in the python script(s) in the backend. The reason the design is like this is a in which multiple webpages used for file conversions were looked at as well as other websites. From which it was found that basic user interfaces are better than complicated ones.

System Design

Prototype 1's design is basic, as can be seen in this program diagram in Appendix E. The program takes the image of the character. The program puts the image through some of pre-processing techniques that are detailed in Chapter 2 to clean up the noise of the image and reshape the image to a 28x28 size where following the pre-processing the image will be passed to the Keras machine learning model which has been trained using the MNIST dataset which will then preform optical character recognition on the image and then output the text in the image.

Prototype 2's design is a bit more complex than prototype 1, as can be seen in this program diagram in Appendix E. The program takes the image of the word it puts the image through some of the pre-processing techniques that are detailed in Chapter 2 such as rotating the image and making sure that noise reduction is applied. Following the pre-processing the image will be passed to the Keras based machine learning model which will then preform optical character recognition on the image and then output the text in the image.

Prototype 3's design builds upon prototype 2 as can be seen in this program diagram in Appendix E. The program takes the file that the user gives it puts the image through some of the pre-processing techniques that are detailed in Chapter 2. Following the pre-processing the image will be passed to The Keras based machine learning model which will then preform optical character recognition on the image and then output the text in the image.

The final system design is very simple. as can be seen in this program diagram in Appendix E. The program takes the file that the user gives it, it will use OpenCV and its libraries to put the image through some of the pre-processing techniques that are detailed in Chapter 2. Following the pre-processing the image will be passed to The Keras based machine learning model which will then preform optical character recognition on the image and then output the text in the image.

Current Artefact Work

Methodology and Procedure

Due to the time constraints applied to the current artefact, the need to be adaptable to problems and to be able to work on other things while the training of the machine learning models was happening were important factors while selecting the development approach. As a result, the

current artefact tis also using the SCRUM methodology. The SCRUM will be broken down into three sprints, each sprint will be focused on one step of development, for the current artefact the first Sprint will focus on getting the pre-processing of images and the segmentation to work using OpenCV, Sprint 2 will focus on training PyTesseract to recognise handwritten characters and words. Sprint 3 will focus on getting PyTesseract and OpenCV to work together so that when the image is segmented it is broken down into manageable chunks which PyTesseract will use as its basis to perform optical character recognition to digitise the text.

Interface Design

The user interface will be very basic, it will be a simple command line interface. The user will have to run the script using the command line and passing the image location into the script via the command line. The reason it will be a basic command line interface is because of the lack of time due to the loss of the original artefact.

System Design

Sprint 1's design is basic, as can be seen in this program diagram in Appendix E. The program takes the image of the text. Using OpenCV program puts the image through a variety of pre-processing techniques that are detailed in Chapter 2 to clean up the noise of the image as well as using contours to find the regions of interests which in this case would be the sentences and words.

Sprint 2's design is a lot less complex than that of Sprint 1, as can be seen in this program diagram in Appendix E. The program takes the image of the word it puts the image through some of the basic pre-processing techniques that are detailed in Chapter 2 such as rotating the image and making sure that noise reduction is applied. Following the pre-processing the image will be passed to PyTesseract which will do the optical character recognition.

Sprint 3's design builds upon both Sprint 1 and 2 as can be seen in this program diagram in Appendix E. The program takes the file that the user gives it puts the image through the pre-processing techniques that are detailed in Sprint 1. Following the pre-processing the image will then be contoured and from that the regions of interest will be chosen, these regions of interest will then be passed to PyTesseract which will then preform optical character recognition on the image and then output the text in the image.

The final system design is very simple. as can be seen in this program diagram in Appendix E the system design is fairly simple. The program takes the file that the user gives it, it will use OpenCV and its libraries to put the image through some of the pre-processing techniques that are detailed in Chapter 2. Following the pre-processing the image will be passed to The Keras based machine

learning model which will then preform optical character recognition on the image and then output the text in the image.

Summary

The design of the original artefact got progressively more and more complex as it went from prototype to prototype with the final program being the most complicated. The reason that the final program is more complicated is due to its implementation of multiple different python libraries as well as a web-based user interface. The current artefact though is a lot more complicated than the original final program, this is due to the difficulties of using PyTesseract for handwriting recognition.

Chapter 5: Development

Introduction

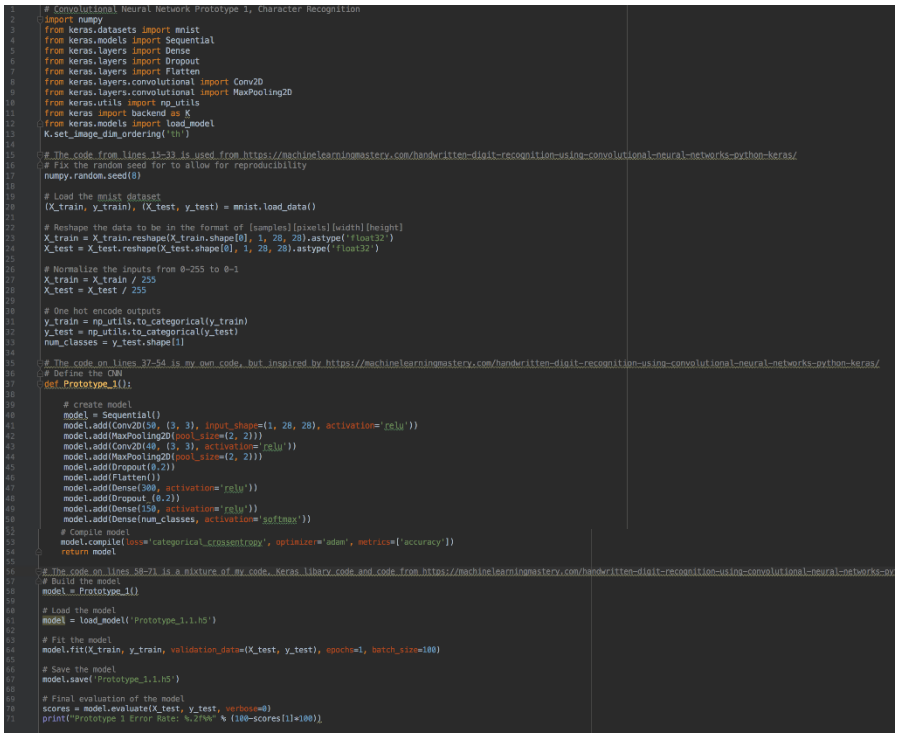
The solution to the problem that is outlined in Chapter1 is outlined here. Using neural networks and word prediction the project will allow people to upload a picture of their notes to a program that will read it and produce a PDF with their notes digitised as well as correcting their spelling. The project is planned in such a way that will there will be 3 prototypes made with each one building on the last until they come together to build the final program. Prototype 1 is focusing on characters and symbols, prototype 2 is focusing on words, prototype 3 is focusing on sentences and small paragraphs, while the final program will combine all of this build up and do full pages of notes. The way the project will be built is as follows:

Original Artefact Work

System Development

Prototype 1 (Character recognition)

Prototype 1 is a test application and therefore has only two non-functional requirement which is to make sure it is reliable and stable. The first step in its development was to find a dataset of just characters and symbols to train the Convolutional Neural Network on. The one that was chosen is a widely used dataset, due to the amount of data in it, and the fact it has both a training and testing part of its dataset. It is called the MNIST dataset. The second step is to code the pre-processing of the images. Since MNIST is a dataset designed for optical character recognition the images are already pre-processed to a degree, however it is necessary to make sure that the images are exactly 28x28 pixels. This can be seen in figure 5.1. The third step is to code the Convolutional Neural Network and create a way for the program to be easily run from the command line, since it will not be used for anything other than testing and training the Convolutional Neural Network it does not need a proper user interface. This can also be see in figure 5.1. The fourth step is to start the Convolutional Neural Network off on training to recognise handwritten characters and symbols, while keep an eye on how quickly it learns and making notes on possible improvements to be made. The final step is to test the Convolutional Neural Network properly once it has finished training. The way the training and testing of the Convolutional Neural Network is that epochs of the MNIST dataset are ran on it. First with 60,000 training images, followed by another 10,000 testing images. The Convolutional Neural Network managed a staggering 98.27% recognition rate, with more time and but more tweaking the Convolutional Neural Network most likely could have gotten a much higher recognition rate.



```
# Convolutional Neural Network Prototype 1, Character Recognition
import numpy
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
from keras.models import load_model
K.set_image_dim_ordering('th')

# The code from lines 15-33 is used from https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/
# Fix the random seed for to allow for reproducibility
numpy.random.seed(0)

# Load the mnist dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Reshape the data to be in the format of [samples][pixels][width][height]
X_train = X_train.reshape(X_train.shape[0], 1, 28, 28).astype('float32')
X_test = X_test.reshape(X_test.shape[0], 1, 28, 28).astype('float32')

# Normalize the inputs from 0-255 to 0-1
X_train = X_train / 255
X_test = X_test / 255

# One hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# The code on lines 37-54 is my own code, but inspired by https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/
# Define the CNN
def Prototype_1():

    # create model
    model = Sequential()
    model.add(Conv2D(50, (3, 3), input_shape=(1, 28, 28), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(40, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(300, activation='relu'))
    model.add(Dropout (0.2))
    model.add(Dense(150, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

# The code on lines 58-71 is a mixture of my code, Keras libary code and code from https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-py
# Build the model
model = Prototype_1()

# Load the model
model = load_model('Prototype_1.1.h5')

# Fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=1, batch_size=100)

# Save the model
model.save('Prototype_1.1.h5')

# Final evaluation of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Prototype 1 Error Rate: %.2f%%" % (100-scores[1]*100))
```

Figure 5.1 Prototype 1 code

Prototype 2 (Word recognition and word prediction)

Prototype 2 is another test application and therefore has one non-functional requirement which is to make sure the program is reliable. First step is to find a dataset of simple handwritten words to train the neural network on, preferably one with spelling mistakes, to test the word prediction. Second step is to make code changes to the neural network from prototype 1 as well as add word prediction. Third step is to start the neural network off on training to recognise handwritten words, while keep an eye on how quickly it learns and making notes on possible improvements to be made it should beat this point on the 05/01/18 and the final step is to test the neural network once its finished training. Should its success rate fall below 80% work out why and fix it, also work on seeing if the word prediction is accurate at guessing misspelt words as well as correctly spelt words.

Prototype 3 (Sentence and paragraph recognition)

Prototype 3 is yet another test application and therefore has one non-functional requirement which is to make sure it is reliable, and it works consistently. First Step is to find a data set of simple handwritten sentences or paragraphs to train the neural network on, preferably one with spelling mistakes. Second step is to make code improvements to the neural network, mainly adding in the ability to process sentences and paragraphs, and the improvements that come from the tests for prototype 2. Third step is to start the neural network off on training to recognise handwritten sentences and paragraphs, while keep an eye on how quickly it learns and making notes on possible

improvements to be made and the final step is to test the neural network once its finished training, and should its success rate fall below 80% work out why and fix it, also work on seeing if the word prediction is accurate at guessing misspelt words as well as correctly spelt words, and checking that the sentences and paragraphs make sense when they come out.

Final program

The final program is the finished product and therefore has non-functional requirements, the first one is reliability, second is usability as users will be using this version. First Step is to find a dataset of handwritten notes to train the neural network on, preferably ones with spelling mistakes, The Rimes dataset was chosen because of this. Second step is to make code improvements to the neural network and make a user- friendly GUI, which will be made so that its simple to use while looking modern and sleek. Third step is to start the neural network off on training to recognise handwritten notes, while keep an eye on how quickly it learns and making notes on possible improvements to be made. The fourth step is to test the neural network once its finished training, and should its success rate fall below 80% work out why and fix it, also work on seeing if the word prediction is accurate at guessing misspelt words as well as correctly spelt words, and checking that the sentences and paragraphs make sense when they come out with making changes and improvements to the neural network.

Interface Design

Resources and Constraints

There sources and constraints will be generally the same for every step. The resources needed to complete this project are a vast number of handwritten datasets and a powerful computer with a lot of hard drive space and one that can be left on for a long period of time. The constraints will be finding appropriate datasets or getting enough data to create my own, having enough hard drive space for the datasets and the program and its backups, finding a computer I can leave running for a large amount of time.

Hardware and Software

Hardware requirements will be better defined once progress is made, as I will need to work out how powerful a PC, how much hard drive space etc I need. Although Dr Chris Bowerman has advised me to use a GPU for training since the training is quite intensive, so a critical component is a good GPU.

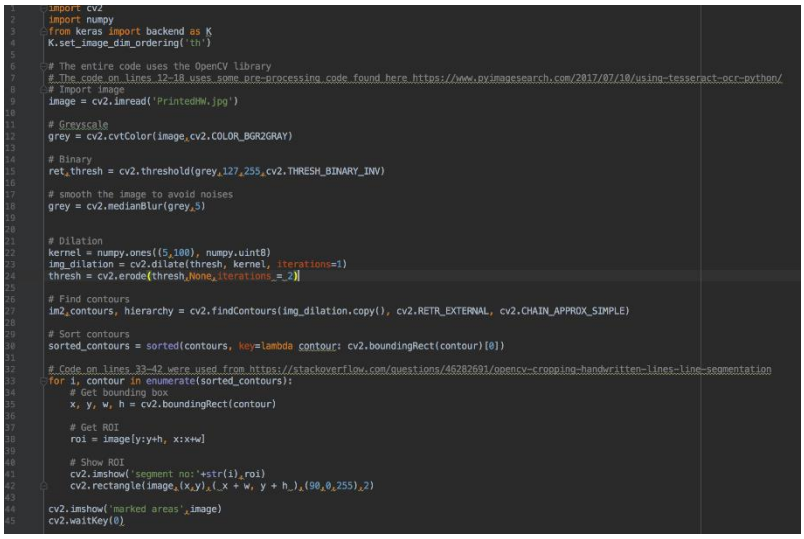
As for software, the programs will be being developed on macOS and Windows, Dr Chris Bowerman recommended to me to use Python and Keras for this project.

Current Artefact Work

System Development

Sprint 1 (Pre-processing and Segmentation)

Sprint 1 is focusing on testing half of an application and therefore has only two non-functional requirement which is to make sure it is reliable and stable. The first step in its development was to find a way of doing the image pre-processing and the segmentation, the best way of doing this is to use a 3rd party library. The one that was chosen is a widely used library, due to the vast compatibility and features it possess, but most importantly because it is open source and easy to use. It is called Open Source Computer Vision Library or OpenCV for short. The second step is to code the pre-processing of the images. Due to the fact that this artefact is designed for a user to give an image to the script, the user cannot be expected to have done any pre-processing of their own on the image. So, the code will treat all images as needing to be pre-processed. This can be seen in figure 5.2 as the first line of code imports the image and then the second line immediately makes it greyscale, then binarize the image and then blur the image to reduce noise. As well as the pre-processing of the image the, the image will also need to be segmented into regions of interest. Regions of interest are what in this case would be the text data, followed by the sentences and words on the image. The final step is to test the pre-processing segmentation. This will be done using white box testing. After each pre-processing step the image will be displayed. After that when the image is past to the segmentation, the overview of the regions of interest will be displayed. Followed by the individual regions of interest, which are displayed in their order of segmentation. Once all of the images have been shown they will be compared to the original image and what similar images look after each pre-processing step. Once that is done the overview of the regions of interest will be done for the image based on what it should be, followed by checking that the individual segments are numbered properly and that they show what they should, this can be seen in Appendix G.



```
import cv2
import numpy
from keras import backend as K
K.set_image_dim_ordering('th')

# The entire code uses the OpenCV library
# The code on lines 12-18 uses some pre-processing code found here https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/
# Import image
image = cv2.imread('PrintedHW.jpg')

# Greyscale
grey = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Binary
ret,thresh = cv2.threshold(grey,127,255,cv2.THRESH_BINARY_INV)

# smooth the image to avoid noises
grey = cv2.medianBlur(grey,5)


# Dilation
kernel = numpy.ones((5,100), numpy.uint8)
img_dilation = cv2.dilate(thresh, kernel, iterations=1)
thresh = cv2.erode(thresh,None,iterations =2)

# Find contours
im2,contours, hierarchy = cv2.findContours(img_dilation.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Sort contours
sorted_contours = sorted(contours, key=lambda contour: cv2.boundingRect(contour)[0])

# Code on lines 33-42 were used from https://stackoverflow.com/questions/46282691/opencv-cropping-handwritten-lines-line-segmentation
for i, contour in enumerate(sorted_contours):
    # Get bounding box
    x, y, w, h = cv2.boundingRect(contour)

    # Get ROI
    roi = image[y:y+h, x:x+w]

    # Show ROI
    cv2.imshow('segment no:'+str(i),roi)
    cv2.rectangle(image,(x,y),(x + w, y + h),(90,0,255),2)

cv2.imshow('marked areas',image)
cv2.waitKey(0)
```

Figure 5.2 Sprint 1 code

Sprint 2 (PyTesseract and Training)

Sprint 2 is focusing on testing half of an application and therefore has only two non-functional requirement which is to make sure it is reliable and stable. Work was started on this Sprint on the 13/04/18. First Step is to find a data set of simple handwritten files to us to test PyTesseract. Second step is to give PyTesseract an image argument to process as can be seen in figure 5.3. Third step is to run PyTesseract on an image to test how accurate it is. Fourth step is starting to train PyTesseract to be able to do OCR on handwritten documents. The first thing to do is to generate a word document which contains a bunch of random words and symbols in fonts which are similar to handwriting as you can see from figure 5.4. Following that using jTessBoxEditor it is then fed into tesseract which guesses at what it thinks each letter and symbol is, using the program the guesses can be corrected if Tesseract has got them wrong as can be seen from figures 5.5 and 5.6. From that training data is generated which is then placed in Tesseract's directory. The final step is to run Tesseract again and see what it's recognition is like, this is done until it is as good as it can be.

```python
# import the necessary packages
from PIL import Image
import pytesseract
import numpy
import cv2
import os

#The code is inspired by and uses some of the code found here https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/
#The code from lines 11-25 and 34-36 is from the OpenCV libary
#Import image
image = cv2.imread('PrintedHW.jpg')

# Greyscale
grey = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Binary
ret,thresh = cv2.threshold(grey,127,255,cv2.THRESH_BINARY_INV)

# Smooth the image to avoid noises
gray = cv2.medianBlur(grey,5)

# write the grayscale image to disk as a temporary file so we can
# apply OCR to it
filename = "{}.png".format(os.getpid())
cv2.imwrite(filename, grey)

# The code on line 28 is from the PyTesseract library
# Load the image as a PIL/Pillow image, apply OCR, and then delete the temporary file
text = pytesseract.image_to_string(Image.open(filename))
os.remove(filename)
print(text)

# show the output images
cv2.imshow("Image", image)
cv2.imshow("Output", grey)
cv2.waitKey(0)
```

