Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines

By

Lai Suk Ling

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2019

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines

Academic Session: JAN 2019

I LAI SUK LING

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________	_________________________
(Author's signature)	(Supervisor's signature)
Address	Dr. Ooi Boon Yaik
_________________________	Supervisor's name
_________________________	
Date: ____________________	**Date**: ____________________

Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines

By

Lai Suk Ling

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A OPTICAL CHARACTER RECOGNITION (OCR) SOLUTION FOR CAPTURING DATA FROM LEGACY MANUFACTURING MACHINES**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________

Name : ______________________________

Date : ______________________________

ACKNOWLEDGEMENTS

I would like to thank to my supervisor, Dr. Ooi Boon Yaik for giving a chance to engage into the Optical Character Recognition (OCR) Solution For Capturing Data From Legacy Manufacturing Machines project. Besides, I would like to express my appreciation to CREST for funding the project.

Lastly, a million thanks to my family and friends for their encouragement and support throughout the whole Final Year Project

ABSTRACT

This project is named Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines. Literally, it is a system that can read data from screen of the legacy machines for the operator for monitoring purpose. In fact, there are still exist some factories and manufacturers applying old machines rather than upgrading the infrastructure to Industrial 4.0 which makes everything connected. This trouble the productivity and monitoring process. However, upgrading the environment to the Industrial 4.0 is not easy as the cost is expensive and the machines are proprietary solutions therefore no customization is allowed. Some machines have to run 24 hours therefore it is unable to off for customization and modification as it will cause loss to the company.

This project would help in solving the problems in the lowest cost without interrupting the operation. This project is to find the most suitable OCR engine, to develop an automated OCR solution with minimal human calibration and to transmit the data to the cloud for data storage. This system will be hanging in front of the machine's screen and collects the data so that operator can have a clear view of the status of the machine's operation. This system able to capture the screen of the machines using the Raspberry Pi camera, recognize the character/data on the screen using existing OCR engine and upload the data to the cloud for storage purpose Google cloud. The OCR engine chosen for the system is Tesseract after doing the researches.

Researching, analyzing and evaluating are a must for a good project. The review on the existing solutions based on the criteria. The criteria are 1) the solution is online or offline, 2) need of cropping, 3) detect the character automatically, 4) font and language limitation, 5) needs of human intervention. OCR engines has been reviewed to find the most suitable engine to the project.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

OCR	Optical Character Recognition
LED	Light Emitting Diode
LCD	Liquid Crystal Display
SSOCR	Seven Segment Optical Character Recognition
ESO	Easy Screen OCR
API	Application Program Interface
GUI	Graphical User Interface
CPU	Central Processing Unit
RAM	Random Access Memory
MB	Mega Byte
HD	High Definition
CLI	Command-line Interface
GCP	Google Cloud Platform
MQTT	Message Queuing Telemetry Transport
ROI	Region of Interest
HTTP	HyperText Transfer Protocol

Chapter 1 Introduction

1.1 Problem Statement

Despite the advancement of OCR technology, the use of OCR for reading LCD from industrial machines from the yesteryears are limited. From our literature review, existing OCR solutions revolves around recognizing text from physical documents. The process involves a) scan the document, b) crop the region of interest and c) recognize the characters and process it. Although the process is convenient for physical documents processing, it is not suitable for capturing readings from industrial machines, which always displaying the reading that we concern on the LCD/LED. It is impossible to implement the OCR system inside the legacy machines since it might be tedious and will lead problems to the factories or industries. This project is motivated by these reasons and it is developed to solve the problem.

This project will propose the solutions for handling the existing problems. The problem statements of this project are as follows:

a. There are many great OCR engines such as “Google OCR” and “Tesseract”. All of them have their own pros and cons. It is not known which of the OCR is best for this project to use to develop the proposed solution.
b. Existing OCR solutions are designed for converting text from documents and images to text. It requires many human intervention therefore it is not suitable for capturing data from LCD/LED screen continuously and automatically.
c. Existing OCR solutions are designed to convert text images to text but the solution is yet completed for IoT solution. The existing solution is not designed to capture data from LCD/LED screen and upload to the cloud for data storage and ease of monitoring.

1.2 Background and Motivation

Although we are now in the era of industrial 4.0, there are still a lot of factories who do not update to the era of industrial 4.0 due to some reasons. The most common reason will be the insufficient of capital to update their machines or system and their environment. They had invested a huge amount of funds into the system when they are in the era of industrial 3.0 or even below. It will cost a lot of money to set up the infrastructure and environment from scratch in order to get the machines and devices connected, via cloud or local server. Besides, there might be some operators who think that they have used to the old machines and solutions and do not want to adapt to the new technology, or perhaps some products can only produce by the proprietary solution which customization is not allowed. Furthermore, the legacy machines are all offline, therefore they might need people to monitor the machines, to observe the value on the LCD/LED display to avoid operation failures. Human always make mistakes, therefore hiring a worker to look at the display is not the best solution towards the problem.

Factories really need to adapt to the concept of industrial 4.0 in order to pull up the productivities. Having worker to monitor the machines by observing the display might lead to many problems. Worker might misread the display value or even miss out because the worker might not 24/7 looking at the display. Some factory might hire few workers to take turn to survey on the display since the machines is running 24/7 to ensure the operation always running. In fact, hiring few workers may cost more than adapting to the new technology in a long run. Besides, hiring people may also cause problems because nobody can make sure that they can focus on the display all the time. The legacy systems mostly have their own proprietary solution and it is not flexible to do customization. It is impossible to shut down the operation just for the purpose of amending the solution. It will make an enormous lost to the company. Therefore, unable to 24/7 tracking the display value is a very critical problem to the factories because the slightly change of the value will cause a huge trouble and may affect the production of the factory.

Optical character recognition (OCR) consider in the field of pattern recognition, artificial intelligent and computer vision. It is also a well-known technology used to read or recognize the text inside an image or scanned document (docparser, n.d.). It can even read the handwriting word if possible. How OCR works is that it will analyze the light and dark area to identify the text and convert those text no matter it is printed or handwritten into the machine encoded text (Rouse, n.d.). There are three steps on how OCR works: (i) open or scan the image that containing the text or the document in the OCR software, (ii) recognizing the text in the document or image, (iii) save the produced document in the preferred format. OCR is a very useful technology exist in the world. It has been treated as a form of information and data entry from the printed paper, which just need to scan through the document instead of entering the data one by one into the machine manually. Nowadays, OCR has been developed and could help in many specific domain applications with OCR. For examples, receipt OCR which use to scan and read the receipt, invoice OCR that could easily read the data in the invoice and other OCR applications.

Since there are so many legacy manufacturing systems from era of below Industry 3.0, there is a need to improve these systems to the era of Industrial 4.0 for productivity purposes. Collecting machineries data manually is definitely not efficient and very often prone to errors. Therefore, getting the reading from the machine's screen automatically with minimal human intervention and upload the data to the cloud for storage will definitely increase the quality of the monitoring process.

Although modifying the existing machines to have the self-monitoring function is possible, the amount of downtime and cost needed for manufacturers to upgrade their existing solutions may not justify such modification. Therefore, the proposed solution of this work provides an alternative for manufactures to retain their legacy machines and yet ready for the industry 4.0 challenges.

1.3 Project Objectives

The main aim of this project is to develop an end-to-end OCR solution that can be used to capture data from the LCD/LED displays of yesteryear machineries with minimal human intervention. Therefore the objectives of this project are:

a. To identify the best OCR engine to this project for recognizing the characters
b. To develop an automated OCR solution which can be used to capture data from the LCD/LED display without much calibration such as image capturing, image cropping and rotating the tilted image.
c. To develop a solution where the captured data are automatically transmitted to the cloud for data storage.

1.4 Proposed Approach/Study

This project will deliver an OCR system that works on LCD/LED screen to capture the data. The system also includes data storage in cloud. The system will be designed a camera to capture the LCD/LED screen automatically and do the calibration automatically with minimal human intervention. Once the image is captured by the OCR the data will then upload to the cloud for storage and ease the monitoring purposes.

1.5 Highlight of What Have Been Achieved

This project will deliver an OCR system that works on LCD/LED screen to capture the data. The system also includes data storage with cloud service. The system will be designed a camera to capture the LCD/LED screen automatically. Once the image is captured, the image will be processed. There is a pair of QR Code at the sides of the region of interest, therefore OCR will only recognize the region to avoid from getting unnecessary noise from the image. The system able to detect more than a ROI if there is multiple pairs of QR Code. If the image is upside down or tilted, the system able to reposition the image so that the data can be recognized with high accuracy. To avoid low accuracy of data, the system will only process those images captured from certain degree. After processing and recognition of data, the data will uploaded to the cloud along with the image capture date and time.

1.6 Report Organization

The report consists of 6 chapters, in addition to the appendix and references. The first chapter is about the problem statement, the background, motivation, objectives, proposed approach, and the highlight of what have been achieved. Chapter 2 is about the literature reviews and the comparison between the existing systems and OCR tools. Chapter 3 is about the system design. In this chapter, there will be diagrams for a better understanding towards the project. The diagrams are use case diagram, activity diagram, sequence diagram and communication diagram. The following chapter, chapter 4 is about to discuss on the methodology and tools using and the implementation of the system. However on chapter 5, testing result on the system will be recorded down.

Chapter 2: Literature Review

2.1 Overview of Existing Solutions

In recent day, many things around us has become digitalized and computerized. There are a lot of the household appliances has the LED or LCD display for the ease of human to operate them (Tekin, et al., 2011). Besides, the number on the display may be very critical to the production flow (Xie, 2009). While in industry area, there are a lot of the machines that are mostly adapted to automation and had reduced man power. There are existent of very user-friendly application that apply the OCR to help the vision impaired person to read. It will convey the message (e.g,: on the book) into the vocal voice, which act like someone is help them to read the text. Although they could read the characters in printed documents or book easily and more accurately, but it does not mean that the result will be the same when read the characters on the LCD/LED display. . There are few solutions toward getting the texts and characters from screen, especially from those legacy machines which customization are not allowed. A visual approach is a smart and diligent way to solve the problem as there will be a direct access to the application through the user interface (Gleichman, et al., 2011). In common, OCR will always have weakness as it could not know the captured image is readable or not at first until the character/s is/are generated, and the OCR processing latency worsen the recognition process, more time is required to determine if the document captured is readable (Cutter & Manduchi, 2015).

In this section, existing OCR solutions are reviewed. The purpose of this review is to evaluate the suitability and identify the functionality gaps of using the existing OCR solutions to achieve our intended purpose.

The evaluation criteria include:

a) Online/Offline – Does the solution requires internet to perform its function.
b) Cropping – Does the solution need user to crop the image to reduce the size of the image.
c) Auto Detect Character – Can the solution automatically detect the text/character without human intervention.

d) Font limitation –Can the solution recognize all type of font but not limited to some type.
e) Multiple language – Does the solution recognizes more than one type of language.
f) Need of human intervention – Does the solution requires human to capture, cropping and other calibration to perform its function.

2.1.1 OCR of seven segments display

Seven segments display means the output shows via 7 bars. The multimeter shows it output via 7 segments display and the reading values wanted to be extract out of the multimeter display. Therefore, image was taken from the multimeter using camera and the image will be processed for converting the reading value in the image into the text. There are so many types of OCR engine in the world, and each will have their own specialty. The OCR engine chose to perform the text recognition is SSOCR (seven segment optical character recognition). SSOCR is a program that has been written for recognizing the seven segments display only (Seven Segment Optical Character Recognition, n.d.).

The image of the multimeter was capture using a mobile phone, then the image was scale to a very small size until the reading is fully filled up the image. The result came up showing the text which almost same as the number in the text, just that the decimal point had left out.

Figure 2.1.1.1: Cropped image used in for the text recognition (jbeale, 2013)

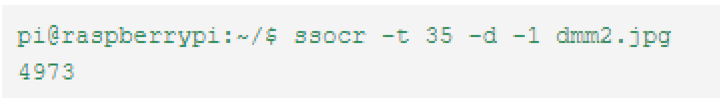


Figure 2.1.1.2: The result (jbeale, 2013)

The limitation of this solution is that the image has to be cropped into a very small size to generate a very accurate result. Besides, the decimal point could not be recognized as part of the text. Furthermore, this solution only can detect 0-9 characters instead of the normal alphabets and it can't recognize the characters that is not in 7 segments format.

