Summary of Comments on Practical Work Report

Student Name:.. Markers Initials ____________

	Ranking			
Structure of the Report (compliance issue) All instruction about format have been followed, and all the prescribed sections are present	5	(4)	3	2 OR 1
Quality of the Report Grammar and spelling are good. The language is fluent, and flows well, no proof reading errors.	5	(4)	3 OR 2	1
Presentation of the Report The report is well focussed, balanced, interesting and contains an appropriate amount of detail. Illustrations are useful.	5	(4) OR 3	2	1
Scope of Report Well-chosen and relevant to a Practical Work Report	(3)	2	1	0
TOTAL VALUE OF MARKS	(15)			

OVERALL Grade for this Report

Total Marks greater than 14	(A)
Total Marks greater than 10	B
Total marks greater than 6	C
You are required to resubmit this report.	D

It is recommended that you contact the Student Learning Unit for advice on the writing of this report

ENGGEN499 – PRACTICAL WORK REPORT

, Department of Mechanical Engineering

Workplace:

Location: , Auckland

Period Worked: 27th November 2017 to 16th February 2018

Report Date: 5th March 2018

Summary

Upon the conclusion of my four years studying towards an Engineering degree, I was employed as an Engineering Intern at , part of the G division of the . The works on developing simulation training systems for military bodies across the world. The New Zealand office employs engineers in the fields of Software, Mechanical, and Electrical engineering to work on laser products for systems.

For my internship, I was employed to carry out a project looking to extract information from a vehicle display interface that could help improve some of their vehicle products. This was a software based project that involved work in the areas of image processing and machine learning. The program that I produced at the end of my internship was able to remove the background from a display image, extract and identify text characters, and decode the required information.

Over the course of my time at I gained exposure to the Agile Software Development methodology, a formal software development process, as well as Test Driven Development and Version Control. The internship concluded with the writing of a formal report and a presentation to members of senior management, which provided an opportunity to develop my presenting skills.

My time at has been a very valuable learning experience and I have been able to develop my skills in problem solving and my ability to plan work and motivate myself. Even though my project was an individual project, I have also gained an appreciation for working together in teams to complete work and helping each other. In addition to this, I have had the privilege of learning from some of senior executives about valuable leadership qualities.

Table of Contents

Table of Figures

Table of Tables

1.0) INTRODUCTION

is one of the three divisions of the ı, the other two being s and s. main headquarters are in , with various other offices around the world, including that in New Zealand. The three divisions are responsible for different areas, as described below:

- The (rks on developing training systems for military and special operations forces as well as mission support services.
- s works on developing communications, command, control, surveillance, and reconnaissance systems for defence, security, intelligence, and commercial mission applications.
- Transportation Systems develops systems for managing transportation, including implementing intelligent payment systems for public transportation, such as London's Oyster card.

The New Zealand office is part of the division that develops training systems for defence forces worldwide. The training systems that develop can be compared to a high tech laser tag game where soldiers use real weapons that have attachments to fire laser beams rather than actual bullets. New Zealand office employs engineers to work on the design and development of the laser training systems, including software, mechanical, and electronical design elements.

(regularly takes on engineering student interns to complete internal Research and Development projects over the course of the summer. For my internship period, I was employed to work on a software development project to extract information from the display interface of a vehicle used in military training. This project mainly involved work in the areas of Image Processing and Machine Learning.

During my time working on my project, I gained exposure to formal software development practices used in industry. I was also able to further develop my problem solving and programming skills and, having completed all papers for my degree, begin my transition from working at university to working in industry.

N.B. Due to privacy issues, I have not been able to take pictures inside the : office of facilities etc.

2.0) Workplace Location and Layout

As shown in Figure 1, s Auckland office is located in Parnell at close to the a and the . The building is shared with New Zealand with occupying two of the four floors in the building, with a carpark on the other floor. This location is close to bus stops on Symonds Street and in Parnell, as well as the . The carpark facilities were not made available due to the engineering student interns for my time working there but the location meant that public transport was a viable option for travel to and from work.

