University of Plymouth
School of Engineering, Computing, and Mathematics

COMP3000
Final Stage Computing Project
2020/2021

ALabDaC - Automated Lab Book Data Collection

Gabryel Mason-Williams
10574626
BSc (Hons) Computer Science

Acknowledgements

I want to thank my supervisor at Plymouth, David Walker, for allowing this project to go ahead, supervising it and for being very helpful during my time at Plymouth.

I also want to thank my supervisors at the Rosalind Franklin Institute (RFI), Laura Shemilt and Mark Basham, for supervising this project and making me an active member of the Artificial Intelligence theme. It has been a joy doing this project with you.

I also want to thank my family and friends who have been here for the ups and downs of this project and my time at Plymouth. As always, I am very grateful to have you in my life.

Last but certainly not least, I would like to thank Frank, my dog, who has been with me every step of the way this year, including interrupting the testing, meetings and pretty much everything. Without him, this project certainly wouldn't have succeeded. Frank is the real MVP.

Abstract

The ALaBDaC system was a collaboration with the Rosalind Franklin Institute that aimed to solve the problem of paper lab books being challenging to access, review, search through, and hard to practically backup. This dissertation covers the software development of the proof of concept system ALaBDaC. Part one outlines the background to the problem with the aims and objectives of the project, the legal, ethical, social and professional issues surrounding its development. Part two discusses and evaluates the hardware and methods used to create the system. Part three provides a postmortem of the development process and assesses the technologies, sprints, methodologies, and testing. Concluding with how the project effectively met the objectives, ergo solving the problem and outlining the area of future development.

Information

Word count: 9842

Github: private repository contact for access `https://github.com/rosalindfranklininstitute/alabdac`

Contents

List of Tables

List of Figures

1 Introduction

This project is in partnership with the Rosalind Franklin Institute (RFI). At RFI, they have laboratories where experiments take place. Under The Good Laboratory Practice Regulations 1999 [1], the Freedom of Information Act 2000 [2], and the Data Protection Act [3], it is essential that Laboratories keep and maintain accurate records of all experiments conducted within the lab [4]. With that paper lab books are a valuable tool used at RFI to record the experiments. However, they have many downsides, such as challenging to access, review, search through, and hard to practically backup. The following report demonstrates and evaluates the software solution that attempted to solve this problem.

2 Background, Objectives and Aims

2.1 Background

This project is in partnership with the Rosalind Franklin Institute (RFI). RFI "is a new national institute, funded by the UK government through UK Research and Innovation (UKRI), dedicated to bringing about transformative changes in life science through interdisciplinary research and technology development" [5]. RFI's "work is focused into five complementary themes, which together will produce technologies which allow us to see the biological world in new ways, from single molecules to entire systems. This insight will speed up drug design and development, and push forward our understanding of human health and disease" [5]. This project is associated with the Artificial Intelligence and Informatics theme but is directly relevant to the other themes at RFI, which use lab books in the lab environment.

Under The Good Laboratory Practice Regulations 1999 [1], the Freedom of Information Act 2000 [2], and the Data Protection Act [3], it is essential that Laboratories keep and maintain accurate records of all experiments conducted within the lab [4]. Therefore, researchers take meticulous notes of their work in the lab, which is usually written into lab books. Electronic lab books such as "Lab Folder"[6] and "labguru" [7] are sometimes used within certain lab environments. However, some labs require strict contamination control within RFI, especially in labs where there are chemical and biological hazards. These labs give practical restrictions on using computers as the sharing of computer hardware such as keyboards or mouses is infeasible due to potential contamination. This means that the document of choice is a paper lab book. However, the downside of using a paper lab book is, of course, that lab books are harder to share and search at a later date, as well as the fact that lab books within strict contamination controlled labs are not allowed to leave the lab, therefore making the lab book data locked to the lab creating an issue where the data is not easily accessible, reviewable, or searchable as well as being difficult to practically back up.

2.2 Aims

The project aims to develop and investigate an automated way to collect lab book information as the user writes and automatically record it to a central database, thus removing the issue of lab book data being locked to the labs and allowing for a digital representation to be stored and searched through at a later date.

2.3 Objectives

This is a proof of concept and research based project with the following being the core objects:

- The project needs to be done in a low-cost way, below £300, so that it can be implemented across RFI.
- It needs to be activated and deactivated with an RFID card reader, as this is the activation method used within RFI.
- There should be a method to display that the device is capturing data.
- Relevant information needs to be collected and extracted from the lab books.
- Results will be uploaded to the RFI's S3 data store with appropriate metadata.
- Extensible and maintainable code.

Ultimately, the objective is see whether these objectives are feasible and possible, as if this is shown to be promising or achievable, then RFI will continue to develop this project further.

3 Legal, Social, Ethical and Professional issues

When designing and creating a system such as ALaBDaC, where the primary objective is to collect data, there are many legal, social, ethical and professional issues to consider. This section goes into detail about potential issues and expresses how they are overcome.

3.1 Legal and Professional

The Data Protection Act of 2018 [3] is the UK's implementation of the General Data Protection Regulation (GDPR) [8]. The Data Protection Act of 2018 states how user data must be maintained, looked after and used within the UK. For the ALaBDaC system, the data is stored and transferred securely to an s3 data store using a write-only key, which means if someone was to get hold of the physical ALaBDaC system, they would not be able to access the user data with those keys, therefore adhering to data protection act. The data is then stored in accordance with RFI's data policy, which follows the FAIR principles set by UKRI [9].

This project will require the use of RFI's computing infrastructure to train machine learning models, store data, and run the project. Therefore, it is essential, professionally and legally, that the computers are used in accordance with the law so that the computer misuse act 1990 [10] is not broken. When using RFI's infrastructure, only computers and data that the developer is authorised to use will be used, and nothing that can facilitate computer misuse will be used/ implemented/developed.

As the project is a collaboration between the developer who represents The University of Plymouth as a student and the British Computer Society (BCS) , as the project is accredited by the BCS, it is essential that throughout the project that any interaction with the student adheres to their respective code of conducts [11], [12]. This will ensure that a healthy, effective and professional relationship is fostered during the collaboration with RFI.

3.2 Social and Ethical

The system has social implications for the aspect of collecting images, where some users may not be comfortable with this data being collected. Thus this system will be opt-in by default and will require users interaction before any recording takes place. This will ensure that users have made an active decision to use the system. This opt-in will only last for as long as the session is. When they end the session they will have effectively opted out and they will have to opt-in to the system every time they want to use it. The access and removal of the data will then be managed by RFI's data policy.

4 Hardware

To be able to meet the objectives of the project hardware was required. This section goes into the reasoning behind the hardware used.

4.1 Components

A single-board computer is required to run the system—there were two compelling options for this project, the Jetson Nano [13] and the "Raspberry Pi model 4 Model B"[14]. The Jetson Nano is a single board computer with a Graphics Processing Unit (GPU) developed by Nvidia for embedded AI applications. The "Raspberry Pi 4 Model B" is a single-board computer with eight GB of RAM. Both were considered; however, it was decided that the lack of the GPU from the "Raspberry Pi 4 Model B" would have lead to a solution where AI inferencing would have to take place in the cloud. Using the cloud would make the project have a continual cost instead of the fixed cost of just the kit. Therefore the Jetson Nano was used with a 128GB SD (Secure Digital) to run the OS and 4 amp power supply.

A camera is required to capture the data; there were two main options the "Raspberry Pi High-Quality Camera Module" (RPIHQCM) [15] and the "Raspberry Pi Camera Module V2" (RPCMv2) [16]. The RPIHQCM offered a 12MP sensor, whereas the RPCMv2 offered an 8mp sensor; therefore, as this project needs to capture high-quality images, the RPIHQCM was used with a 6mm wide-angle lens.

A display is required to show the output from the camera; therefore, a 5-inch screen was used as it was agreed with RFI. This would be an appropriate size to display the camera feed. To be able to read the RFID input, an RFID card reader was also purchased and to connect to the wifi within the lab, a wifi dongle was bought.

The total price of the unit, plus wiring accessories, came to £277.20 at the time of purchase, which met the objective of the unit costing less than £300. Detailed parts list can be seen in Appendix A

5 Methods and Discussion

This chapter goes into detail about the methods implemented, outline the problem and objects for each method, then reviewing the current literature and finally discussing the implementation and its results.

5.1 Intelligent Data Capture

5.1.1 What the problem is and objectives

For this system to effectively extract information and present useful data from the lab books, clear and unobstructed images of the lab book need to be taken, i.e without hands or other objects within the image. The images need to also capture changes made to the book as they are made throughout the book to help record a timeline of changes, that can be later searched through. The objectives are as follows:

- Capture unobstructed images
- Capture images after a change is made to the lab book
- Capture as few images as possible
- Clear and presentable images.

5.1.2 Literature Review

Computer vision is the field of computer science that aims to train computers to recognise and understand a visual stimulus[17]. Computer Vision aims to tackle many areas to do with vision such as; classification, segmentation, object tracking and object detection. Classification is the act of trying to workout what an image is, segmentation is the act of separating an image into constitute parts, object tracking is the act of following an object as it moves across a set of images or frames, and object detection is the act of identifying a specific object and classifying it within the image. There are many different techniques to solve computer vision problems from deep learning[18], to classical image processing methods which include methods such as Canny Edge Detection [19], Hough transform[20] and Hue Saturation and Value (HSV) space colour segmentation[21].

Even though deep learning has had success within many domains of computer vision [22], there are still applications where classical methods work effectively in situations with limited data or small compute resource, such as the report ”Fast and Accurate Document Detection for Scanning” [23] where they use edge detection, then hough transform, to detect the lines on the page, and then computed the intersections to find potential corners. These corners are then processed to find the best possible quadrilateral. This method was able to accurately detect documents for scanning, and was implemented in the Dropbox doc scanner.

5.1.3 Implementation

As a lab environment could not be used for development due to lockdown restrictions [24], a mock environment was created with a white sheet representing the table or desk. The camera was positioned directly above and to aid the user in placing the lab book in the center of the camera, a green border was placed around the lab book, see figure 1. The colour green was used for the border as it is not a colour used within the RFI labs.

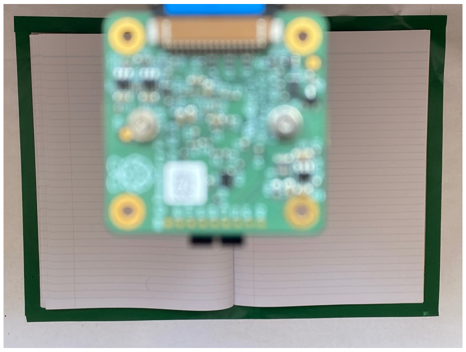


Figure 1: Mock lab setup, with camera directly over the lab book.

For the system to capture clear images of the lab book, it needs to detect when something is happening to the lab book, i.e. a user writing, to know when to stop recording. A supervised object detection method of detecting hands as they enter the scene was considered. However, it was not implemented as RFI did not have a dataset of the scientists writing in lab books from a bird's eye view; other hand datasets such as the Visual Geometry Group hand dataset[25], could have been used for hand detection. Although, due to time constraints, this was not investigated as the scientists have to wear gloves within specific environments, which would have potentially limited the effectiveness of this technique due to the distributional shift problem[26]. Therefore classical image processing methods were used to classify when a page was clear or not, as these would not require a labelled dataset to train and instead, work from the information in the environment created.

As there will always be a green border around the lab book, this information can be used to detect when the lab book is covered, as the user will have to cross this border in order to write inside the lab book. Since the border is the only thing expected to be green, a colour separation technique is implemented to detect the border.

The input stream from the camera has a three colour channel; Red Blue Green (RGB). However, the green border is not consistently the same RGB value all around due to lighting effects; therefore, the frames are colour separated using the HSV colour segmentation technique to find the border instead of an RGB separation technique, as this method allows for the separation of different colours through lighting intensity. This method works by converting the frame into an HSV representation and then thresholding between lower and upper HSV values returning a black and white image, where the white pixels represent the colour captured through segmentation. These white pixels are then counted to form a value of the border.

The majority of the background does not change between frames, only changing when the scientist's hand enters the scene to write in the lab book or change the page. Ergo a background subtraction method can be used to detect when a scientist is writing on/in the lab book. The BackgroundSubtractorKNN method provided by OpenCV [27] is used; this method works by creating a background model that is continually updated such that if the background changes, it can adapt, the current frame is then taken from the background model and passed through a threshold, thus leaving the foreground image. Areas that have not changed are left black, and areas that have changed are white. The white pixels are calculated to create a value for the amount of change in the frame.

As the frames may contain artefacts or random noise, image stacking was used to improve the quality of the images captured, which will make data extraction easier as the images will be clearer. Image stacking is the act of taking the average of series of images to reduce the signal to noise ratio [28]. The image stacking was implemented using a ring buffer of size 5 so that this process does not use too much RAM as, without this limit, the device will crash as it will fill up the memory; it also ensures that the 5 most recent frames can be stacked, after every capture the image stack is cleared.

These methods are used together to form the intelligent data capture; an initial border value is calculated during an initialisation phase, which takes the average of the border over 60 frames; this border value is then used as the baseline. Then the border and frame change is calculated for each process_frame (a smaller resolution of the frame to be stacked); if the border is above the threshold or the frame change value is below the threshold, then the save_frame (full resolution image) is added to the frame stack. Otherwise, the frames are stacked, and an image is returned. This process is shown in figure 2. Its important to note that this method captures the frames before an event so the last change to the lab book will not be captured using this technique. It is also worth noting that this methods effectiveness is subject to the initial thresholding values and will require tuning for each environment it is deployed within.

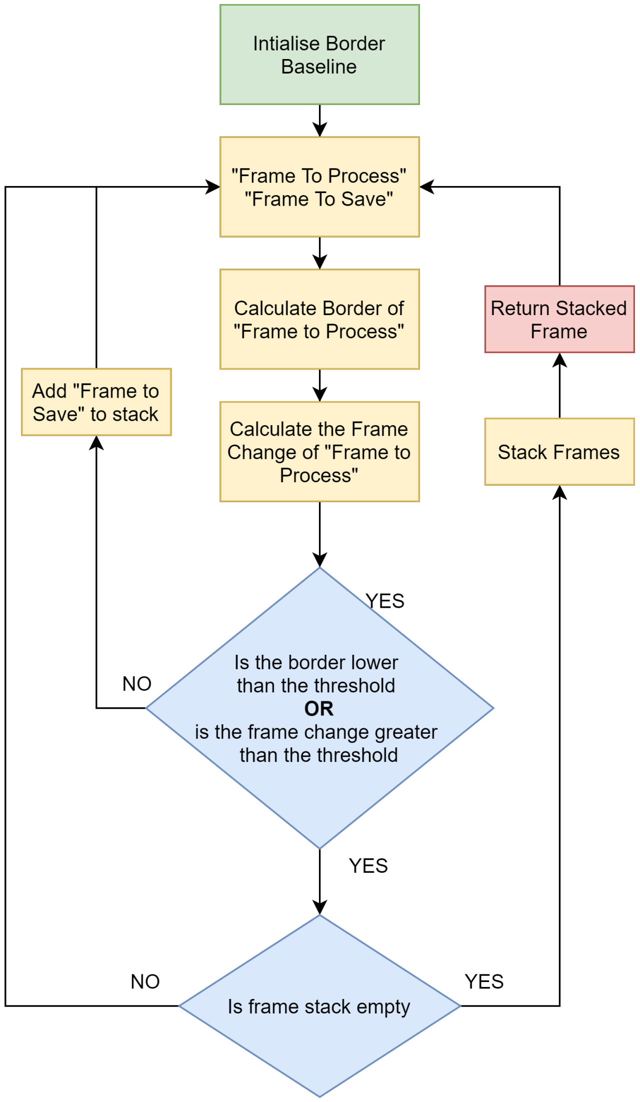

Intialise Border Baseline
"Frame To Process" "Frame To Save"
Calculate Border of "Frame to Process"
Calculate the Frame Change of "Frame to Process"
Return Stacked Frame
Stack Frames
Add "Frame to Save" to stack
YES
NO
Is the border lower than the threshold **OR** is the frame change greater than the threshold
YES
NO
YES
Is frame stack empty

Figure 2: Intelligent Data Capture flow Diagram

5.1.4 Results

This implementation was tested against a set of 5 videos, in which a user writes on a page ”test1, test2, test3, test4” with test[1-2] on the left page and test[3-4] on the right-hand page. There are four key events in these videos (each time the user writes on the lab book). When using a border threshold of 5% and difference in frames threshold of 2500, this method was able to capture all four key events for all the videos, with the outputs, only capturing the lab book, see Appendix B for captures. With those results, this implementation effectively meets the objectives of capturing unobstructed images of the lab book after a change is made to the lab book while keeping the images taken to a minimum and producing clear and presentable images, through the use of classical computer vision techniques.

5.2 Intelligent Data extraction

5.2.1 What the problem is and objectives

Now the data can be effectively collected with a clear lab book view, the task is to try and extract any relevant information from the lab book. In this section the focus is to extract the keywords written within the lab book, as this information will be useful in aiding searching at a later date. The objectives are as follows:

- A method to recognise and extract any keyword from the page

5.2.2 Literature Review

5.2.2.1 Text Recognition

Handwriting recognition is considered a complex problem for machine learning compared to machine text because there is a considerable variation in handwriting styles from person to person [29]. Humans do not always write in a straight line; handwriting can be joined, a mix of capitalisation of letters, etc. This can be summed up if people struggle to read their own and others' handwriting, then how is a computer expected to do any better. There are two different methods of handwriting recognition offline and online [30]. Online methods involve using digital input and using the information as the user writes; often, these methods have a high level of accuracy. Offline methods involve recognising text once written down, so the only information is the image of the written text. Within these methods, there are two techniques-character and word recognition. Character recognition works by recognising individual characters to form a word, whereas word recognition works by recognising the whole word. Offline handwriting recognition problems often use different techniques; however, the highest accuracy is achieved through convolutional neural networks (CNN) [31].

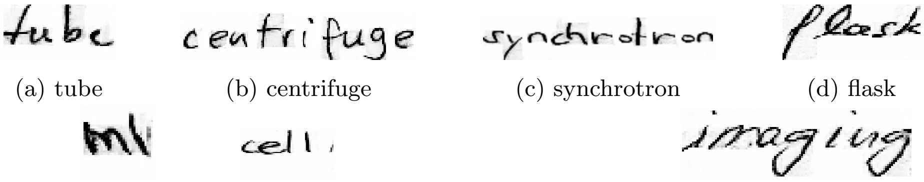
5.2.2.2 Synthetic Dataset Generation

Synthetic dataset generation is the method of creating a dataset that closely represents real world data [32]. Synthetic handwriting datasets have been used been used to produce high recognition accuracy in both character and word classification [33], [34].

5.2.3 Implementation

The objective of the intelligent data extraction method is to recognise and extract relevant keywords from a lab book image. Therefore, an offline computer vision approach is required to identify the keywords. Here an object detection algorithm is appropriate as the problem requires finding and classifying specific image data. Current object detection algorithms use supervised deep learning [22]. Therefore a labelled dataset is needed to train them.

As there is not a pre-existing handwritten dataset for all words in the English language, the human handwriting synthesiser ScrabbleGAN [35] was used. ScrabbleGAN is a state of the art semi-supervised handwriting synthesiser designed to generate images of words of arbitrary length with different handwriting styles. ScrabbleGan was trained for 200 epochs using a dataset formed from the IAM dataset [36] for the handwritten words and the English-words dataset [37] for the lexicon. The training produced a model that was capable of producing images that looked liked realistic handwriting, as demonstrated in figure 3. This model allows for the creation of a dataset of arbitrary size, including any words.



(a) tube (b) centrifuge (c) synchrotron (d) flask

(e) ml (f) cell (g) imaging

Figure 3: Example ScrabbleGan generated words

As shown in figure 3, the ScrabbleGan output data is perfect for a classification machine learning problem. However, the lab book images are not of singular words,they are instead of many words on a page. Thus the data needs to be manipulated to represent the problem the object detection algorithm will face. For an object detection algorithm to perform well outside of the initial training phase, the dataset on which it is trained and tested on should resemble the data it will be used against. Therefore, the ScrabbleGan words need to be manipulated and superimposed onto lab book backgrounds. The manipulations implemented were: the removal of the backgrounds around the words, such that the words have the same background as the lab book, resizing of words to represent different sizing in people's writing, and the rotation of words across the book so that different rotations are taken into account when people write. This method of manipulation is demonstrated in figure 4.