2.1.2 Easy Screen OCR (ESO)

Easy Screen OCR is an OCR software that will do the recognition and conversion online. Like what a normal OCR software does, it will extract the text in the image content and convert it into a format for edit purpose, which make may increase the productivity (Steven, n.d.). There are some key features which make this software so powerful: -

- Multiple languages. It can recognize more than 100 different languages.
- Google OCR as the engine. Google OCR is an advanced engine in OCR area.
- Online OCR process. The process is happened in the cloud, not occupying the user computer resources.
- Hot keys. Shortcut keys make the whole process faster.
- Different image format.

ESO can implement in computer with different operating system. It can be implemented in Windows and Mac. ESO has a very hard accuracy on recognizing the text because it has used the google OCR module of google cloud vision. Google cloud vision API are said to be have high performance is because "The Google Cloud Vision API takes incredibly complex machine learning models centered around image recognition and format it in a simple REST API interface" (Chan, n.d.). ESO has graphical user interface (GUI), which user interface that user will prefer to. This is because command-line interface is not user-friendly, unless the user has much experience with it. Moreover, it allows user to select the region of interest for OCR. It does not fix the user to process the image himself before using the ESO.

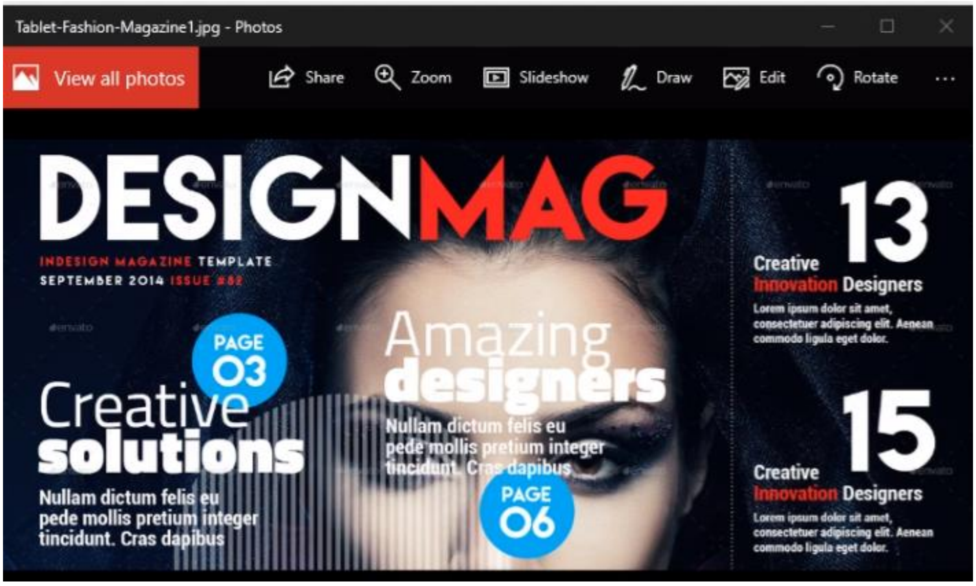


Figure 2.1.2.1: The image for OCR process (snapshot from ESO demo video, n.d.)

Figure 2.1.2.2: The user selects the region of interest instead of scanning the whole image. (snapshot from ESO demo video, n.d.)

Figure 2.1.2.3: The region of interest is under the OCR process. (snapshot from ESO demo video, n.d.)

Figure 2.1.2.4: The result after the OCR process. (snapshot from ESO demo video, n.d.)

Selecting the region of interest is a good feature from ESO. This is because uploading a smaller file size is much easier than the larger file. Smaller file size will required smaller bandwidth therefore the performance can be increased. The whole OCR process is worked in the google cloud, therefore user could save up their computer resource such as CPU for computation, RAM and so on. ESO will only occupy less than 8MB in the computer (Steven, n.d.). Therefore, this solution is suitable to those low-spec computers which may have disaster using the OCR software that needs higher computer-spec.

Using online OCR could have both pros and cons. Although using google OCR could save up computer resources, it requires internet. Requiring internet usage can be a very huge cons. How about if there is internet problem which is not control by the user? This solution is not free, user has to subscribe it either in monthly, half-yearly or yearly basis. Besides, larger image file requires longer time and wider bandwidth of the internet to upload the file to the google cloud for OCR purpose. If there is a lot of large images to be done, the process might slow down and may not as efficient as other offline OCR. Since it is in subscription basis and need internet, it may cost user a lot. Furthermore, ESO does not state any storage issue because uploading files to the cloud will occupy some cloud storage space. How if each user has exceeded the cloud storage limit? If there is add on to the cloud storage, it will cost a lot towards those users that need a large storage.

2.1.3 DaVinci HD/OCR Video Magnifier

DaVinci is a set of assisting tool that is very useful to visual impaired people. It consists of a HD monitor, a 3-in-1 camera and text-to-speech (OCR). DaVinci is a product from Enhanced Vision, which is a leading manufacturer of magnifying solution for the visual impaired in America. With a HD monitor, an excellent color and contrast will deliver a high-quality image, clear picture and vibrant colours (The Low Vision Centers of Indiana, n.d.).

Since DaVinci has a lot of functions, I will focus on one of the features which is the OCR feature. With the OCR features, it will help to read out the text when user pushes a button. It can either take pictures with the HD camera, or either from the screen shot in the monitor. Some people may feel tired on eye when read a lot of text and words, therefore, OCR features will recognize the characters, and convert the text to speech so that the user could still able to continue his reading even if his eyes are not comfortable. DaVinci could read the content using the camera but need human intervention to capture the image by pushing a button. The camera capturing is not an automation process but need human intervention.

Figure 2.1.3.1: DaVinci captures the image for OCR in a guided frame (snapshot from DaVinci User manual, n.d.)

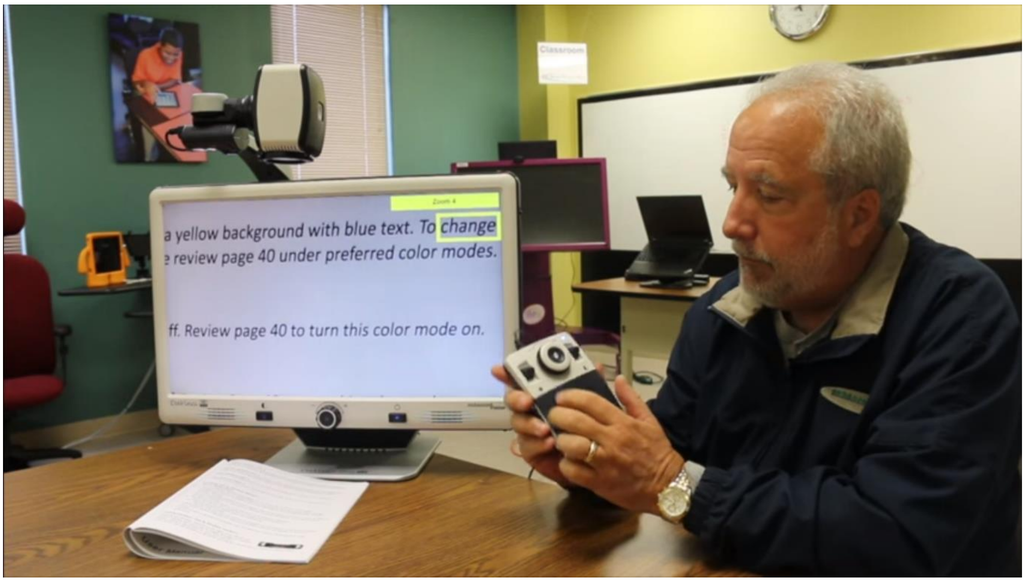


Figure 2.1.3.2: The framing of the words (INDATAProject, 2015)

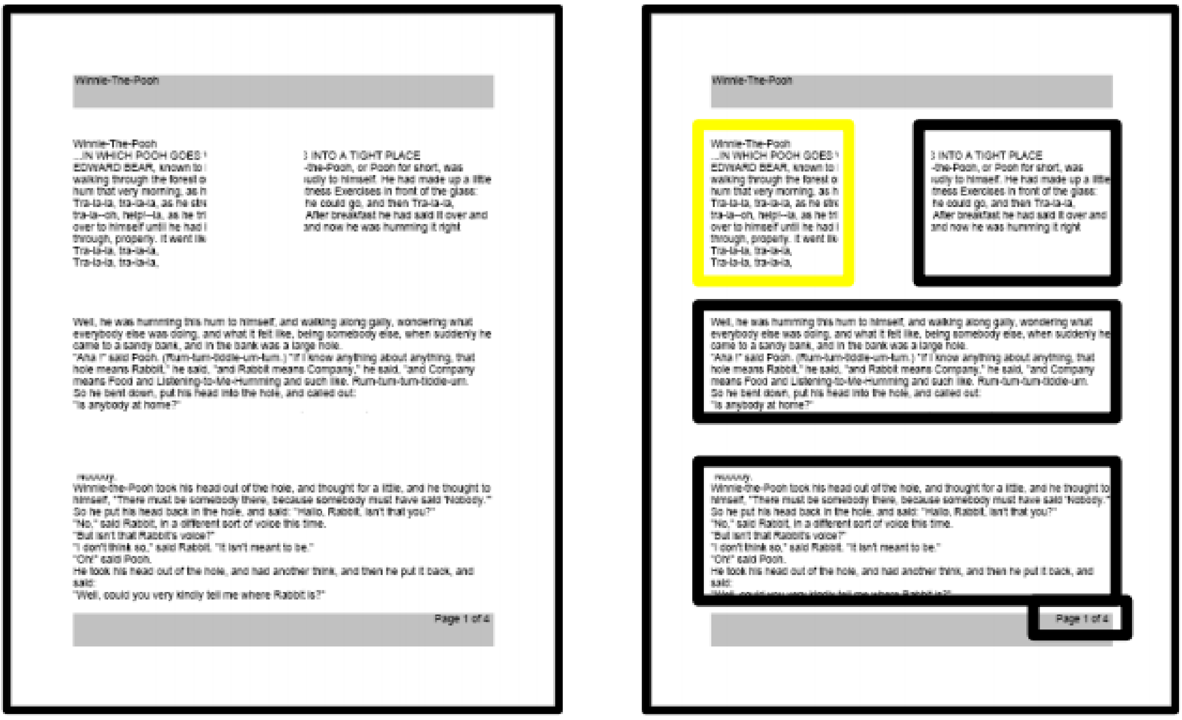


Figure 2.1.3.3: The OCR could segment the content (snapshot from DaVinvi user manual, n.d.)

It will frame the word/character when it read the words. To enter the OCR mode, a button is needs to push so the user could start capturing the image. The image needs to be fit nicely into the guided frame (shows in Figure 2.7). The OCR of the DaVinci is very powerful, it could recognize the text and characters no matter what position of the image is show to the camera. It supports multiple orientations for documents, just that make sure the document is viewable under the camera arm (Enhance Vision, 2018). Last but not least, DaVinci provides a wide range of languages.

The only limitation of the OCR features is that it could not take the picture when the user has some distance from the system. This is because the system requires to push a button in order to capture the image. To solve the problem, the control of the system should not be wired but use the bluetooth remote control instead. Bluetooth is more powerful than Infrared Transmission as bluetooth has wider range than the Infrared Transmission and Infrared need a open path in order to remotely control the devices (Ashish, 2016). Therefore, no matter how far the person from the DaVinci is, he can still use the OCR feature as long as the camera could view the document. There is no big deal about the size when the object is far from the camera. The camera can do magnifying up to 77x. Although the camera can view from an infinitely far object, but the recommended distance is 5-30 feets (Enhance Vision,2018).

2.2 The comparison between the solutions reviewed

After reviewing the solutions based on the criteria, there is differences between the reviewed solutions. Table 2.2.1 shows the comparison between the solutions reviewed.