Figure 5.3 Sprint 2 code

Figure 5.4 Tesseract Training Document

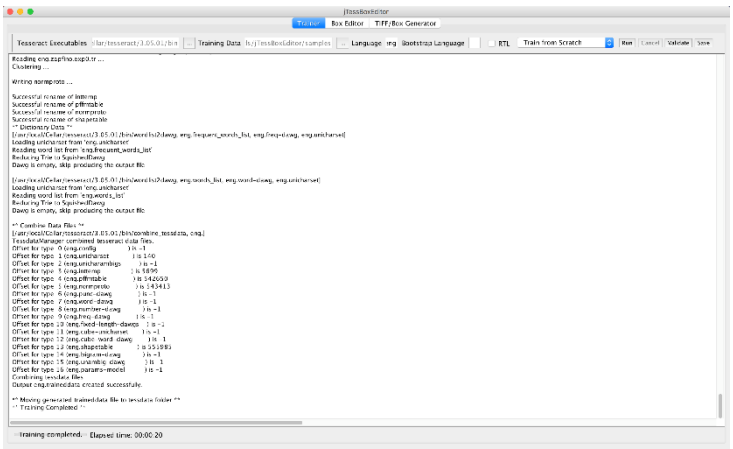


Figure 5.5 Tesseract Training output

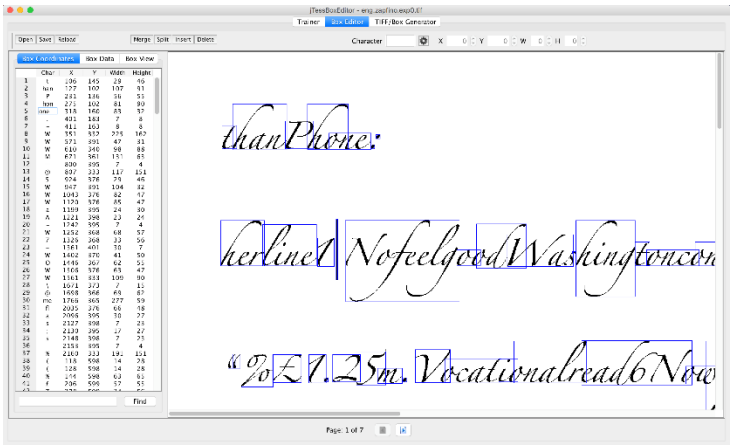
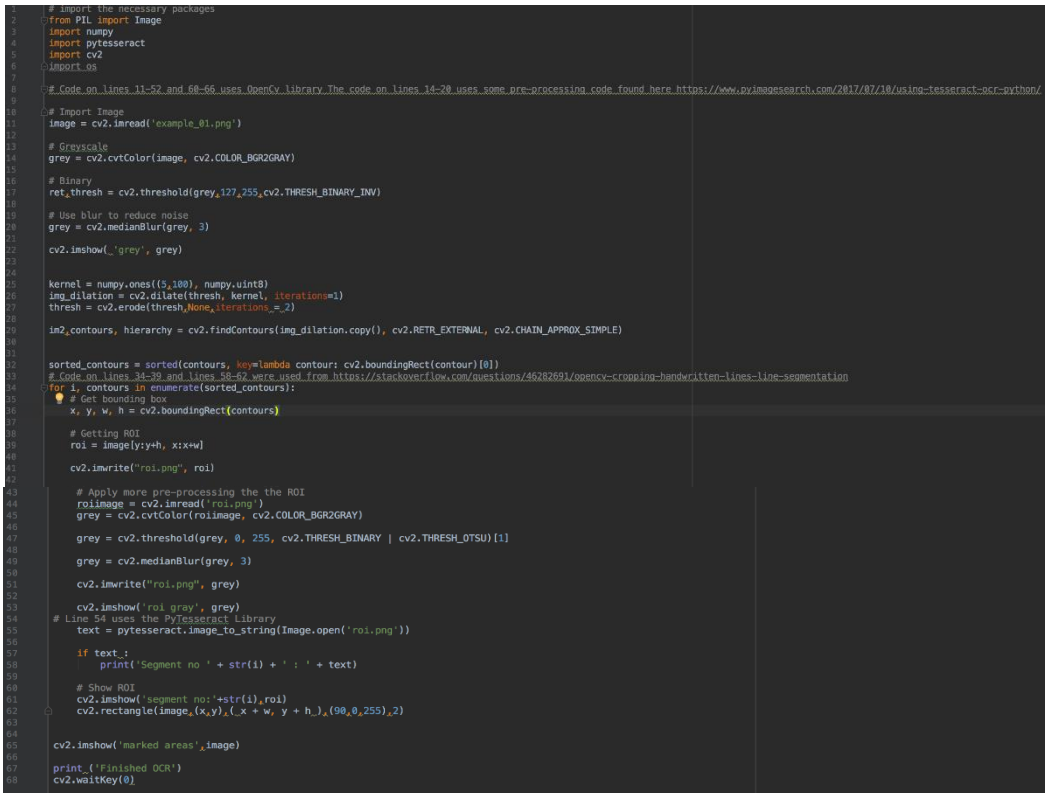


Figure 5.6 Tesseract box training

Sprint 3 (Combining Sprint 1 and 2)

Sprint 3 is focusing on combining two tested halves of an application and therefore has only two non-functional requirement which is to make sure it is reliable and stable. First Step is to copy and paste the work from Sprints 1 and 2 into one python script and fix any errors that occur . Second step is to make code changes were necessary to make sure that the 2 parts come together and can work with each other. Third step is to perform all of the same tests that were performed on Sprint 1, this will be as it was in Sprint 1 by using white box testing. After each pre-processing step the image will be displayed. After that when the image is past to the segmentation, the overview of the regions of interest will be displayed just like in Appendix G. Followed by the individual regions of interest, which are displayed in their order of segmentation. Once all of the images have been shown they will be compared to the original image and what similar images look after each pre-processing step. Once that is done the overview of the regions of interest will be done for the image based on what it should be, followed by checking that the individual segments are numbered properly and that they show what they should. Fourth Step is to test PyTesseract's OCR on the regions of interest and make sure it is getting the regions of interest correct and that it is outputting all of the ones which have text in them. The final step is to make any other alterations that are necessary.



```
# import the necessary packages
from PIL import Image
import numpy
import pytesseract
import cv2
import os

# Code on lines 11-52 and 60-66 uses OpenCv library The code on lines 14-20 uses some pre-processing code found here https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/

# Import Image
image = cv2.imread('example_01.png')

# Greyscale
grey = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Binary
ret,thresh = cv2.threshold(grey,127,255,cv2.THRESH_BINARY_INV)

# Use blur to reduce noise
grey = cv2.medianBlur(grey, 3)

cv2.imshow('grey', grey)


kernel = numpy.ones((5,100), numpy.uint8)
img_dilation = cv2.dilate(thresh, kernel, iterations=1)
thresh = cv2.erode(thresh,None,iterations=2)

im2,contours, hierarchy = cv2.findContours(img_dilation.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)


sorted_contours = sorted(contours, key=lambda contour: cv2.boundingRect(contour)[0])
# Code on lines 34-39 and lines 58-62 were used from https://stackoverflow.com/questions/46282691/opencv-cropping-handwritten-lines-line-segmentation
for i, contours in enumerate(sorted_contours):
    # Get bounding box
    x, y, w, h = cv2.boundingRect(contours)

    # Getting ROI
    roi = image[y:y+h, x:x+w]

    cv2.imwrite("roi.png", roi)

    # Apply more pre-processing the the ROI
    roiimage = cv2.imread('roi.png')
    grey = cv2.cvtColor(roiimage, cv2.COLOR_BGR2GRAY)

    grey = cv2.threshold(grey, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]

    grey = cv2.medianBlur(grey, 3)

    cv2.imwrite("roi.png", grey)

    cv2.imshow('roi gray', grey)
# Line 54 uses the PyTesseract Library
    text = pytesseract.image_to_string(Image.open('roi.png'))

    if text_:
        print('Segment no ' + str(i) + ' : ' + text)

    # Show ROI
    cv2.imshow('segment no:'+str(i),roi)
    cv2.rectangle(image,(x,y),(_x + w, y + h_),(90,0,255),2)


cv2.imshow('marked areas',image)

print_('Finished OCR')
cv2.waitKey(0)
```

Figure 5.7 Sprint 3 code

Interface Development

Due to the time constraint put on the project by the loss of the original artefact there was no time for a good user interface and as a result the user has to use the command line to use the current artefact.

Resources and Constraints

The resources and constraints are generally the same for this artefact as they were in the original artefact. As with the original artefact the resources and constraints are generally the same with every step. The resources needed to complete this project are a few handwritten document examples, as well as and a powerful computer with a lot of hard drive space and one that can be left on for a long period of time. The constraints will be finding appropriate datasets or getting enough data to test OpenCV and PyTesseract and finding a computer I can leave running for a large amount of time.

Hardware and Software

Hardware requirements will be better defined once progress is made, as I will need to work out how powerful a PC, how much hard drive space etc I need. Although Dr Chris Bowerman has advised me to use a GPU for training since the training is quite intensive, so a critical component is a good GPU.

As for software, the programs will be being developed on macOS and Windows, Dr Chris Bowerman recommended to me to use Python and Keras for this project.

Chapter 6: Evaluation

Introduction

This chapter is focused on the evaluation of the artefact as well as an evaluation of the way the project as whole was done. It outlines the way things were done and why they were chosen and then evaluates the way they were done and why they were chosen.

Impact of the literature review on the artefact development

The literature review was very helpful while designing and developing both the original and current artefact as it helped with one of the most important part of the both the original and current artefact which is the pre-processing of the image. The literature review was very helpful at pointing out which pre-processing techniques were the most important to do on an image and which of the ways to do each pre-processing techniques are the best. Ultimately the literature review lead me to a python library called OpenCV which has a lot of the pre-processing techniques in it already which allowed me to speed up development and make help to make sure that the artefact is as good as it can be.

Evaluation of the original artefact

The artefact started as a Keras based machine learning model which was used to identify characters. The first artefact was done on time and to the specifications that were outlined in design. It had a very good recognition rate of 98%. Overall prototype 1 went very well. It was trained using the MNIST dataset.

It then went into its second prototype which was a python script using The Keras based machine learning model using simple pre-processing on words. The second prototype was mostly done to the specifications however there were problems with getting the word prediction to work properly, mainly down to the lack of sufficient training data hence why it was removed from prototype 2 and from the rest of the prototypes.

It then moved onto prototype 3 which was a python script using The Keras based machine learning model using simple pre-processing on sentences and paragraphs. The third prototype was done to the specifications however there were problems with getting the optical character recognition to work properly, mainly down to the segmentation not being good enough for The Keras based machine learning as well as that there was a problem that arose with using NVIDIA's CUDA to train the AI as it caused the python environment to become corrupted as well as losing a lot of python work due to it crashing while training. However this was solved by manually training The Keras based machine learning model to be better at recognising handwritten data which has made it more on a

document scale than out of the box The Keras based machine learning model, although it was still not above a 43% recognition rate.

Ultimately the original artefact has ended with a python script which uses OpenCV and The Keras based machine learning model and has a basic UI. The final program was done to the specifications however there were still problems with getting the optical character recognition to work properly. However, this version of The Keras based machine learning model improved on the previous prototype's. Word prediction was once again tried but made the final program unstable and therefore was removed.

Evaluation of the current artefact

The current artefact started as two different half's that were being worked on. The first was the pre-processing and segmentation half. This half was to provide the pre-processing of the images and to segment them into logical chunks for Optical Character Recognition. It was done by using the OpenCV library for python. This first half was done to specifications and worked perfectly

The second half was the Optical Character Recognition half. This half was to take the segments from the first half and perform Optical Character Recognition on them. It was done by using the PyTesseract python library. The second half was mostly done to specifications however there were problems with getting the optical character recognition to work properly, as PyTesseract is designed to be used on typed text so it's optical character recognition on the handwritten data was hit and miss at best with sometimes it reading nothing from the image.

Ultimately the current artefact has ended with a python script which uses OpenCV and PyTesseract libraries and has a command line UI. The final program was done to the specifications however there were still problems with getting the optical character recognition to work properly. However, this version of PyTesseract improved on the previous version. This was down to more training and the fact that PyTesseract has built in deep learning which meant it got better with every test. Although not part of the original specification, the final artefact has the ability to use Optical Character Recognition on both handwritten text and typed text meaning that both of the key user groups of Students and Adult Workers as students might have handouts from lecturers or they might want to life a passage of text from a book. Similarly Adult Workers will also have need of a typed text Optical Character Recognition as lawyers for example might want to lift a case from a particular common law book.

What does your client think of this?

The Client, Dr Chris Bowerman evaluated the current artefact and said that while it was not as promising as the original artefact and while it missed a few of the requirements it was still a good artefact and that it was a good demonstration and that it had a lot of future potential. This can be seen from the email of the notes between me and Dr Chris Bowerman in Appendix C.

Evaluation of self

When starting work on the project, there were an abundance of different projects that seemed to be interesting and were great ideas to work on, however at that start of the project the amount of time available and the lack of fundamental knowledge on Machine Learning meant that the knowledge required in order to achieve them was too great to achieve within the time frame. After consulting with both the client and supervisor, I was able to significantly narrow the choices of projects down to 3. The follow up talks that took place with the client helped to narrow it down to the current project. Talking with the supervisor helped keep the project reasonable and allowed for a schedule to be easily created which set out a realistic time frame for the project to be completed in.

During the course of the project, background research was conducted at each stage. Each Sprint contained a little bit of background research as due to the nature of the project, parameters changed quickly and additional bits of background research were needed to adapt to them, the best example of this is where Prototype 1's Keras machine learning model was replaced by PyTesseract a python library that is based on Google's Tesseract optical character recognition model. During Sprint 1 there was a lot of background research into current and cutting-edge technologies, mainly focused on image pre-processing, although other areas of study were also looked at within Sprint 1 such as machine learning techniques as well as research into Python and Keras, this research was key in this project. The vast majority of these techniques were visited over the course of development, most of the principles were tried which was very time intensive. The majority of them however, were successful and useful and therefore ended up being implemented in various prototypes and in the final program.

The agile methodology that was used for the project allowed for the project schedule to be easily adapted and reorganised when any changes occurred or needed to be made. The agile approach was really crucial while dealing with development problems, such as the Keras model being difficult to make work on larger pieces of text. It was also crucial in deal with some personal issues which were experienced over the last few months of the project whereby progress on the project was very slow and there were various delays in the project. Despite losing a lot of time to these problems due to the agile way of working I was easily able to alter the schedule to keep the deliverables on track.

Using local version control was very useful as it enabled the project to be split not only into its various prototypes but those various prototypes into specific versions. This was very useful in helping to track the progress being made on the project as well as making it easy to look back and use older versions of different prototypes on the newer prototypes. For example some of the changes that were made to pre-processing between the first version of Prototype 2 and first version of Prototype 3 were not as effective with the longer text formats of Prototype 3 but some of the earlier versions of Prototype 2 had better pre-processing in them which allowed them to be easily implemented into Prototype 3.

There are somethings that could have been done better on the project. Setting specific aims and goals that were realistic given the time available and the project itself as early as possible during the project would have lead to a much easier to development and would have meant that time could have been saved. To help ease the dissertation writing process, Chapters 4, 5 and 6 should have been written while development on the artefact was ongoing where they should have been put into drafts and been checked over by my Client and Supervisor instead of doing it last few weeks of the project. This would have enabled more time to be spent on including more detailed information in Chapters 4, 5 and 6. It would have also been helpful to the supervisor and client to understand the progress and provide concise feedback early and to help with any problems that arose.

Through the course of the project, I have learnt some new skills. These new skills mainly centre around the actual tools I used to create the artefact. The new skills that I have learnt are as following, Python 3.6, Keras, machine learning, image pre-processing, OpenCV and PyTesseract. Learning Python and Keras was essential to the artefact and it will provide me with some good skills for employment. Learning how to set up and train machine learning models is also a great skill, especially since I want to do a machine learning Masters and possibly a PHD. Learning about image pre-processing and implementing it, as well as using an image pre-processing, OpenCV. These are important skills especially in the field of machine learning. As well as that using 3rd party libraries such as OpenCV and PyTesseract are useful Python skills, as Python coding relies upon using a lot of 3rd party libraries. I have also used several skills that I have been taught previously throughout my degree. As can be seen in Appendix A, I have taken a lot of different modules in a lot of different subject areas. These have given me skills which I have used in this project such as CET206 which provided me with the SCRUM skills that I used on the project. As well as this, CET212 gave me the skills to do advanced programming, such as Object Oriented programming as well as prototyping projects.

Summary

The prototypes and the final program that have been created, despite their failings were able to perform their intended tasks. While some prototypes were able to perform their intended task better than others overall the artefact went well.

The way in which the project has been carried out, despite some setbacks has been carried out to a high standard. With the steps that were taken to ensure that this project would succeed such as using agile development and local version control really helped the project to keep on track despite the setbacks.

In conclusion, I believe that the intended aim was achieved by creating a Convolutional Neural Network which had a 98% character recognition rate, and while PyTesseract never got that good at recognition with more time that would be possible however the fact that it still does an ok job means that the project is a success.

Chapter 7: Conclusion and Future Recommendations

Concluding remarks

A detailed evaluation of the project has been made in the previous chapter, what remains to be discussed is the future direction of the work.

Limitations of the project

There were a total of six limitations imposed on the project at the start, three of which were features that needed to be implemented, three were additional features to be added if the project was complete and there was time left. Out of the three necessary features only one was not implemented, this was allowing the user to be able to get a PDF of their notes written up by the computer after it was finished doing the optical character recognition. The reason this was not done was due to personal issues allowing down development which meant that this feature which was the last core feature to be implemented was not implemented.

The three additional features were also not added due to time, but these were all researched in their own right as an explore into how easy they would be to implement. With the first additional feature the creation of a summary of the notes that were uploaded, would have required the time to build, train and test another neural network to summarise text. The second additional feature of allowing images to be stored would have been easy to implement had the original idea of a desktop application been used however due to website being used this would have required work to create a login system as well as databases for storing the images. As with the pervious additional feature the last additional feature which was allowing images to be searchable based on their contents was not possible for the same reasons as well as the fact that the recognition rate would need to be a lot higher than it is in order to make sure this feature worked well.

Future directions

The project has 2 future directions it could go in, as has been shown in this project Neural Networks are proven to be reliable for optical character recognition and with more and more training can be used to get pin point accuracy. By using neural networks such as Convolutional Neural Networks that were used in prototype one, it is possible with the new knowledge of Python, Keras, OpenCV as well as knowledge of neural networks gained over the time on this project that a new one could be created, trained and used to better predict than PyTesseract currently can.

As well as that direction there is a definite direction that the project must go in first and that is to get the output to be outputted to a PDF for easy download by the user as well as to implement as many

of the additional features as possible. Once this has been achieved one of the two directions can be chosen.