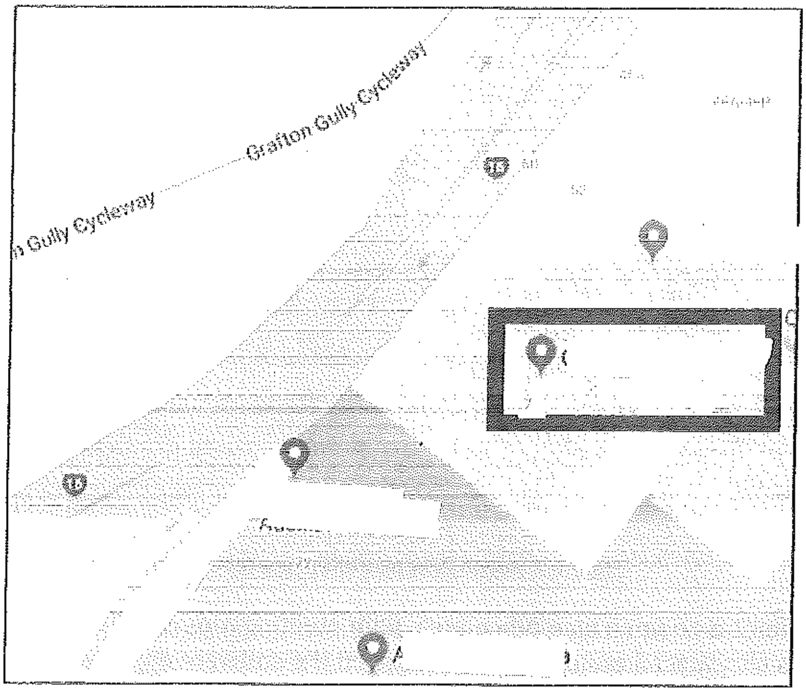

Figure 1: Location in Auckland at

The main office is located on the upper floor that occupies. This contains the following departments:

- HR and Admin
- Executive Staff
- Engineering departments – software, mechanical, systems, electronics/hardware
- Finance and Business Operations

The bottom floor contains some test engineers and the Operations staff, split into the following areas:

- Stores – storage of parts and products
- Purchasing – purchase externally made parts required for products
- Production/Assembly – produce and assemble products that design
- Testing – testing of lasers and assembled products
- Workshop

Almost all of my time at was spent on the upper floor as we did not need to be using any equipment or facilities in the bottom floor. Figure 2 on the next page shows the floor plan of the upper floor. The main lunchroom/cafeteria is also on the upper floor although there is a smaller lunchroom area on the lower floor.

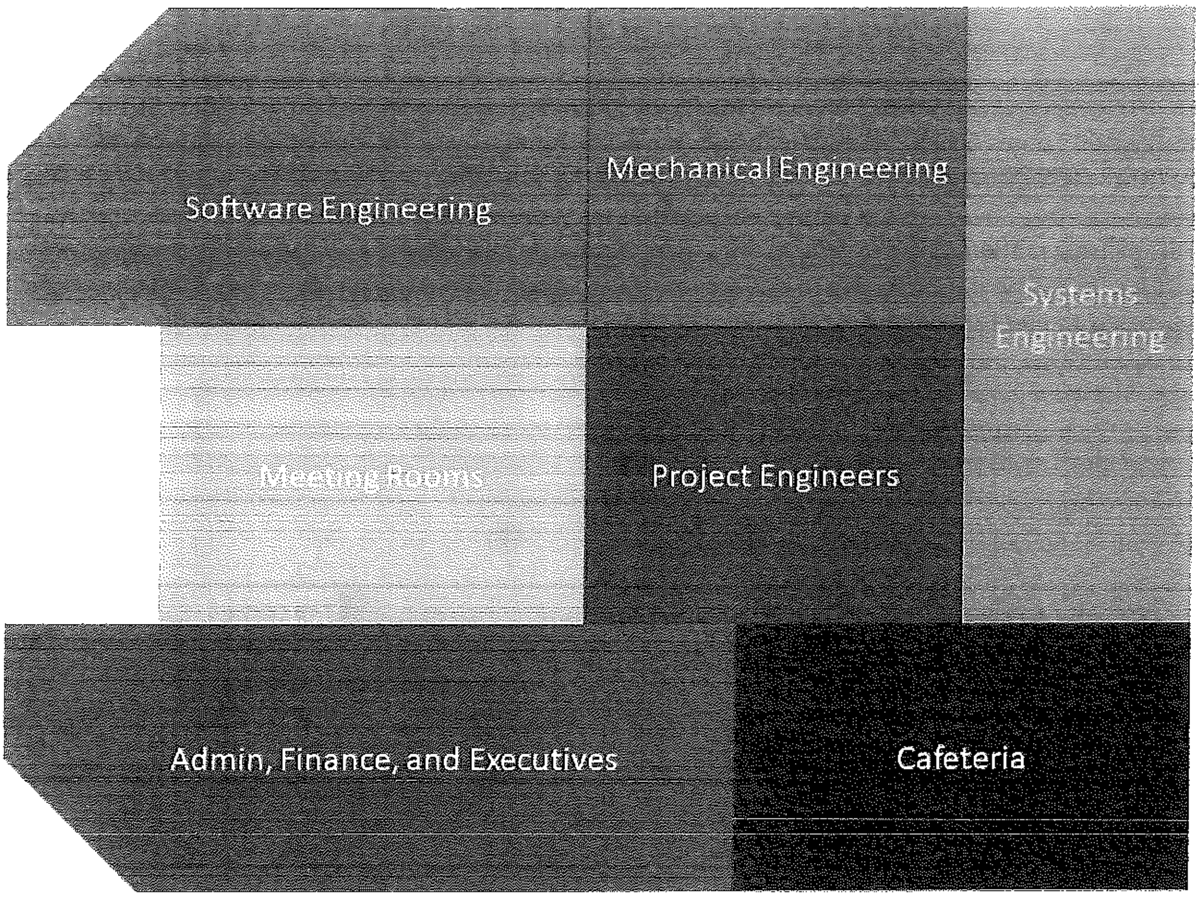

Figure 2: Rough floor plan of the upper floor

The whole upper floor is an open plan office area, with meeting rooms and smaller rooms for test labs or 3D printers also spaced around. I was working in the software engineering department, close to the meeting rooms, surrounded by both the other engineering interns and the permanent software engineers. As my project did not require any other areas of engineering knowledge, I only spent time in the software engineering area and did not need to interact much with other departments for my work.