	OCR of seven segments display (OCR optical character recognition of seven segment displays, n.d.)	Easy Screen OCR (ESO) (Screen OCR, n.d.)	DaVinci (Enhanced Vision, n.d.)
Online/Offline	Offline	Online	Offline
Cropping	Yes	Yes (If want to recognize certain part)	No
Auto detect character	No	Yes	Yes
Font limitation	Yes. Limit to 7 segments display	No	No

Multiple language	No	Yes	Yes
Needs of human intervention	Yes	Yes	Yes

Table 2.2.1: Existing OCR solutions comparison

From the table above, some of the solutions could solved some of the criteria mentioned. Unfortunately, none of the solution could integrate all of the criteria. They have the common problems, which that all of the above involve human intervention and do not further the system for IoT yet. To further explain the problems, firstly, user has to capture the image so that the solutions can process the image for OCR purpose. The systems/solutions could not automatically capture the image for further process. Besides, all the solutions do not have any further process after displaying the results, which mean the existing solutions are not designed for data storage for the ease of monitoring purpose. A system will be very powerful if integrating the OCR and cloud service, which will be suit monitoring the best.

As a conclusion, there is no existing of system that can solve our problems discuss in chapter 1. Developing a system with the criteria above could succeed this project.

2.3 Overview of the OCR engine

There are a lot of OCR engines which helps in many industries. The following section will discuss on the comparison between the popular OCR engines.

2.3.1 Tesseract

Tesseract is an offline open source OCR engine that could run on every operating system and it is considered as a top 3 OCR engine in the world in term of accuracy. (Vincent, n.d.) It runs through a command-line interface (CLI) instead of a graphical user interface (GUI). It has a very good performance in actual character recognition, on the other hand, it has performance problem on multicolumn materials (Vincent, n.d.). Since tesseract was developed from 1985, to be honest, it has some limitation in the older time but for now, tesseract have evolved into so may version.

Now we have Tesseract 4.0, which able to detect characters in many languages. Tesseract does support much language nowadays as many feature libraries are provided (Smith, et al, 2009). In the older day, it could only detect English language (Vincent, n.d.). User have to download the other language training data online before want to use it (James, 2017).

People in recent year are more adapted and exposed to GUI other than command-line interface. It is not user friend since user have to know the parameters and command in order to operate it. Tesseract is different from other commercial OCR, it is intended to serve as a part of another program, it do not provide GUI to user unless integrate with another program (Language Technologies Unit (Canolfan Bedwyr), Bangor University, 2008). Mostly factory operators may not an IT expert, it may bring up some trouble if using the Tesseract without any integration into other program. Last but not least, Tesseract has high accuracy on the black and white documents rather in this fancy modern world. It might be hard for it to maintain its high accuracy if the content of the image is complex with too artistic fonts and many graphics (Leung, 2017).

Tesseract is a good choice for those who want to have some customization on the engine therefore can be applied into different kind of scenario, as the libraries of Tesseract are expandable (Li, et al., 2016)

2.3.2 Google OCR

Google cloud vision is a very power image analysis in recent days. This powerful software provides pretrained model through API (Application Programming Interface) and it do offer flexibility to user, which user could customize the model using AutoML Vision. Google cloud vision helps to analyse the image content and help developers to understand the content of the image. Since it is very powerful, it provides a lot of features such as label detection, optical character recognition, handwriting recognition, face detection and so on. We will only focus on the OCR by google cloud vision in this study.

There are two features support the OCR to work perfectly, which are TEXT_DETECTION and DOCUMENT_TEXT_DETECTION. The TEXT_DETECTION will do text extraction from image while DOCUMENT_TEXT_DETECTION is doing the same things as the

TEXT_DETECTION but it has optimal respond in dense text document. Google OCR will firstly perform a layout analysis on the image, then perform the text recognition analysis for text generation, lastly, correct the errors at post-processing by giving a lot of model or dictionary (Chan, 2015).

Using the vision API is quite convenient. All the recognition will be done online. The images that scanned into the computer need to upload to the google cloud platform (GCP) project in the google cloud for the further OCR process. The google vision OCR is so powerful and do no limit the font to be in 7 segments only. It will easily recognize the text or character in the image and give a high accurate result. Besides, it able to detect the alphabets in the image. There is a difficulty in the old OCR technique which is the text of interest must occupy most of the space of the image whereas the background clutter is compulsory to discard before the display can be read (Tekin, et al., 2011). For OCR of google vision, there is no necessary to have such criteria to make the OCR works. No matter what is the size of the image, google vision OCR able to extract the character unless there are no characters. Google OCR can recognize the characters and returns high accuracy label even the image is in odd angle, unusual font, against complex backgrounds.

As a conclusion for section 2.3, Google OCR will have better performance than Tesseract because Tesseract could not have consistent performance when there not only black and white documents but image with fancy content. On the other hand, the user interface might also become a criterion to judge whether the engine is suitable for anyone. Since Google OCR is GUI based which may be easier to operate than the command-line based Tesseract. Besides, Google OCR is online while Tesseract is offline. Google OCR will be done online which may not consume computer resources, but Tesseract will consume the resources.

Chapter 3: System Design

3.1 Use Case Diagram

Figure below shows the use case diagram.

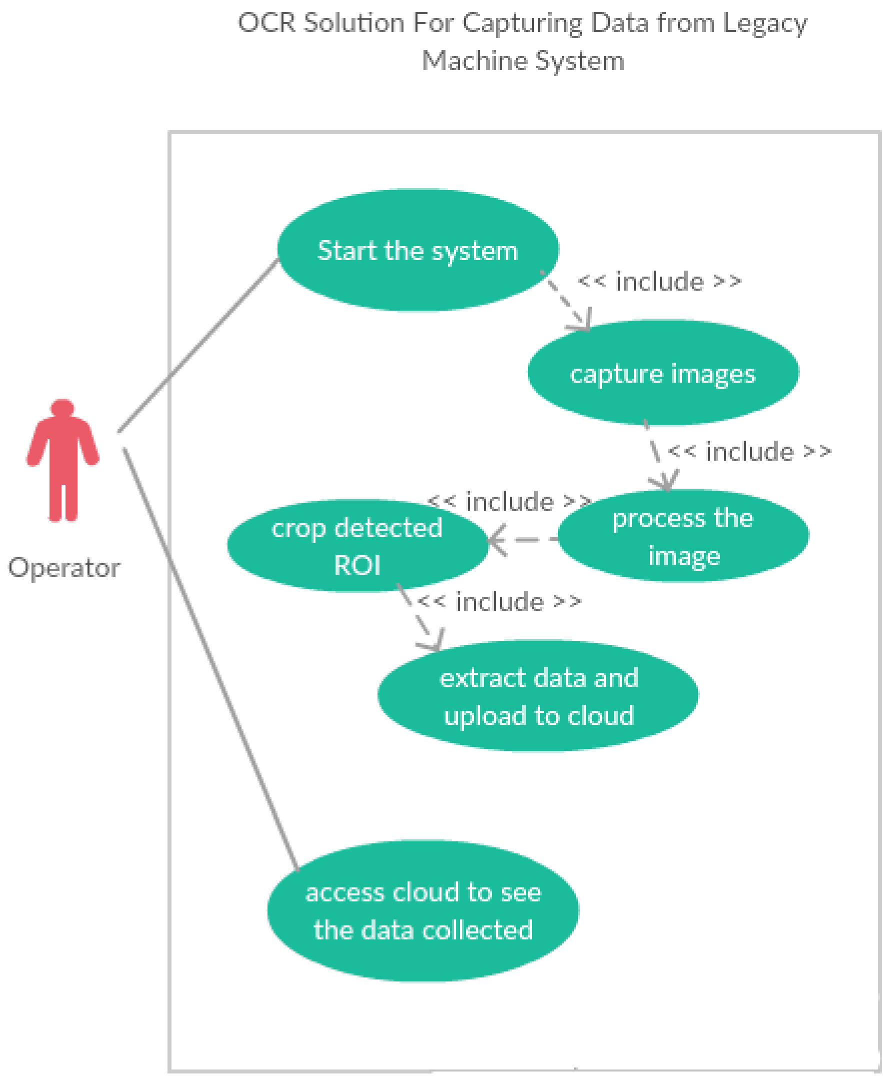


Figure 3.1.1: The use case diagram.

3.2 Use Case Description

The ROI use case is describe in the table 3.2.1

Use Case ID	UC01
Use Case Description	Start the system
Actor	Operator
Precondition	-
Main Flow	1. Operator starts the system. 2. The system starts capturing the image. 3. The system resize the image. 4. The system detect the ROI. 5. The system check if the image can be used. 6. The system repositioning the image. 7. The system get the size of the ROI. 8. The system crop the ROI 9. The system extract and do recognition task. 10. The system delete the image 11. The system upload the data collected
Alternate Flow – Fail to detect ROI	4.1 Delete the image
Alternate Flow – No tilted image	6.1 System skip the repositioning step
Author	Lai Suk Ling

Table 3.2.1: Start the system use case description

Set threshold use case is describe in the table 3.2.2 with ID UC02.

Use Case ID	UC02
Use Case Description	Access cloud to see the data collected
Actor	Operator
Precondition	Internet connection and system was running.
Main Flow	1. Operator access into the cloud. 2. The system display the data.
Author	Lai Suk Ling

Table 3.2.2: Access cloud to see data collected description.

3.3 Sequence Diagram

Figure below shows the sequence diagram of the project.

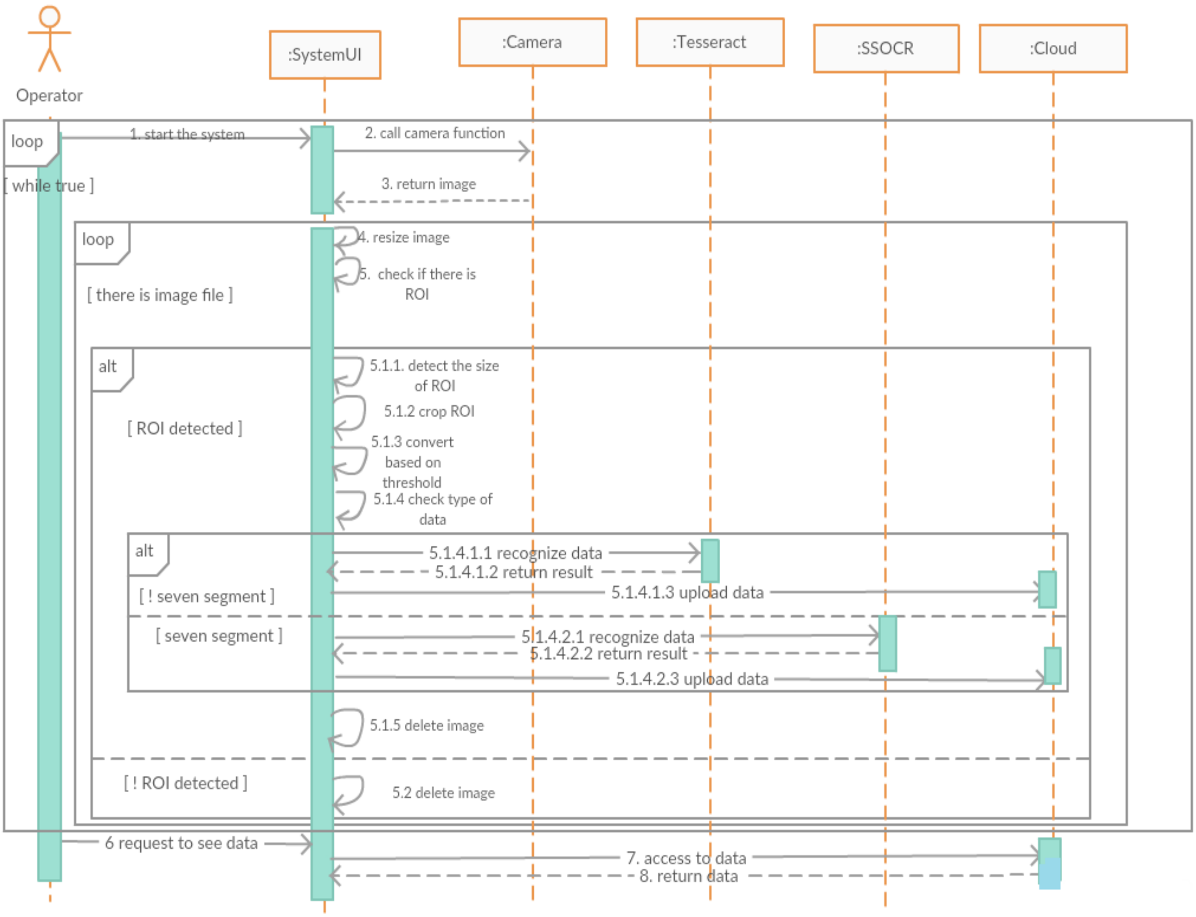


Figure 3.3.1: The sequence diagram of the system.