References

Romano, B. and Silva, A. (2015). Project management using the Scrum agile method: A case study within a small enterprise. Information Technology - New Generations (ITNG), 2015 12th International Conference, [online] pp.774-776. Available at: http://ieeexplore.ieee.org/document/7113578/ [Accessed 23 Oct. 2017].

Graves, A., Fernandez, S., Liwicki, M., Bunke, H. and Schmidhuber, J. (2007). Unconstrained Online Handwriting Recognition with Recurrent Neural Networks. Advances in Neural Information Processing Systems 20 (NIPS 2007). [online] Available at: http://papers.nips.cc/paper/3213- unconstrained-on-line-handwriting-recognition-with [Accessed 1 Nov. 2017].

Tappert, C., Suen, C. and Wakahara, T. (1990). The state of the art in online handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8), pp.787-808.

Sahu, V. and Kubde, B. (2013). Offline Handwritten Character Recognition Techniques using Neural Network: A Review. International Journal of Science and Research, [online] 2(1), pp. 87-94. Available at: https://www.ijsr.net/archive/v2i1/IJSR13010129.pdf [Accessed 5 Nov. 2017].

Pradeep, J., Srinivasan, E. and Himavathi, S. (2010). Diagonal Feature Extraction Based Handwritten Character System Using Neural Network. International Journal of Computer Applications, 8(9), pp.17-22.

Jang, J. and Hong, K. (1999). Binarization of noisy gray-scale character images by thin line modeling. Pattern Recognition, 32(5), pp.743-752.

Shrivakshan, G. and Chandrasekar, C. (2012). A Comparison of various Edge Detection Techniques used in Image Processing. IJCSI International Journal of Computer Science Issues, 9(5), pp. 269-276.

Trier, O. and Jain, A. (1995). Goal-directed evaluation of binarization methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(12), pp.1191-1201.

Pratap, N. and Arya, S. (2012). A Review of Devnagari Character Recognition from Past to Future. International Journal of Computer Science and Telecommunications, 3(6), pp.77-82.

Farooq, F., Govindaraju, V. and Perrone, M. (2005). Pre-processing methods for handwritten Arabic documents. In: Eight International Conference on Document Analysis and Recognition. Seoul: IEEE.

Kavallieratou, E., Fakotakis, N. and Kokkinakis, G. (2001). Slant estimation algorithm for OCR systems. Pattern Recognition, 34(12), pp.2515-2522.

Bag, S., Bhowmick, P. and Harit, G. (2011). Word & Character Segmentation for Bangla Handwriting Analysis & Recognition. In: Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics. Hubli: IEEE, pp.243-246.

Randriamasy, S. and Vincent, L. (1994). Benchmarking Page Segmentation Algorithms. In: Computer Vision and Pattern Recognition. Seattle: IEEE, pp.411-416.

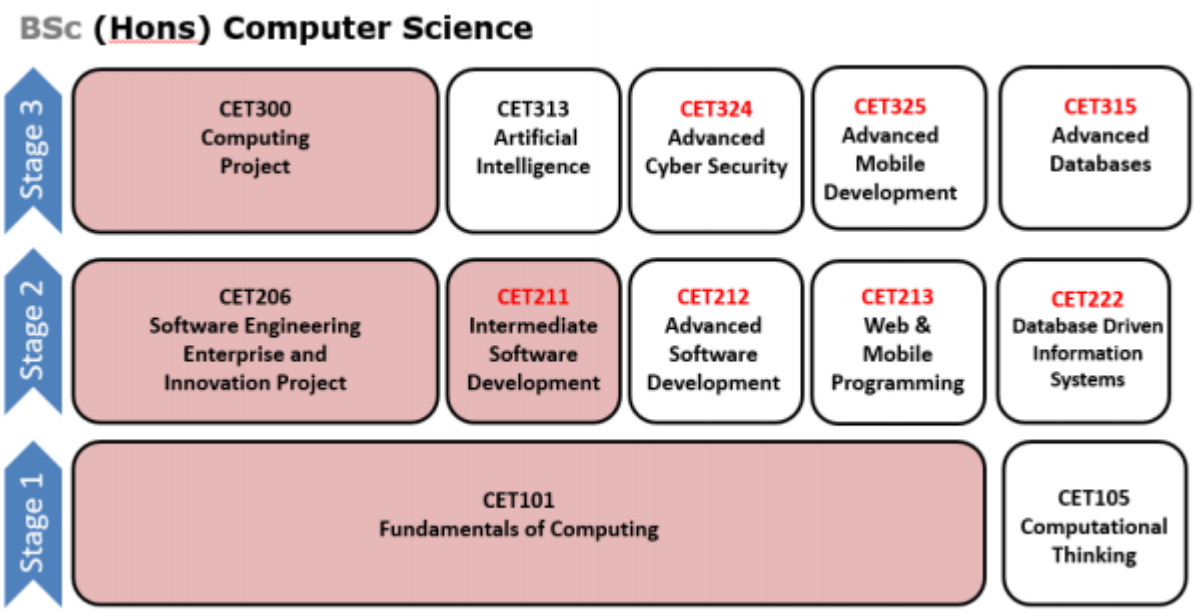
Suen, C., Berthod, M. and Mori, S. (1980). Automatic recognition of handprinted characters—The state of the art. Proceedings of the IEEE, 68(4), pp.469-487.

Arica, N. (1998). An Offline Character Recognition System For Free Style Handwriting. Masters. The Middle East Technical University

EL Melhaoui, O., El Hitmy, M. and Lekhal, F. (2011). Arabic Numerals Recognition based on an Improved Version of the Loci Characteristic. International Journal of Computer Applications, 24(1), pp.36-41.

Mohamed, M. and Gader, P. (1996). Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(5), pp.548-554.

Appendix A


BSc (Hons) Computer Science
Stage 3
CET300 Computing Project
CET313 Artificial Intelligence
CET324 Advanced Cyber Security
CET325 Advanced Mobile Development
CET315 Advanced Databases
Stage 2
CET206 Software Engineering Enterprise and Innovation Project
CET211 Intermediate Software Development
CET212 Advanced Software Development
CET213 Web & Mobile Programming
CET222 Database Driven Information Systems
Stage 1
CET101 Fundamentals of Computing
CET105 Computational Thinking

Appendix B

University of Sunderland

Faculty of Computer Science
Ethics Supervisor Certification Record for projects involving Human Test Participants

Project Details	
Indicative/Short Title:	Image Processing and Computer Vision relating to offline handwriting recognition
Student Name:	Thomas William Gaff
Supervisor Name:	Sheila Garfield

Project Participants and Activities	
1. Will this project ONLY involve HEALTHY Adults answering questionnaires or participating in interviews about non-sensitive topics?	Yes
2. Will this project ONLY involve HEALTHY Adults participating in user-acceptance and/or usability tests with a computer system in a non-sensitive domain?	Yes
3. Will this project involve children or vulnerable adults?	No
Answering YES to question 3 means that an application for approval must be submitted to the University Research Ethics Committee. Answering yes to questions 1 and 2 falls within the remit of supervisor self-certification. However, the project must be conducted within the principles of informed consent and ethical treatment of participants.	

Declarations	
4. Has the supervisor received ethics training?	Yes/~~No~~
5. The student has produced study information sheet for participants	Yes/~~No~~

6. The student has produced a consent form	Yes/~~No~~

Supervisor Signature
Signature: *Sheila Garfield*
Counter signature of Module leader if answer to Question 4 is no:

PARTICIPANT INFORMATION SHEET

Study Title: Looking into the user friendliness of a handwriting recognition application

What is the purpose of the study?

The purpose of this study is to look into how people interact with an application designed to read handwritten notes and produce a digital version of the document and to take feedback on the design of the application.

Who can take part in the study?

Anyone who is of sound mind and is over the age of 18.

Do I have to take part?

Participation is entirely voluntary. If you change your mind about taking part in the study, you can withdraw at any point during the session without giving a reason and without penalty.

What will happen to me if I take part?

You will be asked at your own convenience to access the application and complete a survey which will ask you to do tasks and provide feedback on how easy it was to perform these tasks.

What are the possible disadvantages and risks of taking part?

There are no risks to those involved unless you have any pre-existing medical conditions which would make being on a computer for an hour and a half a risk to your health.

What are the possible benefits of taking part?

The possible befits of taking part in this study are helping a university student improve their dissertation and maybe a future commercial product.

What if something goes wrong?

If you feel unhappy about the conduct of the study, please contact me immediately or the Chairperson of the University of Sunderland Research Ethics Group, whose contact details are given below.

Will my taking part in this study be kept confidential?
If suitable, the results may also be presented at academic conferences and/or written up for publication in peer reviewed academic journals.

What will happen to the results of the research study?
If suitable, the results may also be presented at academic conferences and/or written up for publication in peer reviewed academic journals.

Who is organising and funding the research?
It is being funded and organised by myself.

Who has reviewed the study?
The University of Sunderland Research Ethics Group has reviewed and approved the study.

Contact for further information
Name: Thomas William Gaff (University Student, University of Sunderland)
Email: bg88vx@student.sunderland.ac.uk
Phone: 07703 348811
Name: Doctor John Fulton (Chair of the University of Sunderland Research Ethics Group, University of Sunderland)
Email: john.fulton@sunderland.ac.uk
Phone: 0191 515 2529
Name: Sheila Garfield (Academic Lecturer, University of Sunderland)
Email: sheila.garfield@sunderland.ac.uk
Phone: 0191 515 4253

Consent Form for: Looking into the usability of a handwriting recognition application

Please tick the appropriate boxes	**Yes**	**No**
Taking Part		
I have read and understood the project information sheet dated 09/01/2018.	□	□
I have been given the opportunity to ask questions about the project.	□	□
I agree to take part in the project. Taking part in the project will include a written survey.	□	□
I understand that my taking part is voluntary; I can withdraw from the study at any time and I do not have to give any reasons for why I no longer want to take part.	□	□
Use of the information I provide for this project only		
I understand my personal details such as phone number and address will not be revealed to people outside the project.	□	□

I understand that my words may be quoted in publications, reports, web pages, and other research outputs.	□	□

*Please choose **one** of the following two options:*

I would like my real name used in the above	□	
I would **not** like my real name to be used in the above.	□	

Use of the information I provide beyond this project

I agree for the data I provide to be archived for use in this research.	□	□
I understand that University of Sunderland lectures will have access to this data only if they agree to preserve the confidentiality of the information as requested in this form.	□	□

So we can use the information you provide legally

I agree to assign the copyright I hold in any materials related to this project to Thomas William Gaff.	□	□

_____	_____	_____
Name of participant [printed]	Signature	Date
_____	_____	_____
Researcher [printed]	Signature	Date

Project contact details for further information:

Name: Thomas William Gaff (University Student, University of Sunderland)
Email: bg88vx@student.sunderland.ac.uk
Phone: 07703 348811

Appendix C

Schedule

No.	Task	Hours	Planned Start	Actual Start	Planned Finish	Actual Finish	Deliverable
	Planning& Control	***41***					
1	Draft Proposal	5	25/09/17	25/09/17	05/10/17	05/10/17	Draft Proposal
2	Completed Proposal	3	06/10/17	06/10/17	13/10/17	01/6/10/17	Signed off Proposal
3	Draw up Schedule	4	10/10/17	10/10/17	30/10/17	17/10/17	Draft Schedule
4	Definitive Brief		20/10/17	21/10/17	31/10/17	31/10/17	Definitive Brief
5	Prepare for Supervision	10	02/10/17	02/10/17	16/04/18	16/04/18	Updated eportfolio blog, supervision record and associated documentation
6	Attend Supervision	4	03/10/17	03/10/17	16/03/18	16/03/18	Supervisor feedback comments in eportfolio
7	Find appropriate datasets	15	16/10/17	16/10/17	21/10/17	21/10/17	Sample of dataset.
	Literature Review	***100***					
8	Gather Literature/ Materials	20	04/10/17	05/10/17	06/11/17	12/11/17	Bank of Research Materials
9	Make notes	30	10/10/17	10/10/17	06/11/17	03/11/17	Research Notes
10	Draft notes into chapter	50	07/11/17	07/11/17	15/12/17	15/12/17	Draft Research Chapter

	Development	***156***					
11	Learn Python	15	10/10/17	10/10/17	22/10/17	22/10/17	Notes and a few example programs
12	Learn Keras	10	22/10/17	23/10/17	27/10/17	27/10/111	Notes and a few example programs
13	Find and prepare data	20	10/11/17	10/11/17	20/11/17	20/11/17	New dataset, with example records
14	Interview Client	2	5/10/17	5/10/17	10/03/18	15/04/18	Interview notes
15	Analyse current systems/ situation	10	01/02/18	01/02/18	04/02/18	04/02/18	Analysis Documentation
16	Design & develop prototype 1	20	23/10/17	23/10/17	29/10/17	16/11/17	Design documentation & prototype
17	Setup Training prototype 1	1	24/10/17	16/11/17	24/10/17	16/11/17	Training report
18	Design & develop prototype 2	20	20/12/17	20/12/17	05/01/18	05/01/18	Design documentation & prototype
19	Setup Training prototype 2	1	05/01/18	05/01/18	05/01/18	05/01/18	Training report
20	Design & develop prototype 3	20	20/01/18	20/01/18	01/02/18	01/02/18	Design documentation & prototype
21	Setup Training prototype 3	1	01/02/18	01/02/18	01/02/18	01/02/18	Training report
22	Design & develop final prototype	25	20/02/18	20/02/18	05/03/18	05/03/18	Design documentation & prototype
23	Setup Training final prototype	1	05/03/18	05/03/18	05/03/18	05/03/18	Training report

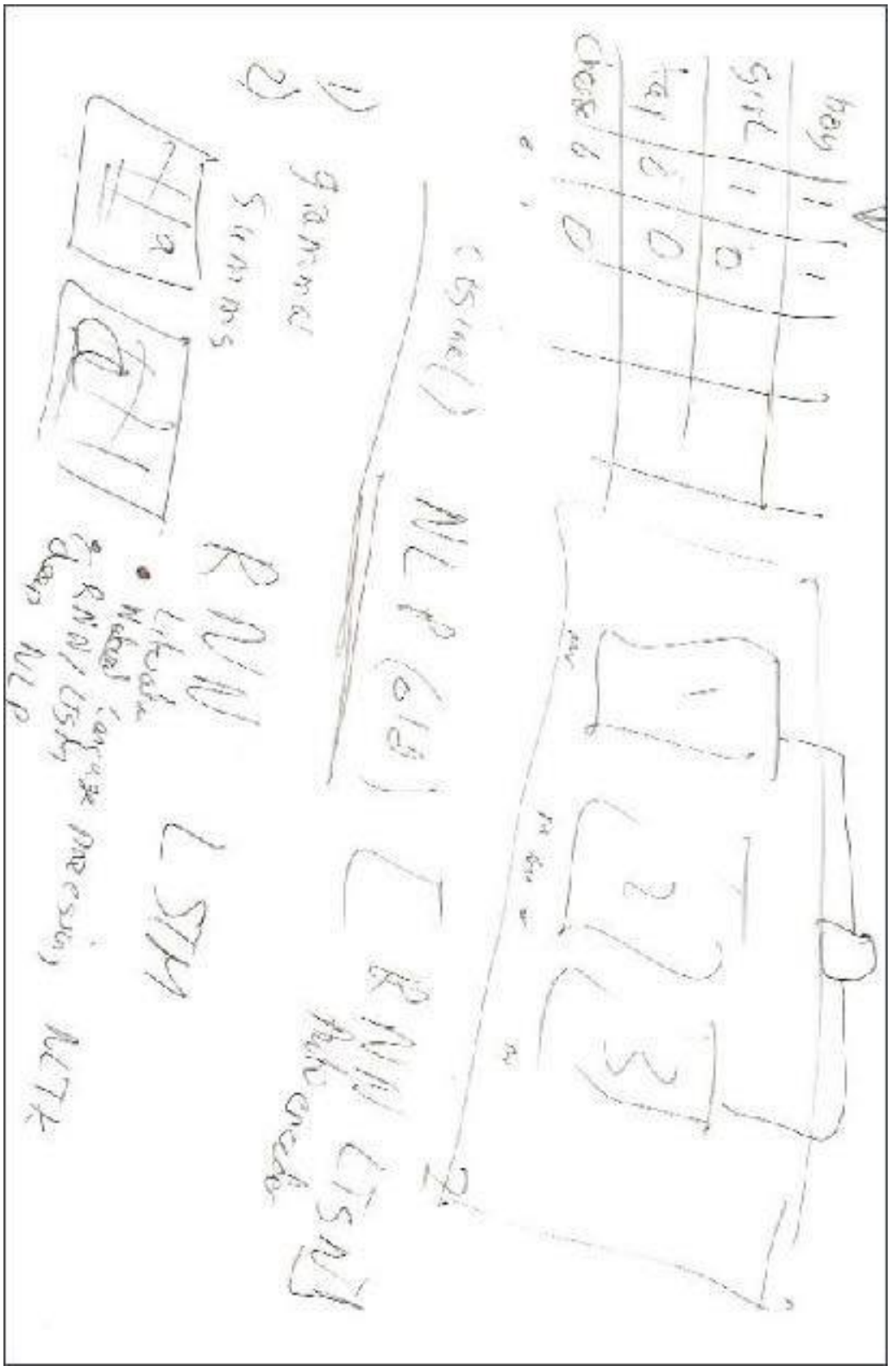
24	Optimise Final prototype	10	10/03/18	10/03/18	13/03/18	13/03/18	Prototype
	Testing	***22***					
25	Test prototype 1 recognition	2	25/10/17	25/10/17	25/10/17	25/10/17	Test report & tests
26	Test prototype 1 text output	2	25/10/17	25/10/17	25/10/17	25/10/17	Test report & tests
27	Test prototype 2 recognition	2	06/01/18	06/01/18	06/01/18	06/01/18	Test report & tests
28	Test prototype 2 text output	2	06/01/18	06/01/18	06/01/18	06/01/18	Test report & tests
29	Test prototype 2 word prediction	2	06/01/18	06/01/18	06/01/18	06/01/18	Test report & tests
30	Test prototype 3 recognition	2	02/02/18	02/02/18	02/02/18	02/02/18	Test report & tests
31	Test prototype 3 text output	2	02/02/18	02/02/18	02/02/18	02/02/18	Test report & tests
32	Test prototype 3 word prediction	2	02/02/18	02/02/18	02/02/18	02/02/18	Test report & tests
33	Test final prototype recognition	2	06/03/18	06/03/18	06/03/18	06/03/18	Test report & tests
34	Test final prototype text output	2	06/03/18	06/03/18	06/03/18	06/03/18	Test report & tests
35	Test final prototype word prediction	2	06/03/18	06/03/18	06/03/18	06/03/18	Test report & tests
	Evaluation	***50***					
36	Evaluate prototypes with client	20	29/10/17	29/10/17	13/03/18	13/03/1	Evaluation report