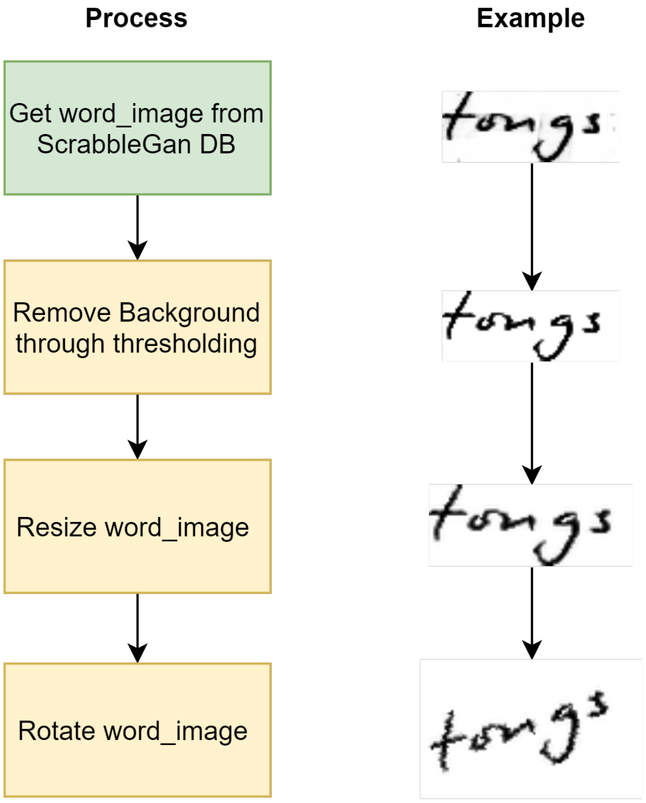
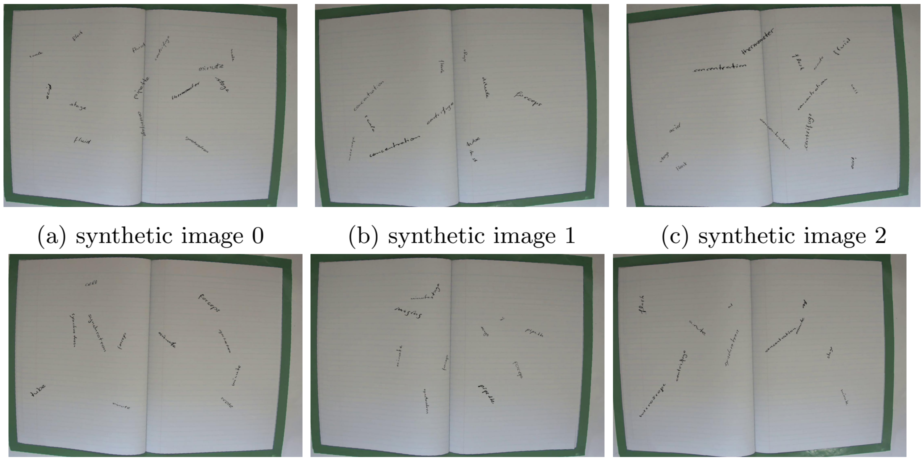

Process
Example
Get word_image from ScrabbleGan DB
Remove Background through thresholding
Resize word_image
Rotate word_image
tongs

Figure 4: Word manipulation process

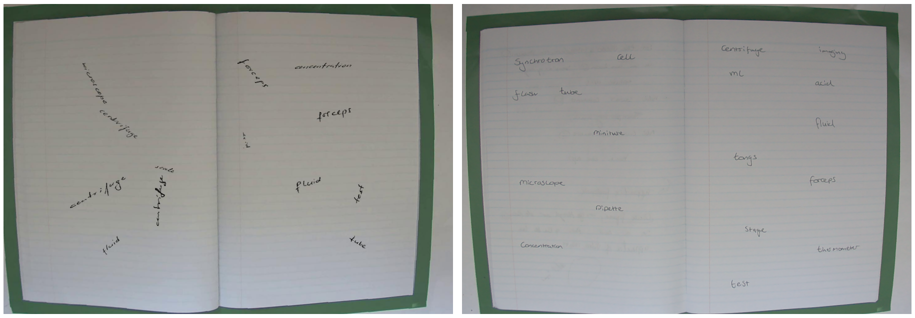
The words are then pseudo-randomly superimposed onto the background images, such that words do not overlap, and the backgrounds are cropped and rotated to represent differences in lab book positioning; figure 5 displays example artificial training or testing data, and figure 6 displays the example data next to real data.



(a) synthetic image 0 (b) synthetic image 1 (c) synthetic image 2

(d) synthetic image 3 (e) synthetic image 4 (f) synthetic image 5

Figure 5: Example synthetic images



(a) Synthetic data (b) Example of words on page

Figure 6: Synthetic data vs Lab book with words written in it.

As the words are superimposed onto the background, the boundary boxes and class annotation labels are already calculated. The resulting data is then stored in the format shown in table 1 . The annotation method used follows the format shown in table 2.

Folder Structure	**Contents**
images/test/*.jpg	Images for testing
images/train/*.jpg	Images for training
annotations/test/*.txt	Associated annotations for testing
annotations/train/*.txt	Associated annotations for training

Table 1: Artificial Dataset image and annotations folder structure, where image/[train/test]/*.jpg maps 1-to-1 to annotations/[train/test]/*.txt

Class	**x_center**	**y_center**	**Width of Box**	**Height of Box**
1	0.23	0.21	0.03	0.02
2	0.14	0.15	0.015	0.0124

Table 2: Example Annotation and format meaning: the centre and width and height values are normalised so they are between 0 and 1, the original values can be returned by multiplying them by the width and height of the test/train image.

Due to the nature of synthetic data generation, the dataset can be, in theory, any size, which is exceptionally advantageous as it allows for deep learning object detection algorithms to be used, which often require a large dataset to be performant [22]. Usually, these algorithms are characterised by their ability to perform better the more data they are given to train [22] meaning, in theory, provided a suitable model and enough training time. The words could be accurately detected with an arbitrary amount of precision. Even though this method can generate a large dataset, it is essential to note that the dataset is artificial and not using any images of human written words and superimposed onto the lab book. Therefore, due to the distributional shift problem[26], the model's performance on the test dataset will not truly reflect its performance when dealing with human handwriting on a lab book background. This implementation yields a word detection pipeline that allows for any set of words to be detected. The pipeline is demonstrated in Figure 7.

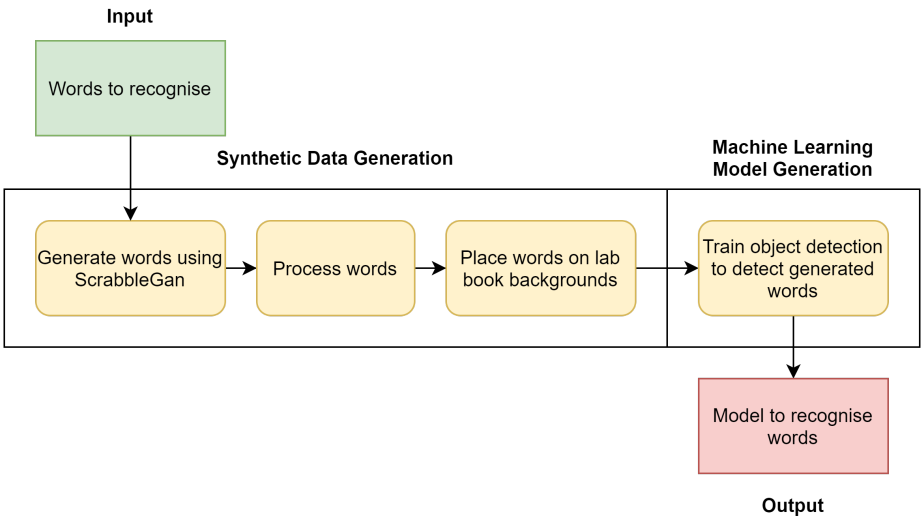
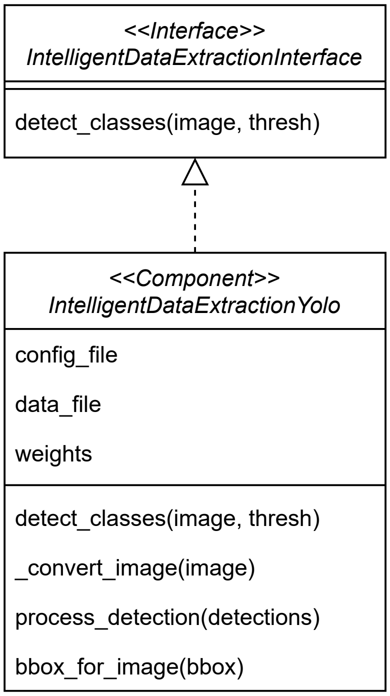

Input
Words to recognise
Synthetic Data Generation
Machine Learning Model Generation
Generate words using ScrabbleGan
Process words
Place words on lab book backgrounds
Train object detection to detect generated words
Model to recognise words
Output

Figure 7: Synthetic data generation pipeline for creating machine learning models to recognise human handwriting.

The object detection algorithm, used within ALaBDaC is YOLOv3 [38]. YOLOv3 was used as it is able to do real time object detection on the Jetson Nano. However, as RFI may in future want to use different object detection algorithms, such as ”Faster R-CNN” [39] or ”Single shot multibox detector” [40], an interface has been implemented as this ensures that future code will have the same functionality; this is demonstrated in figure 8

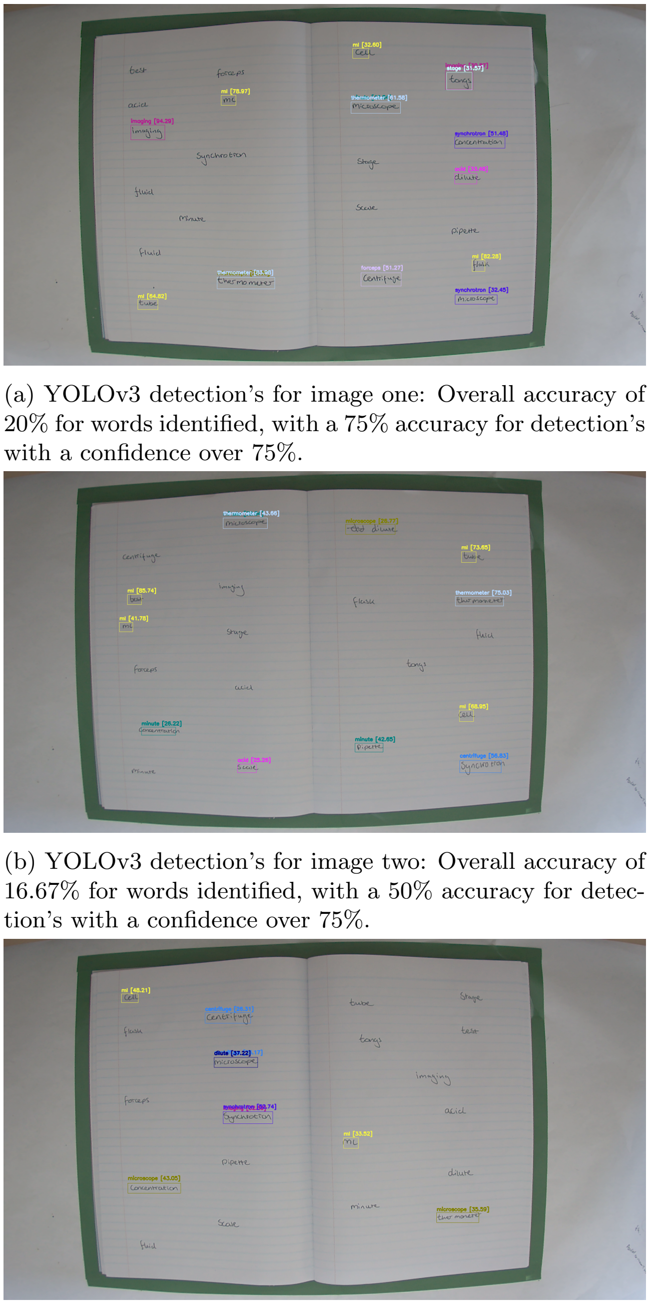


<<Interface>>
IntelligentDataExtractionInterface
detect_classes(image, thresh)
<<Component>>
IntelligentDataExtractionYolo
config_file
data_file
weights
detect_classes(image, thresh)
_convert_image(image)
process_detection(detections)
bbox_for_image(bbox)

Figure 8: Intelligent Data capture UML diagram

5.2.4 Results

This implementation of creating an artificial dataset was then used to generate a dataset consisting of 56,000 training images and 14,000 testing images with 20 keywords [synchrotron, cell, flask, tube, centrifuge, imaging, ml, acid, tongs, forceps, microscope, pipette, concentration, stage, dilute, scale, minute, thermometer, fluid, test], with 10 to 15 words on a background. This dataset was used to train a YOLOv3 until a mean average precision score of 70% due to time constraints. The model was then able to recognise some human written words on a page, as displayed in figure 9.



(a) YOLOv3 detection's for image one: Overall accuracy of 20% for words identified, with a 75% accuracy for detection's with a confidence over 75%.

(b) YOLOv3 detection's for image two: Overall accuracy of 16.67% for words identified, with a 50% accuracy for detection's with a confidence over 75%.

(c) YOLOv3 detection's for image three: Overall accuracy of 33.33% for words identified, no words where detected with confidence over 75%

Figure 9: YOLOv3 detection's against human written words on a lab book

As expected, this method is not perfect due to being trained and tested on different data, to what it is ultimately deployed against. However, this does demonstrate that this implementation does allow for any keywords to be recognised and extracted from the lab book, if not with a few inconsistencies, effectively meeting RFI's requirements. With further improvement to the processing of the words to look more realistic on the lab book background, this synthetic data generation method could potentially achieve better results when deployed against real lab book data. Ultimately this method shows promise and is worth investigating further.

5.3 Data transfer

5.3.1 What the problem is and objectives

Once the image is captured and processed, the image and relevant data (meta-data) (User info, Keywords detected, host machine, etc,) must be exported to the central RFI data-store. RFI uses object storage to store their image data, within an object store buckets/pools are created to partition the data, here an s3 bucket is provided. This encompasses the problem of what file format to use, how to store the relevant data and how to export it. The objectives are as follows:

- Store high-quality images - maintaining original quality
- Image format viewable in a web browser
- Metadata easily read and written
- Export images to an s3 bucket
- Secure data transfer

5.3.2 Literature Review

5.3.2.1 Image formats

Image formats can be split into two main groups lossless and lossy formats, with lossless formats using a compression mechanism that reduces the data footprint while maintaining a perfect representation of the original image, and lossy formats using a compression mechanism that perseveres the representation of the original image however reduces image quality [41]. Examples of lossless formats include TIFF [42], PNG [43] and WEBP [44], with lossy formats including JPEG [45], TIFF[42]. These formats have varying support with different web browsers, with TIFF not being displayed on any modern web browser and WEBP requiring modern web browsers, however, JPEG and PNG have full support from all mainstream web browsers [46].

5.3.2.2 Storing relevant data

Metadata is data the gives and describes information about the data. The paper ”Metadata practices for consumer photos” [47], explains how the Exchangeable Image File Format (EXIF) is the standard for storing administrative metadata in consumer photos and allows for effective searching through images by using metadata tags that stores information such as photo descriptions, instead of searching via the filename. EXIF is supported by JPEG [45], TIFF [42] formats [45] and recently PNG format [43]. The PNG format also offers the addition of metadata through textual data, in the form of text chunks, where the data is stored in a key-value pair, i.e. ”Title”: ”This is a title”, the key-value pairs can be created as needed the key however must be between 1 and 79 bytes long [43].

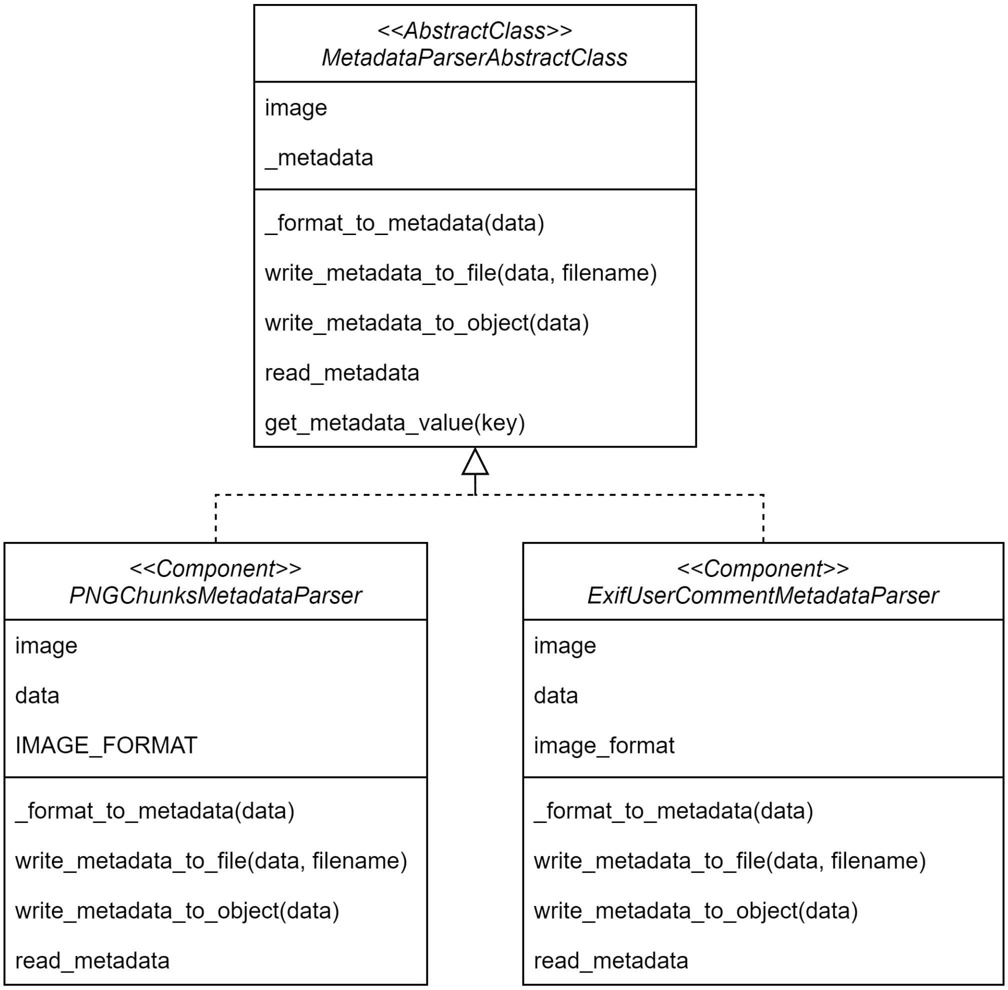
5.3.2.3 Exporting data

S3 is a service for dealing with object storage across the web; it allows for storing and retrieving objects, with many object storage providers provided support for it, including Ceph [48], [49], MinIO [50]. Within S3, the objects are stored in a bucket. An object consists of data and metadata with a unique key for identification within a bucket. Amazon provides a REST S3 API for interacting with the S3 service [51]. With this API, there are many ways data can be exported to a bucket, including mounting the bucket as a file system with an application such as S3FS-Fuse [52], syncing a folder to a bucket with an application such as rclone [53], and using an application programming interface (API) provided by amazon to communicate with the s3 instance such as Boto3 [54].

5.3.3 Implementation

An objective of the data exporter is to store the images in a high-quality format. Therefore, a lossless format is required, which means the JPEG format cannot be used due to doing lossy compression. As these images will also be displayed through a web browser, the format also needs to be supported by modern web browsers; thus, TIFF cannot be used, leaving WEBP and PNG, and as there is more support for PNG, the PNG format is used.

The PNG format offers both text chunks and EXIF for adding metadata to an image. Thus both methods of storing metadata were implemented through an abstract metadata class, as demonstrated in figure 10. An abstract class was used as it offers extensibility in the future and allows additional metadata storing types to be implemented. With this, both and future parsers can expect similar functionality when writing/reading data to/from the image.



<<AbstractClass>>
MetadataParserAbstractClass
image
_metadata
_format_to_metadata(data)
write_metadata_to_file(data, filename)
write_metadata_to_object(data)
read_metadata
get_metadata_value(key)
<<Component>>
PNGChunksMetadataParser
image
data
IMAGE_FORMAT
_format_to_metadata(data)
write_metadata_to_file(data, filename)
write_metadata_to_object(data)
read_metadata
<<Component>>
ExifUserCommentMetadataParser
image
data
image_format
_format_to_metadata(data)
write_metadata_to_file(data, filename)
write_metadata_to_object(data)
read_metadata

Figure 10: Metadata Parser UML

Even though the PNG format supports both text chunks and EXIF, the EXIF format is used within the system to remove reliance on the PNG format. So that if a different file format is required in the future or there is a requirement change, there potentially would not be the need to create another parser due to EXIF's broad support; however, this would be the case if PNG text chunks were used. The metadata is created through an alabdac metadata formatter, as this allows for a consistent metadata format. The alabdac metadata formatter implements the metadata formatter interface so that other formats can be implemented in future. The method create_metadata returns a dictionary that can be used passed in either EXIF parser PNG text chunks parser. The metadata is stored as represented in table 3. When using the EXIF format, the metadata is stored under the 'User Comment' tag as a JSON string, whereas the PNG text chunks stores the data as key-value pairs creating new keys.