3.4 Collaboration Diagram

Figure below shows the collaboration diagram of the system.

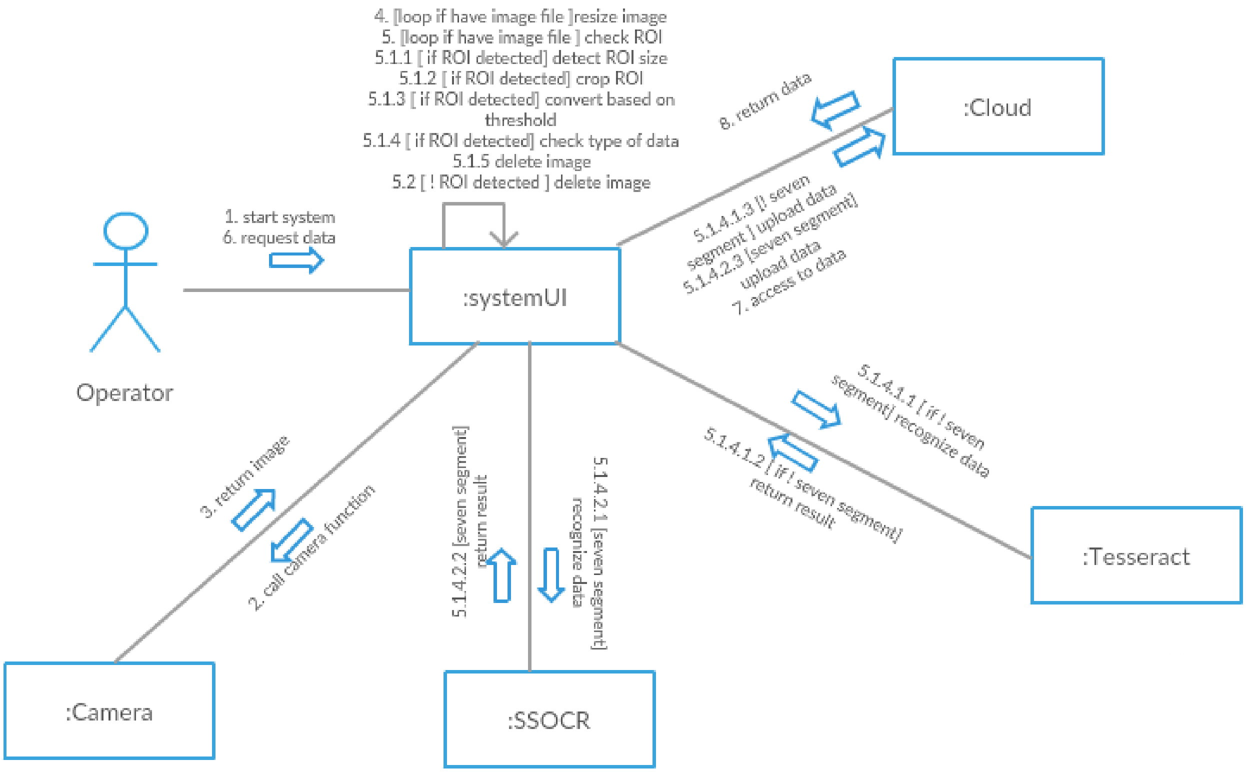


Figure 3.4.1: The collaborative diagram.

3.5 Activity Diagram

Figure below shows the activity diagram of the system.

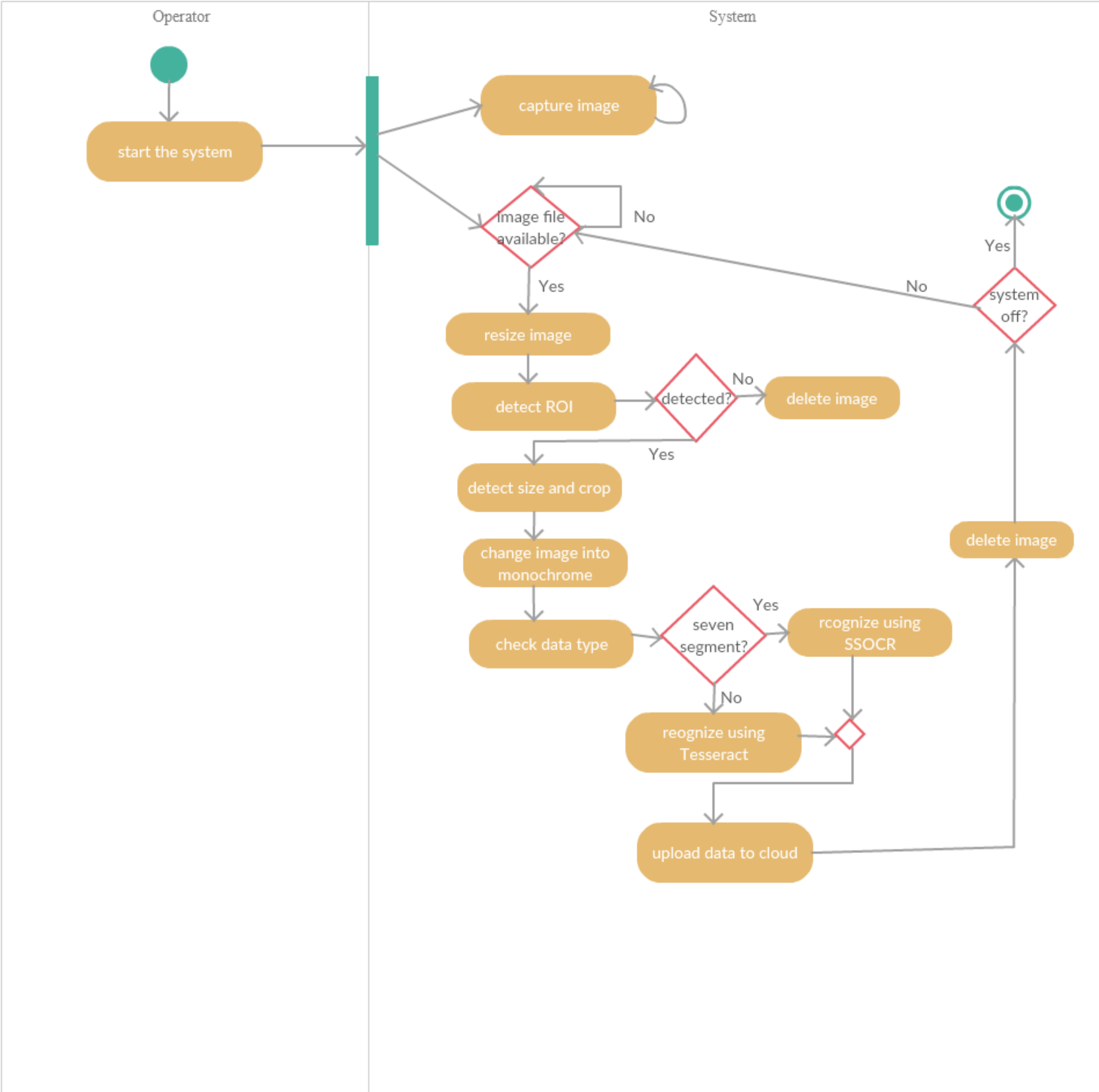


Figure 3.5.1: The activity diagram of the system.

Chapter 4 Methodology

4.1 Methodology and General Work Procedure

The methodology chose to work the project is prototyping. The final system will be delivered after satisfying with the prototypes developed.

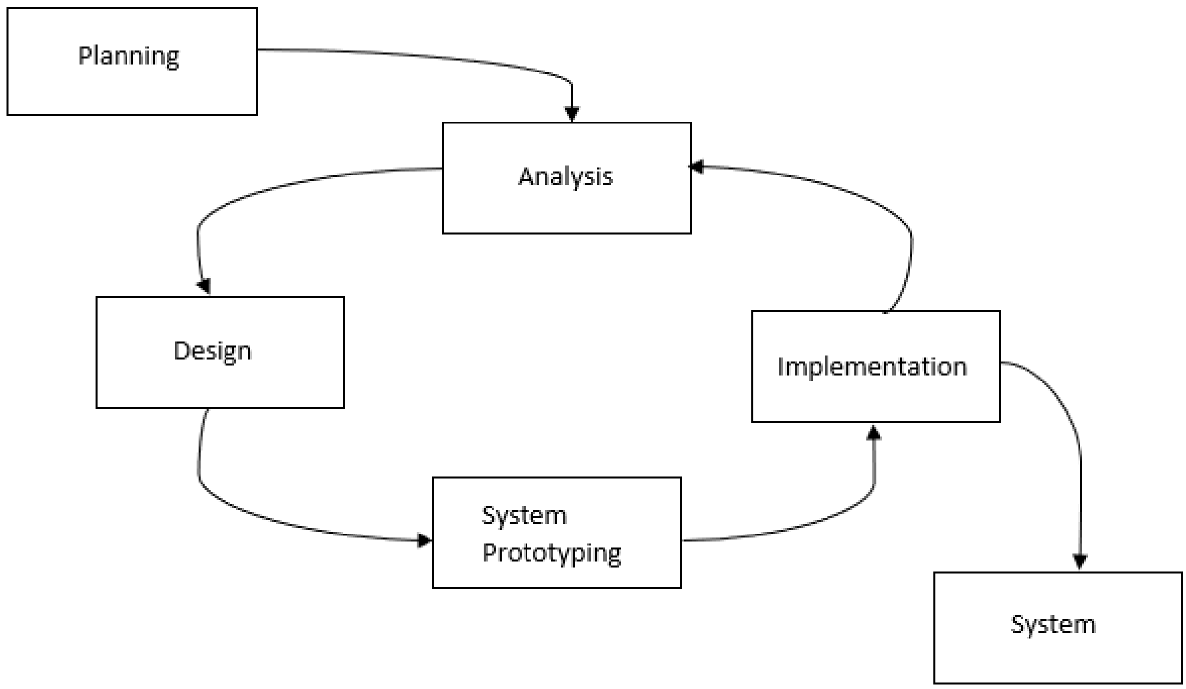


Figure 4.1.1: Prototyping flow

There will have several prototypes for system evaluation and improvement.

1. First prototype is based on the basic analysis and design. The prototype should have the features to capture the image of the LCD/LED screen by camera and the OCR engine is implemented.
2. Second prototype will describe the system where the system has the cloud service for storage on the data collected from the image captured.
3. Third prototype will integrate the first prototype and the second prototype but the camera will able to capture the image and process the data automatically without human to involve in the system.

The steps in delivering the system:

1. Planning: The idea of the project is proposed and discuss with the supervisor. The overview of the system is discussed for a better direction. The scope of the project and the objectives of the project is defined as well. Besides, the requirements of the project are gathered for building the system.

2. Analysis: The requirements are analyzed to build the ideal system. There are many existing systems about OCR and these existing systems have been studied for further understanding. Besides, their strengths and limitations are also been reviewed. These existing systems are compared and the features which could be add into the project system are taken into considerations. Furthermore, the previous prototype develop are analyzed for improvement.

3. Design: The prototypes of the system is design for the ease of doing the system prototyping. There are several prototypes as mention in the above, each prototype will be design in different iteration.

4. System prototyping: The prototype is built into physical. There are 3 prototypes, first with the basic requirements and the next 2 prototypes will add on the additional features.

5. Implementation: The prototype is implemented and tested. The weakness and the insufficient performance of the prototype will be pointed out and become the requirements in the next prototyping. This phase is important for achieving the ideal system project. If the system prototypes are satisfied, then the prototypes will be integrated and deliver as the final product.

4.2 Tools

1. Raspberry Pi 3B+

   Raspberry Pi is the hardware used to build the system. The Operating system is Debian 9.4.

2. Raspberry Pi Camera

   The camera is attached to the Raspberry Pi for capturing image purpose. Before proceeding to process the image, image has to be taken and captured. Camera is used to capture the image.

3. Tesseract

   The software, Tesseract is a very powerful OCR tools, which play a very important role to do recognition work. This OCR tools could give a high accuracy of a result under certain condition.

4. SSOCR

   SSOCR stands for Seven Segment OCR, which literally recognize the seven segment characters. This is because Tesseract is unable to detect the seven segment character.