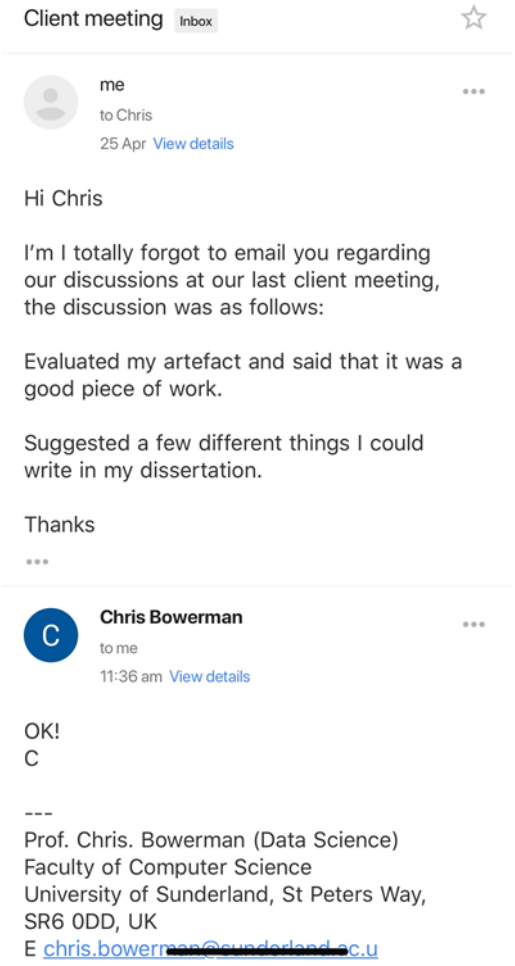
37	Evaluate system with end users	20	14/03/18	14/03/18	16/03/18	16/03/18	Evaluation report
38	Evaluate impact of literature on development	10	17/03/18	17/03/18	19/03/18	19/03/18	Evaluation report
	Documentation	***30***					
39	Draft Introduction chapter	*4*	01/02/18	01/02/18	14/02/18	14/02/18	Draft Chapter
40	Draft design chapter	*4*	15/02/18	15/02/18	28/02/18	28/02/18	Draft Chapter
41	Draft development Chapter	*4*	01/03/18	01/03/18	14/03/18	23/04/18	Draft Chapter
42	Draft evaluation and conclusion Chapter	*4*	15/03/18	15/03/18	*30/03/18*	23/04/18	Draft Chapter
43	Finalise Introduction chapter	4	15/03/18	15/03/18	15/03/18	23/04/18	Finished Chapter
44	Finalise research chapter	4	30/03/18	30/03/18	30/01/18	23/04/18	Finished Chapter
45	Finalise design chapter	4	1/04/18	1/04/18	1/04/18	23/04/18	Finished Chapter
46	Finalise development Chapter	4	5/04/18	5/04/18	5/04/18	23/04/18	Finished Chapter
47	Finalise evaluation and conclusion Chapter	4	10/04/18	10/04/18	22/04/18	23/04/18	Finished Chapter
48	Write Abstract	2	22/03/18	22/03/18	22/03/18	23/04/18	Finished Abstract

49	Generate Table of Contents, List of Figures, Chapter headings etc	1	22/03/18	22/03/18	22/03/18	22/03/18	Finished Dissertation
50	Print, bind and submit dissertation	5	05/04/17	26/04/18	16/04/18	26/04/18	Final dissertation, submitted.
51	Prepare for viva	3					Good presentation
	TOTAL HOURS	***416***					

Supervision sessions can be found on the eportfolio.

Notes from client meetings.





Client meeting Inbox

me
to Chris
25 Apr View details

Hi Chris

I'm I totally forgot to email you regarding our discussions at our last client meeting, the discussion was as follows:

Evaluated my artefact and said that it was a good piece of work.

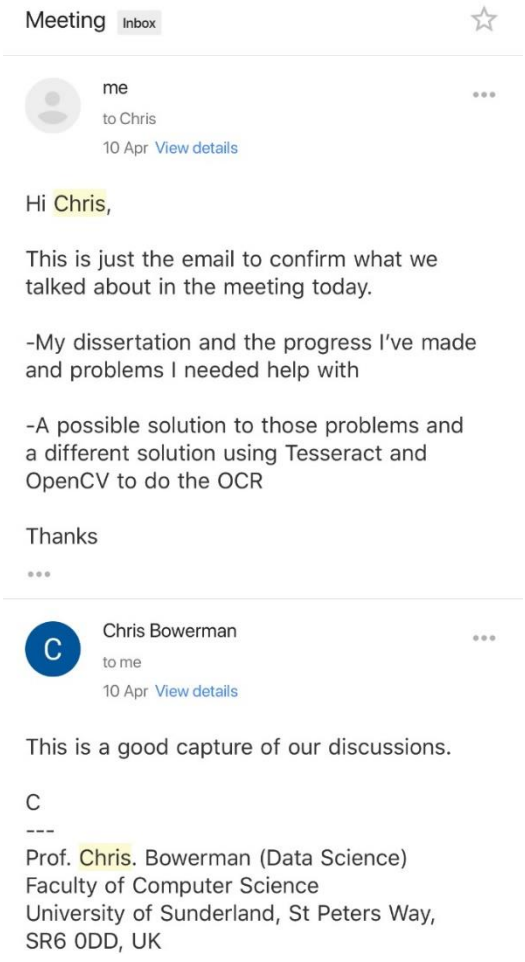
Suggested a few different things I could write in my dissertation.

Thanks

Chris Bowerman
to me
11:36 am View details

OK!
C

Prof. Chris. Bowerman (Data Science)
Faculty of Computer Science
University of Sunderland, St Peters Way,
SR6 0DD, UK
E chris.bowerman@sunderland.ac.u



Meeting Inbox

me
to Chris
10 Apr View details

Hi Chris,

This is just the email to confirm what we talked about in the meeting today.

-My dissertation and the progress I've made and problems I needed help with

-A possible solution to those problems and a different solution using Tesseract and OpenCV to do the OCR

Thanks

Chris Bowerman
to me
10 Apr View details

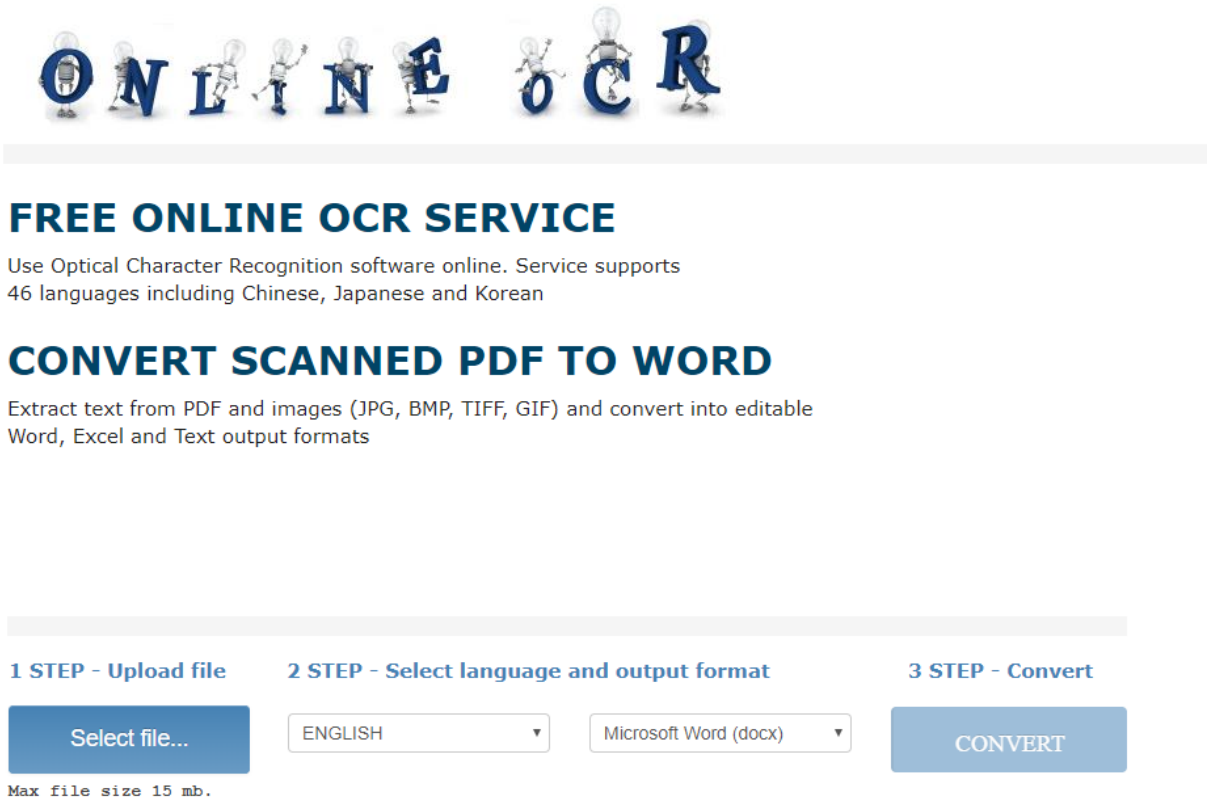
This is a good capture of our discussions.

C

Prof. Chris. Bowerman (Data Science)
Faculty of Computer Science
University of Sunderland, St Peters Way,
SR6 0DD, UK

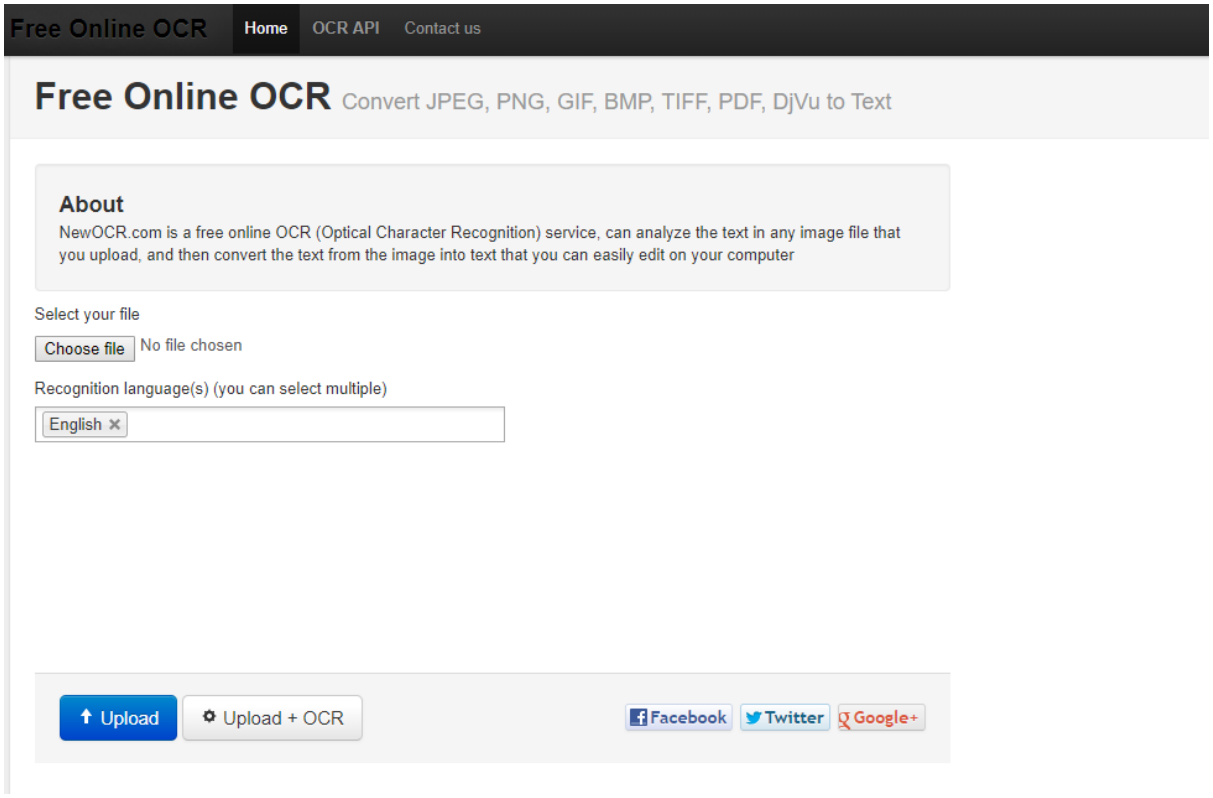
Appendix D

Website analysis



Notes: The website is very simple, with some brief instructions for use as well as a simple step by step guide at the bottom for each step.

Source: Onlineocr.net. (2017). Free Online OCR - convert PDF to Word or Image to text. [online] Available at: https://www.onlineocr.net/ [Accessed 17 Oct. 2017].

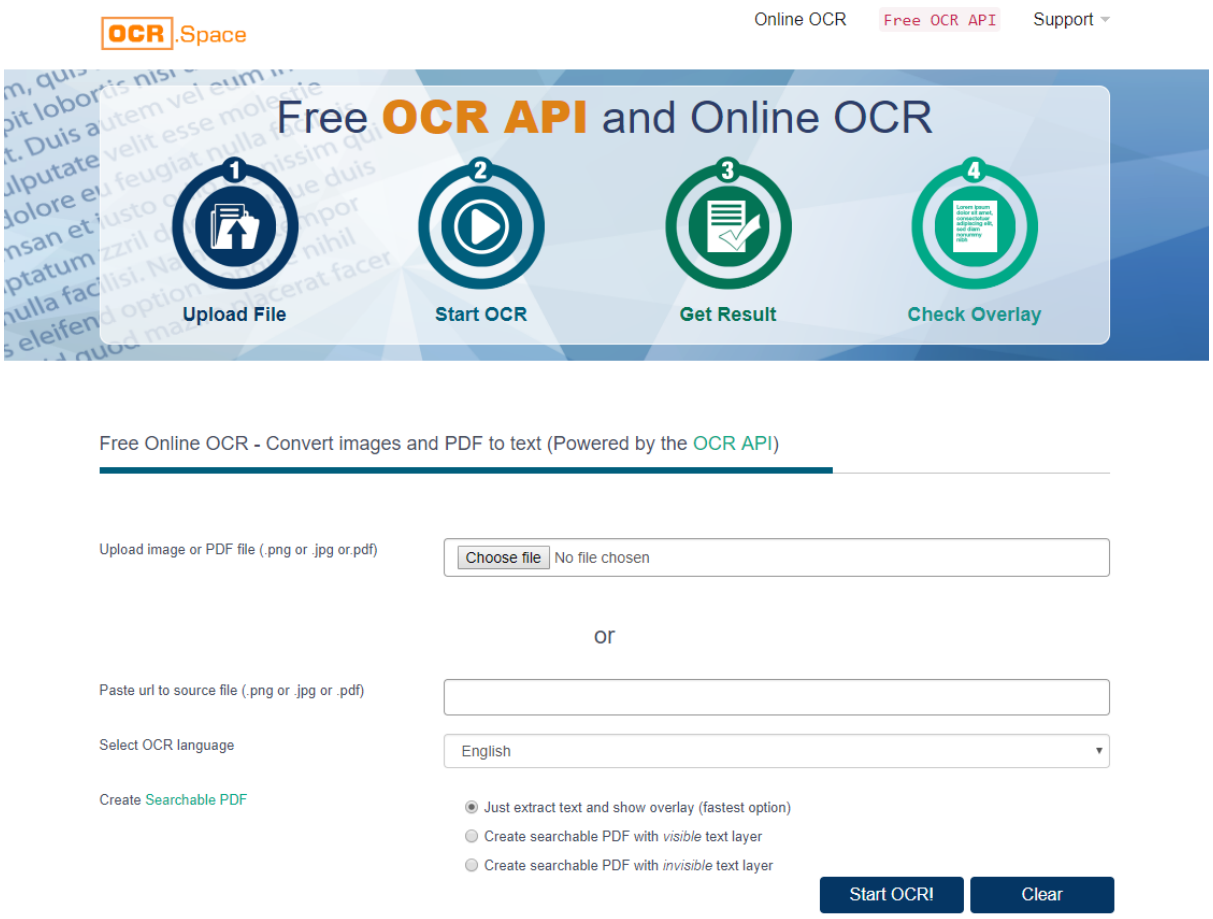


Notes: The website is very simple, just like the first however it features little in the way of instructions and is not very user friendly.

Source: Newocr.com. (2017). Free Online OCR - Convert JPEG, PNG, GIF, BMP, TIFF, PDF, DjVu to Text. [online] Available at: http://www.newocr.com/ [Accessed 17 Oct. 2017].

Notes: The website is very simple, with some detail at the bottom for the user detailing what the website is, the features and what OCR is, the process for the OCR is also laid out in a logical and user friendly manner.

Source: Free-online-ocr.com. (2017). Free Online OCR. [online] Available at: http://www.free-online-ocr.com/ [Accessed 17 Oct. 2017].



Notes: The website is very simple, with some brief instructions for use at the very top.

Source: Ocr.space. (2017). Best Free OCR API, Online OCR, Searchable PDF - Fresh 2018 OCR Software. [online] Available at: https://ocr.space/ [Accessed 17 Oct. 2017].

PACT (People, Activities, Contexts, Technologies) Summary

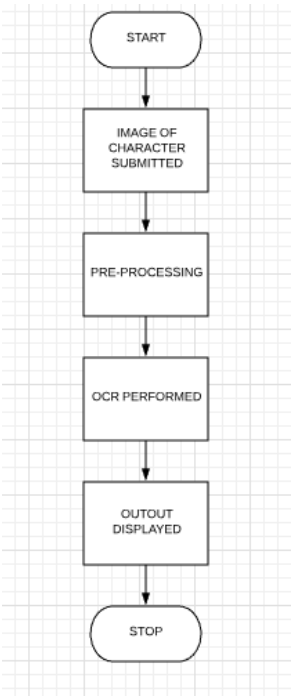
People	**PACT Summary** Students aged between 14-23. Students often take a lot of handwritten notes, especially those in secondary and college level education as they normally do not have the ability to take notes on a computer. University level students often take handwritten notes if they're doing a subject that requires a lot calculations or

	diagrams such as maths or physics so having an application which can digitise the notes for them would allow them to be more effective at studying. Due to most students being computer literate due to growing up with computers, the UI should be designed in a very common way, with common shortcuts built into it for power users to take advantage of. Overall students as a group are heterogenous, due to the variety in their physical, cognitive and usage patterns. Students have a variety of physical disabilities, just like any big group of people. These disabilities could be something such as bad eyesight or such as colour blindness. Hence why this application will use UI scaling so as to allow users with bad eye sight to enlarge the size of text without effecting the application, as well as a very plain UI so as to make it easy for colour blind users to use the application. Due to the cognitive abilities of students varying, the application will be very simple to use so that it will just require common sense to use. Working Adults aged between 18-60. Working Adults in professions such as doctors, lawyers, receptionists/assistants often make a lot of handwritten notes, therefore having an application which can digitise them for them would be a huge advantage. Due to most working adults being computer literate due to growing up with computers as well as a move to use computers over the last 18 years means that most working adults have also received computer training. As a result the UI should be designed in a very common way, with common shortcuts built into it for power users to take advantage of. Overall working adults as a group are heterogenous, due to the variety in their physical, cognitive and usage patterns. Working adults can have a variety of physical disabilities, just like any large group of people. These disabilities could be something such as bad eyesight or such as colour blindness. Hence why this application will use UI scaling so as to allow users with bad eye sight to enlarge the size of text without effecting the application, as well as a very plain UI so as to make it easy for colour blind users to use the application. Due to the cognitive abilities of students varying, the application will be very simple to use so that it will just require common sense to use.