Key	**Value**
USERID	000000000000
KEYWORDS	[cell]
BOUNDARYBOX	[[110,98,115,95]]
CONFIDENCE	[94]
HOSTNAME	alabdac-01
TIME	05-04-2021-13-01-53-02

Table 3: Example key-value pair metadata of an alabdac image; userID, hostname, keywords (word detected during data extraction), boundarybox (the boundary box of the words), confidence (percentage recognition of the word identified) and the time the image is saved.

As stated within the literature review, there are many different ways the data could be exported to an s3 bucket. However, as this implementation needs to be secure, the bucket keys are created as write-only. Write-only keys remove any risk of a user deleting and seeing the information in the bucket if they got access to the keys as they only allow writing. With this as a requirement, using rclone to sync a folder would require the folder to be write-only for the user and read and write for rclone, thus adding a possible attack vector. This would also be a problem using s3fs. Therefore the Boto3 API library has been used to allow for objects (images) from memory to be written to a bucket, thus removing the write-only folder security issue and removing the IO overhead of writing to a file. Boto3 also allows the s3 credentials to be stored as environment variables instead of being passed as parameters to the system, allowing the system to be more secure. It is also worth noting that the Boto3 API also allows metadata to be attached to an object during transfer. Thus a metadata parser could also have been implemented using this method, removing the issue of storing metadata within the image itself, allowing all image formats to be supported. However, this method was not implemented as it increases the dependency on the Boto3 API, and the functionality would need to be supported in other data transfer mechanisms.

As RFI may in future want to talk directly to the object storage provider, in this case, Ceph, an interface has been implemented as this ensures that future code will have the same functionality; this is demonstrated in figure 11

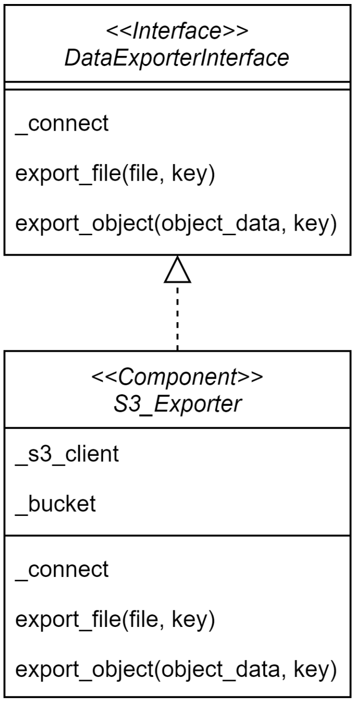


Figure 11: Data exporter package UML diagram

5.3.4 Results

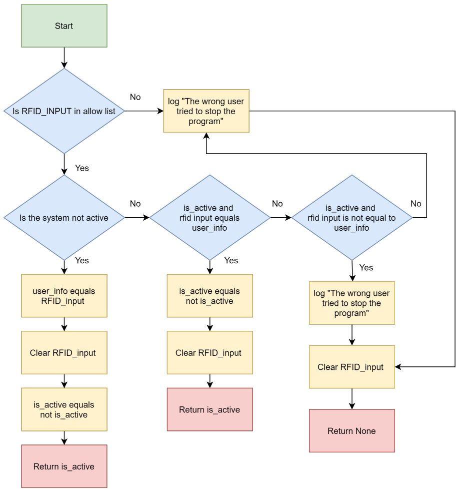
The implementation effectively satisfied the initial objectives by using the lossless PNG image format and the effective use of EXIF, which allows for different file formats to be used in future if the requirements of the system change. The metadata creator creates an easy way to ensure all the data parsed to the metadata parsers in the correct format that is easily readable. The data exporter meets the objective of being able to export images to the s3 bucket effectively and securely, through the use of environment variables and exporting objects from memory.

5.4 The System

This section discusses the system's overall design, activation of data capture, display, and how the different methods come together.

5.4.1 RFID Reader

A requirement of the system is that it is activated through an RFID card. Therefore, an RFID reader is required to start and stop the program. The RFID reader class checks to see if the RFID code is the correct length and then checks to see if it is within the allowed list provided. If that is passed, it then checks if the system is not active (not recording), then the system returns True and changes active to True; if the system is active (recording) and it is the same RFID card, then it returns False and changes active to False. This process is displayed in figure 12.



Start
Is RFID_INPUT in allow list
No
log "The wrong user tried to stop the program"
Yes
Is the system not active
No
is_active and rfid input equals user_info
No
is_active and rfid input is not equal to user_info
No
Yes
user_info equals RFID_input
Clear RFID_input
is_active equals not is_active
Return is_active
Yes
is_active equals not is_active
Clear RFID_input
Return is_active
Yes
log "The wrong user tried to stop the program"
Clear RFID_input
Return None

Figure 12: RFID Reader Flow Diagram

As RFI will in future want to talk directly to their Lightweight Directory Access Protocol (LDAP) server instead of a text file containing the allowed users held on the machine, an interface has been implemented as this ensures that future code will have the same functionality; this is demonstrated in figure 13

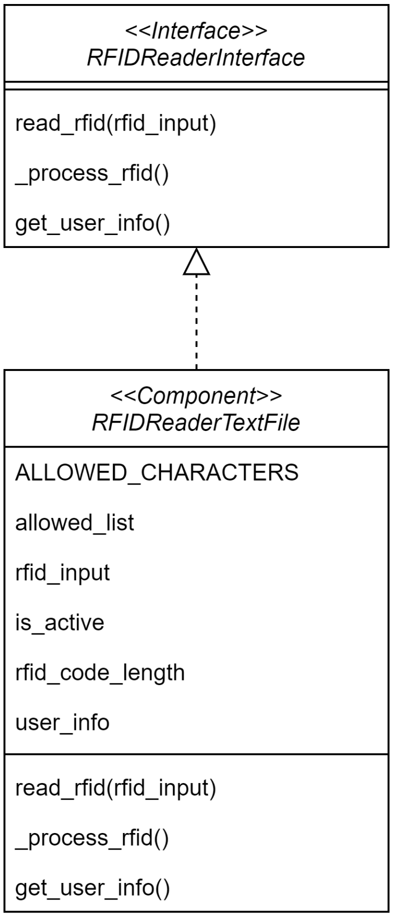


Figure 13: RFID Reader UML

5.4.2 GUI

An objective of the system is to display the camera feed and inform the user when it is recording; a graphical user interface (GUI) is required. A simple GUI was implemented in TKinter[55]. The GUI displays ”TAP to start” as well as a small resolution of the camera feed. When the RFID card is correctly read, it displays ”Recording” and initialises the intelligent data capture method to calculate the initial border, then starting the capture frame thread, which runs the intelligent data capture. This method is shown in Figure 14. When the same RFID card is reread, ”TAP to start” is displayed again. Figure 15 shows the two separate modes of the GUI.

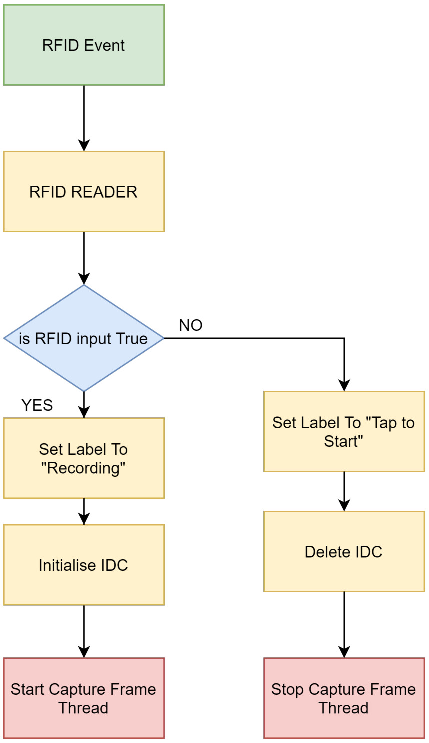
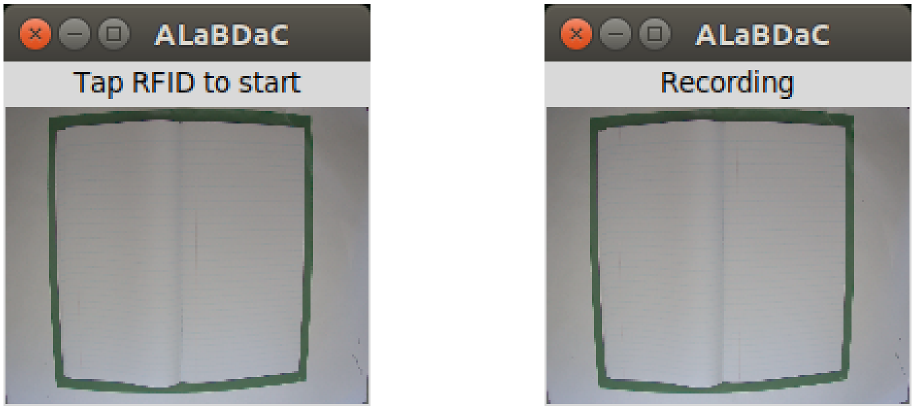


Figure 14: GUI RFID event flow diagram



(a) Display when not recording

(b) Display during recording

Figure 15: GUI Display

5.4.3 Overall Design

ALaBDaC runs solely on the Jetson Nano using two processors, one processor for the GUI and intelligent data capture, and the other for the data processing, including intelligent data extraction, metadata writing, and data exporting. The GUI and intelligent data capture are on the same process as the user must be informed when data is being captured. However, the data processing can be done at any time and can take a while to run, so it is placed on a different processor. This also means that if the GUI crashed, data would still be processed but not captured. When the RFID reader activates the recording, the data captured from the intelligent data capture and the last frame (when recording stops) is passed to the data processing process via a multiprocessor queue; this is displayed in figure 16.

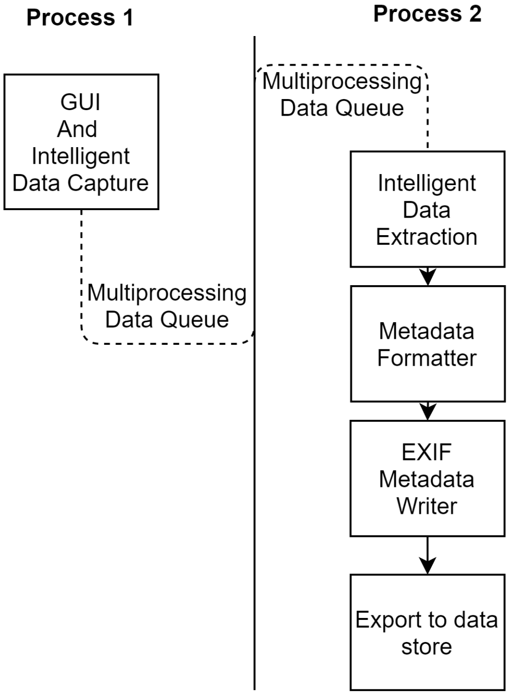

Process 1
Process 2
GUI And Intelligent Data Capture
Multiprocessing Data Queue
Multiprocessing Data Queue
Intelligent Data Extraction
Metadata Formatter
EXIF Metadata Writer
Export to data store

Figure 16: Multiprocessing setup of ALaBDaC

All of the data processing happens in memory on the Jetson Nano such that no data is written to the SD card; once the images are processed,they are then exported to the S3 data store, as shown in figure 17.

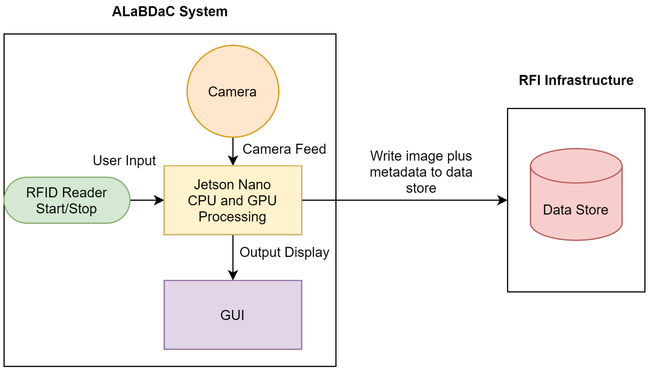


Figure 17: System Architecture: Started and Stopped with an RFID Card, the processing is on the Jetson Nano and storage is in RFI's data store.

6 Project Post-Mortem

6.1 Technologies

There were many different technologies used throughout this project; this section goes into detail about each dependency, disusing what is it, why it was used, and then a reflection on its use.

6.1.1 Hardware

The hardware used in this project was effective, at helping achieve the overall objectives. The GPU on the Jetson Nano was very useful as it allowed the AI inference to be run on the board itself, which was the primary reason the Jetson Nano was used over the Raspberry Pi 4 Model B. The Camera and Lens was effective and collection high resolution images even though it did require hardware and software modifications to work. In future the Arducam Camera IMX477 HQ Camera Module [56] should be used as it does not require the hardware modification step, and does not significantly increase the cost of the unit. The RFID reader worked effectively and achieved the goal of reading the RFID cards. Overall the hardware, selected was good and was effective and within budget.

6.1.2 Programming Language

6.1.2.1 Python

Python is a high level, interpreted programming language [57] and was considered due to the vast amount of packages and widespread support, with PyPI supporting over 300 thousand projects [58]. However, there were questions around Python performance due to limited resources on the Jetson Nano and it being an interpreted language. Therefore the C programming language, which is a low-level compile language [59], was considered as it would have allowed for specific hardware acceleration and would potentially make better use of the hardware. However, ultimately Python was used for this project as it allowed for quick and easy development due to the developer's previous experience with the language.

6.1.2.2 Bash

Bash is a command language, and UNIX shell provided [60] and was used in this project to automate the installation of packages and dependencies. It was used due to the familiarity with the developer as well as ease of creating. In retrospect, a better tool to use would have been Ansible [61] or similar as this would allow easy provisioning of installations across multiple devices.

6.1.3 Image manipulation

6.1.3.1 Pillow

Pillow is an image processing package built for python [62]. It was used throughout this project to do just that, from resizing images to adding PNG metadata. scikit-image [63] could have also been used for image manipulation. However, it was not selected as it didn't support PNG chunks metadata or Exif metadata. Overall, Pillow caused no issues within the project and was a compelling package for the image processing tasks it was used for, helping to meet the project objectives.

6.1.3.2 OpenCV

"OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library" [27]. This library was used as it came pre-installed on the Jetson nano image. It also offered an effective and efficient way to run computer vision functions in Python and allowed the CSI camera to be set up and run efficiently. OpenCV was an essential package in this project. It reduced the complexity of the task by providing effective and efficient algorithms for RGB to HSV conversion and background subtraction to name a few.

6.1.3.3 Numpy

"NumPy is the fundamental package for scientific computing in Python" [64]. NumPy was mainly used in this project to store the images as arrays as the NumPy array is a faster and more efficient method than using python default lists[64].

6.1.3.4 Piexif

Piexif is a simple EXIF manipulator package for python [65]. It was used to modify EXIF metadata. Pillow does allow for EXIF metadata manipulation. However, the developer found this package easier to use as it was solely designed to do EXIF manipulation, so the package was more intuitive than Pillow. The package allowed for the objectives of metadata to be easily read and written to be easily met.

6.1.4 Data Transfer

6.1.4.1 Boto3 and botocore

Boto3 and botocore are python packages provided and developed by Amazon to communicate with the s3 instances [54]. They were used to implement the data transfer from the Jetson Nano to the RFI's data store. These packages were used as amazon is the developer and maintainer of the S3 protocol [51] and provided the correct functionality. Boto3 caused no problems in the project and allowed for an effective solution to be implemented.

6.1.5 Graphical User Interface (GUI)

6.1.5.1 Tkinter

Originally PyGTK [66] was used to implement the GUI as RFI had experience in it. However, there were issues with displaying the camera feed. So Tkinter

[55] was used instead as it is Python's default GUI and is provided by default, and the developer was able to implement a working GUI with minimal hassle.

6.1.6 Machine Learning

6.1.6.1 YOLO

Originally PyTorchYolo [67] was used to implement YOLOv3 [38]. As it used PyTorch [68] which the developer had experience with, and it worked fine on the virtual machine provided by RFI. However, this implementation of YOLO had difficulty running on the Jetson Nano, taking a long time to process and image, and it was complicated to install due to its dependencies not being built for the ARM architecture. Therefore this implementation was changed to use Darknet [69]. Darknet is an open source neural network that provides C++, and Python APIs that provides an implement of the YOLOv3 algorithm. It requires making the project from the source, which is negative; however, a Bash script was implemented to automate this process. In retrospect, PyTorchYolo should have been run and tested on the Jetson Nano first to gauge whether it was suitable and would run efficiently. Fortunately Darknet was a suitable alternative that ran effectively.

6.1.6.2 ScrabbleGan

ScrabbleGan is a handwriting synthesiser [35]. It was used to generate the handwriting datasets, as it was well documented and easy to setup. ScrabbleGan caused no problems and produced realistic data.

6.2 Project management

Project management is an integral part of any project, as effective or ineffective management can ultimately decide the outcome of a project. The ALaBDaC project was managed through various techniques; from meetings with supervisors, version control, code management, and sprint planning, all of which are discussed in detail in this section. The techniques also allowed for the professional standards of the University [11] and BCS [12] to be upheld.

6.2.1 Supervisor Meetings

6.2.1.1 Plymouth

An initial meeting was had with Dr David Walker, where the project scope was proposed and accepted. From that point, there were meetings every other week with fellow students, where we went round in a group and discussed; what had been achieved, what was blocking and the next steps. These meetings were helpful as they provided a way to gauge progression against other people's progress who were doing similar projects; it also offered an opportunity to discuss problems and discussion over the best approaches and offer advice, based on the developers previous and current experience.

6.2.1.2 RFI

At the beginning of the project, a weekly meeting was set up with Dr Mark Basham (Science Director Artificial Intelligence) and Dr Laura Shemilt (Research Software Engineer) from RFI, who were supervising the project from their side. The initial meetings were to discuss the initial problem and potential methods in which it could be tackled; with further meetings being a place to give updates and discuss problems. They followed this format:

- What has been done
- What is planned for next week
- Any questions/ problems that have come up
- Discussion about the project

In addition to this agenda, demos of the project were given to make sure the project was on track and meeting requirements. These meetings were invaluable for helping keep the project on track and offered an opportunity to discuss ideas and different approaches to the project.

As well as these meetings, a weekly meeting with the Artificial Intelligent Theme group was attended. This was to help gain a further understanding of the use of Artificial Intelligence within RFI and a chance to discuss the project with the wider RFI team. These meetings were helpful as they gave insight into different technologies and techniques applicable to the project.

6.2.2 Version Control

Version control is a method of tracking and managing changes to a software code base [70]. To manage and maintain the project, the version control system Git [70] was used. A repository was created on the rosalindfrankininstitute GitHub account, with the name of alabdac. A feature branch and merge methodology was followed, see figure 18 for example, such that a branch would be made for a feature and developed on and then, once complete, merged to the master branch.

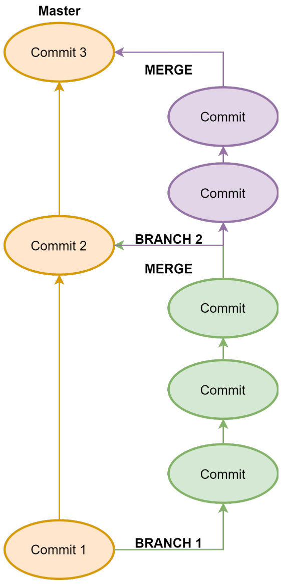
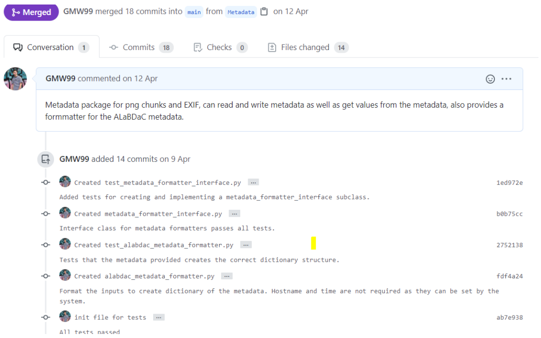


Figure 18: Branch and merge version control diagram

This ensured that minimal merge conflicts occurred as each branch was working on a specific atomic feature. This method also allowed for detailed pull requests that effectively outline the feature being added, allowing the developer to have a detailed history of each feature implemented. As well as this, it allows the developer to see changes and how changes were made throughout the code base through the use of detailed commits that had a title and the description of the changes made, see figure 19 for example, commit.