5. Zbar

   Zbar is a library to use in python. Zbar is use to detect the QR Code in the image, to find the position and location of the QR Code so that the system able to do calibration before recognition work for a higher accuracy of the result.

6. ImageMagick

   A software used to process the image such as crop, rotate and convert image into monochrome according to the provided information such as the size of the image to be crop and the degree to be rotate.

7. Google Sheet (cloud)

   To store the data captured for the ease of tracking and the ease of monitoring purpose.

4.3 Timeline

Figure below shows a Gantt Chart.

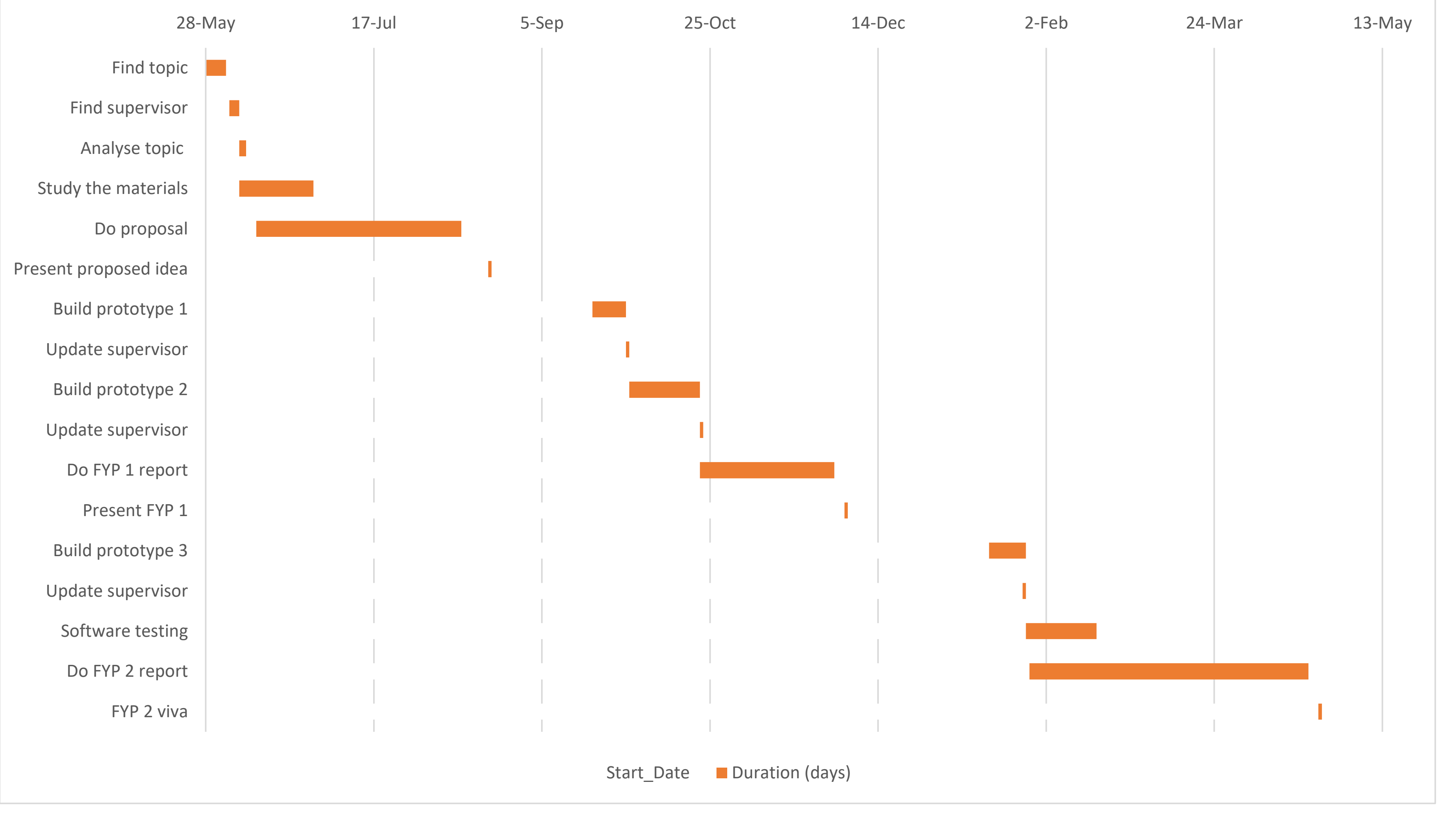


Figure 4.3.1: Gantt Chart

4.4 System Implementation

This project is to capture data from LCD/LED screen of machines and upload to the cloud for ease of monitoring purpose. However, there are a lot of work have to be done in order to pull the accuracy of the recognition task, especially when there is no human calibration and intervention, everything has to be automated.

The hardware of the system is built using Raspberry Pi 3B+ and Raspberry Pi camera. The operating system is Debian 9.4. There are functions implemented into the system. To combine all the functions, bash script and python are used to complete the system. The image taking and processing the image will work in parallel. The image will resize to 50% of the original size to reduce the time needed to process an image.

The first function to discuss is the recognition task. Tesseract is a powerful OCR tool, it is downloaded and tested to check if it really works. Other than Tesseract, SSOCR (Seven Segment Optical Character Recognition) is chose to implement into the system as well because one of the cons of the Tesseract is not able to detect and recognize seven segment characters.

To minimize the human intervention and calibration, the system should able to detect the ROI instead of processing the whole image captured. To know the position and size of the ROI, QR Code is used for detecting the ROI. For the system to detect the QR Code, Zbar library is downloaded and use in python platform. Python file imports the zbar library, and all the calibration is done in the python file to make the system automated. The system able to return the position of the QR Codes, from the position/location returned, the starting position of the ROI is known and the size of the ROI can be calculated, hence the area of the ROI is knwon. The system able to detect multiple ROIs and able to process them as well.

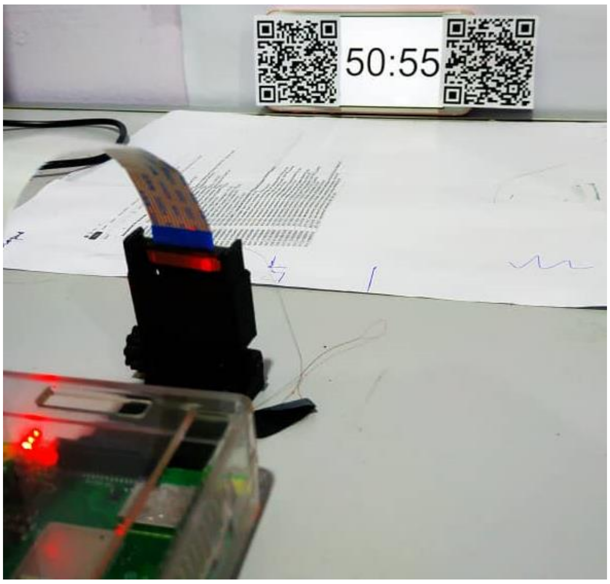


Figure 4.4.1: The QR Code on the screen.

From the figure above, the system able to detect the qrcode and calculate the size of the ROI and output the size into a file.

Other than detecting the ROI, there is a problem with the angle to take pictures. When the camera is set at certain angles from the screen, the result will be inaccurate as the camera is not able to ‘see’ the data from the screen. For example when 180 degree from the screen. To avoid from getting the inaccurate result, those images have to be ignored and not processing them. To let the system know the current angle of the camera, use the area ratio of the qr codes. When the camera is placed right in front of the screen, then ratio should be approximate to 1:1. Therefore, for every image that has the ratio less than 0.7, the image will be ignore and deleted. The testing on finding the angle will discuss in Chapter 5 later.

```
if ( larea >= rarea ):

        if not (rarea/larea >=0.70 and rarea/larea <= 1.00):
                qrcode.remove(left)
                qrcode.remove(right)
                detail.remove(cur2)
                detail.remove(cur)
                flag=True
                break

else :
        if not ( larea/rarea >=0.70 and larea/rarea <= 1.00 ):
                qrcode.remove(left)
                qrcode.remove(right)
                detail.remove(cur2)
                detail.remove(cur)
                flag=True
                break
```

Figure 4.4.2: Code section used to eliminate the fail images.

The code above shows how to eliminate the ROI that do not fulfill the condition.

In addition, the system could not detect and process the image if the image is tilted. With the existing of QR Code, the problem could be solve. When the top left and top right of the QR code are not in a same Y location, meaning that the image is tilted. With the formula of :

$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

the degree of rotation is able to find out and hence the image can be rotate to the position that can perform the recognition task the best. The code to find out the rotation degree is:

```
def checkTilt(left,right):

        a = abs(left.location[0][1] - left.location[3][1])
        b = float(math.sqrt((left.location[0][0] - left.location[3][0])**2 + (left.location[0][1] - left.location[3][1])**
        c = abs(left.location[3][0] - left.location[0][0])
        result = degrees(acos((b * b + c * c - a * a)/(2.0 * b * c)))          # calculate the degree of rotation
        if not ( left.location[0][0] == left.location[1][0] and left.location[0][1] == left.location[1][1] ):

                if( (left.location[0][1] > left.location[1][1]) and (left.location[0][1] > left.location[2][1])
and (left.location[3][1] > left.location[1][1])
and (left.location[3][1] > left.location[2][1])):       #check if the image is upside down

                        os.system('convert '+sys.argv[1]+' -rotate "'+str(180)+'" '+sys.argv[1])

                        if (left.location[0][1] > left.location[3][1]):
                                os.system('convert '+sys.argv[1]+' -rotate "'+str(360 - result)+'" '+sys.argv[1])

                        else :
                                os.system('convert '+sys.argv[1]+' -rotate "'+str(result)+'" '+sys.argv[1])
                else:

                        if (left.location[0][1] < left.location[3][1]):
                                os.system('convert '+sys.argv[1]+' -rotate "'+str(360 - result)+'" '+sys.argv[1])

                        else :
                                os.system('convert '+sys.argv[1]+' -rotate "'+str(result)+'" '+sys.argv[1])
```

Figure 4.4.3: The image rotating code section

The figure above shows the way to find out the degree of rotation.

After the image is done processing, the ImageMagick will continue the task. The ROI will be crop according to the size of the ROI. Since there will have flicker problem when capturing image from LCD/LED screen, the cropped image have to convert into monochrome. After that only proceed to recognition process. The figure 4.4.4 below describe the processes.

Figure 4.4.4: The original image captured by the camera.

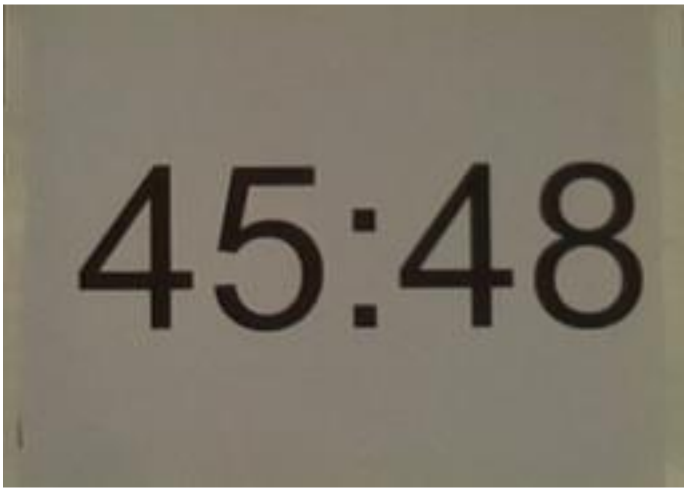


Figure 4.4.5: The cropped image

Figure 4.4.6: The image converted into monochrome.