As the only time I spent on the lower floor was during a brief tour at the start of my internship period, I was not familiar with the layout of staff on the lower floor of the building.

3.0) Workplace Facilities

The main facilities available for staff on the upper floor were:

Desk spaces

In the software department area where I was working, all staff members had a laptop with two extra monitors along with a wireless keyboard and mouse. Many employees had extra equipment at their desks for testing and debugging such as oscilloscopes, multi-meters, test microprocessors, or prototypes of some products.

Meeting rooms

There were 10 meeting rooms, ranging in size from 4 seats to 16 seats. I spent the first day of inductions in one of the larger meeting rooms. Weekly show time meetings were also held in one of these for all the engineering interns. I also had smaller meetings with my project sponsor and project mentor in one of the smaller meeting rooms.

Each meeting room had chairs, desks, and a television screen to which a laptop could be connected in order to present or display material to people at the meeting.

Quiet rooms

There were 8 quiet rooms that did not need to be booked but were free for use if empty. These ranged in size from 1 seat to 4 seats. I did not need to use any of these rooms.

Meeting spaces

There were 5 of these spread across the upper floor in the different departments. These consisted of a long high table for standing and a television screen that could connect to a laptop for presenting or displaying material. Daily stand up meetings for the engineering interns were held at one of these meeting spaces every morning.

In the office, there was also one large movable touchscreen television screen that could be moved across the office to different meeting spaces.

Lunchroom

Staff would mainly spend lunch breaks at the lunchroom, which had several tables and chairs to sit at while having lunch. There was a kitchen area with dishwashers, sinks, hot and cold water taps, and a fridge for staff to store their lunch in. There were also board games and packs of cards in the lunchroom for staff to make use of on lunch breaks.

Staff social get-togethers, such as the Start of Month drinks and the End of Year Barbecue, were held in the lunchroom as well as Monthly Birthday morning teas and "All Hands" meetings. There was an onsite barbecue in an outdoor area that also had chairs and tables, at which staff could also spend lunch breaks. The final presentation for the engineering interns was also held in the lunchroom as it had a projector for displaying presentation slides.

Other facilities

During my time at I also noted other facilities, such as the workshop and laser testing room on the lower floor, as well as 3D printers on the top floor for the mechanical department to use when prototyping. For my project work, I did not need to use any extra equipment so did not become too familiar with other areas.

4.0) STAFF ORGANISATION

As an engineering intern at I was a member of the Software Engineering team, working under the Software Engineering Manager, with the organisational structure shown in Figure 3 below.

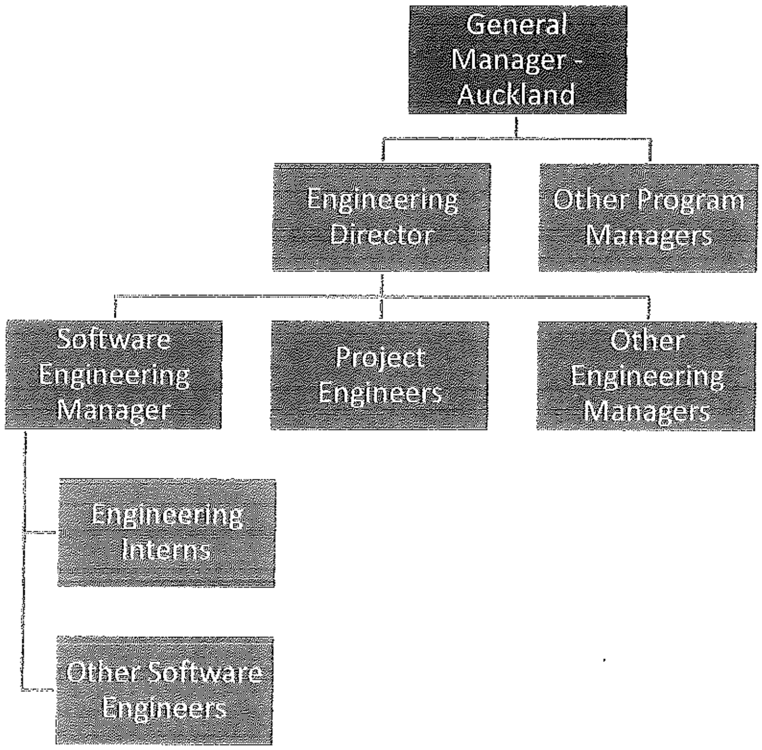

Figure 3: Hierarchical structure of my position as an engineering intern

There were four other engineering interns working with me. Also working under the software engineering manager were around 20 other software engineers, split across embedded software development and software applications.