GMW99 merged 18 commits into main from Metadata on 12 Apr

GMW99 commented on 12 Apr

Metadata package for png chunks and EXIF, can read and write metadata as well as get values from the metadata, also provides a formmatter for the ALaBDaC metadata.

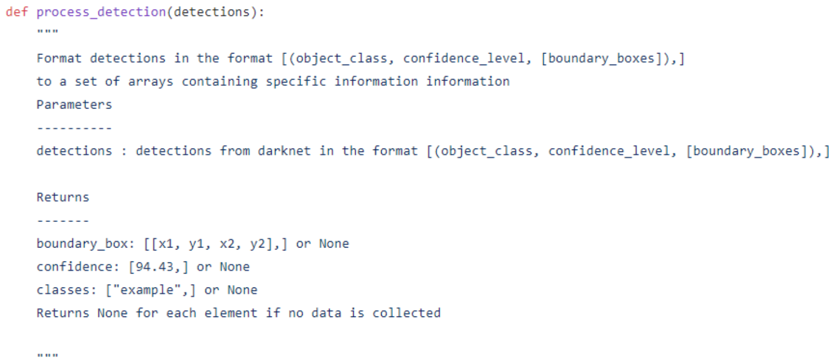
GMW99 added 14 commits on 9 Apr

```
Created test_metadata_formatter_interface.py ...                                  1ed972e
Added tests for creating and implementing a metadata_formatter_interface subclass.
Created metadata_formatter_interface.py ...                                       b0b75cc
Interface class for metadata formatters passes all tests.
Created test_alabdac_metadata_formatter.py ...                                    2752138
Tests that the metadata provided creates the correct dictionary structure.
Created alabdac_metadata_formatter.py ...                                         fdf4a24
Format the inputs to create dictionary of the metadata. Hostname and time are not required as they can be set by the
system.
init file for tests ...                                                           ab7e938
All tests passed
```

Figure 19: Example detailed commits and pull request on ALaBDaC repository

6.2.3 Code Management

As this project required writing and developing code, it is essential to manage and develop correctly. To make sure that the code was appropriately commented, Doc Strings [71] were used. Doc Strings is a string literal that occurs after defining a module, class, method, and function. They outline what the module, class, method or function does. Figure 20 shows an example Docstring. This made the code and project more manageable and maintainable. Especially when returning to old code to either add functionality or use the modules developed, allowing for an easy way to know what the code was meant to do without having to delve into the code, thus saving time.



```
def process_detection(detections):
    """
    Format detections in the format [(object_class, confidence_level, [boundary_boxes]),]
    to a set of arrays containing specific information information
    Parameters
    ----------
    detections : detections from darknet in the format [(object_class, confidence_level, [boundary_boxes]),]

    Returns
    -------
    boundary_box: [[x1, y1, x2, y2],] or None
    confidence: [94.43,] or None
    classes: ["example",] or None
    Returns None for each element if no data is collected

    """
```

Figure 20: Example Doc String from ALaBDaC code base

To manage the codebase and ensure that there was a consistent and well readable style throughout, the auto-formatting tool 'black'[72], was used. Auto formatting enforces the best practices, reducing the time spent refactoring and solving issues that could have been avoided if the code was formatted correctly. The use of black throughout this project highlighted 'code smells' [73] such as too many parameters and excessively long lines of code early. This meant the code smells could be appropriately managed before becoming an issue, thus saving time from fixing potential problems further down the line and creating a more maintainable codebase for future development.

6.2.4 Dependency Management

As the project is being developed over months and has a restricted deadline, the latest packages were used when creating the software. The version was locked to ensure that no issues would be incurred due to major package updates. Dependencies are also kept to a minimum as this reduces the surface for supply chain attacks [74].

6.2.5 Design Principles

Design Patterns and Principles are essential when developing software to ensure it stays readable and well maintainable in the future, which is vital for this project. Therefore the SOLID principles [75] were adhered to throughout the project as demonstrated in the codebase.

6.2.6 Sprint Plans

The approach to this project was agile; therefore, different project requirements were split into sprints. The weekly reports can be seen in Appendix C

6.2.6.1 Sprint Zero: Setup and kit ordering

The Aim: This sprint aimed to work out the exact requirements, define the Minimal Viable Product (MVP) that would allow research to then take place, order the hardware, and discuss the problem with RFI and Dr David Walker.

What was done: First, a meeting between the developer and RFI was set up to discuss the project and define the scope, then the project was proposed to Dr David Walker and accepted. A further meeting was arranged with RFI to specify the project's requirements further and decide what the MVP of the project would to allow for research and investigation to take place once implemented, see Appendix D for requirements and if they where met and see Appendix E for the MVP description. After this, the hardware was selected and bought, and a project backlog was created using the office 365 planner.

Reflection on the sprint: The initial sprint went well; communication with RFI was good, with a mutual understanding of what was expected from the project. The requirements/objectives seemed achievable and exciting problems to solve.

6.2.6.2 Sprint One: Install Setup: GUI and Read RFID, MVP

The Aim: This sprint aimed to install the hardware, set up the GUI and RFID reader to meet the minimal requirements to allow the research aspect of the project to begin.

What was done: The kit arrived, and a mock lab was created with paper so the system could be tested. All the parts were connected. The assumption that the Raspberry Pi High-Quality Camera would work natively with Jetson Nano was wrong, and thus a hardware modification was required to make it work. After discussion with RFI, this was accepted, and the modification was done, and the appropriate drivers were installed to run the camera. This process was then documented for future installations. Then a simple GUI to display the camera feed was implemented in PyGTK. However, this was a complicated process, and it showed the camera with lag, so this was reimplemented in Tkinter, which had no issues. Originally it was planned to get the RFID reader to read the RFI RFID Cards and connect to their LDAP. However, we could not get an RFI RFID Card made due to the developer not being an employee. So the requirement was instead changed to read the RFID cards provided and check them against a list of allowed RFID cards in a read-only allowed file as this was not the project's primary objective and would require internal work from RFI. Then the program was set up to store a frame on the device every 10 seconds to meet the minimum requirements effectively.

Reflection on the sprint: This sprint went well. It was frustrating that the Raspberry Pi High-Quality Camera did not work straight away and required a hardware modification. After searching the problem, there was a replacement camera that had this modification already done. The agile ability to change the requirements due to unexpected issues showed good project management and good communication with RFI. The completion of the MVP has allowed for the research and investigation part of the project to begin.

6.2.6.3 Sprint Two: Data Capture

The aim: This sprint aimed to investigate current methods and develop a way to capture data intelligently.

What was done: Initially, the approach to this problem was to use object detection techniques to recognise when a hand was on the page. However, it was realised that this method would be challenging to implement as RFI could not collect a dataset that could be used to train such an algorithm. After this realisation, unsupervised and classical computer vision methods were investigated and tried, covering methods from canny edge detection [19] and hough transforms [20] to colour segmentation [21] and background subtraction [76]. Once the colour segmentation and background subtraction method was working, a demo was given to RFI to validate the method and check they were happy with the approach taken.

Reflection on the sprint: There were several challenges with this stage of the project, as the developer had expected this part to be solved through modern supervised machine learning techniques, so time had been spent researching such techniques to find that a dataset couldn't be collected or effectively created. It made this problem a lot harder as the current literature covered supervised methods of object detection [39], [40], [38]. This required the developer to change the approach and instead look into unsupervised and classical techniques, which needed further literature review. This took a long time as these methods were harder to implement/effectively tune and evaluate, as there was no previous/objective data to test against. However, overall this sprint went well and resulted in a suitable method with the time taken researching different techniques being an efficient use of the developers time. It allowed for many different techniques to be tried and tested to develop a solution that ultimately worked.

6.2.6.4 Sprint Three: Data Extraction

The Aim: This sprint aimed to create a method to extract keywords from the lab book.

What was done: It was initially assumed that there would be a package or pre-existing system for handwriting detection, like there was for optical character recognition, with systems such as Tesseract [77], [78]. This not being the case made the problem significantly harder as not being able to make or having a dataset was an issue again, as these supervised object detection requires a labelled dataset. An investigation was done into handwriting datasets; however, none contained keywords scientific terms, and due to the project's objective, this would not suit. At this point, synthetic datasets were looked into, and the YOLO algorithm was trained on a synthetic dataset of numbers on a white background generated using the MNIST dataset [79] to recognise digits on a page. This produced promising results and was able to identify some digits. This gave the developer enough confidence to look for a handwriting synthesiser to generate handwriting samples. To begin with, ScrabbleGan was trained on the developers home machine for five epochs to see the results, which looked promising, after calculating that it would take approximately ten days if the machine were left alone. This was not feasible due to other requirements. Therefore a virtual machine with an NVIDIA V100 TENSOR CORE GPU was provided from RFI for training, which reduced the training time down to 2.5 days. A module was then made that would make a synthetic dataset, using the words generated by ScrabbleGan and placing them on a white background in different sizes. However, after training YOLO, it was clear that it struggled with words of a similar length, and it would detect words within the lab book background and not the writing itself. Then the program was updated to include backgrounds and rotation of the words to help the algorithm detect the difference between the words as the synthetic data better represented the real-world data.

Reflection on the sprint: The developer was pleased with how this sprint went, as with the initial assumption being wrong, it turned what seemed to be an easy task into a difficult one. With more time, more could have been done to make the synthetic data look more realistic, i.e. adding greyness, words in sentences. However, given the results, the developer was happy that this method had shown promise. With the ending solution being an elegant and practical way forward to the problem presented.

6.2.6.5 Sprint Four: Data exporter

The Aim: To transfer the image and relevant data (metadata) to the central RFI data store.

What was done: Initially, the system was designed so that this objective would be managed by the RFI-FILE monitor [80]. The RFI file monitor would have meant the data could have been placed in a folder on the system, and the data would have automatically transferred. However, during the project, it was made clear that this would not be ready in time. Therefore several different approaches were considered before Boto3 was used. This leads to an interface class development, outlining the functions a data transfer class should have. So that RFI could extend the functionality in the future easily. Once data could be transferred, a discussion with RFI was had over what data was relevant to be stored in the metadata. With this, a metadata formatter was implemented to provide a way to keep the format consistent. Then, as this sprint was running smoothly, the PNG chunks and EXIF metadata storage formats were implemented.

Reflection on the sprint: This sprint had no significant issues and went very well, mainly due to the developers' previous experience interacting and accessing objects with s3 and using effective packages to help with implementing the metadata part. This sprint ended with an effective solution being developed.

6.2.6.6 Sprint Five: All Together

The Aim: Bring everything together, remove redundant dependencies and write everything up.

What was done: The final stage of the project was bringing the parts of the project together. No new functionality was added; instead, quality of life improvements were made to make sure RFI could easily maintain it in the future, and all relevant code was added to the repository that hadn't previously. With this, the dependency for PyTorchYolo was replaced with Darknet due to its difficulty to install and run on the Jetson Nano, and the installation documentation was written up. During this time, the developer gave a talk to the RFI team and the Theme Advisory Panel (TAP) (a panel that aims to check that the theme is on-topic) outlining what had been achieved during the project. The report was then written up in neat.

Reflection on the sprint: This sprint went very well. The talk with the TAP was very well received, with a lot of interest, in the implementation, and all code was in the repository, worked and well documented. The report writing was smooth, with no significant issues.

6.3 Methods

In this section the methods are evaluated, with potential areas to investigate or implement in future.

6.3.1 Intelligent Data Capture

Although the tuning is difficult, this was an exemplary implementation, which could have been improved by creating a set of tools to display what the different setting would produce. Given more time and a larger budget, further techniques could have been investigated to reduce the dependence on colour segmentation and background detection. Some methods considered include using an infrared camera to detect a hand entering the lab book as it can be assumed that the hand will have a higher operating temperature than the lab book, a two-camera setup to perform depth perception as the hand will be closer to the camera than the lab book.

6.3.2 Intelligent Data Extraction

This implementation allows RFI to create a dataset of any words that can then be used to train an object detection algorithm that works effectively meeting the objective. Given more time, the developer would have investigated the performance of different object detection algorithms as well as using ScrabbleGan to generate a dataset of letters. To get the object detection algorithm to detect the letters on the page and then use Natural Language Processing (NLP) techniques such as word segmentation to stitch the letters together to workout the keywords [81], to compare the effectiveness of word vs character recognition.

6.3.3 Data transfer

The data transfer implementation effectively met all of the requirements set out for that section of the project. In the future it would be interesting to interact with the RFI-File monitor for the data transfer, as this would give RFI more control over how and when the data is transferred.

6.3.4 System

Overall the system implemented all of the initial objectives, apart from connectivity with LDAP server, due to reasons mentioned in the sprint plans. The future work for the system would be to add this connectivity to LDAP, so that the allowed list of users is not stored on the machine itself.

6.4 Testing and validation

This project followed a test-driven development approach. This meant that the tests were written before the functionality was implemented to ensure that the implementation would meet the requirements.

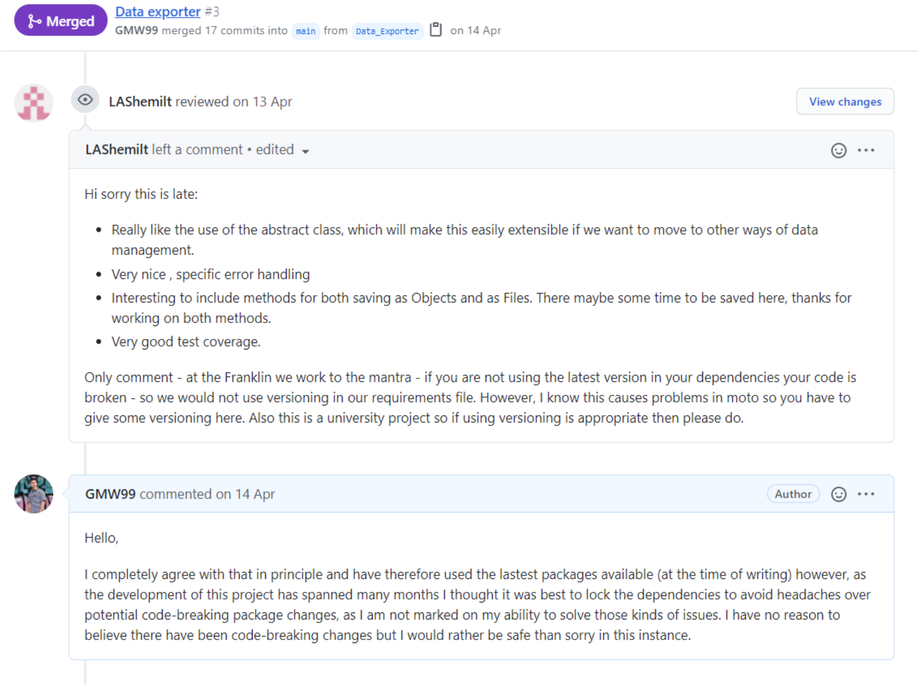
The main type of testing undertaken in this project was unit testing. Unit testing is a method of testing a small bit of code in isolation to check if it functions correctly[82]. Mock unit testing was used to test the S3_exporter class to check that it correctly interacted with an S3 data store by mocking an S3 instance. There was a total of 72 tests which covered 92% of the lines in the codebase, a 100% of the tests passed.

Manual testing was also taken out on parts that either required user input, i,e the RFID reader or required specific hardware, such as the camera or specific parameters to be able to run. This consisted of visually checking that the camera feed was displayed and checking that the RFID reader correctly read the input and that intelligent data capture worked on a series of test videos, The set of tests can be found in table 4

Test	**Expected Result**
Running the GUI displays the camera feed	Camera Feed is shown
Check RFID reader reads cards within the allowed list.	Change in GUI display to ”Recording.”
Check RFID reader that the system doesn't start when using a card, not within the allowed list.	No change in GUI display
Check that the intelligent data extraction loads appropriately.	Appropriate detections with a test image.
Check that covering the border by the amount specified leads to image capture.	Previous frames captured
Check that causing a frame difference to the amount specified leads to image capture.	Previous frames captured
Check installation script works	Everything that is needed is installed.
Closing GUI doesn't instanly stop multiprocessing process	Terminal will ouput ”Finished: Exited cleanly”
Pass arguments to the system	Expected modifications to the system are made

Table 4: Manual Tests

Usability testing, in the common sense, was not feasible in this project due to practical and legal reasons, e.g. lockdown restrictions due to coronavirus [24], which meant that the developer could not set the project up within a lab-based environment, and neither could RFI. The developer was also studying remotely, so they could not ask fellow students or family members covered under the ethics review form either. Fortunately, the aim of this project was to create an automated way to collect lab book data whilst not disrupting the workflow of the scientist as much as possible, so their only direct interaction with the system is RFID reader, which is an accepted norm within RFI as most things are activated or require an RFID card to use within the RFI Building. However, the data exporter aspect of the project was evaluated internally by RFI. The metadata format was evaluated by Dr Laura Shemilt and passed as being usable and acceptable for being processed within RFI Scicat cataloguer [83]. As well as this the S3 data exporter code was reviewed by Dr Laura Shemilt to check that it did not expose any confidential information surrounding the internal workings of RFI, which it passed, this is shown in figure 21.



Data exporter #3
GMW99 merged 17 commits into main from Data_Exporter on 14 Apr

LAShemilt reviewed on 13 Apr

LAShemilt left a comment • edited

Hi sorry this is late:

- Really like the use of the abstract class, which will make this easily extensible if we want to move to other ways of data management.
- Very nice , specific error handling
- Interesting to include methods for both saving as Objects and as Files. There maybe some time to be saved here, thanks for working on both methods.
- Very good test coverage.

Only comment - at the Franklin we work to the mantra - if you are not using the latest version in your dependencies your code is broken - so we would not use versioning in our requirements file. However, I know this causes problems in moto so you have to give some versioning here. Also this is a university project so if using versioning is appropriate then please do.

GMW99 commented on 14 Apr

Hello,

I completely agree with that in principle and have therefore used the lastest packages available (at the time of writing) however, as the development of this project has spanned many months I thought it was best to lock the dependencies to avoid headaches over potential code-breaking package changes, as I am not marked on my ability to solve those kinds of issues. I have no reason to believe there have been code-breaking changes but I would rather be safe than sorry in this instance.

Figure 21: Data exporter pull request code review.

The overall feedback of the project from RFI was very positive with them being happy with the system implemented, see Appendix F for detailed feedback.

6.5 Overall

This project was managed well, with good and effective solutions being implemented for the problems. The reasoning and practical selection of the technologies and methods used allowed for the project to run smoothly. Overall, the practical project management of this project allowed it to succeed.

7 Conclusion

This project was in collaboration with the RFI. The key objective was to investigate and develop an automated way to collect lab book information as the user writes into the lab book and record it to a central data store. To solve the inherent issues of using paper lab books, including; accessing, reviewing, searching, and being hard to practically backup.

The system presented in the body of this report effectively met that objective and solved the problem. It solved the problem using classical computer vision methods to capture data, machine learning to extract the data, and effective software development to export data and run the system. That, in turn, created a usable proof of concept piece that RFI are delighted with and plan to use in production within their labs.

In the future, the area that needs further investigation and improvement is the synthetic dataset generation. It has shown potential as described by the main body of this report but could be improved if made to look more realistic through further image processing.

Overall, the developer is very pleased with this project's outcome. Given its inherent difficulties, no datasets, no preexisting solutions and hardware limitations, all of which increased the complexity of the problem. However, these issues lead to creating a novel and attractive solution that worked around them, ultimately solving the problem presented.

References

[1] *The good laboratory practice regulations 1999*, 1999. [Online]. Available: https://www.legislation.gov.uk/uksi/1999/3106/contents (visited on 04/02/2021).

[2] *Freedom of information act 2000*, 2000. [Online]. Available: https://www.legislation.gov.uk/ukpga/2000/36/contents (visited on 04/02/2021).

[3] *Data protection act 2018*, 2018. [Online]. Available: https://www.legislation.gov.uk/ukpga/2018/12/contents (visited on 04/02/2021).

[4] J. Prewett, *Laboratory notebooks*, 2018. [Online]. Available: https://warwick.ac.uk/services/ris/research_integrity/code_of_practice_and_policies/research_code_of_practice/datacollection_retention/laboratory_notebooks/ (visited on 04/02/2021).

[5] R. F. Institute, 2020. [Online]. Available: https://www.rfi.ac.uk/about/ (visited on 10/27/2020).