After processing the image and get the data, the image will be delete to save the memory space. Besides, the metadata of the image will be extract to let the operator know when the image is captured and upload to the google sheet (cloud) along with the data. The update of the google sheet do not need human intervention, it will update the google sheet automatically with ‘curl’ command.

```
&& curl "https://script.google.com/macros/s/AKfycbw1qXmQPy0Y9NuC0XN97cvxjoMWcq9oARVAb97fI3-CUIiYj4yd/exec?CaptureTime="+$time+"&data="+$dat
```

Figure 4.4.7 : The CURL command to update the google sheet.

```
function doGet(e){
  try {
    //update the doc and sheet variable according to your configuration
    var doc = SpreadsheetApp.openById("1jYiFG5fGZBEgSkfnXXhFTtXLf4MWA-lvzdSOEPPXQ8c");
    var sheet = doc.getSheetByName("Sheet1");

    // we'll assume header is in row 1
    var headRow = e.parameter.header_row || 1;
    var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];
    var nextRow = sheet.getLastRow()+1; // get next row
    var row = [];
    // loop through the header columns
    for (i in headers){
      if (headers[i] == "timestamp"){ // Include a 'Timestamp' column for time
        row.push(new Date());
      } else { // else use header name to get data
        Logger.log(e.parameter[headers[i]]);
        row.push(e.parameter[headers[i]]);
      }
    }
    // more efficient to set values as [][] array than individually
    sheet.getRange(nextRow, 1, 1, row.length).setValues([row]);
    // return json success results
    return ContentService
          .createTextOutput(JSON.stringify({"result":"success", "row": nextRow}))
          .setMimeType(ContentService.MimeType.JSON);
  } catch(e){
    // if error return this
    return ContentService
          .createTextOutput(JSON.stringify({"result":"error", "error": e}))
          .setMimeType(ContentService.MimeType.JSON);
  }
}
```

Figure 4.4.8: The google script to update the data to the cloud.

Chapter 5: Testing and Result

5.1 Overview

In this chapter, the result of before and after implementing the proposed solution will be discussed. Before the proposed system solution, the Tesseract shows it weaknesses, such as unable to detect the ROI as it will recognize all the characters in the image, unable to recognize when the image is tilted and upside down, unable to detect seven segment as others. With the proposed solution, the system able to achieve the objectives stated, which identified the suitable OCR tool that suit the system the best, automatically capture the image, calibrate the image without human intervention, accurate data recognition and able to upload the data to the cloud for storage and easy for monitoring purpose.

5.2 Comparison On Before and After The Proposed Solution

Before Implementing The Project Solution

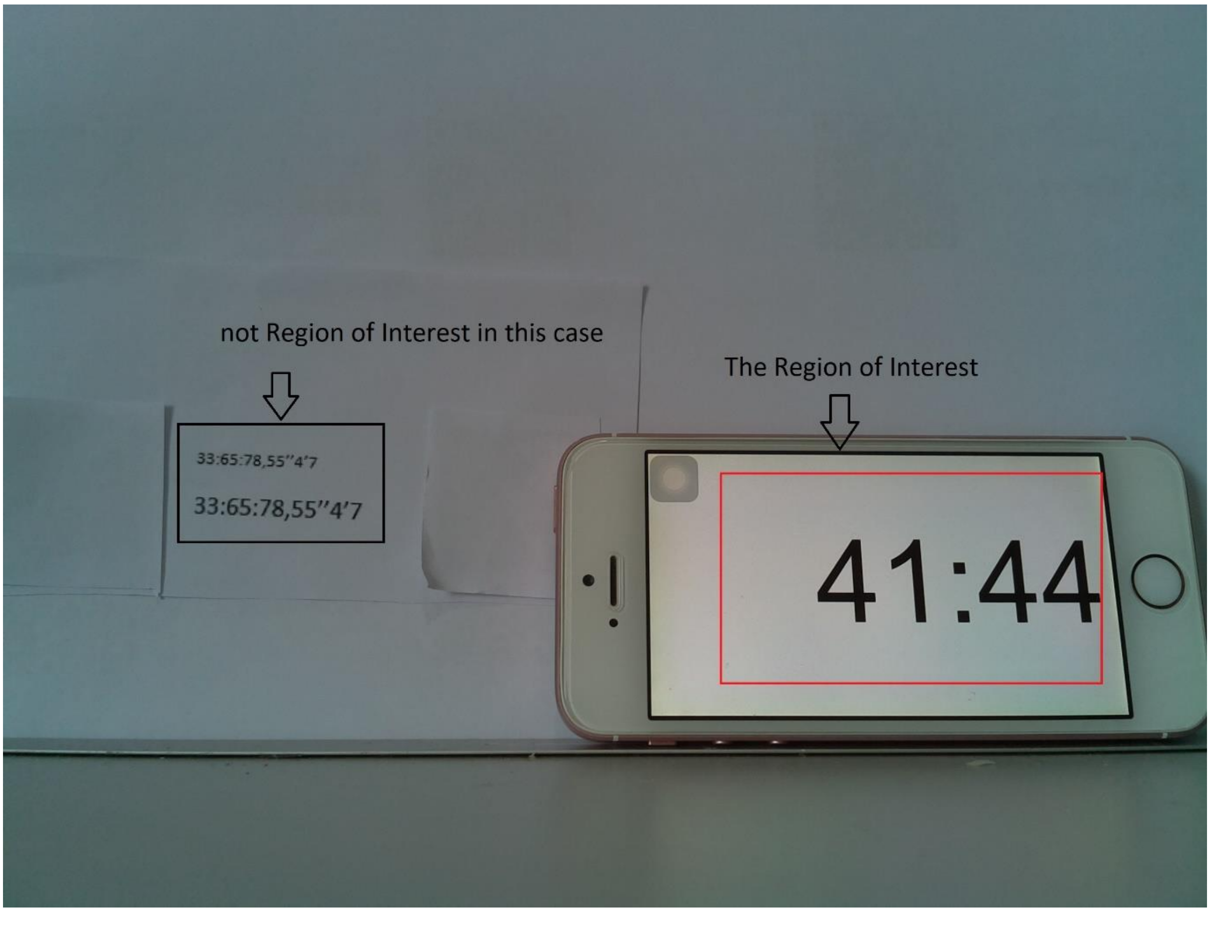


Figure 5.2.1: The image captured by the camera.

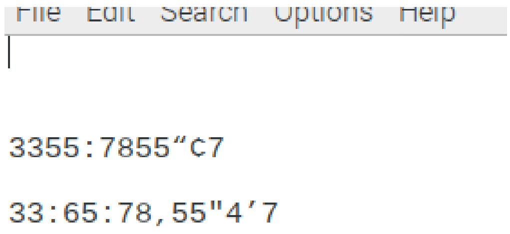


Figure 5.2.2 : The result of the image capture after recognition task

From the Figure 5.2.1 and Figure 5.2.2, Tesseract not able to detect the ROI but it recognize the non ROI. The result is not the correct and this will lead to a big problem for the monitoring purpose. This is because Tesseract not able to recognize characters from the LED/LCD screen, due to the flicker problem.

Figure 5.2.3: The tilted image

Figure 5.2.4: The tilted image

Figure 5.2.5: The tilted image

Figure 5.2.6: The upside down image

For the Figure 5.2.3, Figure 5.2.4, Figure 5.2.5 and Figure 5.2.6, the Tesseract not able to detect and recognize the characters from the screen as well although the image has been process to eliminate the flicker problem. Since the Tesseract does not have the feature to upload result to the cloud, the result is output to a text file. However, the text file is empty as it does not have the ability to recognize characters from these images.

Figure 5.2.7: Image captured with seven segment characters

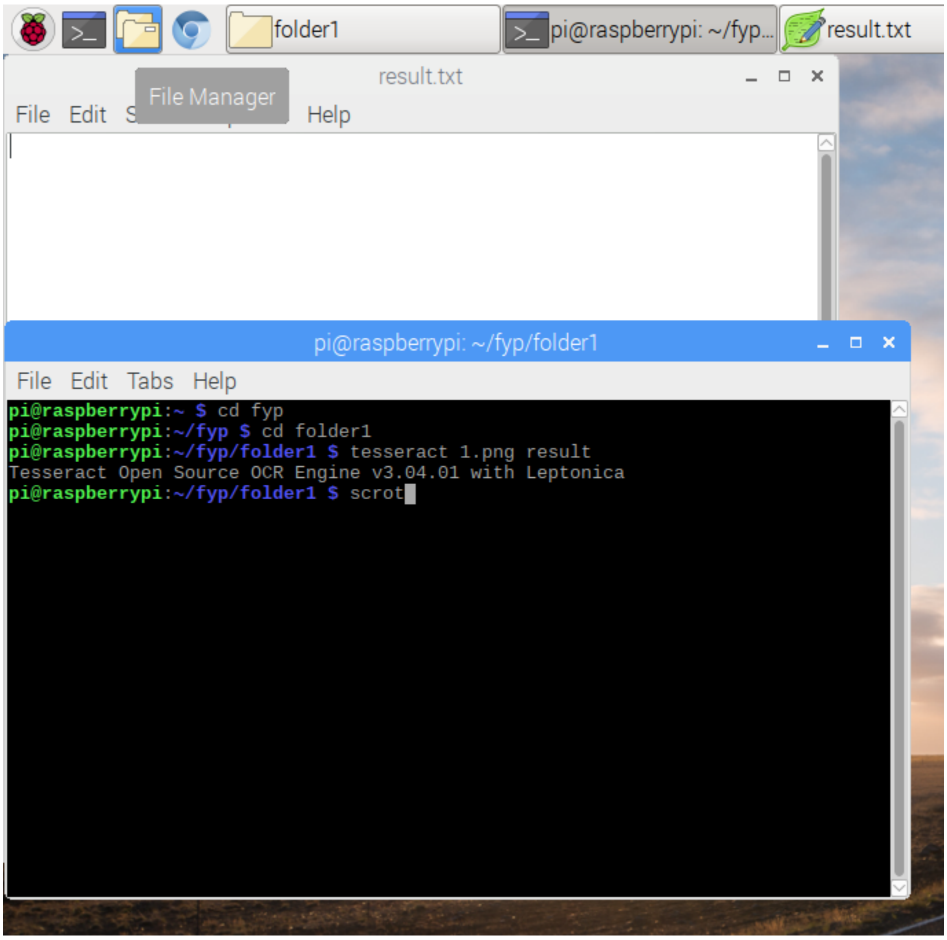


Figure 5.2.8: The empty output from Tesseract.

From the figures above prove Tesseract could not recognize the seven segment characters.

All the result is only output to a text file, but do not upload to cloud for data storage.

After Implementing the Proposed Solution

In this project, the problems mentioned had been solved and the objectives are achieved. There is no human intervention along the capturing, recognizing, and uploading process. Besides, the system able to recognize characters more accurately from tilted image and upside down image. Besides, the system could know if the image capture from the current angle can be used to generate the data or not. By using the ratio of area of two QR Code, if the ratio is below the threshold set, then the image will be delete and proceed to next image. In addition, the data captured will be upload to the cloud for ease of monitoring along with the image captured time.

The system able to take the image automatically. The figures below show the images taken by the system and process the images.

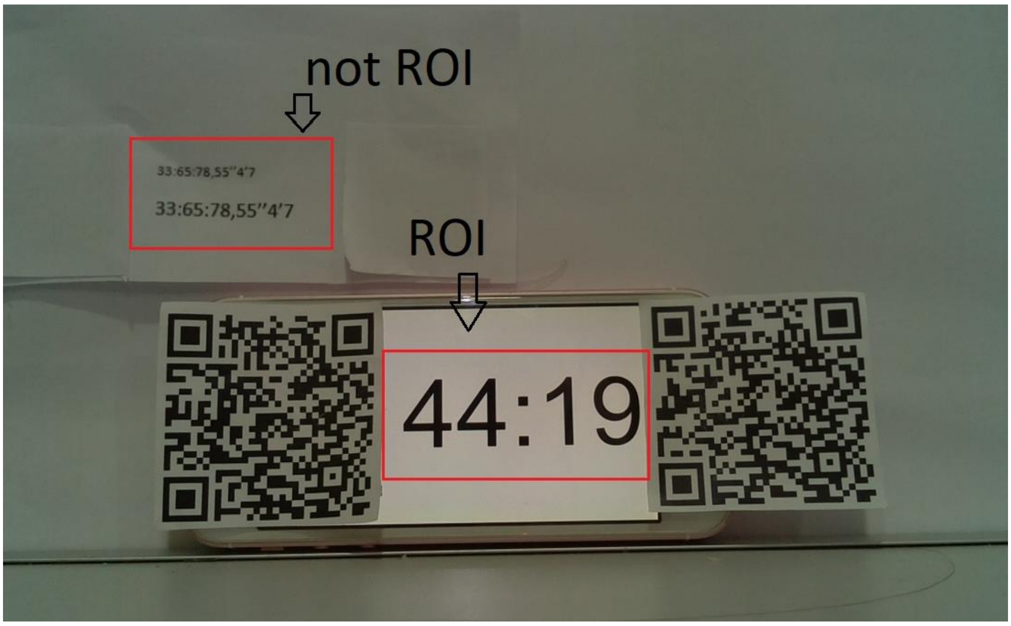


Figure 5.2.9: The first image took by the system.