Under the Engineering Director, there are managers for the other engineering departments, Technical Writing, Optics, Systems, and Hardware (Mechanical and Electronics), with around 25 other employees working across these departments. There are also around 10 other project engineers working under the Engineering Director.

The Engineering Director reports to the General Manager along with around 5 other program managers, such as those for Administration and for Health and Safety. Around 5 further employees work under these program managers for different departments. There is also a team in Denmark working under the General Manager. The General Manager himself reports to the executive team in the USA.

Also in the Auckland office is the Operations team, with about 15 staff, and around 15 more staff members spread across IT, Finance, HR, and Business Operations. This gives a total of around 100 employees at office in Auckland.

N.B. These numbers have been rounded to the nearest multiple of 5.

5.0) WORK COMPLETED

5.1) Vehicle Interface Image Recognition Project

For the time I was at C ., I was working on an internal R&D project. The aim of this project was to create a program to analyse a video stream that is coming from a vehicle's internal video display interface and to extract information that could be used to help improve Cubic's vehicle products.

This was an entirely software related project that relied heavily on image processing techniques and machine learning. Real life footage was not available so I was using footage from a vehicle in a video game that my mentor acquired for me before I started.

Both image processing and machine learning models were available from the OpenCV library for C++. I was initially running my program with a Linux Virtual Machine but also moved to running it from a microcontroller, as would be the case if this program were implemented on a real life vehicle used in defence training exercises.

I split the project into four main stages for processing an image of a vehicle display:

1. Background removal
2. Character region extraction
3. Character identification
4. Information extraction

5.1.1) Background removal

The first step was to remove the background to produce an image that only contained the information available from the video stream. I developed four different methods to perform this.

For the first method, using the fact that text was in a green colour, I created a program to increase the intensity of brighter green areas on the image so that they would stand out. From there, the image can be thresholded so that pixels with a green value above a certain value become black while other pixels become white. The threshold needed to vary for each image so I created a formula using the maximum green intensity for pixels in the image.

The next method I tried was to convert the image to black and white and extract areas where there was bright white on the image. Each extracted area was then thresholded based on the maximum intensity of pixels in the image using a formula. An issue with this method is that bright areas on the image that were in a colour that wasn't green would also be picked up. Therefore I changed the earlier green method to process smaller areas of green as I found that some dimmer areas of text could be missed due to the presence of brighter areas on the same image.

The final method that I implemented was to use a different colour space at the suggestion of my mentor – the HLS colour space rather than using RGB values. Of the three values, I focused on using the L (lightness) and H (hue) values. The L value gives an indication of the brightness of the pixel while the H value represents the colour of the pixel on a red-green-blue spectrum. Brighter areas were extracted and thresholded based on the brightness and hue values, targeting text in green. This method proved to be the most effective at removing the background and was the method implemented for my final solution.

Figure 4 below shows the results of applying each of the four different background methods to a segment of an image. Note that the HLS method produced the most desirable results.