[6] labforward gmbh, *labfolder*. 2021. [Online]. Available: https://www.labfolder.com/ (visited on 04/02/2021).

[7] B. Inc, *Electronic lab notebook-eln-labguru*, 2021. [Online]. Available: https://www.labguru.com/ (visited on 04/02/2021).

[8] E. Regulations, *General data protection regulation (gdpr)*, 2018. [Online]. Available: https://gdpr-info.eu/ (visited on 04/02/2021).

[9] *Open research*, 2021. [Online]. Available: https://www.ukri.org/about-us/policies-standards-and-data/good-research-resource-hub/open-research/ (visited on 04/20/2021).

[10] *Computer misuse act 1990*, 2007. [Online]. Available: https://www.legislation.gov.uk/ukpga/1990/18/contents (visited on 04/24/2021).

[11] L. Beahan, *Student code of conduct and disciplinary procedure*, version 3.1, 2020. [Online]. Available: https://www.plymouth.ac.uk/uploads/production/document/path/17/17958/Code_of_conduct_Aug_2020.pdf (visited on 04/05/2021).

[12] B. The Chartered Institute for IT, *Bcs code of conduct*, 2021. [Online]. Available: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/ (visited on 04/05/2021).

[13] N. LTD, *Jetson nano developer kit*. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[14] R. P. Foundation, *Raspberry pi 4 computer model b 2021*, 2021. [Online]. Available: https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf (visited on 04/02/2021).

[15] ——, *Raspberry pi high quality camera*, 2020. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry_Pi_HQ_Camera_Product_Brief.pdf (visited on 04/02/2021).

[16] ——, *Camera module v2*, 2016. [Online]. Available: `https://www.raspberrypi.org/products/camera-module-v2/` (visited on 10/16/2020).

[17] R. Szeliski, *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

[18] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[20] P. Mukhopadhyay and B. B. Chaudhuri, "A survey of hough transform," *Pattern Recognition*, vol. 48, no. 3, pp. 993–1010, 2015, ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2014.08.027`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0031320314003446`.

[21] L. Shuhua and G. Gaizhi, "The application of improved hsv color space model in image processing," in *2010 2nd International Conference on Future Computer and Communication*, vol. 2, 2010, pp. V2–10–V2–13. DOI: `10.1109/ICFCC.2010.5497299`.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[23] Y. Xiong, *Fast and accurate document detection for scanning*, 2016. [Online]. Available: `https://dropbox.tech/machine-learning/fast-and-accurate-document-detection-for-scanning` (visited on 11/20/2020).

[24] *Coronavirus (covid-19): Guidance and support 2021*, 2021. [Online]. Available: `https://www.gov.uk/coronavirus` (visited on 03/15/2021).

[25] A. Mittal, A. Zisserman, and P. H. Torr, "Hand detection using multiple proposals.," in *Bmvc*, vol. 2, 2011, p. 5.

[26] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, *Concrete problems in ai safety*, 2016. arXiv: `1606.06565 [cs.AI]`.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] J. Balsam, H. A. Bruck, Y. Kostov, and A. Rasooly, "Image stacking approach to increase sensitivity of fluorescence detection using a low cost complementary metal-oxide-semiconductor (cmos) webcam," *Sensors and Actuators B: Chemical*, vol. 171-172, pp. 141–147, 2012, ISSN: 0925-4005. DOI: `https://doi.org/10.1016/j.snb.2012.02.003`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0925400512001177`.

[29] *Why is handwriting recognition so difficult for ai?* [Online]. Available: `https://www.thinkautomation.com/bots-and-ai/why-is-handwriting-recognition-so-difficult-for-ai/`.

[30] W. Kacalak, K. D. Stuart, and M. Majewski, "Selected problems of intelligent handwriting recognition," in *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, P. Melin, O. Castillo, E. G. Ramírez, J. Kacprzyk, and W. Pedrycz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 298–305, ISBN: 978-3-540-72432-2. DOI: 10.1007/978-3-540-72432-2_30. [Online]. Available: https://doi.org/10.1007/978-3-540-72432-2_30.

[31] S Preetha, I. Afrid, P Karthik Hebbar, and S. Nishchay, "Machine learning for handwriting recognition," *International Journal of Computer (IJC)*, vol. 38, no. 1, pp. 93–101, 2020.

[32] G. Albuquerque, T. Lowe, and M. Magnor, "Synthetic generation of high-dimensional datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2317–2324, 2011. DOI: 10.1109/TVCG.2011.237.

[33] S. Gaur, S. Sonkar, and P. P. Roy, "Generation of synthetic training data for handwritten indic script recognition," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 491–495. DOI: 10.1109/ICDAR.2015.7333810.

[34] P. P. Roy, A. Mohta, and B. B. Chaudhuri, *Synthetic data generation for indic handwritten text recognition*, 2018. arXiv: 1804.06254 [cs.CV].

[35] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman, "Scrabblegan: Semi-supervised varying length handwritten text generation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[36] U.-V. Marti and H. Bunke, "The iam-database: An english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[37] D. INC, *List of english words*, https://github.com/dwyl/english-words, 2020.

[38] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, 2018. arXiv: 1804.02767 [cs.CV].

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[41] R. H. Wiggins, H. C. Davidson, H. R. Harnsberger, J. R. Lauman, and P. A. Goede, "Image file formats: Past, present, and future," *Radiographics*, vol. 21, no. 3, pp. 789–798, 2001.

[42] A. D. Association *et al.*, *Tiff revision 6.0*, 1992. [Online]. Available: https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf (visited on 03/15/2021).

[43] M. Adler, T. Boutell, J. Bowler, C. Brunschen, A. M. Costello, L. D. Crocker, A. Dilger, O. Fromme, J.-l. Gailly, C. Herborth, A. Jakulin, N. Kettler, T. Lane, A. Lehmann, C. Lilley, D. Martindale, O. Mortensen, K. S. Pickens, R. P. Poole, G. Randers-Pehrson, G. Roelofs, W. v. Schaik, G. Schalnat, P. Schmidt, M. Stokes, T. Wegner, and J. Wohl, *Portable network graphics (png) specification (second edition)*, 2003. [Online]. Available: `https://www.w3.org/TR/2003/REC-PNG-20031110/` (visited on 03/15/2021).

[44] G. Developers, "A new image format for the web," *Internet publication: https://developers.google.com/speed/webp*, 2020. (visited on 03/15/2021).

[45] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[46] *Image file type and format guide 2021*, 2021. [Online]. Available: `https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types` (visited on 03/15/2021).

[47] J. Tesic, "Metadata practices for consumer photos," *IEEE MultiMedia*, vol. 12, no. 3, pp. 86–92, 2005. DOI: `10.1109/MMUL.2005.50`.

[48] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, ser. OSDI '06, Seattle, Washington: USENIX Association, 2006, 307–320, ISBN: 1931971471.

[49] *Ceph object gateway s3 api — ceph documentation*, 2021. [Online]. Available: `https://docs.ceph.com/en/latest/radosgw/s3/` (visited on 03/02/2021).

[50] *Minio — learn how to deploy minio with amazon s3*, 2021. [Online]. Available: `https://docs.min.io/docs/minio-gateway-for-s3.html` (visited on 03/02/2021).

[51] *Amazon simple storage service*, 2006. [Online]. Available: `https://docs.aws.amazon.com/AmazonS3/latest/API/s3-api.pdf` (visited on 03/22/2021).

[52] R. Rizun, D. Moore, A. Petrescu, B. LeMasurier, T. Nakatani, and A. Gaul, *S3fs-fuse*, 2019. [Online]. Available: `https://github.com/s3fs-fuse/s3fs-fuse` (visited on 03/02/2021).

[53] N. Craig-Wood, *Rclone*, 2021. [Online]. Available: `https://rclone.org/` (visited on 11/01/2020).

[54] *Boto3 documentation*, 2021. [Online]. Available: `https://boto3.amazonaws.com/v1/documentation/api/latest/index.html` (visited on 03/02/2021).

[55] G. Van Rossum, *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.

[56] Arducam, *Imx477 hq camera module datasheet*$_2$021, 2021. [Online]. Available: https://www.uctronics.com/download/Amazon/B0249_IMX477_HQ_Camera_for_Jetson_Datasheet.pdf (visited on 05/02/2021).

[57] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[58] *Python package index - pypi.* [Online]. Available: https://pypi.org/ (visited on 03/28/2021).

[59] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd. Prentice Hall Professional Technical Reference, 1988, ISBN: 0131103709.

[60] P GNU, *Free software foundation. bash (3.2. 48)[unix shell program]*, 2007.

[61] Ansible, *Ansible documentation*, 2021. [Online]. Available: https://docs.ansible.com/ansible/latest/user_guide/index.html (visited on 12/02/2020).

[62] H. van Kemenade, wiredfool, A. Murray, A. Clark, A. Karpinsky, O. Baranovič, C. Gohlke, J. Dufresne, B. Crowell, D. Schmidt, K. Kopachev, A. Houghton, S. Mani, S. Landey, vashek, J. Ware, J. Douglas, D. Caro, U. Martinez, S. Kossouho, R. Lahd, S. T., A. Lee, E. W. Brown, O. Tonnhofer, M. Bonfill, P. R. (), F. Al-Saidi, G. Novikov, and M. Górny, *Python-pillow/pillow: 8.2.0*, version 8.2.0, Apr. 2021. DOI: 10.5281/zenodo.4659051. [Online]. Available: https://doi.org/10.5281/zenodo.4659051.

[63] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, e453, Jun. 2014, ISSN: 2167-8359. DOI: 10.7717/peerj.453. [Online]. Available: https://doi.org/10.7717/peerj.453.

[64] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[65] hMatoba, *Piexif*, version 1.1.3, 2019. [Online]. Available: https://piexif.readthedocs.io/.

[66] PyGObject, *Pygobject documentation*, 2021. [Online]. Available: https://pygobject.readthedocs.io/en/latest/ (visited on 04/02/2021).

[67] L.-N. Erik and F. Vahl, *Eriklindernoren/pytorch-yolov3*, 2021. [Online]. Available: https://github.com/eriklindernoren/PyTorch-YOLOv3 (visited on 04/10/2021).

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[69] J. Redmon, *Darknet: Open source neural networks in c*, `http://pjreddie.com/darknet/`, 2013–2016.

[70] S. Chacon and B. Straub, *Pro git*. Springer Nature, 2014.

[71] D. Goodger and G. van Rossum, *Pep 257 – docstring conventions*, 2001. [Online]. Available: `https://www.python.org/dev/peps/pep-0257/` (visited on 05/18/2021).

[72] Langa, C. Willing, C. Meyer, J. Zijlstra, M. Naylor, Z. Dollenstein, C. Lees, R. Si, A.-R. Janhangeer, A. Johnson, A. Williamson, A. Huynh, A. Vandiver, A. Simon, A.-P. Ljungquist, A. Thorp, A. Zhou, Andrey, A. Freeland, A. Sottile, A. Buijk, A. Borbornah, A. Malyshev, A. H. Drewsen, A. Fackler, A. KC, B. Taşkaya, B. Wohlwend, B. Woodruff, B. Raghunathan, B. Bucher, B. Cannon, B. Bugyi, B. Forbes, C. Lind, Charles, C. Reid, C. Clauss, C. Heimes, C. Wooters, C. Rose, C. Oxley, Cong, C. R. Lees, D. Davison, D. Hahler, D. M. Capella, D. Esposti, D. Hotham, D. Lukes, D. Szotten, D. Laxalde, D. Thor, dylanjblack, E. Treuherz, E. Hessman, F. Kohlgrüber, F. Thiery, Francisco, G. Tagliabue, G. Gandenberger, G. P. Smith, G. Camargo, hauntsaninja, H. Alqattan, Heaford, H. Barrera, H. v. Kemenade, H. Schlawack, I. Katanić, J. Kadlubiec, J. Warczarek, J. Hnátek, J. Fried, J. Friedland, jgirardet, J. Brännlund, J. Jia, J. Antonakakis, J. Dufresne, J. Obrist, J. Wareing, J. Nazario, J. Larson, J. Bode, J. Holland, J. Cannon, J. Padilla, J. L. C. Rodríguez, kaiix, K. McLaughlin, K. Leinweber, K. Smiley, K. Ralph, K. Kirsche, K. Hausmann, K. Sunden, L. Chan, L. Groh, L. Carvalho, L. Sterbic, LukasDrude, M. Hossam, Mariatta, M. VanEseltine, M. Clapp, M. Walster, M. Smolens, M. Aquilina, M. Flaxman, M. J. Sullivan, M. McClimon, M. Gaiowski, Mike, mikehoyio, M. h. Kim, M. Shubernetskiy, MomIsBestFriend, N. Goldbaum, N. Hunt, Neraste, N. Waxweiler, O. Lev, O. Daniel, otstrel, P. Galindo, P. Ganssle, P. Meinhardt, P. Bengtsson, P. Grayson, P. Stensmyr, pmacosta, Q. Pradet, R. Schmitt, R. Valles, R. Fearn, R. Jha, R. Bedford, R. Davis, R. Verschelde, S. Salonen, S. Cormier-Iijima, S. Dasgupta, Sergi, S. Stevenson, Shantanu, shaoran, S. Fujino, springstan, S. Korokithakis, S. Rosen, S. M. Vascellaro, S. Kapil, S. Eustace, T. Amuyal, Terrance, T. Lu, T. Grainger, T. Gates, T. Swast, Timo, T. Fleming, T. Christie, T. Narlock, T. Hombashi, T. Chandra, T.-p. Chung, U. Shah, utsav dbx, vezeli, V. Skyttä, V. B. Sharma, V. Emelianov, williamfzc,

w. bolsterlee, Yazdan, Y. Høiseth, Y. Karabas, and Z. Hatfield-Dodds, *Black the uncompromising python code formatter*, 2021. [Online]. Available: `https://github.com/psf/black` (visited on 05/02/2021).

[73] A. Yamashita and L. Moonen, "Do developers care about code smells? an exploratory survey," in *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013, pp. 242–251. DOI: `10.1109/WCRE.2013.6671299`.

[74] *Supply chain security guidance*, 2018. [Online]. Available: `https://www.ncsc.gov.uk/collection/supply-chain-security` (visited on 04/18/2021).

[75] T. Gilb and S. Finzi, *Principles of software engineering management.* Addison-wesley Reading, MA, 1988, vol. 11.

[76] M. Piccardi, "Background subtraction techniques: A review," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, 2004, 3099–3104 vol.4. DOI: `10.1109/ICSMC.2004.1400815`.

[77] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, IEEE, vol. 2, 2007, pp. 629–633.

[78] R. Smith, A. Abdulkader, R. Antonova, N. Beato, J. Breidenbach, S. Charron, P. Cheatle, S. Crouch, D. Eger, S. Huddleston, D. Johnson, R. Katikam, T. Kielbus, D.-S. Lee, R. Liu Zongyi (Joe)and Moss, C. Newton, M. Reimer, M. Renn, R. Romano, C. Russon, S. Saxena, M. Seaman, F. Shafait, H. Takenaka, R. Unnikrishnan, J. Wanke, P. P. Xiu, A. Ziem, O. Zuniga, Z. Podobný, J. Regan, J. R. Barlow, S. Brechtken, T. Breuel, A. Dovev, M. Ettl, S. D. Kumar, N. Metzger, T. Morris, T. Müller, E. Pugin, R. Sachunsky, R. Schietekat, S. M. Vaidya, R. Watts, S. Weil, N. White, and A. Zaitsev, *Tesseract-ocr/tesseract*, 2019. [Online]. Available: `https://github.com/tesseract-ocr/tesseract` (visited on 01/05/2021).

[79] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. DOI: `10.1109/MSP.2012.2211477`.

[80] T. Schoonjans and L. Shemilt, *Rfi-file-monitor*, 2021. [Online]. Available: `https://rosalindfranklininstitute.github.io/rfi-file-monitor/` (visited on 04/15/2021).

[81] J. R. Van Aken, "A statistical learning algorithm for word segmentation," *arXiv preprint arXiv:1105.6162*, 2011.

[82] B. Okken, *Python testing with Pytest: simple, rapid, effective, and scalable.* Pragmatic Bookshelf, 2017.

[83] *Scicat project*, 2021. [Online]. Available: `https://scicatproject.github.io/` (visited on 11/04/2020).

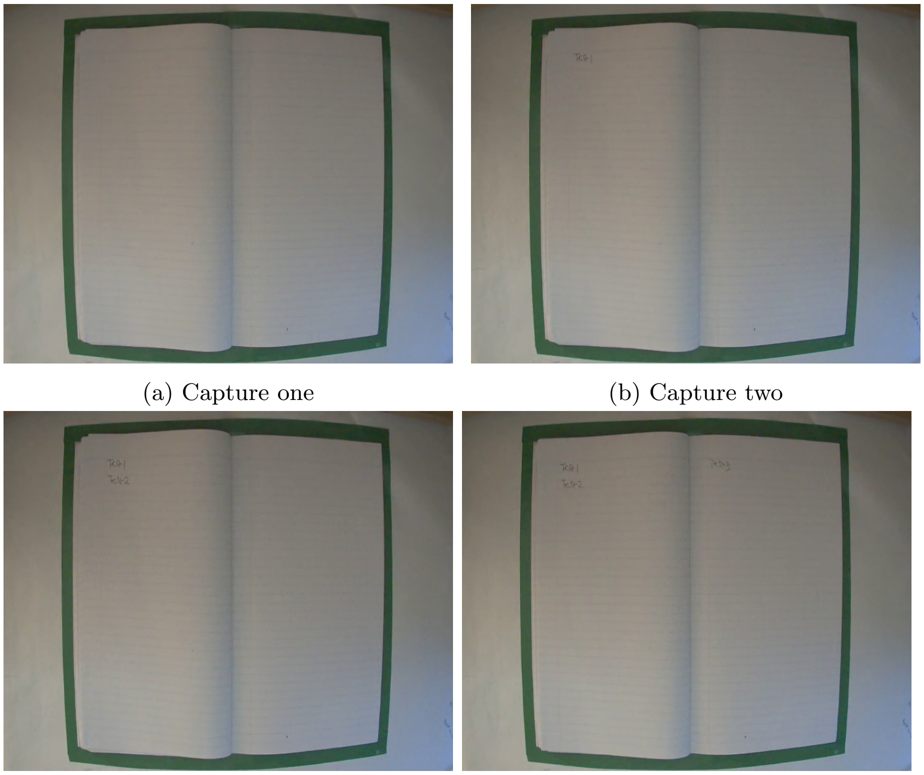
Appendix

A Parts

Item	Price	Where
Jetson Nano	£99	`https://www.okdo.com/p/nvidia-jetson-nano-development-kit/`
Power Supply	£12.99	`https://www.amazon.co.uk/4000mA-Switching-Adaptor-Supply-5-5mmx2-1mm/dp/B07Y8MS5HM`
Raspberry Pi High Quality Camera Lens 6mm Wide Angle	£25	`https://thepihut.com/products/raspberry-pi-high-quality-camera-lens`
Raspberry Pi High Quality Camera Module	£49.50	`https://thepihut.com/products/raspberry-pi-high-quality-camera-module`
SD card 128GB	£27.96	`https://www.amazon.co.uk/Samsung-MB-ME32GA-EU-Select-Adaptor/dp/B085FGMQS6/`
RFID card reader	£11.99	`https://www.amazon.co.uk/KKmoon-Proximity-Sensor-Reader-Interface/dp/B018TXQWRE/`
5inch HDMI LCD (H)	£35.26	`https://www.waveshare.com/5inch-hdmi-lcd-h.htm`
Jumper Wires	£3.50	`https://thepihut.com/products/rpi-premium-jumper-wires-40pk-male-female-100mm`
Wifi Dongle	£11	`https://thepihut.com/products/mini-usb-wifi-module-rtl8188eu-802-11b-g-n`
Jumper Shunt	£1	`https://thepihut.com/products/jumper-shunt-with-handle-0-1-2-54mm-10-pack`
Total	**£277.20**	