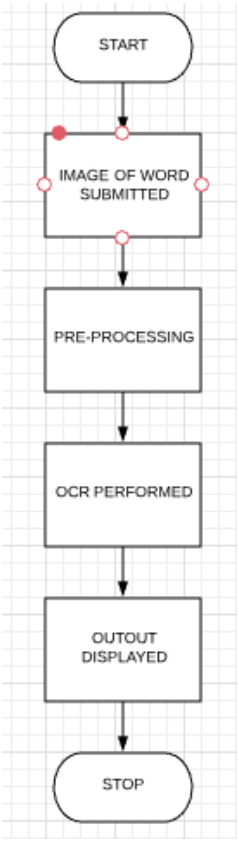
Activities	**Submit an image of notes to be digitised**. This is one part of the key functionality of the application as this is what the users will be using the application for. Each image of notes will be different so therefore the application will have to handle different image file types as well as be able to handle long handwritten documents. **Access digitised notes**. This is the other part of the key functionality of the program as the user will want to have access to their digitised notes. As a result, the application will automatically put the notes in a PDF file for the user to easily access and use for however they want.
Contexts	Both activities are performed on computer, modern smartphone, tablet and laptop. Both activities can be performed in various places which include both indoor and outdoor such as inside a classroom and outside such as outside a coffee shop, the user would have to connected to the internet whether that be through WIFI, cellular connection or ethernet. As a result image pre-processing and segmentation will be required in order to make sure that the Optical Character Recognition performed on it is as accurate as possible.
Technologies	Input **Image Upload**. When the user wishes to upload an image of their handwritten notes to have OCR performed on it. Output **File download**. The user will want to access their digitised notes and so they will have to download a PDF of them. Machine Learning **OCR.** In order to digitise the text. OCR or Optical Character Recognition will have to be applied to the text. This will be done using machine learning and deep learning. Image Processing **Image pre-processing and Segmentation.** These are essential technologies in order to get the image that is uploaded as clear as possible to reduce the error rate of the OCR as much as possible.

Appendix E

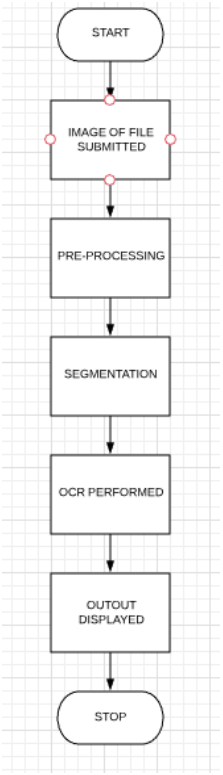
Prototype 1 program flow diagram



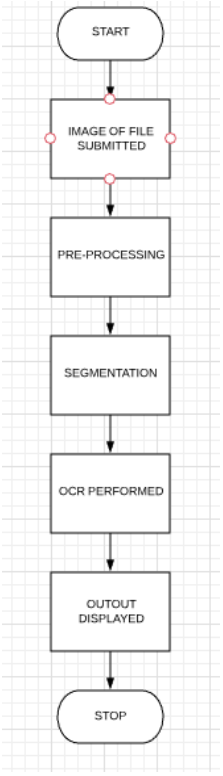
Prototype 2 program flow diagram



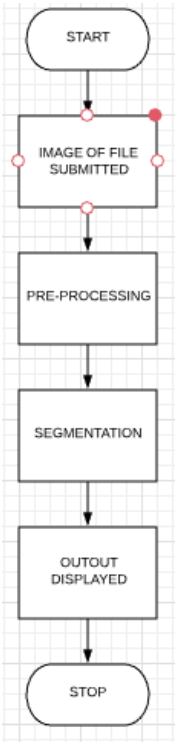
Prototype 3 program flow diagram



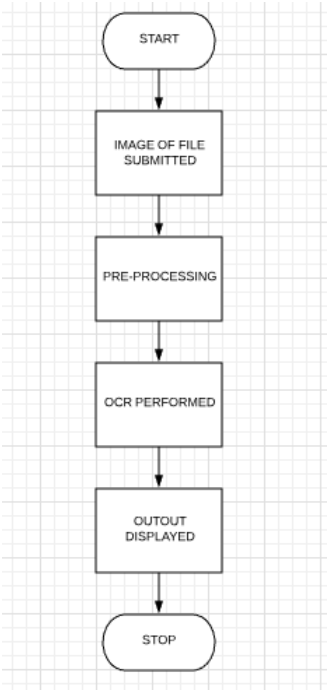
Final program flow diagram



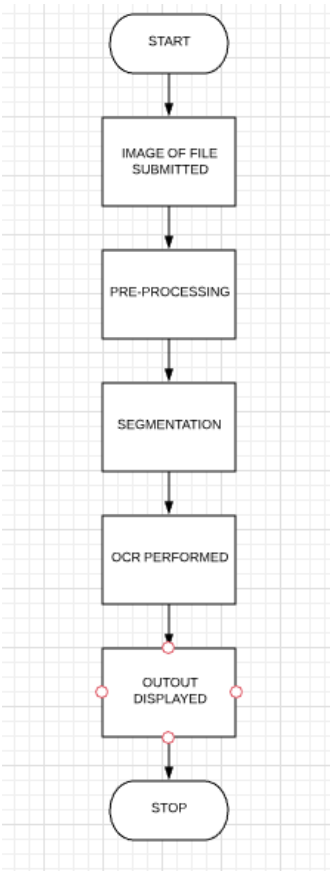
Sprint 1 program flow diagram



Sprint 2 program flow diagram



Sprint 3 program flow diagram


START
IMAGE OF FILE SUBMITTED
PRE-PROCESSING
SEGMENTATION
OCR PERFORMED
OUTOUT DISPLAYED
STOP

Appendix F

<table>
<tr><th colspan="2">SCRUM - Month October</th></tr>
<tr><td>This month's <u>reflection</u>

This month I have been working on…
Learning python, learning Keras, designing and developing prototype 1, test prototype 1 and evaluating the prototype with my client

How many hours did I spend (and on which bits)?
I spent about 23 hours on it.

What did I learn? Did I resolve any problems?
I learnt a lot about what I didn't want to do as well as a lot of ways to go about doing OCR in python and different ways to use python and Keras.</td><td>Next month's Daily Meeting statements - <u>sum up</u>:

Last month I worked on…
Learning python, learning Keras, designing and developing prototype 1, test prototype 1 and evaluating the prototype with my client which I spent 23 hours on, I learnt a lot about what I didn't want to do as well as a lot of ways to go about doing OCR in python and different ways to use python and Keras.</td></tr>
<tr><td>Next month I will work on…
Finishing off leaning Keras and prototype 1</td><td>This month I will work on
Finishing off leaning Keras and prototype 1</td></tr>
</table>

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Methods of how to do the OCR, which I will speak to my Client about.	Lack of knowledge about python and OCR.

SCRUM - Month *November*	
This month's *reflection*	**Next month's Daily Meeting statements - *sum up*:**
This month I have been working on… **Finishing off leaning Keras and prototype 1**	*Last month I worked on…* **Finishing off leaning Keras and prototype 1**
How many hours did I spend (and on which bits)? **I spent about 5 hours on it.**	
What did I learn? Did I resolve any problems? **I learnt a lot about Keras, I resolved a few problems I had with Keras**	
Next month I will work on… **Start prototype 2**	*This month I will work on* **Finishing off prototype 2**

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Nothing.	Lack of knowledge on how to do OCR.

SCRUM - Month *December*	
This month's *reflection*	**Next month's Daily Meeting statements - *sum up*:**
This month I have been working on… **Prototype 2**	*Last month I worked on…* **Prototype 2 I spent 26 hours on it and I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. I ran into problems with word prediction.**
How many hours did I spend (and on which bits)? **I spent about 26 hours on it.**	
What did I learn? Did I resolve any problems? **I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. I ran into problems with word prediction.**	
Next month I will work on… **Prototype 3**	*This month I will work on* **Prototype 3**

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Getting word prediction to work without causing instability. I need to ask my client and the internet about it.	Lack of knowledge on how to do OCR.

SCRUM - Month *January*	
This month's *reflection* *This month I have been working on…* **Prototype 3** *How many hours did I spend (and on which bits)?* **I spent about 23 hours on it.** *What did I learn? Did I resolve any problems?* **I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. Yet again I ran into problems with word prediction.**	**Next month's Daily Meeting statements - *sum up*:** *Last month I worked on…* **Prototype 3 I spent 23 hours on it and I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. I ran into problems with word prediction again.**
Next month I will work on… **Prototype 3**	*This month I will work on* **Prototype 3**

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Getting word prediction to work without causing instability. I need to ask my client and the internet about it.	Lack of knowledge on how to do OCR.

SCRUM - Month *February*	
This month's *reflection*	**Next month's Daily Meeting statements - *sum up*:**
This month I have been working on… **Prototype 3**	*Last month I worked on…* **Prototype 3 I spent 6 hours on it and I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. I ran into problems with word prediction again.**
How many hours did I spend (and on which bits)? **I spent about 6 hours on it.**	
What did I learn? Did I resolve any problems? **I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. Yet again I ran into problems with word prediction.**	
Next month I will work on… **The final program**	*This month I will work on* **The finals program**

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Getting word prediction to work without causing instability. I need to ask my client and the internet about it.	Lack of knowledge on how to do OCR.

SCRUM - Month *March*	
This month's *reflection* *This month I have been working on…* **The final program** *How many hours did I spend (and on which bits)?* **I spent about 26 hours on it.** *What did I learn? Did I resolve any problems?* **I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. As well as how to create a web UI for python.**	**Next month's Daily Meeting statements - *sum up*:** *Last month I worked on…* **The final program I spent 26 hours on it and I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. As well as how to create a web UI for python.**
Next month I will work on… **The final program**	*This month I will work on* **The final program**

What's bothering me/what do I need to ask - and who do I need to ask this question?	*What might prevent me from making further progress?*
Nothing.	Lack of knowledge on how to do OCR.

SCRUM - Month *April*	
This month's *reflection*	**Next month's Daily Meeting statements - *sum up*:**
This month I have been working on… **The final program, Sprint 1, 2 and 3's programs** *How many hours did I spend (and on which bits)?* **I spent about 40 hours on it.** *What did I learn? Did I resolve any problems?* **I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. How to create a web UI for python and how to use and train PyTesseract**	*Last month I worked on…* **The final program, Sprint 1, 2 and 3's programs I spent 40 hours on it and I learnt a lot of about OpenCV and how to implement pre-processing techniques in Python. How to create a web UI for python and how to use and train PyTesseract**
Next month I will work on… **Nothing**	*This month I will work on* **Nothing**

What'sbothering me/whatdoIneedtoask -andwho do Ineed to ask this question?	*Whatmightpreventmefrom makingfurtherprogress?*
Nothing.	Lack of time.