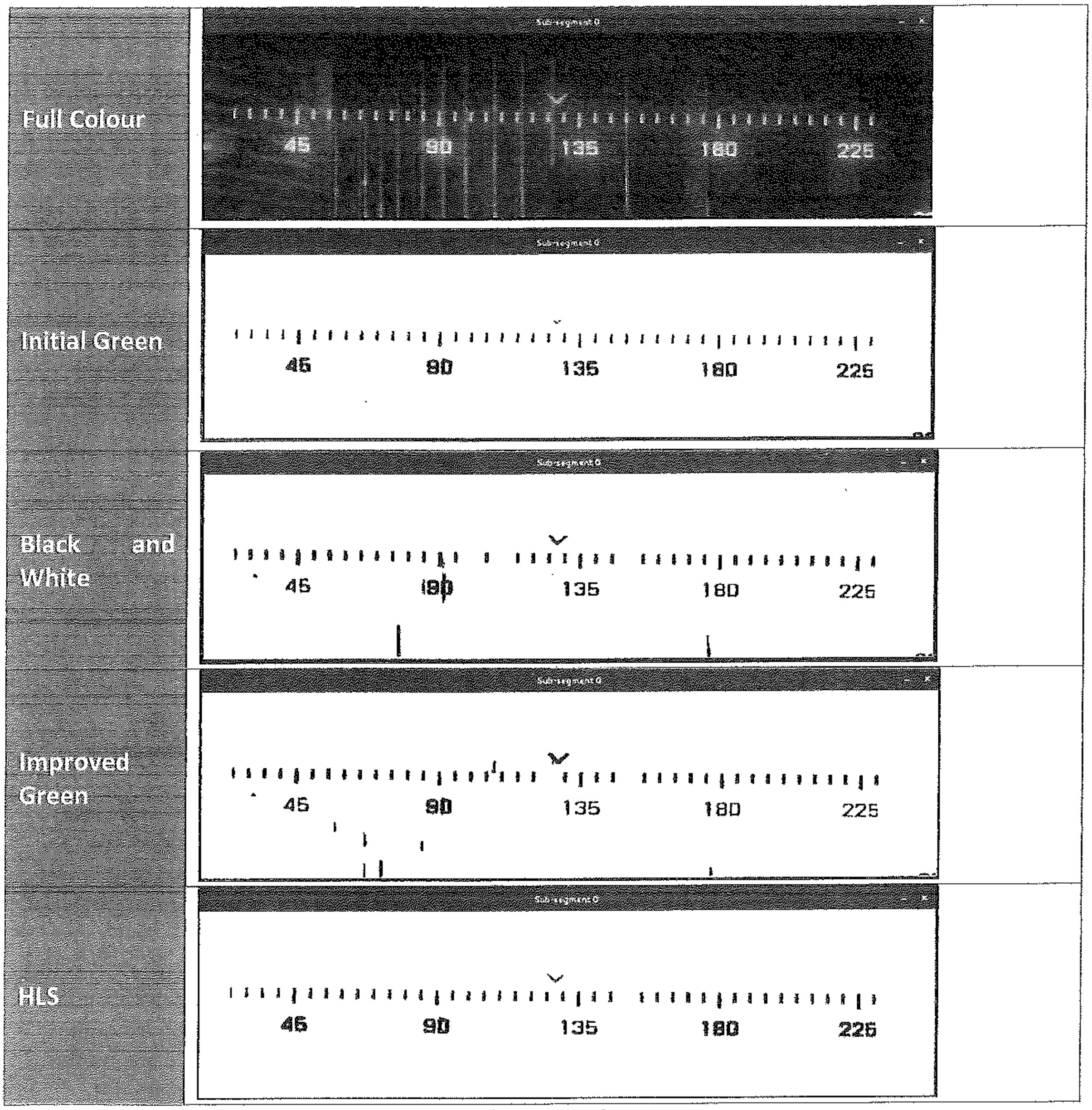

Figure 4: The different background removal methods applied to part of an image

5.1.2) Character region extraction

Once the background had been removed, the next step was to identify the areas that contained text. Since I was not able to produce a perfect background removal program that could fully filter out unwanted parts of the image, I needed to make the extraction method able to reject noise parts that weren't properly removed.

The extraction process went as follows:

1. The black and white image was eroded to increase the size of black areas
2. Contours were generated around each black area
3. Rectangle objects were generated for each contour
4. The area of the original black and white image before erosion was extracted.
5. Characters were extracted from each identified area

Several different methods were considered for extracting each individual character, described below in Table 1:

Table 1: Different character segmentation methods

Name	Description
Pixel Connectivity	Pixels that are connected are labelled. Those with similar size and aspect ratios are deemed to be characters.
Projection Profiles	The sum of black pixels is found for each row/column to give an idea on where new spaces between characters and new lines are.
Hardcoding size parameters	If the height, width, and separation of characters are already known, these can be used to split characters and lines.
Character contours	Using the shape of contours in comparison to actual letters.

The method selected was using projection profiles as it was the least complex to implement and did not require hard coding, which I wanted to avoid. The code I wrote was able to split up characters from a black and white image and divide them into words and lines.

5.1.3) Character identification

After all the character areas had been extracted, I needed to be able to determine which character was contained in that region of the image. I was able to draw ideas from research into Licence Plate Recognition, a common application for Optical Character Recognition.

From my research, I identified two methods to recognise characters – template matching or machine learning models.

Template matching involves matching a set of parameters from the unknown character to the same parameters for each of a series of templates. Each different character possible would have its own template. I tried a series of different parameters to compare templates, described below in Table 2.

Table 2: Methods to match template for character recognition

No.	Feature	Brief Description
I	Number of matching pixels	The number of pixels between two images that are the same
II	Horizontal/vertical projections	Finding the sum of black pixels for each row and column of an image
III	Number of black pixels per block	Finding the number of black pixels per each n x m block of the full image.
IV	Hausdorff distance	Comparing how different two contours are from each other in space – the longest distance between a point on one contour to the closest point on the other contour
V	Hamming distance	The number of pixels between two images that are different
VI	Normalised cross correlation	A measure of similarity based on the difference between each pixel in both images and the average values of each image

The best method was using the box fill method as it produced the most accurate results. However, I also explored using machine learning methods for classification, namely Neural Networks and a k Nearest Neighbours algorithm.

A neural network consists of a series of layers consisting of nodes. Each node calculates a weighted sum of its inputs to determine an output. For the k Nearest Neighbours algorithm, an unknown point is classified based on the most dominant class in the k nearest points from a training set to the unknown.

I tested different methods for generating feature vectors for the machine learning models, shown below in Table 3: Different feature vectors used for character recognition

Table 3: Different feature vectors used for character recognition

No.	Feature	Brief Description
I	Number of black pixels per block	Formed from the number of black pixels per each n x m block of the full image
II	Horizontal and vertical projections	Finding the sum of black pixels for each row and column of an image
III	All pixels	Formed by placing the values of each pixel in the image together
IV	Hu Moments	Seven parameters that represent different qualities of a shape – invariant of size, position, or orientation

Once again, the best performing method was using the box fill method. However, there were still misclassifications such as identifying the letter 'O' as the number '0' and the number '6' as the letter 'G'. I was able to work around this by specifying what characters were expected from the block of text. Such information would be known about each area of the vehicle display so can be used to increase the program's accuracy.