Table 5: Detailed Parts List

B Intelligent Data Capture captures



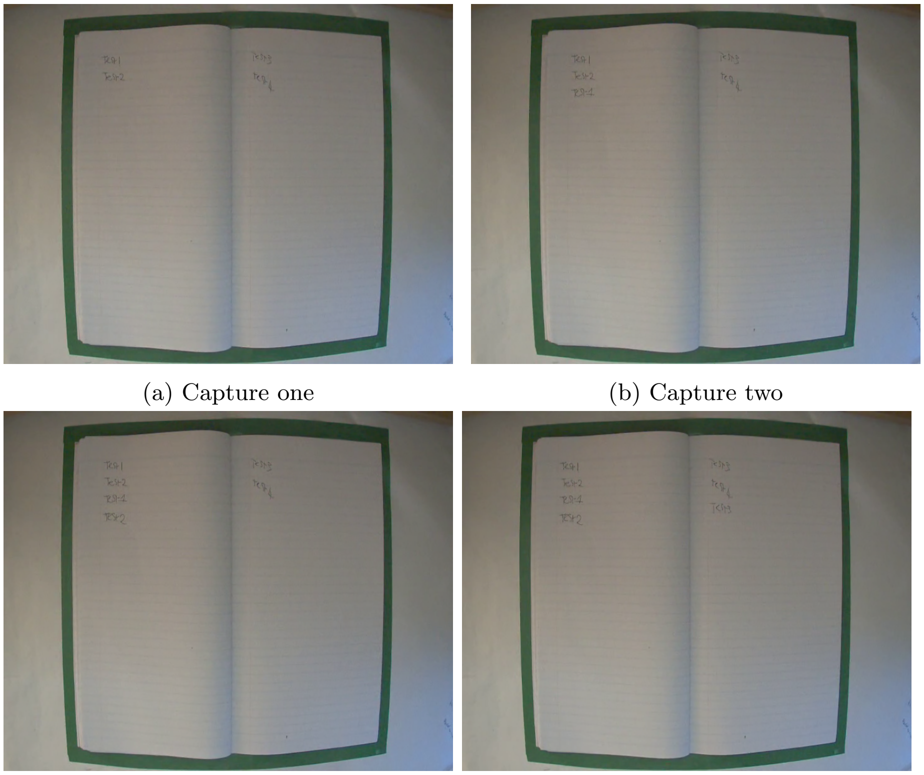
(a) Capture one

(b) Capture two

(c) Capture three

(d) Capture four

Figure 22: Video one captures with border threshold of 5% and frame change threshold of 2500



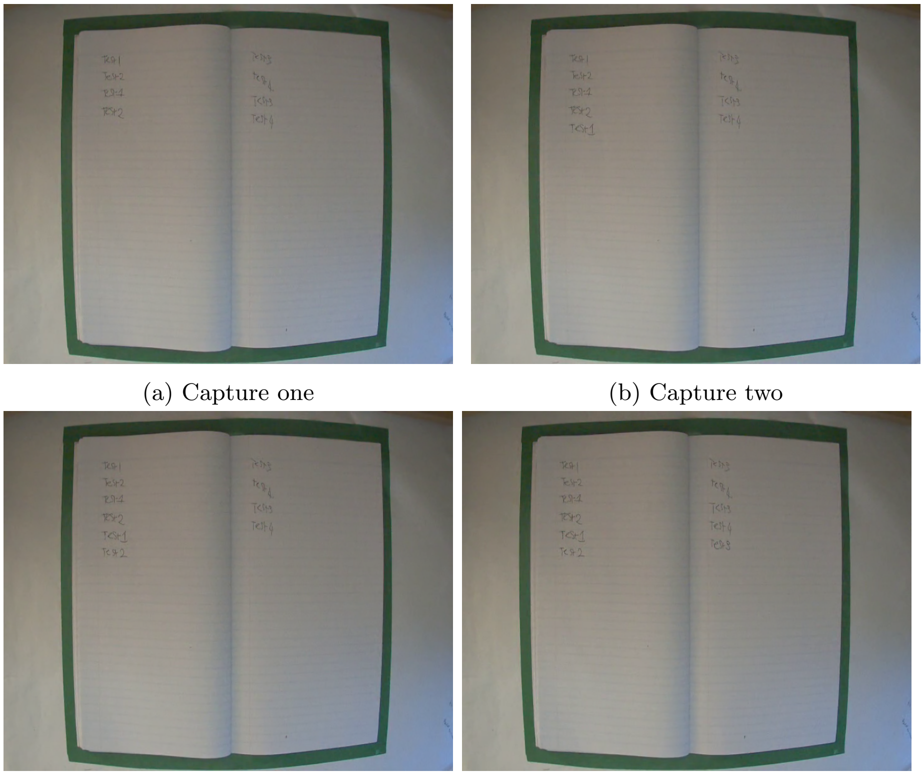
(a) Capture one

(b) Capture two

(c) Capture three

(d) Capture four

Figure 23: Video two captures with border threshold of 5% and frame change threshold of 2500



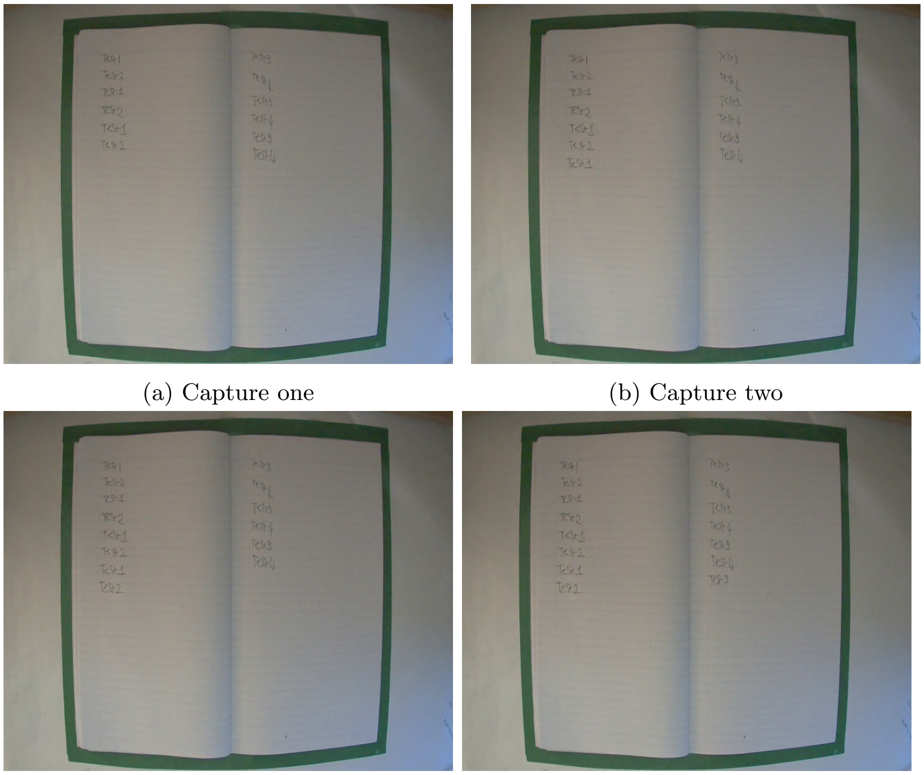
(a) Capture one

(b) Capture two

(c) Capture three

(d) Capture four

Figure 24: Video three captures with border threshold of 5% and frame change threshold of 2500



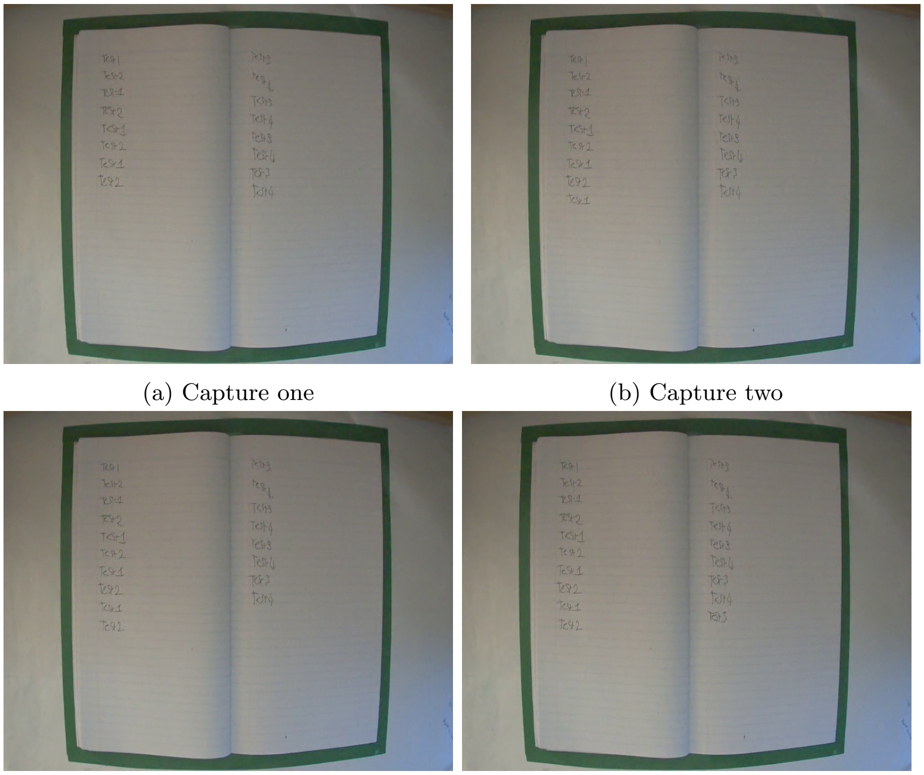
(a) Capture one

(b) Capture two

(c) Capture three

(d) Capture four

Figure 25: Video four captures with border threshold of 5% and frame change threshold of 2500



(a) Capture one

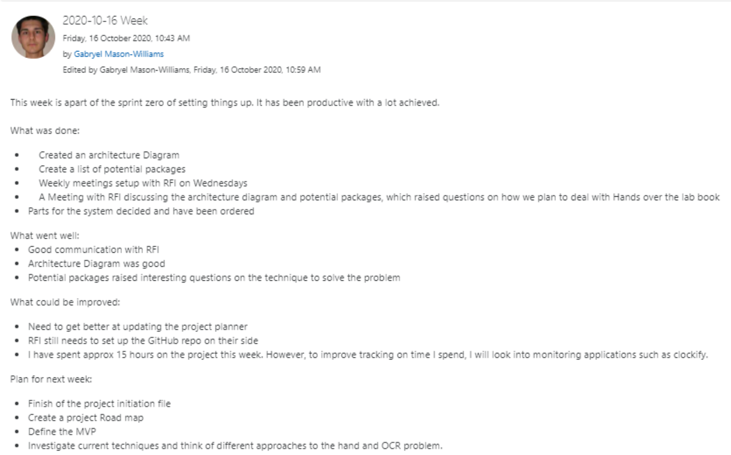
(b) Capture two

(c) Capture three

(d) Capture four

Figure 26: Video five captures with border threshold of 5% and frame change threshold of 2500

C Sprints - Weekly reporting



2020-10-16 Week

Friday, 16 October 2020, 10:43 AM

by Gabryel Mason-Williams

Edited by Gabryel Mason-Williams, Friday, 16 October 2020, 10:59 AM

This week is apart of the sprint zero of setting things up. It has been productive with a lot achieved.

What was done:

- Created an architecture Diagram
- Create a list of potential packages
- Weekly meetings setup with RFI on Wednesdays
- A Meeting with RFI discussing the architecture diagram and potential packages, which raised questions on how we plan to deal with Hands over the lab book
- Parts for the system decided and have been ordered

What went well:

- Good communication with RFI
- Architecture Diagram was good
- Potential packages raised interesting questions on the technique to solve the problem

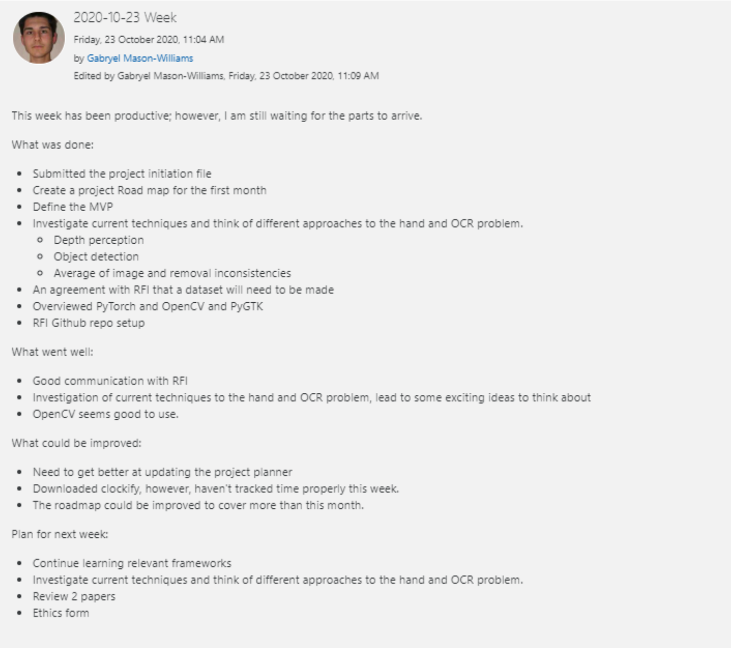
What could be improved:

- Need to get better at updating the project planner
- RFI still needs to set up the GitHub repo on their side
- I have spent approx 15 hours on the project this week. However, to improve tracking on time I spend, I will look into monitoring applications such as clockify.

Plan for next week:

- Finish of the project initiation file
- Create a project Road map
- Define the MVP
- Investigate current techniques and think of different approaches to the hand and OCR problem.

Sprint 0



2020-10-23 Week

Friday, 23 October 2020, 11:04 AM

by Gabryel Mason-Williams

Edited by Gabryel Mason-Williams, Friday, 23 October 2020, 11:09 AM

This week has been productive; however, I am still waiting for the parts to arrive.

What was done:

- Submitted the project initiation file
- Create a project Road map for the first month
- Define the MVP
- Investigate current techniques and think of different approaches to the hand and OCR problem.
  - Depth perception
  - Object detection
  - Average of image and removal inconsistencies
- An agreement with RFI that a dataset will need to be made
- Overviewed PyTorch and OpenCV and PyGTK
- RFI Github repo setup

What went well:

- Good communication with RFI
- Investigation of current techniques to the hand and OCR problem, lead to some exciting ideas to think about
- OpenCV seems good to use.

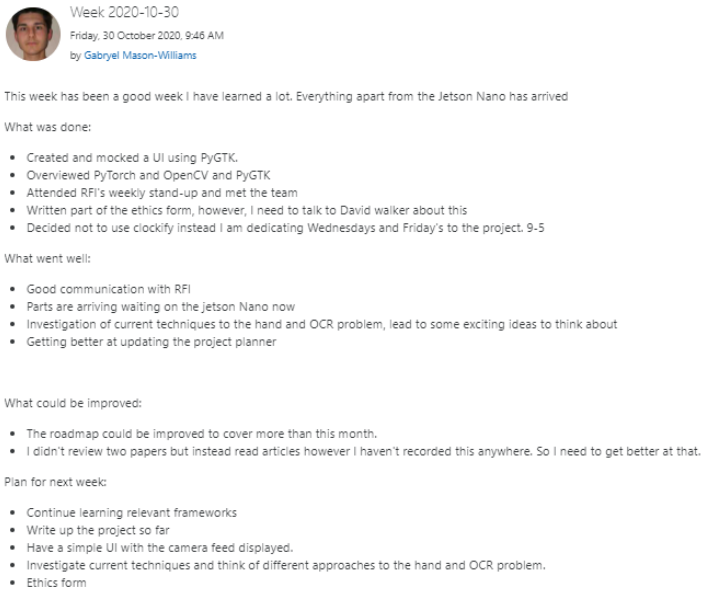
What could be improved:

- Need to get better at updating the project planner
- Downloaded clockify, however, haven't tracked time properly this week.
- The roadmap could be improved to cover more than this month.

Plan for next week:

- Continue learning relevant frameworks
- Investigate current techniques and think of different approaches to the hand and OCR problem.
- Review 2 papers
- Ethics form

Sprint 1



Week 2020-10-30

Friday, 30 October 2020, 9:46 AM

by Gabryel Mason-Williams

This week has been a good week I have learned a lot. Everything apart from the Jetson Nano has arrived

What was done:

- Created and mocked a UI using PyGTK.
- Overviewed PyTorch and OpenCV and PyGTK
- Attended RFI's weekly stand-up and met the team
- Written part of the ethics form, however, I need to talk to David walker about this
- Decided not to use clockify instead I am dedicating Wednesdays and Friday's to the project. 9-5

What went well:

- Good communication with RFI
- Parts are arriving waiting on the jetson Nano now
- Investigation of current techniques to the hand and OCR problem, lead to some exciting ideas to think about
- Getting better at updating the project planner

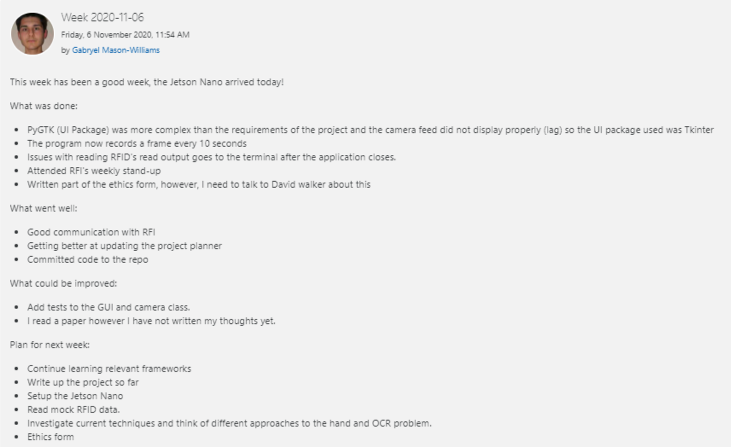
What could be improved:

- The roadmap could be improved to cover more than this month.
- I didn't review two papers but instead read articles however I haven't recorded this anywhere. So I need to get better at that.

Plan for next week:

- Continue learning relevant frameworks
- Write up the project so far
- Have a simple UI with the camera feed displayed.
- Investigate current techniques and think of different approaches to the hand and OCR problem.
- Ethics form

Sprint 2



Week 2020-11-06

Friday, 6 November 2020, 11:54 AM

by Gabryel Mason-Williams

This week has been a good week, the Jetson Nano arrived today!

What was done:

- PyGTK (UI Package) was more complex than the requirements of the project and the camera feed did not display properly (lag) so the UI package used was Tkinter
- The program now records a frame every 10 seconds
- Issues with reading RFID's read output goes to the terminal after the application closes.
- Attended RFI's weekly stand-up
- Written part of the ethics form, however, I need to talk to David walker about this

What went well:

- Good communication with RFI
- Getting better at updating the project planner
- Committed code to the repo

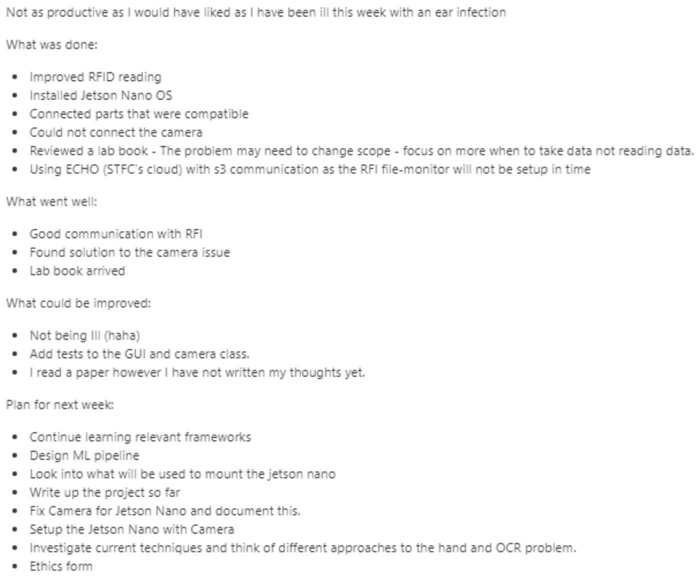
What could be improved:

- Add tests to the GUI and camera class.
- I read a paper however I have not written my thoughts yet.

Plan for next week:

- Continue learning relevant frameworks
- Write up the project so far
- Setup the Jetson Nano
- Read mock RFID data.
- Investigate current techniques and think of different approaches to the hand and OCR problem.
- Ethics form

Sprint 3



Not as productive as I would have liked as I have been ill this week with an ear infection

What was done:

- Improved RFID reading
- Installed Jetson Nano OS
- Connected parts that were compatible
- Could not connect the camera
- Reviewed a lab book - The problem may need to change scope - focus on more when to take data not reading data.
- Using ECHO (STFC's cloud) with s3 communication as the RFI file-monitor will not be setup in time

What went well:

- Good communication with RFI
- Found solution to the camera issue
- Lab book arrived

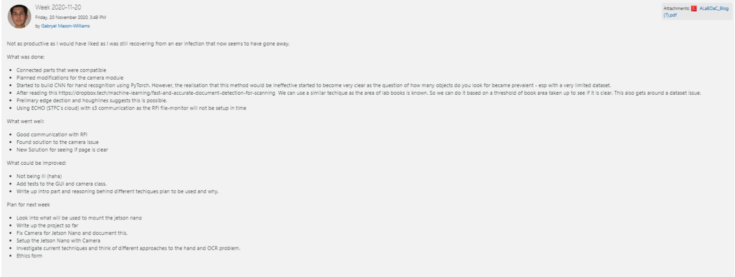
What could be improved:

- Not being Ill (haha)
- Add tests to the GUI and camera class.
- I read a paper however I have not written my thoughts yet.