Figure 5.2.10: The second image took by the system

:

Figure 5.2.11: The third image took by the system.

timestamp	CaptureTime	data
4/6/2019 17:31:08	20190406-173055	44:19
4/6/2019 17:31:18	20190406-173057	44:17
4/6/2019 17:31:29	20190406-173059	44:15
4/6/2019 17:31:40	20190406-173101	44:13
4/6/2019 17:31:50	20190406-173103	44:11

Figure 5.2.12: The result from the image after processed by the system.

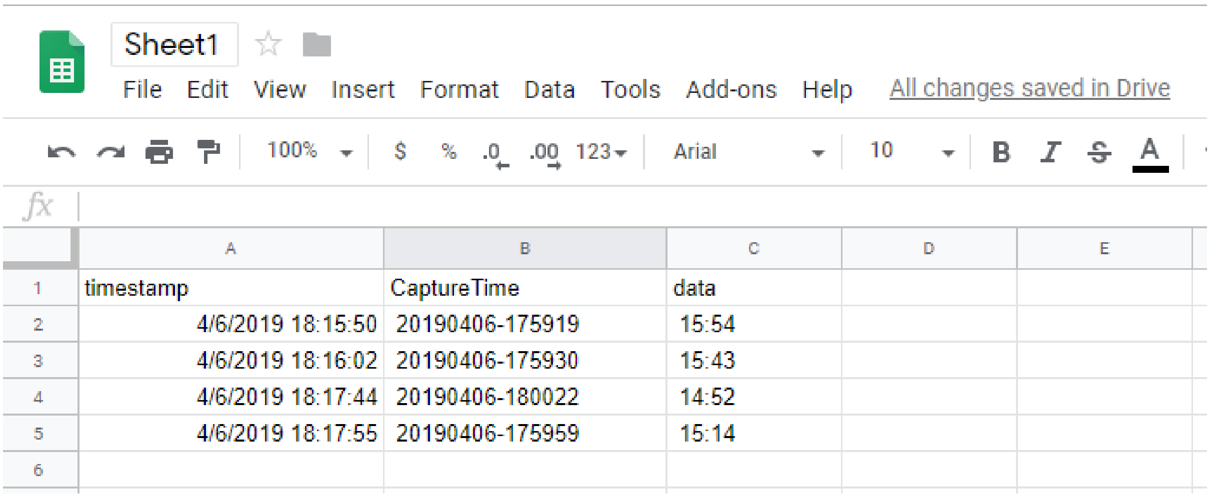
The figures below show the tilted and upside down images to prove that this developed system could process and recognize the characters and able to upload to the cloud as well.

Figure 5.2.13: The tilted image.

Figure 5.2.14: The tilted image.

Figure 5.2.15: The 90 degree rotated screen image.

Figure 5.2.16: The upside down screen image.



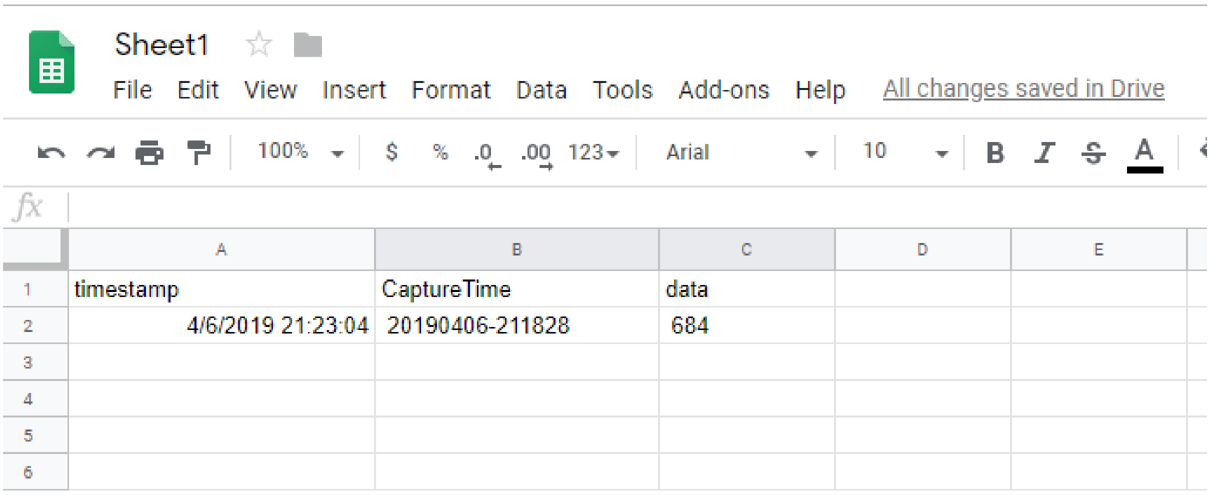
	A	B	C	D	E
1	timestamp	CaptureTime	data		
2	4/6/2019 18:15:50	20190406-175919	15:54		
3	4/6/2019 18:16:02	20190406-175930	15:43		
4	4/6/2019 18:17:44	20190406-180022	14:52		
5	4/6/2019 18:17:55	20190406-175959	15:14		
6					

Figure 5.2.17: The result data uploaded to the cloud

As the Figure 5.2.13 to Figures 5.2.16 show, the tilted and upside down image able to be detected accurately. The 'timestamp' indicating the time of data push into cloud, the 'CaptureTime' indicates the image captured date and time, for example: 20190406-175919 mean the image is captured at 6th of April 2019, 17:59:19. The 'data' indicating the result from recognition process.

The system able to detect seven segment characters, which the Tesseract tool could not.

Figure 5.2.18: The image captured with seven segment characters



	A	B	C	D	E
1	timestamp	CaptureTime	data		
2	4/6/2019 21:23:04	20190406-211828	684		
3					
4					
5					
6					

Figure 5.2.19: The result of the image in Figure 5.2.18

The developed OCR solution uses Tesseract as the OCR engines as it is very powerful and does not need internet access like Google OCR. Besides, the solution able to capture, crop, rotating and other calibration without human intervention. Lastly, the solution able to transmit the data to cloud automatically, along with the metadata of the image. Therefore, the objectives of the project are achieved.

5.3 Testing on the Tesseract engine and Proposed System

5.3.1 Testing on the Accuracy Against Angle From The Screen On The Tesseract Engine and The Proposed OCR Solution

Testing on the accuracy against angle from the screen on the Tesseract engine

The test is conducted under 74.7 Lux. The LCD/LED screen is put in front of the camera, with the distance of 30cm from the screen. This is a test on the Tesseract engine alone, to study the accuracy of engine when capturing from the screen. 20 images will captured and be processed for recognition. The test angle range is from 10-170 degree.

The table below shows the result from the test.

Angle	Accuracy (Max = 1.00)
10	0
20	0
30	0
40	0
50	0
60	0
70	0
80	0
90	0
100	0
110	0
120	0
130	0
140	0
150	0
160	0
170	0

Table 5.3.1.1: The result of the test on Tesseract engine

The figure below shows the graph of the accuracy against angle from the test conducted.

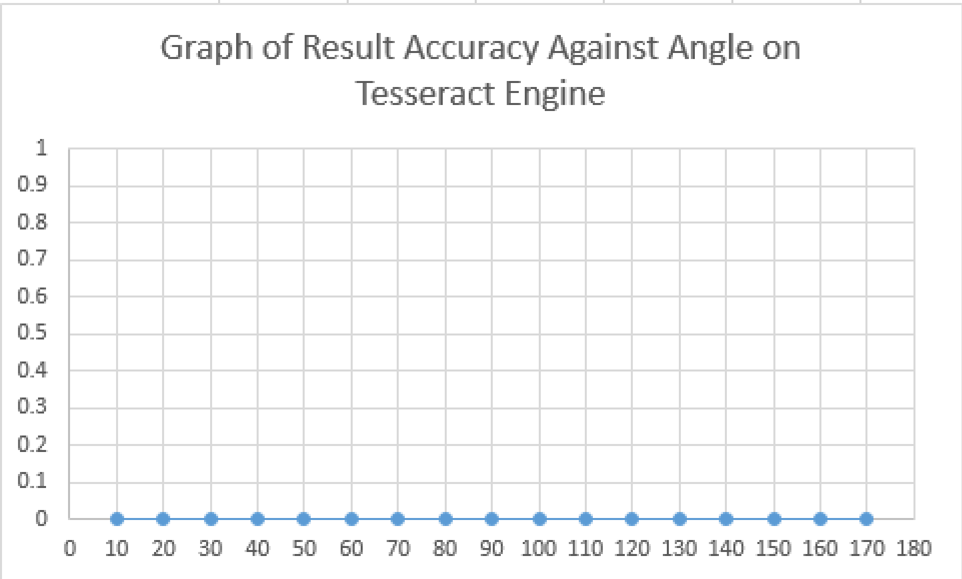


Figure 5.3.1.1: The graph of accuracy against angle of the test

The test above shows Tesseract could not give accurate result when the image is captured from the LCD/LED screen. No matter capture the image from any angle, Tesseract could not recognize the characters in from the LCD/LED screen.

Testing On the Accuracy against Angle From The Screen On The Proposed OCR Solution

This test is conducted under 74.7 Lux, camera is set 30cm from the screen. Each angle will capture 20 images and then to process for recognition task. The results are recorded and the graph is plotted. The angle to be tested is from 10 degree to 170 degree. The QR Codes are attach at the end of the ROI for the system to know the position of the ROI.

The table below showing the result of the angle test.

angle (degree)	accuracy (max = 1)	ratio of QR Codes
10	0	cannot detect
20	0	cannot detect
30	0	cannot detect
40	0	cannot detect
50	0.9	0.59
60	1	0.65
70	1	0.68
80	1	0.82
90	1	0.99
100	1	0.95
110	1	0.72
120	0.9	0.69
130	0	cannot detect
140	0	cannot detect
150	0	cannot detect
160	0	cannot detect
170	0	cannot detect

Table 5.3.1.2: The table of the result of the test.

The graph below shows the accuracy against angle, the test is carried out under 74.7 Lux, and the position of the screen are constant.

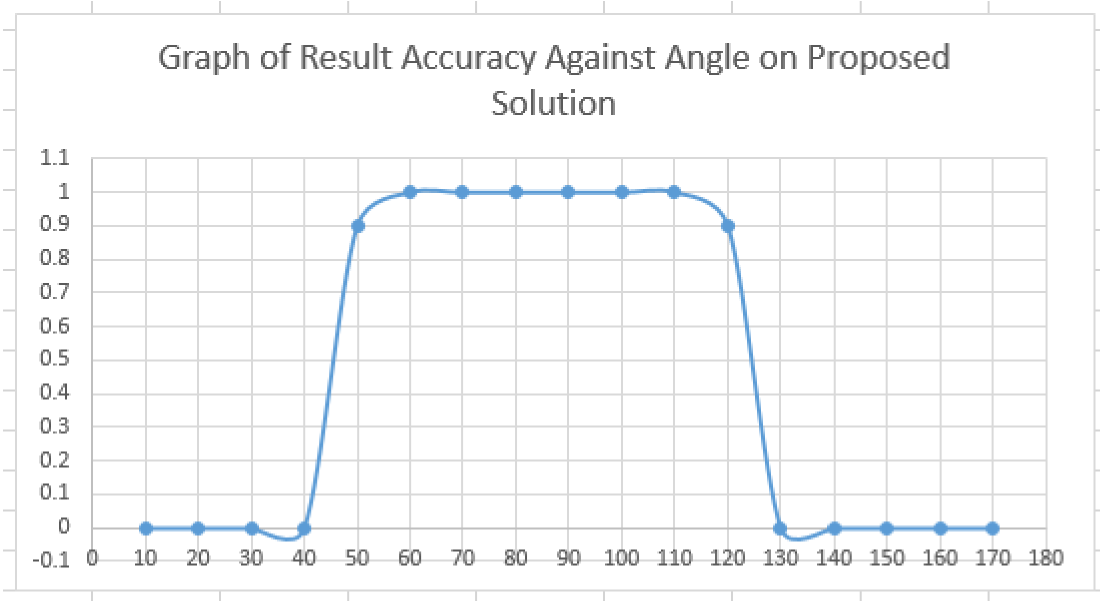


Figure 5.3.1.2: Graph of Accuracy Against Angle

The test shows the system able to give accurate result when the camera is captured from 60-110 degree from the screen. Therefore, for the system to eliminate the image, image QR Code ratio should be less than 0.70, to ensure the accuracy of the result data.