Appendix G

Prototype 1 Testing readout

```
/Users/Tom/anaconda2/envs/dissertation/bin/python /Users/Tom/PycharmProjects/Dissertation/Prototype1.py
/Users/Tom/anaconda2/envs/dissertation/lib/python3.5/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
/Users/Tom/anaconda2/envs/dissertation/lib/python3.5/importlib/_bootstrap.py:222: RuntimeWarning: compiletime version 3.6 of module 'tensorflow.python.framework.fast_tensor_util' does not match runtime version 3.5
  return f(*args, **kwds)
2018-04-26 11:18:02.805123: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
Train on 60000 samples, validate on 10000 samples
Epoch 1/1

  100/60000 [..............................] - ETA: 3:41 - loss: 0.0395 - acc: 0.9800
  200/60000 [..............................] - ETA: 2:49 - loss: 0.0199 - acc: 0.9900
  300/60000 [..............................] - ETA: 2:33 - loss: 0.0145 - acc: 0.9933
  400/60000 [..............................] - ETA: 2:24 - loss: 0.0109 - acc: 0.9950
  500/60000 [..............................] - ETA: 2:19 - loss: 0.0088 - acc: 0.9960
  600/60000 [..............................] - ETA: 2:15 - loss: 0.0074 - acc: 0.9967
  700/60000 [..............................] - ETA: 2:13 - loss: 0.0063 - acc: 0.9971
  800/60000 [..............................] - ETA: 2:11 - loss: 0.0058 - acc: 0.9975
  900/60000 [..............................] - ETA: 2:09 - loss: 0.0064 - acc: 0.9967
 1000/60000 [..............................] - ETA: 2:08 - loss: 0.0074 - acc: 0.9960
 1100/60000 [..............................] - ETA: 2:07 - loss: 0.0067 - acc: 0.9964
 1200/60000 [..............................] - ETA: 2:06 - loss: 0.0062 - acc: 0.9967
 1300/60000 [..............................] - ETA: 2:05 - loss: 0.0104 - acc: 0.9954
 1400/60000 [..............................] - ETA: 2:04 - loss: 0.0097 - acc: 0.9957
 1500/60000 [..............................] - ETA: 2:03 - loss: 0.0093 - acc: 0.9960
 1600/60000 [..............................] - ETA: 2:03 - loss: 0.0087 - acc: 0.9963
 1700/60000 [..............................] - ETA: 2:02 - loss: 0.0085 - acc: 0.9965
 1800/60000 [..............................] - ETA: 2:02 - loss: 0.0081 - acc: 0.9967
 1900/60000 [..............................] - ETA: 2:01 - loss: 0.0077 - acc: 0.9968
 2000/60000 [>.............................] - ETA: 2:01 - loss: 0.0074 - acc: 0.9970
 2100/60000 [>.............................] - ETA: 2:00 - loss: 0.0074 - acc: 0.9971
 2200/60000 [>.............................] - ETA: 2:00 - loss: 0.0071 - acc: 0.9973
 2300/60000 [>.............................] - ETA: 2:00 - loss: 0.0074 - acc: 0.9970
 2400/60000 [>.............................] - ETA: 1:59 - loss: 0.0071 - acc: 0.9971
 2500/60000 [>.............................] - ETA: 1:59 - loss: 0.0069 - acc: 0.9972
 2600/60000 [>.............................] - ETA: 1:59 - loss: 0.0066 - acc: 0.9973
 2700/60000 [>.............................] - ETA: 1:58 - loss: 0.0068 - acc: 0.9970
 2800/60000 [>.............................] - ETA: 1:58 - loss: 0.0075 - acc: 0.9964
 2900/60000 [>.............................] - ETA: 1:58 - loss: 0.0077 - acc: 0.9962
 3000/60000 [>.............................] - ETA: 1:57 - loss: 0.0075 - acc: 0.9963
 3100/60000 [>.............................] - ETA: 1:57 - loss: 0.0074 - acc: 0.9965
 3200/60000 [>.............................] - ETA: 1:57 - loss: 0.0073 - acc: 0.9966
 3300/60000 [>.............................] - ETA: 1:56 - loss: 0.0071 - acc: 0.9967
 3400/60000 [>.............................] - ETA: 1:56 - loss: 0.0072 - acc: 0.9965
 3500/60000 [>.............................] - ETA: 1:56 - loss: 0.0073 - acc: 0.9963
 3600/60000 [>.............................] - ETA: 1:55 - loss: 0.0073 - acc: 0.9964
 3700/60000 [>.............................] - ETA: 1:55 - loss: 0.0080 - acc: 0.9962
 3800/60000 [>.............................] - ETA: 1:55 - loss: 0.0078 - acc: 0.9963
 3900/60000 [>.............................] - ETA: 1:55 - loss: 0.0077 - acc: 0.9964
 4000/60000 [=>............................] - ETA: 1:54 - loss: 0.0077 - acc: 0.9965
 4100/60000 [=>............................] - ETA: 1:54 - loss: 0.0075 - acc: 0.9966
 4200/60000 [=>............................] - ETA: 1:54 - loss: 0.0074 - acc: 0.9967
 4300/60000 [=>............................] - ETA: 1:54 - loss: 0.0073 - acc: 0.9967
 4400/60000 [=>............................] - ETA: 1:53 - loss: 0.0072 - acc: 0.9968
 4500/60000 [=>............................] - ETA: 1:53 - loss: 0.0070 - acc: 0.9969
 4600/60000 [=>............................] - ETA: 1:53 - loss: 0.0073 - acc: 0.9967
```

```
 4700/60000 [=>.....] - ETA: 1:52 - loss: 0.0073 - acc: 0.9966
 4800/60000 [=>.....] - ETA: 1:52 - loss: 0.0076 - acc: 0.9965
 4900/60000 [=>.....] - ETA: 1:52 - loss: 0.0076 - acc: 0.9965
 5000/60000 [=>.....] - ETA: 1:52 - loss: 0.0083 - acc: 0.9960
 5100/60000 [=>.....] - ETA: 1:52 - loss: 0.0087 - acc: 0.9959
 5200/60000 [=>.....] - ETA: 1:51 - loss: 0.0086 - acc: 0.9960
 5300/60000 [=>.....] - ETA: 1:51 - loss: 0.0085 - acc: 0.9960
 5400/60000 [=>.....] - ETA: 1:51 - loss: 0.0085 - acc: 0.9961
 5500/60000 [=>.....] - ETA: 1:51 - loss: 0.0086 - acc: 0.9960
 5600/60000 [=>.....] - ETA: 1:50 - loss: 0.0085 - acc: 0.9961
 5700/60000 [=>.....] - ETA: 1:50 - loss: 0.0087 - acc: 0.9960
 5800/60000 [=>.....] - ETA: 1:50 - loss: 0.0088 - acc: 0.9959
 5900/60000 [=>.....] - ETA: 1:50 - loss: 0.0098 - acc: 0.9956
 6000/60000 [==>.....] - ETA: 1:49 - loss: 0.0097 - acc: 0.9957
 6100/60000 [==>.....] - ETA: 1:49 - loss: 0.0096 - acc: 0.9957
 6200/60000 [==>.....] - ETA: 1:49 - loss: 0.0094 - acc: 0.9958
 6300/60000 [==>.....] - ETA: 1:49 - loss: 0.0095 - acc: 0.9959
 6400/60000 [==>.....] - ETA: 1:48 - loss: 0.0098 - acc: 0.9958
 6500/60000 [==>.....] - ETA: 1:48 - loss: 0.0097 - acc: 0.9958
 6600/60000 [==>.....] - ETA: 1:48 - loss: 0.0095 - acc: 0.9959
 6700/60000 [==>.....] - ETA: 1:48 - loss: 0.0097 - acc: 0.9958
 6800/60000 [==>.....] - ETA: 1:48 - loss: 0.0097 - acc: 0.9957
 6900/60000 [==>.....] - ETA: 1:47 - loss: 0.0097 - acc: 0.9958
 7000/60000 [==>.....] - ETA: 1:47 - loss: 0.0097 - acc: 0.9957
 7100/60000 [==>.....] - ETA: 1:47 - loss: 0.0103 - acc: 0.9955
 7200/60000 [==>.....] - ETA: 1:47 - loss: 0.0109 - acc: 0.9954
 7300/60000 [==>.....] - ETA: 1:47 - loss: 0.0115 - acc: 0.9953
 7400/60000 [==>.....] - ETA: 1:46 - loss: 0.0118 - acc: 0.9951
 7500/60000 [==>.....] - ETA: 1:46 - loss: 0.0116 - acc: 0.9952
 7600/60000 [==>.....] - ETA: 1:46 - loss: 0.0115 - acc: 0.9953
 7700/60000 [==>.....] - ETA: 1:46 - loss: 0.0114 - acc: 0.9953
 7800/60000 [==>.....] - ETA: 1:45 - loss: 0.0116 - acc: 0.9953
 7900/60000 [==>.....] - ETA: 1:45 - loss: 0.0114 - acc: 0.9953
 8000/60000 [===>.....] - ETA: 1:45 - loss: 0.0113 - acc: 0.9954
 8100/60000 [===>.....] - ETA: 1:45 - loss: 0.0112 - acc: 0.9954
 8200/60000 [===>.....] - ETA: 1:45 - loss: 0.0114 - acc: 0.9952
 8300/60000 [===>.....] - ETA: 1:44 - loss: 0.0117 - acc: 0.9952
 8400/60000 [===>.....] - ETA: 1:44 - loss: 0.0118 - acc: 0.9951
 8500/60000 [===>.....] - ETA: 1:44 - loss: 0.0120 - acc: 0.9951
 8600/60000 [===>.....] - ETA: 1:44 - loss: 0.0120 - acc: 0.9951
 8700/60000 [===>.....] - ETA: 1:44 - loss: 0.0121 - acc: 0.9951
 8800/60000 [===>.....] - ETA: 1:43 - loss: 0.0121 - acc: 0.9950
 8900/60000 [===>.....] - ETA: 1:43 - loss: 0.0121 - acc: 0.9951
 9000/60000 [===>.....] - ETA: 1:43 - loss: 0.0119 - acc: 0.9951
 9100/60000 [===>.....] - ETA: 1:43 - loss: 0.0123 - acc: 0.9948
 9200/60000 [===>.....] - ETA: 1:42 - loss: 0.0124 - acc: 0.9948
 9300/60000 [===>.....] - ETA: 1:42 - loss: 0.0123 - acc: 0.9948
 9400/60000 [===>.....] - ETA: 1:42 - loss: 0.0122 - acc: 0.9949
 9500/60000 [===>.....] - ETA: 1:42 - loss: 0.0123 - acc: 0.9948
 9600/60000 [===>.....] - ETA: 1:42 - loss: 0.0128 - acc: 0.9947
 9700/60000 [===>.....] - ETA: 1:41 - loss: 0.0127 - acc: 0.9947
 9800/60000 [===>.....] - ETA: 1:41 - loss: 0.0132 - acc: 0.9947
 9900/60000 [===>.....] - ETA: 1:41 - loss: 0.0135 - acc: 0.9946
10000/60000 [====>.....] - ETA: 1:41 - loss: 0.0134 - acc: 0.9947
10100/60000 [====>.....] - ETA: 1:40 - loss: 0.0134 - acc: 0.9948
10200/60000 [====>.....] - ETA: 1:40 - loss: 0.0132 - acc: 0.9948
10300/60000 [====>.....] - ETA: 1:40 - loss: 0.0132 - acc: 0.9949
10400/60000 [====>.....] - ETA: 1:40 - loss: 0.0131 - acc: 0.9949
10500/60000 [====>.....] - ETA: 1:40 - loss: 0.0130 - acc: 0.9950
10600/60000 [====>.....] - ETA: 1:39 - loss: 0.0129 - acc: 0.9950
10700/60000 [====>.....] - ETA: 1:39 - loss: 0.0128 - acc: 0.9950
10800/60000 [====>.....] - ETA: 1:39 - loss: 0.0127 - acc: 0.9951
10900/60000 [====>.....] - ETA: 1:39 - loss: 0.0129 - acc: 0.9950
11000/60000 [====>.....] - ETA: 1:39 - loss: 0.0128 - acc: 0.9951
11100/60000 [====>.....] - ETA: 1:38 - loss: 0.0127 - acc: 0.9951
```

```
11200/60000 [====>.....] - ETA: 1:38 - loss: 0.0126 - acc: 0.9952
11300/60000 [====>.....] - ETA: 1:38 - loss: 0.0129 - acc: 0.9951
11400/60000 [====>.....] - ETA: 1:38 - loss: 0.0135 - acc: 0.9950
11500/60000 [====>.....] - ETA: 1:38 - loss: 0.0133 - acc: 0.9950
11600/60000 [====>.....] - ETA: 1:37 - loss: 0.0132 - acc: 0.9951
11700/60000 [====>.....] - ETA: 1:37 - loss: 0.0131 - acc: 0.9951
11800/60000 [====>.....] - ETA: 1:37 - loss: 0.0132 - acc: 0.9951
11900/60000 [====>.....] - ETA: 1:37 - loss: 0.0131 - acc: 0.9950
12000/60000 [=====>.....] - ETA: 1:36 - loss: 0.0130 - acc: 0.9951
12100/60000 [=====>.....] - ETA: 1:36 - loss: 0.0129 - acc: 0.9951
12200/60000 [=====>.....] - ETA: 1:36 - loss: 0.0128 - acc: 0.9952
12300/60000 [=====>.....] - ETA: 1:36 - loss: 0.0128 - acc: 0.9952
12400/60000 [=====>.....] - ETA: 1:36 - loss: 0.0127 - acc: 0.9952
12500/60000 [=====>.....] - ETA: 1:35 - loss: 0.0126 - acc: 0.9953
12600/60000 [=====>.....] - ETA: 1:35 - loss: 0.0125 - acc: 0.9953
12700/60000 [=====>.....] - ETA: 1:35 - loss: 0.0125 - acc: 0.9953
12800/60000 [=====>.....] - ETA: 1:35 - loss: 0.0125 - acc: 0.9953
12900/60000 [=====>.....] - ETA: 1:35 - loss: 0.0124 - acc: 0.9953
13000/60000 [=====>.....] - ETA: 1:34 - loss: 0.0124 - acc: 0.9953
13100/60000 [=====>.....] - ETA: 1:34 - loss: 0.0123 - acc: 0.9953
13200/60000 [=====>.....] - ETA: 1:34 - loss: 0.0122 - acc: 0.9954
13300/60000 [=====>.....] - ETA: 1:34 - loss: 0.0122 - acc: 0.9954
13400/60000 [=====>.....] - ETA: 1:34 - loss: 0.0122 - acc: 0.9954
13500/60000 [=====>.....] - ETA: 1:33 - loss: 0.0121 - acc: 0.9955
13600/60000 [=====>.....] - ETA: 1:33 - loss: 0.0120 - acc: 0.9955
13700/60000 [=====>.....] - ETA: 1:33 - loss: 0.0120 - acc: 0.9955
13800/60000 [=====>.....] - ETA: 1:33 - loss: 0.0119 - acc: 0.9955
13900/60000 [=====>.....] - ETA: 1:33 - loss: 0.0119 - acc: 0.9955
14000/60000 [======>.....] - ETA: 1:32 - loss: 0.0118 - acc: 0.9955
14100/60000 [======>.....] - ETA: 1:32 - loss: 0.0118 - acc: 0.9955
14200/60000 [======>.....] - ETA: 1:32 - loss: 0.0117 - acc: 0.9956
14300/60000 [======>.....] - ETA: 1:32 - loss: 0.0116 - acc: 0.9956
14400/60000 [======>.....] - ETA: 1:32 - loss: 0.0115 - acc: 0.9956
14500/60000 [======>.....] - ETA: 1:31 - loss: 0.0115 - acc: 0.9957
14600/60000 [======>.....] - ETA: 1:31 - loss: 0.0114 - acc: 0.9957
14700/60000 [======>.....] - ETA: 1:31 - loss: 0.0115 - acc: 0.9956
14800/60000 [======>.....] - ETA: 1:31 - loss: 0.0114 - acc: 0.9957
14900/60000 [======>.....] - ETA: 1:30 - loss: 0.0113 - acc: 0.9957
15000/60000 [======>.....] - ETA: 1:30 - loss: 0.0113 - acc: 0.9957
15100/60000 [======>.....] - ETA: 1:30 - loss: 0.0112 - acc: 0.9958
15200/60000 [======>.....] - ETA: 1:30 - loss: 0.0112 - acc: 0.9958
15300/60000 [======>.....] - ETA: 1:30 - loss: 0.0113 - acc: 0.9957
15400/60000 [======>.....] - ETA: 1:30 - loss: 0.0112 - acc: 0.9957
15500/60000 [======>.....] - ETA: 1:30 - loss: 0.0112 - acc: 0.9957
15600/60000 [======>.....] - ETA: 1:29 - loss: 0.0112 - acc: 0.9957
15700/60000 [======>.....] - ETA: 1:29 - loss: 0.0111 - acc: 0.9957
15800/60000 [======>.....] - ETA: 1:29 - loss: 0.0111 - acc: 0.9958
15900/60000 [======>.....] - ETA: 1:29 - loss: 0.0110 - acc: 0.9958
16000/60000 [=======>.....] - ETA: 1:29 - loss: 0.0111 - acc: 0.9958
16100/60000 [=======>.....] - ETA: 1:29 - loss: 0.0110 - acc: 0.9958
16200/60000 [=======>.....] - ETA: 1:29 - loss: 0.0109 - acc: 0.9958
16300/60000 [=======>.....] - ETA: 1:29 - loss: 0.0109 - acc: 0.9958
16400/60000 [=======>.....] - ETA: 1:28 - loss: 0.0109 - acc: 0.9959
16500/60000 [=======>.....] - ETA: 1:28 - loss: 0.0108 - acc: 0.9959
16600/60000 [=======>.....] - ETA: 1:28 - loss: 0.0107 - acc: 0.9959
16700/60000 [=======>.....] - ETA: 1:28 - loss: 0.0107 - acc: 0.9959
16800/60000 [=======>.....] - ETA: 1:28 - loss: 0.0107 - acc: 0.9959
16900/60000 [=======>.....] - ETA: 1:28 - loss: 0.0106 - acc: 0.9959
17000/60000 [=======>.....] - ETA: 1:28 - loss: 0.0106 - acc: 0.9959
17100/60000 [=======>.....] - ETA: 1:27 - loss: 0.0108 - acc: 0.9958
17200/60000 [=======>.....] - ETA: 1:27 - loss: 0.0107 - acc: 0.9959
17300/60000 [=======>.....] - ETA: 1:27 - loss: 0.0107 - acc: 0.9959
17400/60000 [=======>.....] - ETA: 1:27 - loss: 0.0106 - acc: 0.9959
17500/60000 [=======>.....] - ETA: 1:27 - loss: 0.0105 - acc: 0.9959
17600/60000 [=======>.....] - ETA: 1:27 - loss: 0.0105 - acc: 0.9960
```