Plan for next week:

- Continue learning relevant frameworks
- Design ML pipeline
- Look into what will be used to mount the jetson nano
- Write up the project so far
- Fix Camera for Jetson Nano and document this.
- Setup the Jetson Nano with Camera
- Investigate current techniques and think of different approaches to the hand and OCR problem.
- Ethics form

Sprint 4



Week 2020-11-20

Friday, 20 November 2020, 3:49 PM

by Gabryel Mason-Williams

Attachments: ALaBDaC_Blog (7).pdf

Not as productive as I would have liked as I was still recovering from an ear infection that now seems to have gone away.

What was done:

- Connected parts that were compatible
- Planned modifications for the camera module
- Started to build CNN for hand recognition using PyTorch. However, the realisation that this method would be ineffective started to become very clear as the question of how many objects do you look for became prevalent - esp with a very limited dataset.
- After reading this https://dropbox.tech/machine-learning/fast-and-accurate-document-detection-for-scanning We can use a similar techique as the area of lab books is known. So we can do it based on a threshold of book area taken up to see if it is clear. This also gets around a dataset issue.
- Prelimary edge dection and houghlines suggests this is possible.
- Using ECHO (STFC's cloud) with s3 communication as the RFI file-monitor will not be setup in time

What went well:

- Good communication with RFI
- Found solution to the camera issue
- New Solution for seeing if page is clear

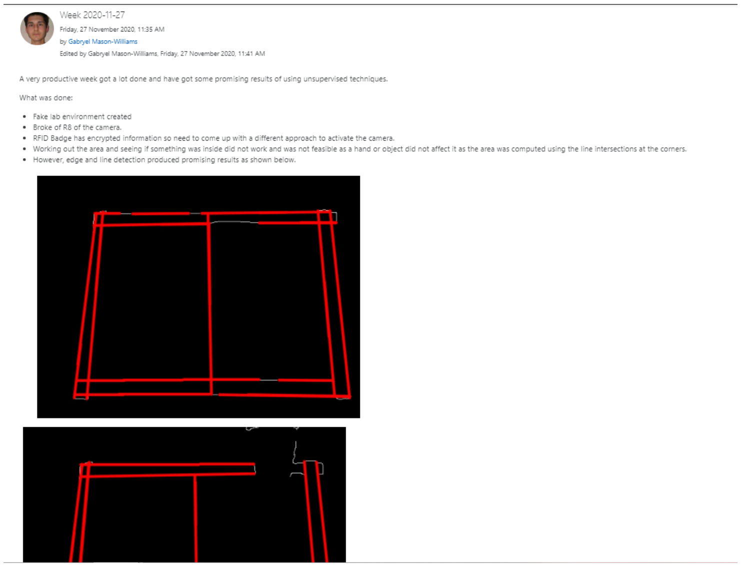
What could be improved:

- Not being Ill (haha)
- Add tests to the GUI and camera class.
- Write up intro part and reasoning behind different techiques plan to be used and why.

Plan for next week

- Look into what will be used to mount the jetson nano
- Write up the project so far
- Fix Camera for Jetson Nano and document this.
- Setup the Jetson Nano with Camera
- Investigate current techniques and think of different approaches to the hand and OCR problem.
- Ethics form

Sprint 5



Week 2020-11-27

Friday, 27 November 2020, 11:35 AM

by Gabryel Mason-Williams

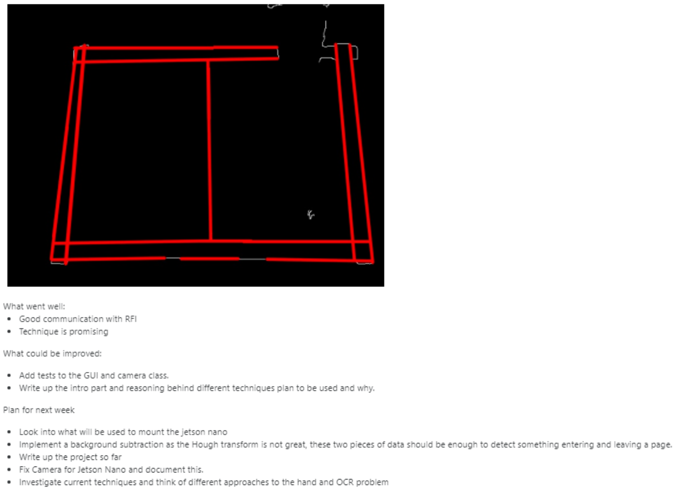
Edited by Gabryel Mason-Williams, Friday, 27 November 2020, 11:41 AM

A very productive week got a lot done and have got some promising results of using unsupervised techniques.

What was done:

- Fake lab environment created
- Broke of R8 of the camera.
- RFID Badge has encrypted information so need to come up with a different approach to activate the camera.
- Working out the area and seeing if something was inside did not work and was not feasible as a hand or object did not affect it as the area was computed using the line intersections at the corners.
- However, edge and line detection produced promising results as shown below.

Sprint 6-1



What went well:

- Good communication with RFI
- Technique is promising

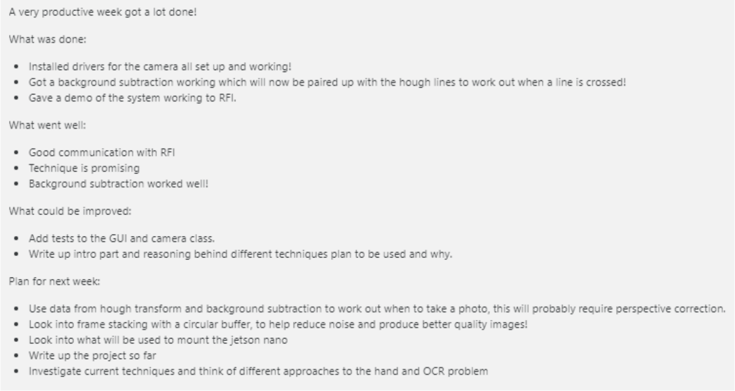
What could be improved:

- Add tests to the GUI and camera class.
- Write up the intro part and reasoning behind different techniques plan to be used and why.

Plan for next week

- Look into what will be used to mount the jetson nano
- Implement a background subtraction as the Hough transform is not great, these two pieces of data should be enough to detect something entering and leaving a page.
- Write up the project so far
- Fix Camera for Jetson Nano and document this.
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 6-2



A very productive week got a lot done!

What was done:

- Installed drivers for the camera all set up and working!
- Got a background subtraction working which will now be paired up with the hough lines to work out when a line is crossed!
- Gave a demo of the system working to RFI.

What went well:

- Good communication with RFI
- Technique is promising
- Background subtraction worked well!

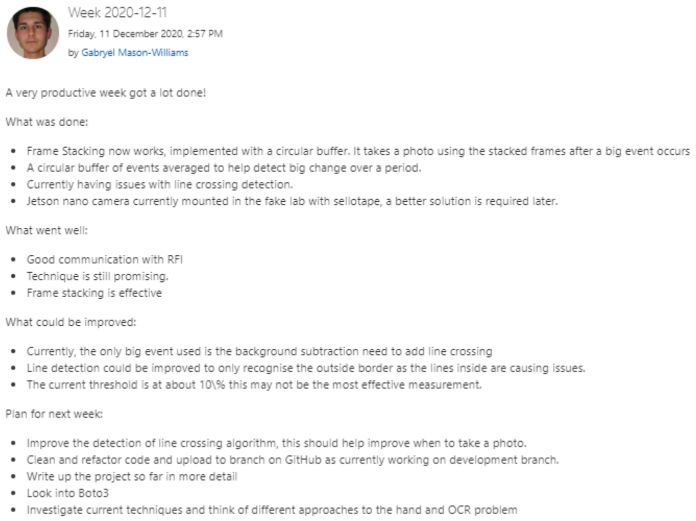
What could be improved:

- Add tests to the GUI and camera class.
- Write up intro part and reasoning behind different techniques plan to be used and why.

Plan for next week:

- Use data from hough transform and background subtraction to work out when to take a photo, this will probably require perspective correction.
- Look into frame stacking with a circular buffer, to help reduce noise and produce better quality images!
- Look into what will be used to mount the jetson nano
- Write up the project so far
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 7



Week 2020-12-11

Friday, 11 December 2020, 2:57 PM

by Gabryel Mason-Williams

A very productive week got a lot done!

What was done:

- Frame Stacking now works, implemented with a circular buffer. It takes a photo using the stacked frames after a big event occurs
- A circular buffer of events averaged to help detect big change over a period.
- Currently having issues with line crossing detection.
- Jetson nano camera currently mounted in the fake lab with sellotape, a better solution is required later.

What went well:

- Good communication with RFI
- Technique is still promising.
- Frame stacking is effective

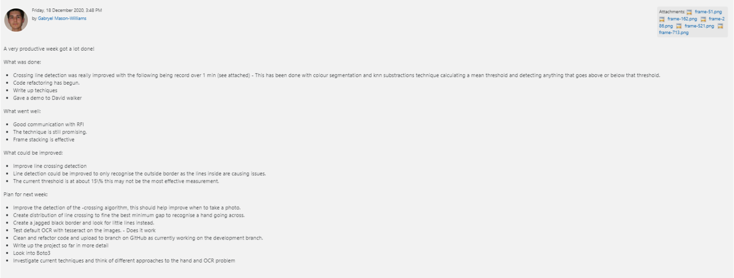
What could be improved:

- Currently, the only big event used is the background subtraction need to add line crossing
- Line detection could be improved to only recognise the outside border as the lines inside are causing issues.
- The current threshold is at about 10\% this may not be the most effective measurement.

Plan for next week:

- Improve the detection of line crossing algorithm, this should help improve when to take a photo.
- Clean and refactor code and upload to branch on GitHub as currently working on development branch.
- Write up the project so far in more detail
- Look into Boto3
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 8



Friday, 18 December 2020, 3:48 PM
by Gabryel Mason-Williams

Attachments: frame-51.png frame-162.png frame-286.png frame-521.png frame-713.png

A very productive week got a lot done!

What was done:

- Crossing line detection was really improved with the following being record over 1 min (see attached) - This has been done with colour segmentation and knn substractions technique calculating a mean threshold and detecting anything that goes above or below that threshold.
- Code refactoring has begun.
- Write up techiques
- Gave a demo to David walker

What went well:

- Good communication with RFI
- The technique is still promising.
- Frame stacking is effective

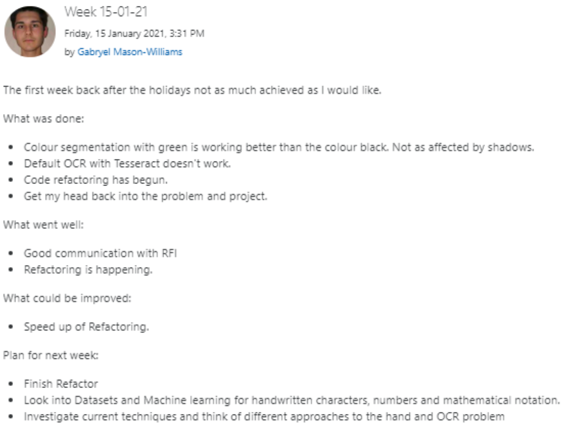
What could be improved:

- Improve line crossing detection
- Line detection could be improved to only recognise the outside border as the lines inside are causing issues.
- The current threshold is at about 15\% this may not be the most effective measurement.

Plan for next week:

- Improve the detection of the -crossing algorithm, this should help improve when to take a photo.
- Create distribution of line crossing to fine the best minimum gap to recognise a hand going across.
- Create a jagged black border and look for little lines instead.
- Test default OCR with tesseract on the images. - Does it work
- Clean and refactor code and upload to branch on GitHub as currently working on the development branch.
- Write up the project so far in more detail
- Look into Boto3
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 9



Week 15-01-21
Friday, 15 January 2021, 3:31 PM
by Gabryel Mason-Williams

The first week back after the holidays not as much achieved as I would like.

What was done:

- Colour segmentation with green is working better than the colour black. Not as affected by shadows.
- Default OCR with Tesseract doesn't work.
- Code refactoring has begun.
- Get my head back into the problem and project.

What went well:

- Good communication with RFI
- Refactoring is happening.

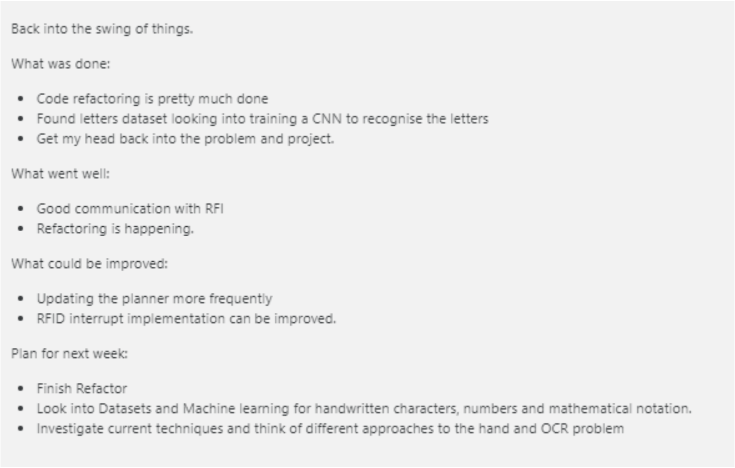
What could be improved:

- Speed up of Refactoring.

Plan for next week:

- Finish Refactor
- Look into Datasets and Machine learning for handwritten characters, numbers and mathematical notation.
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 10



Back into the swing of things.

What was done:

- Code refactoring is pretty much done
- Found letters dataset looking into training a CNN to recognise the letters
- Get my head back into the problem and project.

What went well:

- Good communication with RFI
- Refactoring is happening.

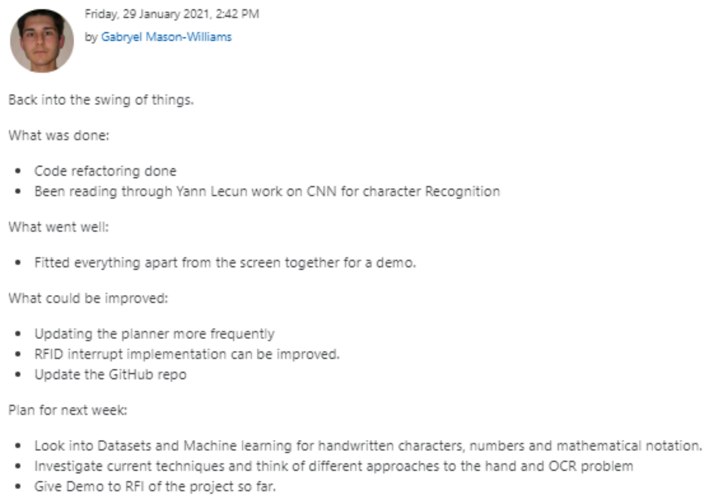
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.

Plan for next week:

- Finish Refactor
- Look into Datasets and Machine learning for handwritten characters, numbers and mathematical notation.
- Investigate current techniques and think of different approaches to the hand and OCR problem

Sprint 11



Friday, 29 January 2021, 2:42 PM

by Gabryel Mason-Williams

Back into the swing of things.

What was done:

- Code refactoring done
- Been reading through Yann Lecun work on CNN for character Recognition

What went well:

- Fitted everything apart from the screen together for a demo.

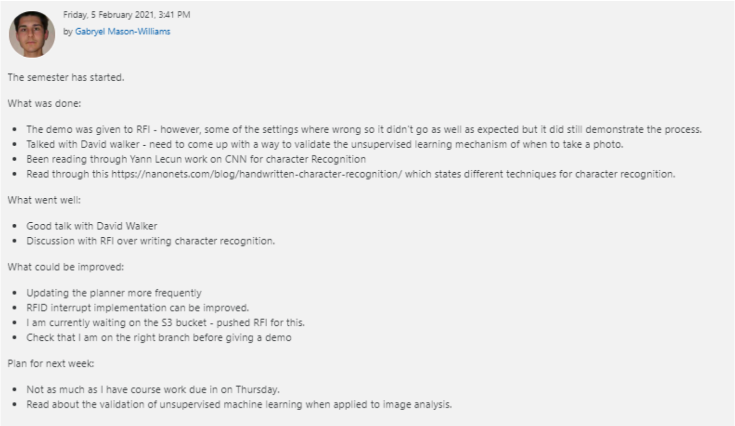
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.
- Update the GitHub repo

Plan for next week:

- Look into Datasets and Machine learning for handwritten characters, numbers and mathematical notation.
- Investigate current techniques and think of different approaches to the hand and OCR problem
- Give Demo to RFI of the project so far.

Sprint 12



Friday, 5 February 2021, 3:41 PM

by Gabryel Mason-Williams

The semester has started.

What was done:

- The demo was given to RFI - however, some of the settings where wrong so it didn't go as well as expected but it did still demonstrate the process.
- Talked with David walker - need to come up with a way to validate the unsupervised learning mechanism of when to take a photo.
- Been reading through Yann Lecun work on CNN for character Recognition
- Read through this https://nanonets.com/blog/handwritten-character-recognition/ which states different techniques for character recognition.

What went well:

- Good talk with David Walker
- Discussion with RFI over writing character recognition.

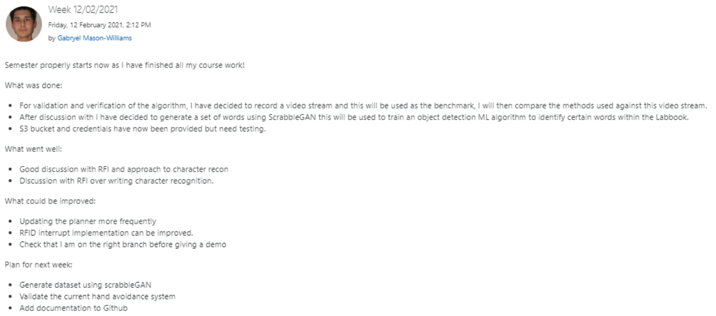
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.
- I am currently waiting on the S3 bucket - pushed RFI for this.
- Check that I am on the right branch before giving a demo

Plan for next week:

- Not as much as I have course work due in on Thursday.
- Read about the validation of unsupervised machine learning when applied to image analysis.

Sprint 13



Week 12/02/2021

Friday, 12 February 2021, 2:12 PM

by Gabryel Mason-Williams

Semester properly starts now as I have finished all my course work!

What was done:

- For validation and verification of the algorithm, I have decided to record a video stream and this will be used as the benchmark, I will then compare the methods used against this video stream.
- After discussion with I have decided to generate a set of words using ScrabbleGAN this will be used to train an object detection ML algorithm to identify certain words within the Labbook.
- S3 bucket and credentials have now been provided but need testing.

What went well:

- Good discussion with RFI and approach to character recon
- Discussion with RFI over writing character recognition.

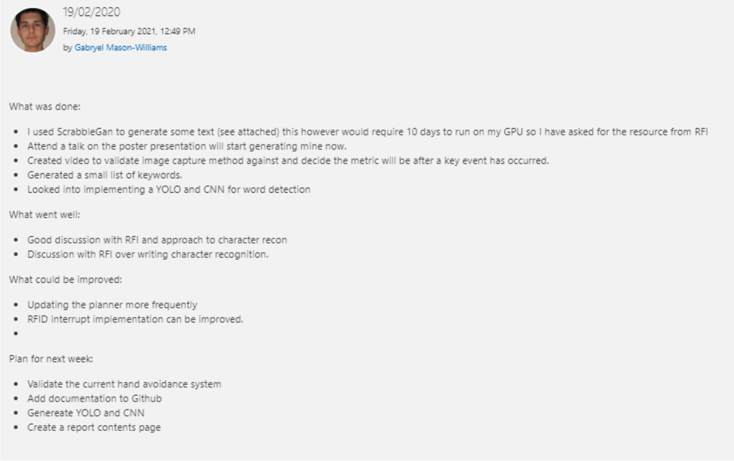
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.
- Check that I am on the right branch before giving a demo

Plan for next week:

- Generate dataset using scrabbleGAN
- Validate the current hand avoidance system
- Add documentation to Github

Sprint 14



19/02/2020

Friday, 19 February 2021, 12:49 PM

by Gabryel Mason-Williams

What was done:

- I used ScrabbleGan to generate some text (see attached) this however would require 10 days to run on my GPU so I have asked for the resource from RFI
- Attend a talk on the poster presentation will start generating mine now.
- Created video to validate image capture method against and decide the metric will be after a key event has occurred.
- Generated a small list of keywords.
- Looked into implementing a YOLO and CNN for word detection

What went well:

- Good discussion with RFI and approach to character recon
- Discussion with RFI over writing character recognition.