Both models required large training sets to increase their accuracy. This required me to go through the available footage to obtain example images of possible characters. I was able to generate a set of 20 examples for each character, which produced good results when using the k Nearest Neighbours algorithm with k = 7.

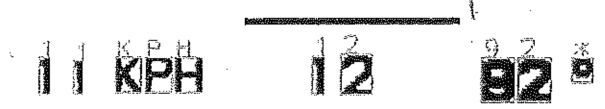

Figure 5: Examples of character recognition from my program

5.1.4) Information extraction

From there, I needed to be able to extract useful information from the characters that had been identified. For most of the cases on the game footage, information was purely text based. However, for the bearing reading at the top of the screen, I needed to be able to estimate the bearing of the vehicle.

I determined the bearing by calculating the distance (in pixels) between the closest numbers on either side of the bearing symbol and the distance between one of the numbers and the bearing symbol and using the ratio of the two. This was multiplied by the difference between the two numbers and added to the smaller of the two numbers. I also included methods to overcome noise interfering with the number identification so that the calculation was more reliable.

For the other areas, I needed to be able to make the information extraction resistant to noise. I was able to reject false characters based on the dimensions of the character area as noisy black areas on the image would sometimes be detected as text areas.

5.1.5) Program Implementation

Once I had completed the steps for extracting the information from an image, I needed to be able to run this on an embedded microcomputer. My mentor supplied me with a Raspberry Pi so that I could test the timing of the information extraction process. This came out at about ten times the time that was measured when running off a laptop.

I was then supplied with an ASUS Tinker Board, which has a faster processing speed than a Raspberry Pi. This proved to be faster than the Raspberry Pi but still took a long time. I then sought to analyse the timing of each step. The longest time was absorbed by the background removal step due to the large size of the image and the many steps needed to process the image.

The video game footage was not the best for this since the image was of a high resolution and the display was not steady, resulting in the area for text identification needing to be large in order to be sure of capturing the whole information segment. Images from a real vehicle would be of a lower resolution and information would be stable on the position of the screen, so with smaller areas being processed, the program would be able to run faster.

Another feature I implemented was the ability to store firing information for the vehicle when a button was pressed. At the "FIRE" button press, the program aims to extract range and weapon information and prints it to the screen, seen in Figure 6. Information of this type can be sent to the laser product attached to the vehicle to cause the laser to fire.

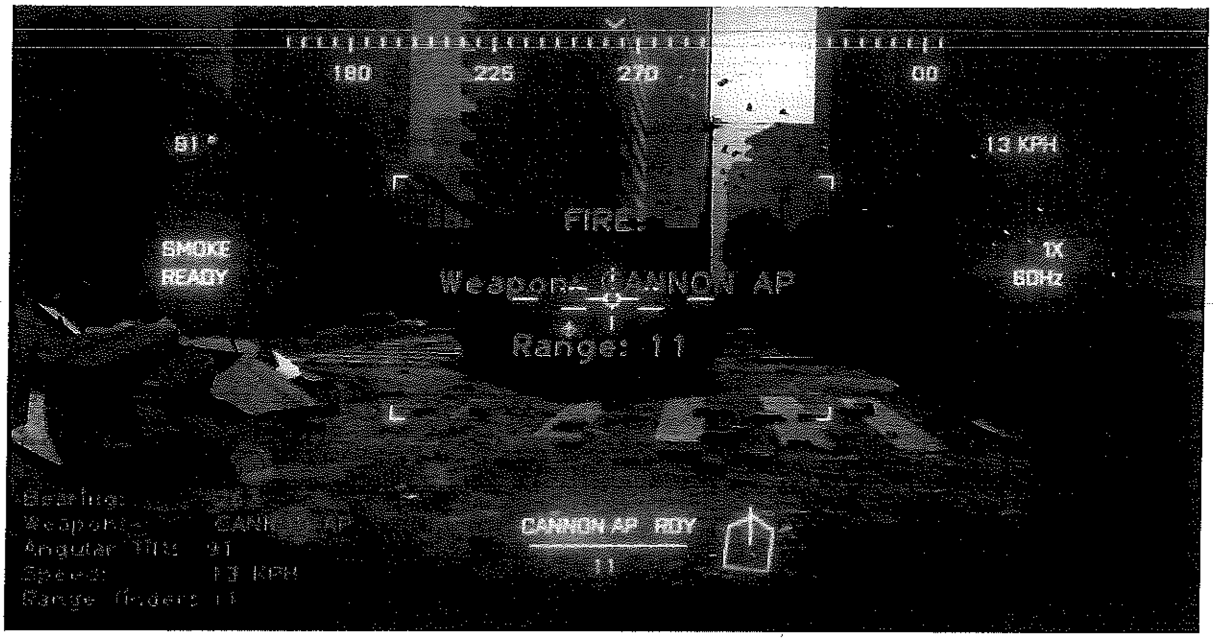

Figure 6: Screenshot showing extracted information at the bottom left with a fire command displayed:

5.2) Software Coding Practices

5.2.1) Test Driven Development

A coding practice we were encouraged to use was creating tests for our code, referred to as unit tests. Ideally these are written before writing code to perform tasks as they can provide an end goal for the code that is to be written. Each piece of new code needed a set of unit tests to validate its performance and ensure it was carrying out the desired functionality. Whenever a new bit of code was written, unit tests for the other parts were used to make sure that the program still had the desired functionality.