17700/60000 [=======>.....] - ETA: 1:27 - loss: 0.0105 - acc: 0.9960
17800/60000 [=======>.....] - ETA: 1:26 - loss: 0.0105 - acc: 0.9960
17900/60000 [=======>.....] - ETA: 1:26 - loss: 0.0106 - acc: 0.9959
18000/60000 [========>.....] - ETA: 1:26 - loss: 0.0105 - acc: 0.9959
18100/60000 [========>.....] - ETA: 1:26 - loss: 0.0105 - acc: 0.9960
18200/60000 [========>.....] - ETA: 1:26 - loss: 0.0105 - acc: 0.9959
18300/60000 [========>.....] - ETA: 1:25 - loss: 0.0105 - acc: 0.9960
18400/60000 [========>.....] - ETA: 1:25 - loss: 0.0105 - acc: 0.9959
18500/60000 [========>.....] - ETA: 1:25 - loss: 0.0105 - acc: 0.9959
18600/60000 [========>.....] - ETA: 1:25 - loss: 0.0105 - acc: 0.9959
18700/60000 [========>.....] - ETA: 1:25 - loss: 0.0104 - acc: 0.9959
18800/60000 [========>.....] - ETA: 1:24 - loss: 0.0104 - acc: 0.9960
18900/60000 [========>.....] - ETA: 1:24 - loss: 0.0103 - acc: 0.9960
19000/60000 [========>.....] - ETA: 1:24 - loss: 0.0104 - acc: 0.9959
19100/60000 [========>.....] - ETA: 1:24 - loss: 0.0104 - acc: 0.9959
19200/60000 [========>.....] - ETA: 1:24 - loss: 0.0106 - acc: 0.9959
19300/60000 [========>.....] - ETA: 1:24 - loss: 0.0106 - acc: 0.9959
19400/60000 [========>.....] - ETA: 1:24 - loss: 0.0105 - acc: 0.9959
19500/60000 [========>.....] - ETA: 1:24 - loss: 0.0105 - acc: 0.9959
19600/60000 [========>.....] - ETA: 1:24 - loss: 0.0104 - acc: 0.9960
19700/60000 [========>.....] - ETA: 1:24 - loss: 0.0104 - acc: 0.9959
19800/60000 [========>.....] - ETA: 1:23 - loss: 0.0104 - acc: 0.9960
19900/60000 [========>.....] - ETA: 1:23 - loss: 0.0103 - acc: 0.9960
20000/60000 [=========>.....] - ETA: 1:23 - loss: 0.0103 - acc: 0.9960
20100/60000 [=========>.....] - ETA: 1:23 - loss: 0.0105 - acc: 0.9960
20200/60000 [=========>.....] - ETA: 1:23 - loss: 0.0105 - acc: 0.9959
20300/60000 [=========>.....] - ETA: 1:22 - loss: 0.0105 - acc: 0.9959
20400/60000 [=========>.....] - ETA: 1:22 - loss: 0.0105 - acc: 0.9959
20500/60000 [=========>.....] - ETA: 1:22 - loss: 0.0105 - acc: 0.9960
20600/60000 [=========>.....] - ETA: 1:22 - loss: 0.0105 - acc: 0.9959
20700/60000 [=========>.....] - ETA: 1:22 - loss: 0.0105 - acc: 0.9959
20800/60000 [=========>.....] - ETA: 1:22 - loss: 0.0104 - acc: 0.9960
20900/60000 [=========>.....] - ETA: 1:22 - loss: 0.0104 - acc: 0.9960
21000/60000 [=========>.....] - ETA: 1:22 - loss: 0.0104 - acc: 0.9960
21100/60000 [=========>.....] - ETA: 1:21 - loss: 0.0104 - acc: 0.9960
21200/60000 [=========>.....] - ETA: 1:21 - loss: 0.0105 - acc: 0.9958
21300/60000 [=========>.....] - ETA: 1:21 - loss: 0.0106 - acc: 0.9958
21400/60000 [=========>.....] - ETA: 1:21 - loss: 0.0106 - acc: 0.9958
21500/60000 [=========>.....] - ETA: 1:21 - loss: 0.0106 - acc: 0.9959
21600/60000 [=========>.....] - ETA: 1:21 - loss: 0.0106 - acc: 0.9959
21700/60000 [=========>.....] - ETA: 1:21 - loss: 0.0107 - acc: 0.9958
21800/60000 [=========>.....] - ETA: 1:20 - loss: 0.0107 - acc: 0.9958
21900/60000 [=========>.....] - ETA: 1:20 - loss: 0.0107 - acc: 0.9958
22000/60000 [==========>.....] - ETA: 1:20 - loss: 0.0107 - acc: 0.9958
22100/60000 [==========>.....] - ETA: 1:20 - loss: 0.0107 - acc: 0.9958
22200/60000 [==========>.....] - ETA: 1:20 - loss: 0.0107 - acc: 0.9958
22300/60000 [==========>.....] - ETA: 1:20 - loss: 0.0106 - acc: 0.9958
22400/60000 [==========>.....] - ETA: 1:19 - loss: 0.0106 - acc: 0.9958
22500/60000 [==========>.....] - ETA: 1:19 - loss: 0.0106 - acc: 0.9958
22600/60000 [==========>.....] - ETA: 1:19 - loss: 0.0106 - acc: 0.9958
22700/60000 [==========>.....] - ETA: 1:19 - loss: 0.0105 - acc: 0.9959
22800/60000 [==========>.....] - ETA: 1:19 - loss: 0.0106 - acc: 0.9958
22900/60000 [==========>.....] - ETA: 1:18 - loss: 0.0106 - acc: 0.9958
23000/60000 [==========>.....] - ETA: 1:18 - loss: 0.0105 - acc: 0.9958
23100/60000 [==========>.....] - ETA: 1:18 - loss: 0.0105 - acc: 0.9958
23200/60000 [==========>.....] - ETA: 1:18 - loss: 0.0105 - acc: 0.9959
23300/60000 [==========>.....] - ETA: 1:18 - loss: 0.0105 - acc: 0.9958
23400/60000 [==========>.....] - ETA: 1:17 - loss: 0.0105 - acc: 0.9958
23500/60000 [==========>.....] - ETA: 1:17 - loss: 0.0105 - acc: 0.9958
23600/60000 [==========>.....] - ETA: 1:17 - loss: 0.0106 - acc: 0.9958
23700/60000 [==========>.....] - ETA: 1:17 - loss: 0.0105 - acc: 0.9958
23800/60000 [==========>.....] - ETA: 1:17 - loss: 0.0105 - acc: 0.9958
23900/60000 [==========>.....] - ETA: 1:16 - loss: 0.0105 - acc: 0.9958
24000/60000 [===========>.....] - ETA: 1:16 - loss: 0.0104 - acc: 0.9958
24100/60000 [===========>.....] - ETA: 1:16 - loss: 0.0105 - acc: 0.9958

```
24200/60000 [===========>..................] - ETA: 1:16 - loss: 0.0106 - acc: 0.9958
24300/60000 [===========>..................] - ETA: 1:15 - loss: 0.0106 - acc: 0.9958
24400/60000 [===========>..................] - ETA: 1:15 - loss: 0.0106 - acc: 0.9958
24500/60000 [===========>..................] - ETA: 1:15 - loss: 0.0106 - acc: 0.9958
24600/60000 [===========>..................] - ETA: 1:15 - loss: 0.0105 - acc: 0.9958
24700/60000 [===========>..................] - ETA: 1:15 - loss: 0.0105 - acc: 0.9958
24800/60000 [===========>..................] - ETA: 1:14 - loss: 0.0104 - acc: 0.9958
24900/60000 [===========>..................] - ETA: 1:14 - loss: 0.0104 - acc: 0.9959
25000/60000 [===========>..................] - ETA: 1:14 - loss: 0.0104 - acc: 0.9959
25100/60000 [===========>..................] - ETA: 1:14 - loss: 0.0103 - acc: 0.9959
25200/60000 [===========>..................] - ETA: 1:14 - loss: 0.0103 - acc: 0.9959
25300/60000 [===========>..................] - ETA: 1:13 - loss: 0.0107 - acc: 0.9959
25400/60000 [===========>..................] - ETA: 1:13 - loss: 0.0108 - acc: 0.9959
25500/60000 [===========>..................] - ETA: 1:13 - loss: 0.0108 - acc: 0.9959
25600/60000 [===========>..................] - ETA: 1:13 - loss: 0.0107 - acc: 0.9959
25700/60000 [===========>..................] - ETA: 1:12 - loss: 0.0107 - acc: 0.9959
25800/60000 [===========>..................] - ETA: 1:12 - loss: 0.0107 - acc: 0.9959
25900/60000 [===========>..................] - ETA: 1:12 - loss: 0.0107 - acc: 0.9959
26000/60000 [============>.................] - ETA: 1:12 - loss: 0.0107 - acc: 0.9959
26100/60000 [============>.................] - ETA: 1:12 - loss: 0.0107 - acc: 0.9959
26200/60000 [============>.................] - ETA: 1:11 - loss: 0.0107 - acc: 0.9959
26300/60000 [============>.................] - ETA: 1:11 - loss: 0.0106 - acc: 0.9959
26400/60000 [============>.................] - ETA: 1:11 - loss: 0.0106 - acc: 0.9959
26500/60000 [============>.................] - ETA: 1:11 - loss: 0.0106 - acc: 0.9960
26600/60000 [============>.................] - ETA: 1:10 - loss: 0.0105 - acc: 0.9960
26700/60000 [============>.................] - ETA: 1:10 - loss: 0.0105 - acc: 0.9960
26800/60000 [============>.................] - ETA: 1:10 - loss: 0.0105 - acc: 0.9960
26900/60000 [============>.................] - ETA: 1:10 - loss: 0.0104 - acc: 0.9960
27000/60000 [============>.................] - ETA: 1:10 - loss: 0.0105 - acc: 0.9960
27100/60000 [============>.................] - ETA: 1:09 - loss: 0.0104 - acc: 0.9960
27200/60000 [============>.................] - ETA: 1:09 - loss: 0.0105 - acc: 0.9960
27300/60000 [============>.................] - ETA: 1:09 - loss: 0.0104 - acc: 0.9960
27400/60000 [============>.................] - ETA: 1:09 - loss: 0.0104 - acc: 0.9960
27500/60000 [============>.................] - ETA: 1:09 - loss: 0.0104 - acc: 0.9960
27600/60000 [============>.................] - ETA: 1:08 - loss: 0.0103 - acc: 0.9961
27700/60000 [============>.................] - ETA: 1:08 - loss: 0.0103 - acc: 0.9961
27800/60000 [============>.................] - ETA: 1:08 - loss: 0.0103 - acc: 0.9961
27900/60000 [============>.................] - ETA: 1:08 - loss: 0.0103 - acc: 0.9961
28000/60000 [=============>................] - ETA: 1:07 - loss: 0.0102 - acc: 0.9961
28100/60000 [=============>................] - ETA: 1:07 - loss: 0.0103 - acc: 0.9961
28200/60000 [=============>................] - ETA: 1:07 - loss: 0.0103 - acc: 0.9961
28300/60000 [=============>................] - ETA: 1:07 - loss: 0.0103 - acc: 0.9961
28400/60000 [=============>................] - ETA: 1:07 - loss: 0.0103 - acc: 0.9961
28500/60000 [=============>................] - ETA: 1:06 - loss: 0.0102 - acc: 0.9961
28600/60000 [=============>................] - ETA: 1:06 - loss: 0.0102 - acc: 0.9961
28700/60000 [=============>................] - ETA: 1:06 - loss: 0.0102 - acc: 0.9961
28800/60000 [=============>................] - ETA: 1:06 - loss: 0.0101 - acc: 0.9961
28900/60000 [=============>................] - ETA: 1:06 - loss: 0.0101 - acc: 0.9962
29000/60000 [=============>................] - ETA: 1:05 - loss: 0.0101 - acc: 0.9962
29100/60000 [=============>................] - ETA: 1:05 - loss: 0.0101 - acc: 0.9962
29200/60000 [=============>................] - ETA: 1:05 - loss: 0.0101 - acc: 0.9961
29300/60000 [=============>................] - ETA: 1:05 - loss: 0.0101 - acc: 0.9961
29400/60000 [=============>................] - ETA: 1:04 - loss: 0.0102 - acc: 0.9961
29500/60000 [=============>................] - ETA: 1:04 - loss: 0.0102 - acc: 0.9961
29600/60000 [=============>................] - ETA: 1:04 - loss: 0.0102 - acc: 0.9960
29700/60000 [=============>................] - ETA: 1:04 - loss: 0.0102 - acc: 0.9961
29800/60000 [=============>................] - ETA: 1:04 - loss: 0.0102 - acc: 0.9961
29900/60000 [=============>................] - ETA: 1:03 - loss: 0.0101 - acc: 0.9961
30000/60000 [==============>...............] - ETA: 1:03 - loss: 0.0101 - acc: 0.9961
30100/60000 [==============>...............] - ETA: 1:03 - loss: 0.0101 - acc: 0.9961
30200/60000 [==============>...............] - ETA: 1:03 - loss: 0.0100 - acc: 0.9961
30300/60000 [==============>...............] - ETA: 1:03 - loss: 0.0100 - acc: 0.9961
30400/60000 [==============>...............] - ETA: 1:02 - loss: 0.0100 - acc: 0.9962
30500/60000 [==============>...............] - ETA: 1:02 - loss: 0.0099 - acc: 0.9962
30600/60000 [==============>...............] - ETA: 1:02 - loss: 0.0100 - acc: 0.9962
```

```
30700/60000 [==============>.............] - ETA: 1:02 - loss: 0.0099 - acc: 0.9962
30800/60000 [==============>.............] - ETA: 1:01 - loss: 0.0099 - acc: 0.9962
30900/60000 [==============>.............] - ETA: 1:01 - loss: 0.0099 - acc: 0.9962
31000/60000 [==============>.............] - ETA: 1:01 - loss: 0.0099 - acc: 0.9962
31100/60000 [==============>.............] - ETA: 1:01 - loss: 0.0099 - acc: 0.9962
31200/60000 [==============>.............] - ETA: 1:01 - loss: 0.0100 - acc: 0.9962
31300/60000 [==============>.............] - ETA: 1:00 - loss: 0.0100 - acc: 0.9962
31400/60000 [==============>.............] - ETA: 1:00 - loss: 0.0100 - acc: 0.9962
31500/60000 [==============>.............] - ETA: 1:00 - loss: 0.0099 - acc: 0.9962
31600/60000 [==============>.............] - ETA: 1:00 - loss: 0.0099 - acc: 0.9962
31700/60000 [==============>.............] - ETA: 1:00 - loss: 0.0099 - acc: 0.9962
31800/60000 [==============>.............] - ETA: 59s - loss: 0.0099 - acc: 0.9962
31900/60000 [==============>.............] - ETA: 59s - loss: 0.0098 - acc: 0.9962
32000/60000 [===============>.............] - ETA: 59s - loss: 0.0098 - acc: 0.9963
32100/60000 [===============>.............] - ETA: 59s - loss: 0.0098 - acc: 0.9963
32200/60000 [===============>.............] - ETA: 59s - loss: 0.0098 - acc: 0.9963
32300/60000 [===============>.............] - ETA: 58s - loss: 0.0097 - acc: 0.9963
32400/60000 [===============>.............] - ETA: 58s - loss: 0.0097 - acc: 0.9963
32500/60000 [===============>.............] - ETA: 58s - loss: 0.0097 - acc: 0.9963
32600/60000 [===============>.............] - ETA: 58s - loss: 0.0096 - acc: 0.9963
32700/60000 [===============>.............] - ETA: 57s - loss: 0.0096 - acc: 0.9963
32800/60000 [===============>.............] - ETA: 57s - loss: 0.0096 - acc: 0.9963
32900/60000 [===============>.............] - ETA: 57s - loss: 0.0096 - acc: 0.9964
33000/60000 [===============>.............] - ETA: 57s - loss: 0.0096 - acc: 0.9963
33100/60000 [===============>.............] - ETA: 57s - loss: 0.0096 - acc: 0.9963
33200/60000 [===============>.............] - ETA: 56s - loss: 0.0095 - acc: 0.9964
33300/60000 [===============>.............] - ETA: 56s - loss: 0.0096 - acc: 0.9963
33400/60000 [===============>.............] - ETA: 56s - loss: 0.0095 - acc: 0.9963
33500/60000 [===============>.............] - ETA: 56s - loss: 0.0095 - acc: 0.9964
33600/60000 [===============>.............] - ETA: 56s - loss: 0.0095 - acc: 0.9964
33700/60000 [===============>.............] - ETA: 55s - loss: 0.0095 - acc: 0.9964
33800/60000 [===============>.............] - ETA: 55s - loss: 0.0094 - acc: 0.9964
33900/60000 [===============>.............] - ETA: 55s - loss: 0.0094 - acc: 0.9964
34000/60000 [================>.............] - ETA: 55s - loss: 0.0095 - acc: 0.9964
34100/60000 [================>.............] - ETA: 54s - loss: 0.0095 - acc: 0.9964
34200/60000 [================>.............] - ETA: 54s - loss: 0.0094 - acc: 0.9964
34300/60000 [================>.............] - ETA: 54s - loss: 0.0094 - acc: 0.9964
34400/60000 [================>.............] - ETA: 54s - loss: 0.0094 - acc: 0.9964
34500/60000 [================>.............] - ETA: 54s - loss: 0.0094 - acc: 0.9964
34600/60000 [================>.............] - ETA: 53s - loss: 0.0093 - acc: 0.9964
34700/60000 [================>.............] - ETA: 53s - loss: 0.0094 - acc: 0.9964
34800/60000 [================>.............] - ETA: 53s - loss: 0.0094 - acc: 0.9964
34900/60000 [================>.............] - ETA: 53s - loss: 0.0094 - acc: 0.9964
35000/60000 [================>.............] - ETA: 53s - loss: 0.0093 - acc: 0.9964
35100/60000 [================>.............] - ETA: 52s - loss: 0.0093 - acc: 0.9964
35200/60000 [================>.............] - ETA: 52s - loss: 0.0093 - acc: 0.9964
35300/60000 [================>.............] - ETA: 52s - loss: 0.0093 - acc: 0.9965
35400/60000 [================>.............] - ETA: 52s - loss: 0.0092 - acc: 0.9965
35500/60000 [================>.............] - ETA: 52s - loss: 0.0092 - acc: 0.9965
35600/60000 [================>.............] - ETA: 51s - loss: 0.0092 - acc: 0.9965
35700/60000 [================>.............] - ETA: 51s - loss: 0.0092 - acc: 0.9965
35800/60000 [================>.............] - ETA: 51s - loss: 0.0092 - acc: 0.9965
35900/60000 [================>.............] - ETA: 51s - loss: 0.0092 - acc: 0.9965
36000/60000 [=================>.............] - ETA: 50s - loss: 0.0092 - acc: 0.9965
36100/60000 [=================>.............] - ETA: 50s - loss: 0.0092 - acc: 0.9965
36200/60000 [=================>.............] - ETA: 50s - loss: 0.0092 - acc: 0.9964
36300/60000 [=================>.............] - ETA: 50s - loss: 0.0092 - acc: 0.9964
36400/60000 [=================>.............] - ETA: 50s - loss: 0.0092 - acc: 0.9965
36500/60000 [=================>.............] - ETA: 49s - loss: 0.0092 - acc: 0.9965
36600/60000 [=================>.............] - ETA: 49s - loss: 0.0092 - acc: 0.9965
36700/60000 [=================>.............] - ETA: 49s - loss: 0.0091 - acc: 0.9965
36800/60000 [=================>.............] - ETA: 49s - loss: 0.0091 - acc: 0.9965
36900/60000 [=================>.............] - ETA: 49s - loss: 0.0091 - acc: 0.9965
37000/60000 [=================>.............] - ETA: 48s - loss: 0.0091 - acc: 0.9965
37100/60000 [=================>.............] - ETA: 48s - loss: 0.0091 - acc: 0.9965
```

```
37200/60000 [=================>.............] - ETA: 48s - loss: 0.0091 - acc: 0.9965
37300/60000 [=================>.............] - ETA: 48s - loss: 0.0090 - acc: 0.9965
37400/60000 [=================>.............] - ETA: 47s - loss: 0.0091 - acc: 0.9965
37500/60000 [=================>.............] - ETA: 47s - loss: 0.0091 - acc: 0.9965
37600/60000 [=================>.............] - ETA: 47s - loss: 0.0090 - acc: 0.9965
37700/60000 [=================>.............] - ETA: 47s - loss: 0.0090 - acc: 0.9965
37800/60000 [=================>.............] - ETA: 47s - loss: 0.0090 - acc: 0.9965
37900/60000 [=================>.............] - ETA: 46s - loss: 0.0092 - acc: 0.9965
38000/60000 [==================>............] - ETA: 46s - loss: 0.0092 - acc: 0.9964
38100/60000 [==================>............] - ETA: 46s - loss: 0.0092 - acc: 0.9965
38200/60000 [==================>............] - ETA: 46s - loss: 0.0093 - acc: 0.9964
38300/60000 [==================>............] - ETA: 46s - loss: 0.0094 - acc: 0.9964
38400/60000 [==================>............] - ETA: 45s - loss: 0.0095 - acc: 0.9964
38500/60000 [==================>............] - ETA: 45s - loss: 0.0096 - acc: 0.9964
38600/60000 [==================>............] - ETA: 45s - loss: 0.0095 - acc: 0.9964
38700/60000 [==================>............] - ETA: 45s - loss: 0.0095 - acc: 0.9964
38800/60000 [==================>............] - ETA: 44s - loss: 0.0095 - acc: 0.9964
38900/60000 [==================>............] - ETA: 44s - loss: 0.0095 - acc: 0.9964
39000/60000 [==================>............] - ETA: 44s - loss: 0.0097 - acc: 0.9964
39100/60000 [==================>............] - ETA: 44s - loss: 0.0097 - acc: 0.9964
39200/60000 [==================>............] - ETA: 44s - loss: 0.0097 - acc: 0.9964
39300/60000 [==================>............] - ETA: 43s - loss: 0.0097 - acc: 0.9964
39400/60000 [==================>............] - ETA: 43s - loss: 0.0097 - acc: 0.9964
39500/60000 [==================>............] - ETA: 43s - loss: 0.0097 - acc: 0.9964
39600/60000 [==================>............] - ETA: 43s - loss: 0.0097 - acc: 0.9964
39700/60000 [==================>............] - ETA: 43s - loss: 0.0096 - acc: 0.9964
39800/60000 [==================>............] - ETA: 42s - loss: 0.0096 - acc: 0.9964
39900/60000 [==================>............] - ETA: 42s - loss: 0.0096 - acc: 0.9964
40000/60000 [==================>............] - ETA: 42s - loss: 0.0096 - acc: 0.9964
40100/60000 [==================>............] - ETA: 42s - loss: 0.0098 - acc: 0.9964
40200/60000 [==================>............] - ETA: 42s - loss: 0.0098 - acc: 0.9964
40300/60000 [==================>............] - ETA: 41s - loss: 0.0098 - acc: 0.9964
40400/60000 [==================>............] - ETA: 41s - loss: 0.0098 - acc: 0.9964
40500/60000 [==================>............] - ETA: 41s - loss: 0.0098 - acc: 0.9964
40600/60000 [==================>............] - ETA: 41s - loss: 0.0098 - acc: 0.9964
40700/60000 [==================>............] - ETA: 40s - loss: 0.0098 - acc: 0.9964
40800/60000 [==================>............] - ETA: 40s - loss: 0.0099 - acc: 0.9964
40900/60000 [==================>............] - ETA: 40s - loss: 0.0099 - acc: 0.9964
41000/60000 [==================>............] - ETA: 40s - loss: 0.0099 - acc: 0.9964
41100/60000 [==================>............] - ETA: 40s - loss: 0.0099 - acc: 0.9964
41200/60000 [==================>............] - ETA: 39s - loss: 0.0098 - acc: 0.9964
41300/60000 [==================>............] - ETA: 39s - loss: 0.0098 - acc: 0.9964
41400/60000 [==================>............] - ETA: 39s - loss: 0.0098 - acc: 0.9964
41500/60000 [==================>............] - ETA: 39s - loss: 0.0098 - acc: 0.9964
41600/60000 [==================>............] - ETA: 39s - loss: 0.0098 - acc: 0.9964
41700/60000 [==================>............] - ETA: 38s - loss: 0.0098 - acc: 0.9964
41800/60000 [==================>............] - ETA: 38s - loss: 0.0098 - acc: 0.9964
41900/60000 [==================>............] - ETA: 38s - loss: 0.0098 - acc: 0.9964
42000/60000 [===================>...........] - ETA: 38s - loss: 0.0098 - acc: 0.9964
42100/60000 [===================>...........] - ETA: 37s - loss: 0.0098 - acc: 0.9964
42200/60000 [===================>...........] - ETA: 37s - loss: 0.0098 - acc: 0.9964
42300/60000 [===================>...........] - ETA: 37s - loss: 0.0098 - acc: 0.9964
42400/60000 [===================>...........] - ETA: 37s - loss: 0.0097 - acc: 0.9964
42500/60000 [===================>...........] - ETA: 37s - loss: 0.0097 - acc: 0.9964
42600/60000 [===================>...........] - ETA: 36s - loss: 0.0098 - acc: 0.9964
42700/60000 [===================>...........] - ETA: 36s - loss: 0.0098 - acc: 0.9964
42800/60000 [===================>...........] - ETA: 36s - loss: 0.0098 - acc: 0.9964
42900/60000 [===================>...........] - ETA: 36s - loss: 0.0098 - acc: 0.9964
43000/60000 [===================>...........] - ETA: 36s - loss: 0.0098 - acc: 0.9963
43100/60000 [===================>...........] - ETA: 35s - loss: 0.0098 - acc: 0.9964
43200/60000 [===================>...........] - ETA: 35s - loss: 0.0098 - acc: 0.9963
43300/60000 [===================>...........] - ETA: 35s - loss: 0.0098 - acc: 0.9964
43400/60000 [===================>...........] - ETA: 35s - loss: 0.0098 - acc: 0.9964
43500/60000 [===================>...........] - ETA: 35s - loss: 0.0097 - acc: 0.9964
43600/60000 [===================>...........] - ETA: 34s - loss: 0.0097 - acc: 0.9964
```

```
43700/60000 [====================>.....] - ETA: 34s - loss: 0.0097 - acc: 0.9964
43800/60000 [====================>.....] - ETA: 34s - loss: 0.0097 - acc: 0.9964
43900/60000 [====================>.....] - ETA: 34s - loss: 0.0100 - acc: 0.9964
44000/60000 [====================>.....] - ETA: 33s - loss: 0.0100 - acc: 0.9964
44100/60000 [====================>.....] - ETA: 33s - loss: 0.0100 - acc: 0.9964
44200/60000 [====================>.....] - ETA: 33s - loss: 0.0100 - acc: 0.9964
44300/60000 [====================>.....] - ETA: 33s - loss: 0.0100 - acc: 0.9964
44400/60000 [====================>.....] - ETA: 33s - loss: 0.0099 - acc: 0.9964
44500/60000 [====================>.....] - ETA: 32s - loss: 0.0100 - acc: 0.9964
44600/60000 [====================>.....] - ETA: 32s - loss: 0.0100 - acc: 0.9964
44700/60000 [====================>.....] - ETA: 32s - loss: 0.0100 - acc: 0.9964
44800/60000 [====================>.....] - ETA: 32s - loss: 0.0100 - acc: 0.9964
44900/60000 [====================>.....] - ETA: 32s - loss: 0.0100 - acc: 0.9964
45000/60000 [====================>.....] - ETA: 31s - loss: 0.0100 - acc: 0.9964
45100/60000 [====================>.....] - ETA: 31s - loss: 0.0100 - acc: 0.9964
45200/60000 [====================>.....] - ETA: 31s - loss: 0.0099 - acc: 0.9964
45300/60000 [====================>.....] - ETA: 31s - loss: 0.0099 - acc: 0.9964
45400/60000 [====================>.....] - ETA: 30s - loss: 0.0099 - acc: 0.9964
45500/60000 [====================>.....] - ETA: 30s - loss: 0.0099 - acc: 0.9964
45600/60000 [====================>.....] - ETA: 30s - loss: 0.0100 - acc: 0.9964
45700/60000 [====================>.....] - ETA: 30s - loss: 0.0099 - acc: 0.9964
45800/60000 [====================>.....] - ETA: 30s - loss: 0.0099 - acc: 0.9964
45900/60000 [====================>.....] - ETA: 29s - loss: 0.0100 - acc: 0.9964
46000/60000 [====================>.....] - ETA: 29s - loss: 0.0100 - acc: 0.9964
46100/60000 [====================>.....] - ETA: 29s - loss: 0.0100 - acc: 0.9964
46200/60000 [====================>.....] - ETA: 29s - loss: 0.0100 - acc: 0.9964
46300/60000 [====================>.....] - ETA: 29s - loss: 0.0099 - acc: 0.9964
46400/60000 [====================>.....] - ETA: 28s - loss: 0.0099 - acc: 0.9964
46500/60000 [====================>.....] - ETA: 28s - loss: 0.0099 - acc: 0.9965
46600/60000 [====================>.....] - ETA: 28s - loss: 0.0099 - acc: 0.9965
46700/60000 [====================>.....] - ETA: 28s - loss: 0.0099 - acc: 0.9965
46800/60000 [====================>.....] - ETA: 27s - loss: 0.0099 - acc: 0.9965
46900/60000 [====================>.....] - ETA: 27s - loss: 0.0100 - acc: 0.9964
47000/60000 [====================>.....] - ETA: 27s - loss: 0.0099 - acc: 0.9964
47100/60000 [====================>.....] - ETA: 27s - loss: 0.0099 - acc: 0.9965
47200/60000 [====================>.....] - ETA: 27s - loss: 0.0099 - acc: 0.9965
47300/60000 [====================>.....] - ETA: 26s - loss: 0.0099 - acc: 0.9965
47400/60000 [====================>.....] - ETA: 26s - loss: 0.0099 - acc: 0.9965
47500/60000 [====================>.....] - ETA: 26s - loss: 0.0099 - acc: 0.9965
47600/60000 [====================>.....] - ETA: 26s - loss: 0.0099 - acc: 0.9965
47700/60000 [====================>.....] - ETA: 26s - loss: 0.0099 - acc: 0.9965
47800/60000 [====================>.....] - ETA: 25s - loss: 0.0099 - acc: 0.9965
47900/60000 [====================>.....] - ETA: 25s - loss: 0.0099 - acc: 0.9965
48000/60000 [====================>.....] - ETA: 25s - loss: 0.0099 - acc: 0.9965
48100/60000 [====================>.....] - ETA: 25s - loss: 0.0099 - acc: 0.9965
48200/60000 [====================>.....] - ETA: 25s - loss: 0.0099 - acc: 0.9965
48300/60000 [====================>.....] - ETA: 24s - loss: 0.0099 - acc: 0.9965
48400/60000 [====================>.....] - ETA: 24s - loss: 0.0099 - acc: 0.9965
48500/60000 [====================>.....] - ETA: 24s - loss: 0.0099 - acc: 0.9965
48600/60000 [====================>.....] - ETA: 24s - loss: 0.0099 - acc: 0.9965
48700/60000 [====================>.....] - ETA: 23s - loss: 0.0099 - acc: 0.9965
48800/60000 [====================>.....] - ETA: 23s - loss: 0.0099 - acc: 0.9965
48900/60000 [====================>.....] - ETA: 23s - loss: 0.0099 - acc: 0.9965
49000/60000 [====================>.....] - ETA: 23s - loss: 0.0098 - acc: 0.9965
49100/60000 [====================>.....] - ETA: 23s - loss: 0.0098 - acc: 0.9965
49200/60000 [====================>.....] - ETA: 22s - loss: 0.0098 - acc: 0.9965
49300/60000 [====================>.....] - ETA: 22s - loss: 0.0098 - acc: 0.9965
49400/60000 [====================>.....] - ETA: 22s - loss: 0.0098 - acc: 0.9965
49500/60000 [====================>.....] - ETA: 22s - loss: 0.0098 - acc: 0.9965
49600/60000 [====================>.....] - ETA: 22s - loss: 0.0098 - acc: 0.9965
49700/60000 [====================>.....] - ETA: 21s - loss: 0.0098 - acc: 0.9965
49800/60000 [====================>.....] - ETA: 21s - loss: 0.0099 - acc: 0.9965
49900/60000 [====================>.....] - ETA: 21s - loss: 0.0099 - acc: 0.9965
50000/60000 [====================>.....] - ETA: 21s - loss: 0.0099 - acc: 0.9965
50100/60000 [====================>.....] - ETA: 21s - loss: 0.0098 - acc: 0.9965
```

```
50200/60000 [=========================>.....] - ETA: 20s - loss: 0.0098 - acc: 0.9965
50300/60000 [=========================>.....] - ETA: 20s - loss: 0.0098 - acc: 0.9965
50400/60000 [=========================>.....] - ETA: 20s - loss: 0.0098 - acc: 0.9965
50500/60000 [=========================>.....] - ETA: 20s - loss: 0.0098 - acc: 0.9966
50600/60000 [=========================>.....] - ETA: 19s - loss: 0.0098 - acc: 0.9966
50700/60000 [=========================>.....] - ETA: 19s - loss: 0.0098 - acc: 0.9966
50800/60000 [=========================>.....] - ETA: 19s - loss: 0.0098 - acc: 0.9966
50900/60000 [=========================>.....] - ETA: 19s - loss: 0.0097 - acc: 0.9966
51000/60000 [=========================>.....] - ETA: 19s - loss: 0.0097 - acc: 0.9966
51100/60000 [=========================>.....] - ETA: 18s - loss: 0.0097 - acc: 0.9966
51200/60000 [=========================>.....] - ETA: 18s - loss: 0.0098 - acc: 0.9965
51300/60000 [=========================>.....] - ETA: 18s - loss: 0.0098 - acc: 0.9965
51400/60000 [=========================>.....] - ETA: 18s - loss: 0.0098 - acc: 0.9966
51500/60000 [=========================>.....] - ETA: 18s - loss: 0.0098 - acc: 0.9965
51600/60000 [=========================>.....] - ETA: 17s - loss: 0.0098 - acc: 0.9966
51700/60000 [=========================>.....] - ETA: 17s - loss: 0.0097 - acc: 0.9966
51800/60000 [=========================>.....] - ETA: 17s - loss: 0.0097 - acc: 0.9966
51900/60000 [=========================>.....] - ETA: 17s - loss: 0.0097 - acc: 0.9966
52000/60000 [=========================>....] - ETA: 17s - loss: 0.0097 - acc: 0.9966
52100/60000 [=========================>....] - ETA: 16s - loss: 0.0097 - acc: 0.9966
52200/60000 [=========================>....] - ETA: 16s - loss: 0.0097 - acc: 0.9966
52300/60000 [=========================>....] - ETA: 16s - loss: 0.0096 - acc: 0.9966
52400/60000 [=========================>....] - ETA: 16s - loss: 0.0096 - acc: 0.9966
52500/60000 [=========================>....] - ETA: 15s - loss: 0.0096 - acc: 0.9966
52600/60000 [=========================>....] - ETA: 15s - loss: 0.0096 - acc: 0.9966
52700/60000 [=========================>....] - ETA: 15s - loss: 0.0096 - acc: 0.9966
52800/60000 [=========================>....] - ETA: 15s - loss: 0.0096 - acc: 0.9966
52900/60000 [=========================>....] - ETA: 15s - loss: 0.0096 - acc: 0.9966
53000/60000 [=========================>....] - ETA: 14s - loss: 0.0096 - acc: 0.9966
53100/60000 [=========================>....] - ETA: 14s - loss: 0.0096 - acc: 0.9966
53200/60000 [=========================>....] - ETA: 14s - loss: 0.0096 - acc: 0.9966
53300/60000 [=========================>....] - ETA: 14s - loss: 0.0096 - acc: 0.9966
53400/60000 [=========================>....] - ETA: 14s - loss: 0.0096 - acc: 0.9966
53500/60000 [=========================>....] - ETA: 13s - loss: 0.0096 - acc: 0.9966
53600/60000 [=========================>....] - ETA: 13s - loss: 0.0096 - acc: 0.9966
53700/60000 [=========================>....] - ETA: 13s - loss: 0.0096 - acc: 0.9966
53800/60000 [=========================>....] - ETA: 13s - loss: 0.0095 - acc: 0.9966
53900/60000 [=========================>....] - ETA: 12s - loss: 0.0096 - acc: 0.9966
54000/60000 [=========================>...] - ETA: 12s - loss: 0.0096 - acc: 0.9966
54100/60000 [=========================>...] - ETA: 12s - loss: 0.0096 - acc: 0.9966
54200/60000 [=========================>...] - ETA: 12s - loss: 0.0096 - acc: 0.9966
54300/60000 [=========================>...] - ETA: 12s - loss: 0.0096 - acc: 0.9966
54400/60000 [=========================>...] - ETA: 11s - loss: 0.0096 - acc: 0.9966
54500/60000 [=========================>...] - ETA: 11s - loss: 0.0096 - acc: 0.9966
54600/60000 [=========================>...] - ETA: 11s - loss: 0.0096 - acc: 0.9966
54700/60000 [=========================>...] - ETA: 11s - loss: 0.0096 - acc: 0.9966
54800/60000 [=========================>...] - ETA: 11s - loss: 0.0096 - acc: 0.9966
54900/60000 [=========================>...] - ETA: 10s - loss: 0.0096 - acc: 0.9966
55000/60000 [=========================>...] - ETA: 10s - loss: 0.0096 - acc: 0.9966
55100/60000 [=========================>...] - ETA: 10s - loss: 0.0096 - acc: 0.9966
55200/60000 [=========================>...] - ETA: 10s - loss: 0.0096 - acc: 0.9966
55300/60000 [=========================>...] - ETA: 10s - loss: 0.0096 - acc: 0.9966
55400/60000 [=========================>...] - ETA: 9s - loss: 0.0097 - acc: 0.9966
55500/60000 [=========================>...] - ETA: 9s - loss: 0.0097 - acc: 0.9966
55600/60000 [=========================>...] - ETA: 9s - loss: 0.0097 - acc: 0.9966
55700/60000 [=========================>...] - ETA: 9s - loss: 0.0097 - acc: 0.9966
55800/60000 [=========================>...] - ETA: 8s - loss: 0.0097 - acc: 0.9966
55900/60000 [=========================>...] - ETA: 8s - loss: 0.0096 - acc: 0.9966
56000/60000 [=========================>..] - ETA: 8s - loss: 0.0096 - acc: 0.9966
56100/60000 [=========================>..] - ETA: 8s - loss: 0.0096 - acc: 0.9966
56200/60000 [=========================>..] - ETA: 8s - loss: 0.0096 - acc: 0.9966
56300/60000 [=========================>..] - ETA: 7s - loss: 0.0096 - acc: 0.9966
56400/60000 [=========================>..] - ETA: 7s - loss: 0.0096 - acc: 0.9966
56500/60000 [=========================>..] - ETA: 7s - loss: 0.0096 - acc: 0.9966
56600/60000 [=========================>..] - ETA: 7s - loss: 0.0096 - acc: 0.9966
```