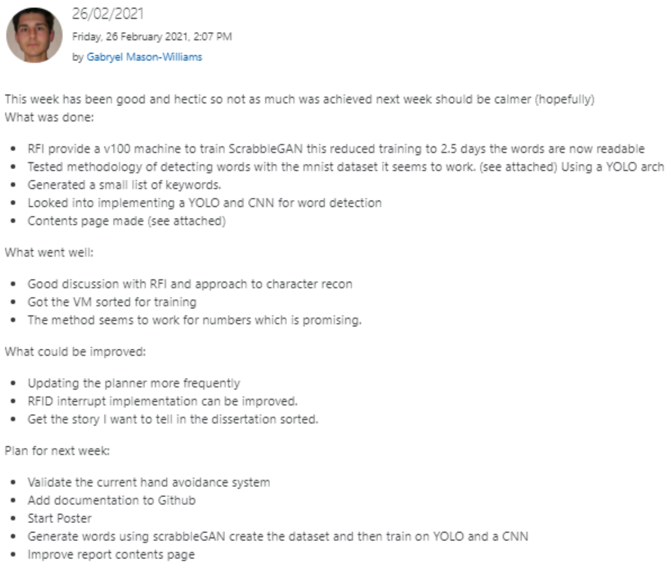
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.
-

Plan for next week:

- Validate the current hand avoidance system
- Add documentation to Github
- Genereate YOLO and CNN
- Create a report contents page

Sprint 15



26/02/2021

Friday, 26 February 2021, 2:07 PM

by Gabryel Mason-Williams

This week has been good and hectic so not as much was achieved next week should be calmer (hopefully)
What was done:

- RFI provide a v100 machine to train ScrabbleGAN this reduced training to 2.5 days the words are now readable
- Tested methodology of detecting words with the mnist dataset it seems to work. (see attached) Using a YOLO arch
- Generated a small list of keywords.
- Looked into implementing a YOLO and CNN for word detection
- Contents page made (see attached)

What went well:

- Good discussion with RFI and approach to character recon
- Got the VM sorted for training
- The method seems to work for numbers which is promising.

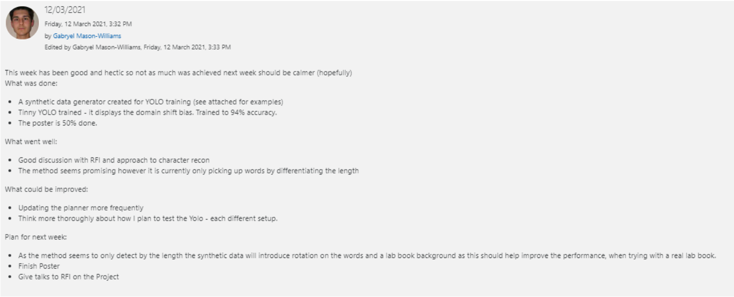
What could be improved:

- Updating the planner more frequently
- RFID interrupt implementation can be improved.
- Get the story I want to tell in the dissertation sorted.

Plan for next week:

- Validate the current hand avoidance system
- Add documentation to Github
- Start Poster
- Generate words using scrabbleGAN create the dataset and then train on YOLO and a CNN
- Improve report contents page

Sprint 16



12/03/2021

Friday, 12 March 2021, 3:32 PM

by Gabryel Mason-Williams

Edited by Gabryel Mason-Williams, Friday, 12 March 2021, 3:33 PM

This week has been good and hectic so not as much was achieved next week should be calmer (hopefully)
What was done:

- A synthetic data generator created for YOLO training (see attached for examples)
- Tinny YOLO trained - it displays the domain shift bias. Trained to 94% accuracy.
- The poster is 50% done.

What went well:

- Good discussion with RFI and approach to character recon
- The method seems promising however it is currently only picking up words by differentiating the length

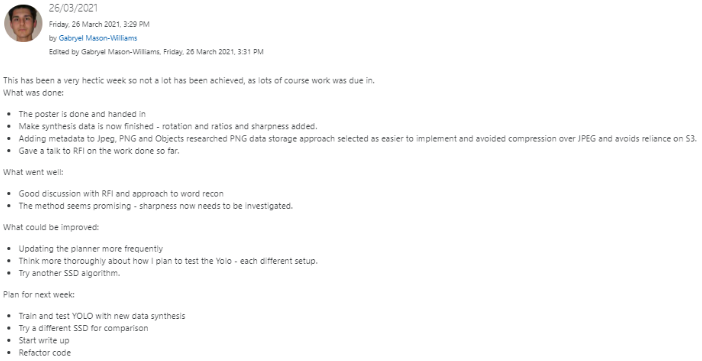
What could be improved:

- Updating the planner more frequently
- Think more thoroughly about how I plan to test the Yolo - each different setup.

Plan for next week:

- As the method seems to only detect by the length the synthetic data will introduce rotation on the words and a lab book background as this should help improve the performance, when trying with a real lab book.
- Finish Poster
- Give talks to RFI on the Project

Sprint 17



26/03/2021

Friday, 26 March 2021, 3:29 PM

by Gabryel Mason-Williams

Edited by Gabryel Mason-Williams, Friday, 26 March 2021, 3:31 PM

This has been a very hectic week so not a lot has been achieved, as lots of course work was due in.
What was done:

- The poster is done and handed in
- Make synthesis data is now finished - rotation and ratios and sharpness added.
- Adding metadata to Jpeg, PNG and Objects researched PNG data storage approach selected as easier to implement and avoided compression over JPEG and avoids reliance on S3.
- Gave a talk to RFI on the work done so far.

What went well:

- Good discussion with RFI and approach to word recon
- The method seems promising - sharpness now needs to be investigated.

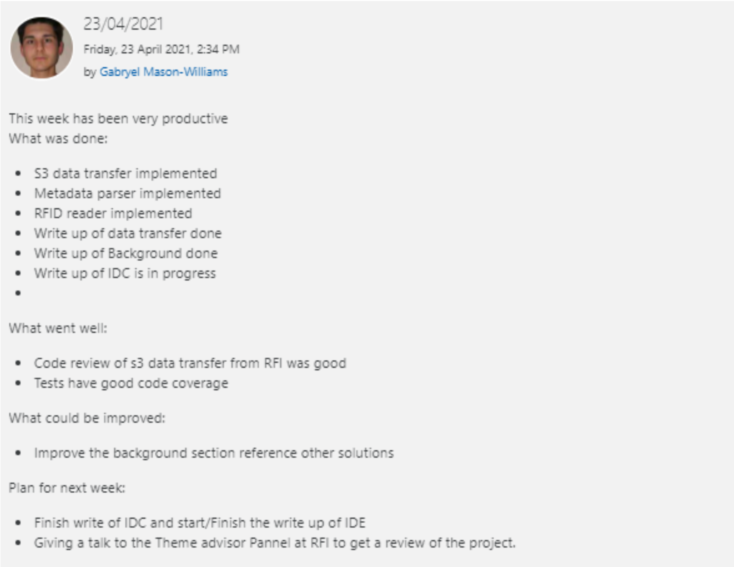
What could be improved:

- Updating the planner more frequently
- Think more thoroughly about how I plan to test the Yolo - each different setup.
- Try another SSD algorithm.

Plan for next week:

- Train and test YOLO with new data synthesis
- Try a different SSD for comparison
- Start write up
- Refactor code

Sprint 18



23/04/2021

Friday, 23 April 2021, 2:34 PM

by Gabryel Mason-Williams

This week has been very productive
What was done:

- S3 data transfer implemented
- Metadata parser implemented
- RFID reader implemented
- Write up of data transfer done
- Write up of Background done
- Write up of IDC is in progress
-

What went well:

- Code review of s3 data transfer from RFI was good
- Tests have good code coverage

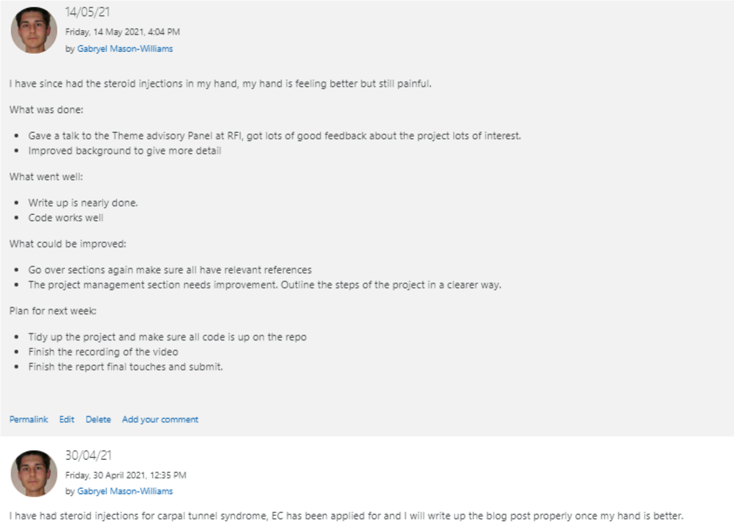
What could be improved:

- Improve the background section reference other solutions

Plan for next week:

- Finish write of IDC and start/Finish the write up of IDE
- Giving a talk to the Theme advisor Pannel at RFI to get a review of the project.

Sprint 19



14/05/21

Friday, 14 May 2021, 4:04 PM

by Gabryel Mason-Williams

I have since had the steroid injections in my hand, my hand is feeling better but still painful.

What was done:

- Gave a talk to the Theme advisory Panel at RFI, got lots of good feedback about the project lots of interest.
- Improved background to give more detail

What went well:

- Write up is nearly done.
- Code works well

What could be improved:

- Go over sections again make sure all have relevant references
- The project management section needs improvement. Outline the steps of the project in a clearer way.

Plan for next week:

- Tidy up the project and make sure all code is up on the repo
- Finish the recording of the video
- Finish the report final touches and submit.

30/04/21

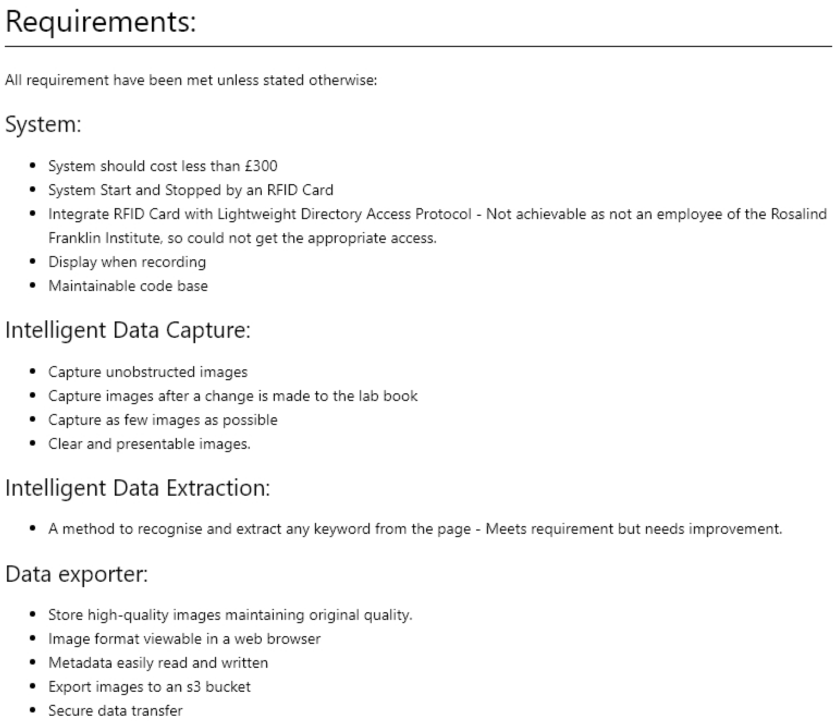
Friday, 30 April 2021, 12:35 PM

by Gabryel Mason-Williams

I have had steroid injections for carpal tunnel syndrome, EC has been applied for and I will write up the blog post properly once my hand is better.

Sprint 20

D Requirements



Requirements:

All requirement have been met unless stated otherwise:

System:

- System should cost less than £300
- System Start and Stopped by an RFID Card
- Integrate RFID Card with Lightweight Directory Access Protocol - Not achievable as not an employee of the Rosalind Franklin Institute, so could not get the appropriate access.
- Display when recording
- Maintainable code base

Intelligent Data Capture:

- Capture unobstructed images
- Capture images after a change is made to the lab book
- Capture as few images as possible
- Clear and presentable images.

Intelligent Data Extraction:

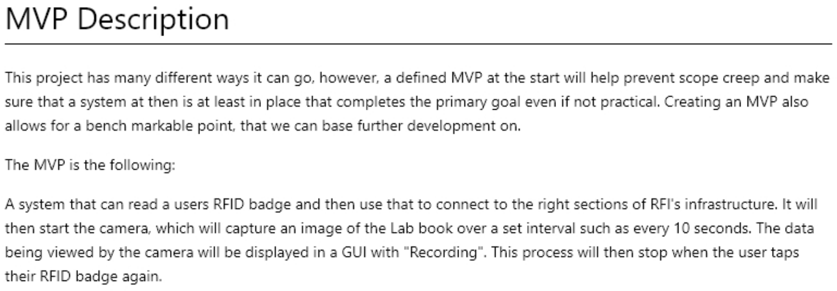
- A method to recognise and extract any keyword from the page - Meets requirement but needs improvement.

Data exporter:

- Store high-quality images maintaining original quality.
- Image format viewable in a web browser
- Metadata easily read and written
- Export images to an s3 bucket
- Secure data transfer

Figure 27: Project Requirements list

E Minimal viable product description



MVP Description

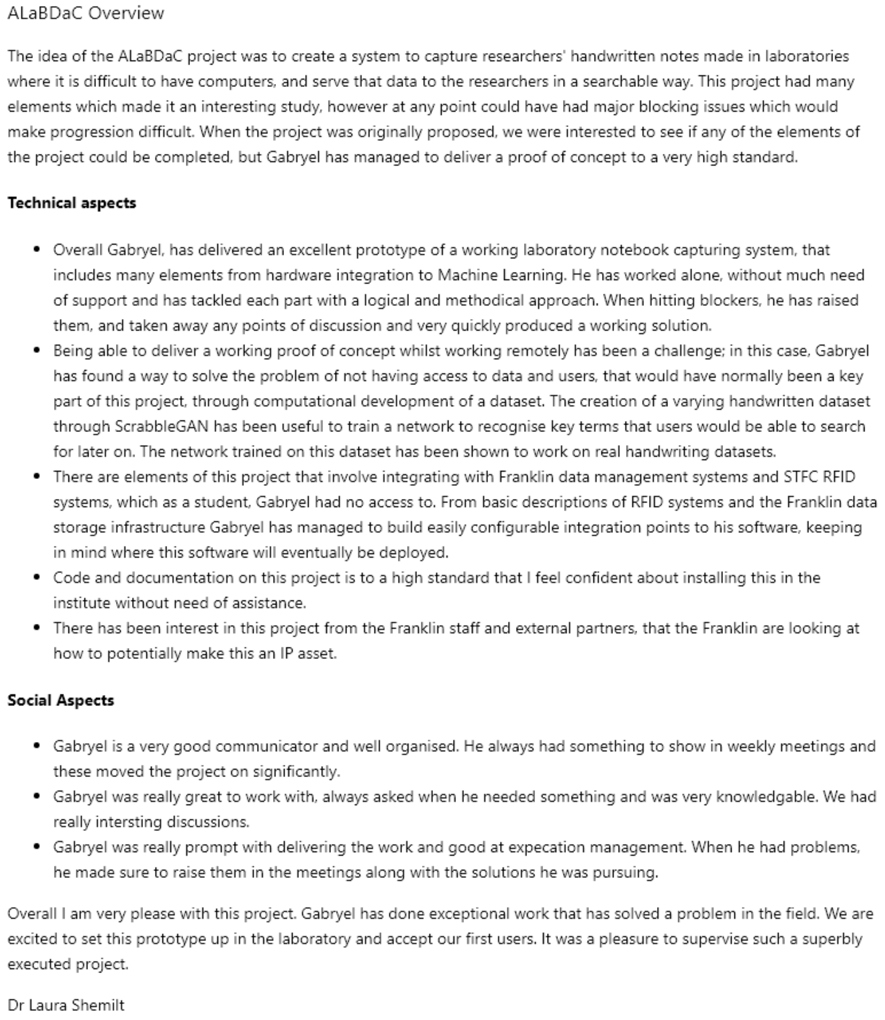
This project has many different ways it can go, however, a defined MVP at the start will help prevent scope creep and make sure that a system at then is at least in place that completes the primary goal even if not practical. Creating an MVP also allows for a bench markable point, that we can base further development on.

The MVP is the following:

A system that can read a users RFID badge and then use that to connect to the right sections of RFI's infrastructure. It will then start the camera, which will capture an image of the Lab book over a set interval such as every 10 seconds. The data being viewed by the camera will be displayed in a GUI with "Recording". This process will then stop when the user taps their RFID badge again.

Figure 28: Minimal Viable Product (MVP) outline, to allow for research and investigation of techniques to improve the system to begin.

F Review from the Rosalind Franklin Institute



ALaBDaC Overview

The idea of the ALaBDaC project was to create a system to capture researchers' handwritten notes made in laboratories where it is difficult to have computers, and serve that data to the researchers in a searchable way. This project had many elements which made it an interesting study, however at any point could have had major blocking issues which would make progression difficult. When the project was originally proposed, we were interested to see if any of the elements of the project could be completed, but Gabryel has managed to deliver a proof of concept to a very high standard.

Technical aspects

- Overall Gabryel, has delivered an excellent prototype of a working laboratory notebook capturing system, that includes many elements from hardware integration to Machine Learning. He has worked alone, without much need of support and has tackled each part with a logical and methodical approach. When hitting blockers, he has raised them, and taken away any points of discussion and very quickly produced a working solution.
- Being able to deliver a working proof of concept whilst working remotely has been a challenge; in this case, Gabryel has found a way to solve the problem of not having access to data and users, that would have normally been a key part of this project, through computational development of a dataset. The creation of a varying handwritten dataset through ScrabbleGAN has been useful to train a network to recognise key terms that users would be able to search for later on. The network trained on this dataset has been shown to work on real handwriting datasets.
- There are elements of this project that involve integrating with Franklin data management systems and STFC RFID systems, which as a student, Gabryel had no access to. From basic descriptions of RFID systems and the Franklin data storage infrastructure Gabryel has managed to build easily configurable integration points to his software, keeping in mind where this software will eventually be deployed.
- Code and documentation on this project is to a high standard that I feel confident about installing this in the institute without need of assistance.
- There has been interest in this project from the Franklin staff and external partners, that the Franklin are looking at how to potentially make this an IP asset.

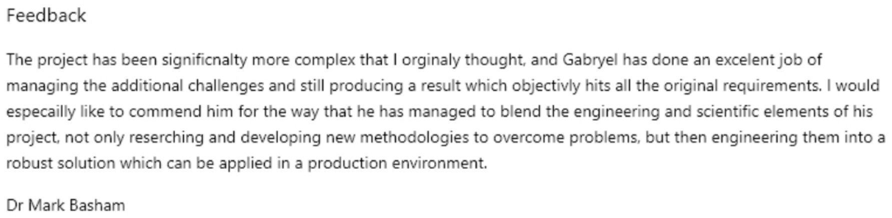
Social Aspects

- Gabryel is a very good communicator and well organised. He always had something to show in weekly meetings and these moved the project on significantly.
- Gabryel was really great to work with, always asked when he needed something and was very knowledgable. We had really intersting discussions.
- Gabryel was really prompt with delivering the work and good at expecation management. When he had problems, he made sure to raise them in the meetings along with the solutions he was pursuing.

Overall I am very please with this project. Gabryel has done exceptional work that has solved a problem in the field. We are excited to set this prototype up in the laboratory and accept our first users. It was a pleasure to supervise such a superbly executed project.

Dr Laura Shemilt

Figure 29: Feedback from Dr Laura Shemilt - Research Software Engineer at RFI



Feedback

The project has been significnalty more complex that I orginaly thought, and Gabryel has done an excelent job of managing the additional challenges and still producing a result which objectivly hits all the original requirements. I would especailly like to commend him for the way that he has managed to blend the engineering and scientific elements of his project, not only reserching and developing new methodologies to overcome problems, but then engineering them into a robust solution which can be applied in a production environment.

Dr Mark Basham

Figure 30: Feedback from Dr Mark Basham - Science Director Artificial Intelligence at RFI