5.2.2) Version Control

To manage different versions of our code (Version Control), we used a system known as Git. Each project had its own repository on Git to allow us to store versions of our code as we worked on it. All work that we uploaded to Git was first placed in a branch off the master code. Each branch was used to represent a different functionality that was being implemented

Once we had finished working on the code, we sent out a "pull request" where we allow people to look at our code and review it. Sub-branches can be merged into the master branch when the pull request has been improved. Merging can also only happen if all unit tests pass and the code builds successfully.

Using pull requests allowed for the other interns and our mentors to review our code before we could join it into our main program. They could write comments or suggestions on how we could improve the code we had written. This allowed for the whole intern team to help each other with each other's projects and we could gain exposure to the way other people code and approach solving problems.

5.3) Agile Methodology

As a Mechatronics engineering student, I had not been exposed to any formal software engineering practices. The methodology that the software engineers use at Cubic is known as Agile Software Development.

Agile Software Development breaks work tasks for a project into short blocks called sprints. At the start of each sprint, we assigned to ourselves smaller tasks, called issues, which we aimed to complete in the sprint. At the end of the sprint, we had a show time session where we would present to the other interns, the intern mentors, and the project sponsors what work we had completed during the sprint. The project sponsors would then ask us questions about our work and provide feedback on our progress so far and what we could look at working on for the next sprint. Figure 7 below summarises the meetings included for each sprint.

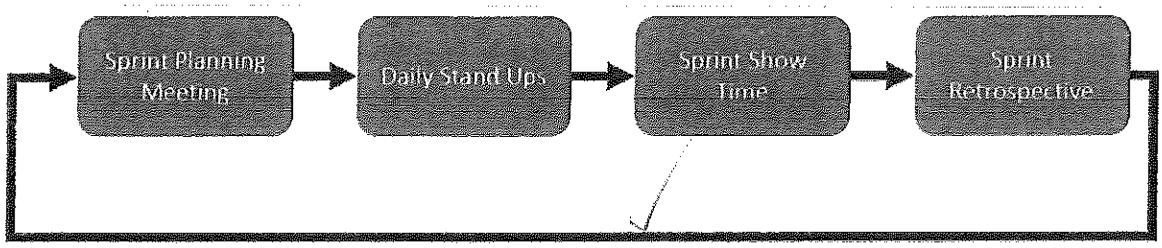

Figure 7: Meetings included in each sprint

Sprint Planning

Initial planning occurred at the start of the project when I worked with my mentor to break down the tasks into smaller issues that could be spread across the ten weeks. Each issue needed a clear indication of success, such as code that performed a certain function, or for research tasks, a summary document on findings. Over the course of the project, issues were added after weekly planning meetings with my project sponsor and where I thought it necessary to work on certain areas.

Daily Stand Ups

In addition to show time sessions, we had daily stand up meetings. These were short informal meetings where we would briefly tell the other interns and the intern mentors what we had been

working on and if we were struggling to complete certain tasks. At each stand up, we would review our progress with the sprint board to make sure that we were on track with our work.
At each daily stand up, we would aim to talk about:

- What we had done
- What we were currently doing and planning to do next
- If there were anything preventing progress with our current work

Progress was kept track using the sprint board. The sprint board splits issues into four categories: To Do, In Progress, In Review, and Completed

- Tasks under "To Do" have been assigned to someone for them to complete during the sprint.
- When starting work on an issue, it is moved to the "In Progress" category.
- Once we have completed the issue, it is moved to the "In Review" category for someone else to check. To make our work ready for checking, we would create a pull request for our work to merge the branch we had been working on to the master branch for our repository.
- When this had been approved, the issue could be set to be "Completed" and another issue could be worked on.

Each sprint had a different board with different issues assigned to that sprint. Although we aimed to complete all issues during that sprint, since it was sometimes difficult to estimate the length of time tasks will take, issues could be taken to the next sprint if not complete. This was also the case when we had short weeks, such as after the Christmas break.

Retrospective

At the end of each sprint, we also had a small retrospective meeting which was added to the end of the show time session. These were short discussions where we would talk about what we thought was going well and what we thought we could improve on for the next sprint.

5.4) Presentation of Results

At the conclusion of my project, I was required to write a report and to give a presentation to the executive members of the company and the project engineers.

This gave me valuable practice at improving my writing and reporting skills as well as applying what I had learnt from my Part Four Project presentation and putting that into practice to presenting in a more industrial environment rather than to a university audience.