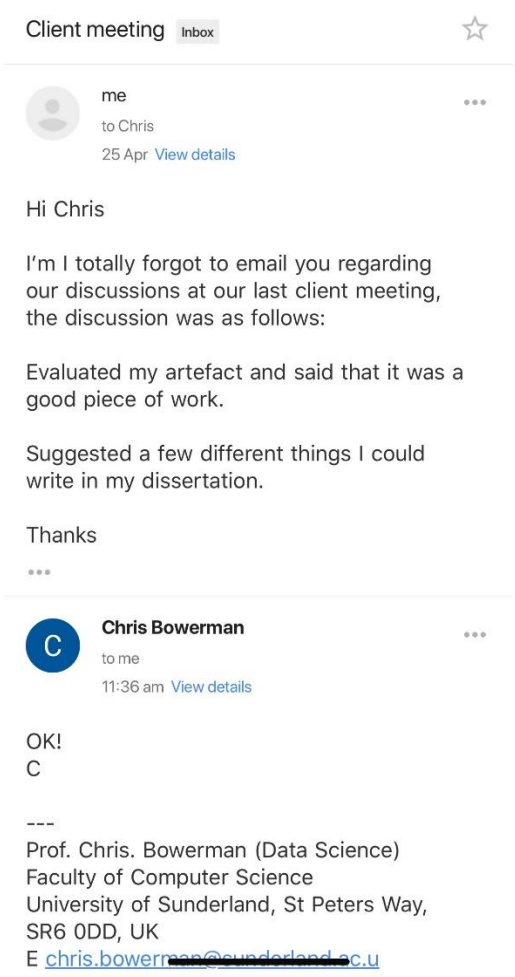
```
56700/60000 [============================>..] - ETA: 7s - loss: 0.0096 - acc: 0.9966
56800/60000 [============================>..] - ETA: 6s - loss: 0.0096 - acc: 0.9966
56900/60000 [============================>..] - ETA: 6s - loss: 0.0096 - acc: 0.9966
57000/60000 [============================>..] - ETA: 6s - loss: 0.0096 - acc: 0.9966
57100/60000 [============================>..] - ETA: 6s - loss: 0.0096 - acc: 0.9966
57200/60000 [============================>..] - ETA: 5s - loss: 0.0096 - acc: 0.9966
57300/60000 [============================>..] - ETA: 5s - loss: 0.0096 - acc: 0.9966
57400/60000 [============================>..] - ETA: 5s - loss: 0.0096 - acc: 0.9966
57500/60000 [============================>..] - ETA: 5s - loss: 0.0095 - acc: 0.9966
57600/60000 [============================>..] - ETA: 5s - loss: 0.0095 - acc: 0.9966
57700/60000 [============================>..] - ETA: 4s - loss: 0.0095 - acc: 0.9966
57800/60000 [============================>..] - ETA: 4s - loss: 0.0095 - acc: 0.9966
57900/60000 [============================>..] - ETA: 4s - loss: 0.0095 - acc: 0.9966
58000/60000 [============================>.] - ETA: 4s - loss: 0.0095 - acc: 0.9966
58100/60000 [============================>.] - ETA: 4s - loss: 0.0095 - acc: 0.9966
58200/60000 [============================>.] - ETA: 3s - loss: 0.0095 - acc: 0.9966
58300/60000 [============================>.] - ETA: 3s - loss: 0.0095 - acc: 0.9966
58400/60000 [============================>.] - ETA: 3s - loss: 0.0095 - acc: 0.9966
58500/60000 [============================>.] - ETA: 3s - loss: 0.0095 - acc: 0.9966
58600/60000 [============================>.] - ETA: 2s - loss: 0.0095 - acc: 0.9966
58700/60000 [============================>.] - ETA: 2s - loss: 0.0095 - acc: 0.9966
58800/60000 [============================>.] - ETA: 2s - loss: 0.0095 - acc: 0.9966
58900/60000 [============================>.] - ETA: 2s - loss: 0.0095 - acc: 0.9966
59000/60000 [============================>.] - ETA: 2s - loss: 0.0095 - acc: 0.9966
59100/60000 [============================>.] - ETA: 1s - loss: 0.0096 - acc: 0.9966
59200/60000 [============================>.] - ETA: 1s - loss: 0.0095 - acc: 0.9966
59300/60000 [============================>.] - ETA: 1s - loss: 0.0095 - acc: 0.9966
59400/60000 [============================>.] - ETA: 1s - loss: 0.0095 - acc: 0.9966
59500/60000 [============================>.] - ETA: 1s - loss: 0.0095 - acc: 0.9966
59600/60000 [============================>.] - ETA: 0s - loss: 0.0095 - acc: 0.9966
59700/60000 [============================>.] - ETA: 0s - loss: 0.0096 - acc: 0.9966
59800/60000 [============================>.] - ETA: 0s - loss: 0.0096 - acc: 0.9966
59900/60000 [============================>.] - ETA: 0s - loss: 0.0096 - acc: 0.9965
60000/60000 [==============================] - 139s 2ms/step - loss: 0.0097 - acc: 0.9965 - val_loss:
0.0468 - val_acc: 0.9883
Prototype 1 Error Rate: 1.17%
```

Sprint 1 & 3 testing output


gray
Noisy image
to test
Tesseract OCR
(x=293, y=7) ~ L:255
marked areas
Noisy image
to test
Tesseract OCR
(x=234, y=249) ~ R:252 G:255 B:255
segment no:1
Noisy image
segment no:2
to test
(x=11, y=46) ~ R:192 G:192 B:192
segment no:0
Tesseract OCR
(x=308, y=52) ~ R:252 G:255 B:255

Appendix H

Client feedback



Client meeting Inbox

me
to Chris
25 Apr View details

Hi Chris

I'm I totally forgot to email you regarding our discussions at our last client meeting, the discussion was as follows:

Evaluated my artefact and said that it was a good piece of work.

Suggested a few different things I could write in my dissertation.

Thanks

C **Chris Bowerman**
to me
11:36 am View details

OK!
C

Prof. Chris. Bowerman (Data Science)
Faculty of Computer Science
University of Sunderland, St Peters Way,
SR6 0DD, UK
E chris.bowerman@sunderland.ac.u

Appendix I

See folder Dissertation on memory stick or in zip file