6.0) Reflections on Work

Overall, I thoroughly enjoyed my time working at and felt it was a very valuable experience in terms of understanding how engineering work is formally done in industry. This was my first real experience working in an engineering related role for a company and I learnt much, both about myself and about adapting to working outside university.

6.1) Impressions of the company

During my time at I found that everyone was approachable and willing to help with any issues that I may have had. One such example is my mentor, , who at one point was helping out with two other intern projects as the other intern mentors were called to work overseas for a period. Whenever I needed help, I could ask my mentor and he was always very welcoming in helping me. I feel that this sort of an attitude in a company's employees helps everyone to work better together.

I enjoyed the formal structure of carrying out work with the Agile Development methodology. Something I felt what worked well over my internship period was the feedback given to us at Daily Stand Ups and weekly Show Times. Our project sponsors gave us valuable advice and suggested things we could try that we may not have thought of previously. This brought to me a sense of working in a team and not just individual interns working on individual projects. Having our code reviewed before it could be merged into our programs also gave us a more team approach to our projects rather than working on them individually.

Something that I was not expecting was visits from senior leadership members from headquarters in the US. We had two "All Hands" meetings where all employees met in the main lunchroom to have a meeting with those senior executive members. After a talk about different points that they had and their thoughts and plan for the company's future, they asked for any questions from any employee. Everyone was treated equally and with respect, whether they were a manager, a member of the engineering team, or a member of the finance team. I feel that encouraging an open environment such as this can be very beneficial to a company's success.

From listening to their speeches, I was able to learn a few techniques to being a good leader. I noted that the senior executives, when addressing the meeting, would use words like "us" and "the team", something which made me feel, even though only an intern for the summer, like a part of a united team working together. I feel that encouraging a team environment like this is a useful leadership skill and something which I shall remember if ever I enter into a leadership role in the future.

The head of mentioned words along the lines of "every good idea from a manager has been thought of before –and it often comes from the employees". Allowing for feedback from employees at lower levels of the business hierarchy, who may know how aspects of their own work can be improved or what works well based on their own experiences, can be a good way for those higher up to improve the working of the company. I'm not sure if this is a common thing in companies that have many offices spread around the world, but, in my opinion, this is something that can help a large company to grow and improve over time.

6.2) Comments on my own work

For my own personal development, I enjoyed the transition from university work to working in industry. After four years of university work, working on an engineering project in an industrial environment was something different for me. A key difference I found is how I could go about getting help when I met an obstacle and unsure on how to progress. At university, this would require sending an email to a lecturer with an uncertain time on how soon I would be able to get assistance or wait until the next day. When working, asking for help is simply walking across the office to find someone to ask.

From my internship project, I felt I was able to develop my skills in problem solving. Several times when I was unsure on how to approach a problem, I found a useful method for me was to write down what information I had, the problem I wanted to solve, and then brainstorm methods to solve the problem. Particularly when writing software code, stepping away from the code and writing my thoughts down was very helpful for me.

Both from feedback given from the other interns and the project mentors, I was able to develop my programming knowledge. The other interns had a bit more experience in coding large projects than I had had and their feedback on how I could improve my code, both for readability and functionality, was very useful for me. I can take such experiences to help me in the future if ever I work on projects that require a large amount of programming.

Building upon my experience from my summer research project, the ability to motivate myself to work was also important over my time at . I feel that I was not driven by achieving deadlines but by a genuine desire to explore new possibilities and improve and build on the work I had already completed. I feel that enjoying the work I was doing was very important for helping my self-motivation. For me, enjoying what I am doing makes more eager to carry on with my work.

Something else I was able to develop was my ability to plan my work. Since we had planning meetings and deadlines every week, I needed to plan my work in one week blocks rather than as a large ten week project. Needing to plan my work in shorter blocks helped me to stay on track as I could focus on smaller goals rather than the larger, final goal of the project. I found this was a useful way of staying on top of work.

Finally, after my final year project presentation, I felt that there were things that I could have done better to been more effective. From the presentation at the end of the internship, I was able to work on my presenting skills and target these areas. In the future, the ability to present my work well will be something important to being successful and I value any chance to practise and improve my presenting skills.

7.0) CONCLUSIONS

My time at has been a very enjoyable learning experience for me. From my internship project, I have been able to develop and broaden my knowledge in:

- Image processing techniques
- Machine learning applications
- Formal software development practices, namely Agile Software Development
- Problem solving and logical thinking to overcome obstacles
- Planning and time management skills
- Software coding skills
- Reporting and presenting details of my work

I have also gained an insight into life working in the industry and have begun my transition away from university work. Overall, I feel this internship has been valuable for meeting new people and developing skills that I will need to draw upon in the future.

8.0) ACKNOWLEDGEMENTS

I would like to thank the following people who gave me help and advice and supported me throughout my time at c:

- J(ı – my project mentor, who helped me become familiar with formal software engineering practices and was always willing to provide assistance
- n – the software engineering manager, who provided feedback and gave valuable insight and guidance into formal software practices and working in general
- – my project sponsor, who provided useful feedback at weekly meetings on my progress

Well written & presented!