From the results above, the proposed solution could perform better than the Tesseract engine. The proposed OCR solution able to recognize the characters from the screen and able to eliminate the image captured from angle below 60 degree and above 110 degree to ensure the accuracy of the data. Therefore, the conclusion can be made from the second test, if the ratio of both QR Code is below 0.70, the image can be eliminate and delete.

5.3.2 Testing Using the 2 by 2 Table on Tesseract Engine and Proposed OCR Solution

Testing on the Tesseract Engine

The test is conducted under 74.7Lux, by setting the Tesseract engine 30cm from and 90 degree to the screen to indicate the true positive, true negative, false positive and false negative. For true positive and false negative test, the LCD/LED screen will be placed as the input of the test. 20 image will be captured as the input for each test. The input will be the data from the LCD/LED screen. The output is the correct result of input.

The table below shows the result from the test.

	Input	No Input
Output	0	0
No Output	20	20

Table 5.3.2.1: The result from the test.

Rate of True Positive = $\frac{TP}{TP+FN} = \frac{0}{0+20} = 0$

Rate of True Negative = $\frac{TN}{TN+FP} = \frac{20}{20+0} = 1$

Rate of False Positive = $\frac{FP}{TN+FP} = \frac{0}{20+0} = 0$

Rate of False Negative = $\frac{FN}{TP+FN} = \frac{20}{20+0} = 1$

Testing on the Proposed OCR Solution

The test on true positive, true negative, false positive and false negative is carried out under 74.7 Lux, setting the proposed OCR solution 30cm distance from and 90 degree to the screen with the noise in background. The input of the test is the data from the LCD/LED screen.

The table below shows the result from the test.

	Input	No Input
Output	20	0
No Output	0	20

Table 5.3.2.2 : The result of the test.

Rate of True Positive = $\frac{TP}{TP+FN} = \frac{20}{0+20} = 1$

Rate of True Negative = $\frac{TN}{TN+FP} = \frac{20}{20+0} = 1$

Rate of False Positive = $\frac{FP}{TN+FP} = \frac{0}{20+0} = 0$

Rate of False Negative = $\frac{FN}{TP+FN} = \frac{0}{20+0} = 0$

Chapter 6 Conclusion

6.1 Project Review, Discussion and Conclusion

There is no OCR solution to capture the image from the screen and extracts the data from image. Most of the existing solutions are mainly on extracting the characters from the physical documents instead of a LCD/LED screen. Besides, the reviewed solutions are yet completed for IoT solution, which could not perform the data analytic. Also, there is no OCR solution could run without human intervention to capture the data automatically and ensure the accuracy of the result. Furthermore, there are so many OCR engine in the world but every OCR engines have their pros and cons, so the most suitable OCR engines has to find out to place in the proposed system.

The objectives of the project are to identify the OCR engine that suit the project the best, to develop an automated OCR solution which can used to capture data from the LCD/LED display without much calibration such as capturing, imaging cropping, image rotating for tilted image. Lastly, to develop a solution where the captured data are automatically transmitted to the cloud.

The OCR solution in this project able to capture the image from the screen continuously and automatically. Next, each image captured will be process to detect the ROI and to check if the image is tilted, captured from a good-positioned camera, and do calibration for an accurate output automatically without human calibration. The system will identify the type of data through decoding the QR Code, then deciding the engine to recognize the processed image, either Tesseract or SSOCR. Hence the system able to detect seven segment characters as well. After getting the data, the data will be transmitted to the cloud for data storage. Hence the solution hit the last objectives of the project as well. In short, the objectives of the project are achieved.

The problem encountered in this project is to think of a solution to detect the ROI and calibrate the image automatically. QR Code helps a lot in the solution which able to know the ROI and the tilted image. It can also help to eliminate the image took from some angle.

In conclusion, this project is focusing on to develop an OCR solution which run automatically and without human intervention to ease the monitoring task. The automated calibration and capturing is the essence of the project. Although the project implementing the existing OCR engine but it enhances the performance to ensure its accuracy for the data.

6.2 Future Work

There are weaknesses in this project. The camera capture the image continuously, which mean it will capture the consecutive same data and upload it again and again, and causing the storage waste. The performance of the system/solution is not good enough. Lastly, there is no data visualization which give operator a clear sight of the status of the machines.

The future work will be able to capture the image if there is a change from previous data. This could help in saving the memory space and also the space in cloud. This is because uploading the same data for couple of times is pointless. Therefore, the system could be into a next level if the camera able to capture image if there is any change. Next, the performance of the system should be increase. It takes long time to process and recognize the data, consider not efficient enough. Lastly, able the data visualization and data analytic could help in the monitoring purpose, in order to push the productivity of the factories.

Bibliography

Chan, E., 2015. *Hackernoon*. Available from: https://hackernoon.com/optical-character-recognition-with-google-cloud-vision-api-255bb8241235 . [1 August 2018].

Chan, E., 2016. *Optical Character Recognition With Google*. Available from: https://hackernoon.com/optical-character-recognition-with-google-cloud-vision-api-255bb8241235 . [1 August 2018].

Cutter, M. & Manduchi, R., 2015. *Towards Mobile OCR:How To Take a Good Picture of a Document Without Sight*. pp. 75-84.

Enhance Vision, 2018. *DaVinci Pro: User Manual*. Available from: https://www.enhancedvision.com/downloads/user-manual/DaVinciPro/davinci-pro-user-manual-english.pdf . [3 August 2018].

Enhance Vision, n.d.. *DaVinci HD/OCR*. Available from: https://www.enhancedvision.com/shop/davinci-hd . [3 August 2018].

Gleichman, S. et al., 2011. *Detection and Segmentation of Antialiased Text in Screen Images*. pp. 424-428.

How Easy Screen OCR Works, n.d. Available from: https://easyscreenocr.com/wp-content/uploads/2018/03/how-to-use-easy-screen-ocr.mp4 . [1 August 2018] .

INDATAProject, 2015. *DaVinci desktop magnifier*. Available from: https://www.youtube.com/watch?v=FlZEDgEch_M&t=294s . [1 August 2018].

James, J.I., 2017. *Digital Forensic Science*. Available from: https://dfir.science/2017/04/tesseract-ocr-extract-text-from-images.html . [31 July 2018].

Jbeale, 2013. Available from: https://lh4.googleusercontent.com/-KzWoCRUHAO8/UlSRgP47JLI/AAAAAAAADKI/dc9C5ZflPgM/s100/dmm-100px.jpg . [1 August 2018].

Language Technologies Unit (Canolfan Bedwyr), Bangor University, 2008. *An Overview of the Tesseract OCR (Optical Character Recognition) engine, and its possible enhancement for use in Wales in pre-competitive research stage*. Available from: https://tesseract-ocr.repairfaq.org/downloads/saltcymru_document5.pdf .[31 July 2018].

Leung, H., 2017. *Tesseract vs Google Vision, Round 1, Fight! The Text extraction Wars!*. Available from: https://www.linkedin.com/pulse/tesseract-vs-google-vision-round-1-fight-text-extraction-hanley-leung .[4 August 2018].

Li, Q., An, W., Zhou, A. & Ma, L., 2016. *Recognition of Offline Handwritten Chinese Characters Using the Tesseract Open*. pp. 452-456.

OCR for Reading Multimeter?, n.d. Available from: https://www.raspberrypi.org/forums/viewtopic.php?t=28021. [1 August 2018].

OCR optical character recognition of seven segment displays, n.d., Available from: https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=57727&p=434958#p434782. [1 August 2018].

Rouse, M., n.d.. OCR (Optical Character Recognition). Available from: https://searchcontentmanagement.techtarget.com/definition/OCR-optical-character-recognition .[31 July 2018].

Screen OCR, n.d. *Easy Screen OCR.* Available from: https://easyscreenocr.com/ .[1 August 2018].

Smith, R., Antonova, D. & Lee, D.-S., 2009. "Adapting the Tesseract Open Source OCR Engine for Multilingual OCR", *Proceedings of the International Workshop on Multilingual OCR MOCR '09*, pp. 1:1-1:8.

Steven, S., n.d. *Screen OCR Super Easy To Use*. Available from: https://easyscreenocr.com/how-to-convert-screenshot-to-text-with-a-free-ocr-software/ . [1 August 2018].

Tekin, E., Coughlan, J. M. & Shen, H., 2011. Real-Time Detection and Reading of LED/LCD Displays for Visually Impaired Persons. Proc IEEE Workshop Appl Comput Vis, pp. 491-496.

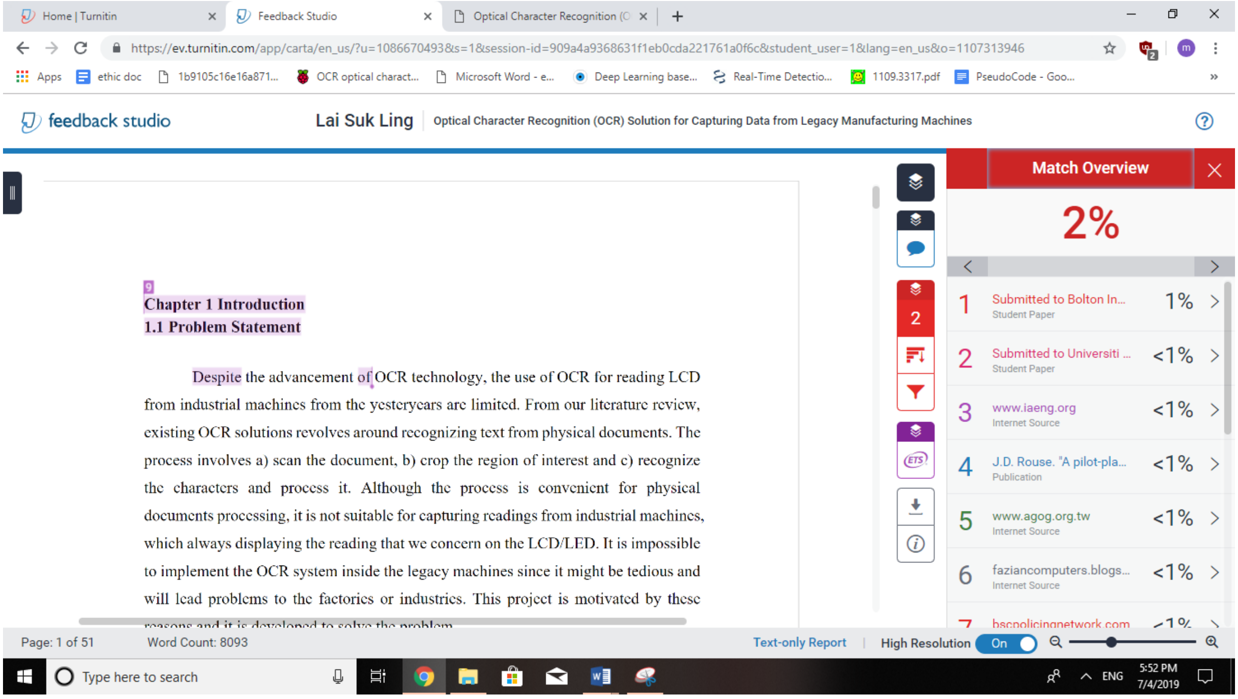
The Low Vision Centers Of Indiana, n.d.. *DaVinci HD/OCR Video Magnifier.* Available from: http://www.eyeassociates.com/davinci-hdocr-video-magnifier/ . [3 August 2018].

Vincent, L., n.d.. *Google Code Blog*. Available from: https://web.archive.org/web/20061026075310/http://google-code-updates.blogspot.com/2006/08/announcing-tesseract-ocr.html .[29 July 2018].

Wachenfeld, S., Klein, H.-U. & Jiang, X., 2006. *Recognition of Screen-Rendered Text*.

Xie, J., 2009. *Optical Character Recognition Based on Least Square Support Vector Machine*. pp. 626-629.

Appendix A Turnitin Result



Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines

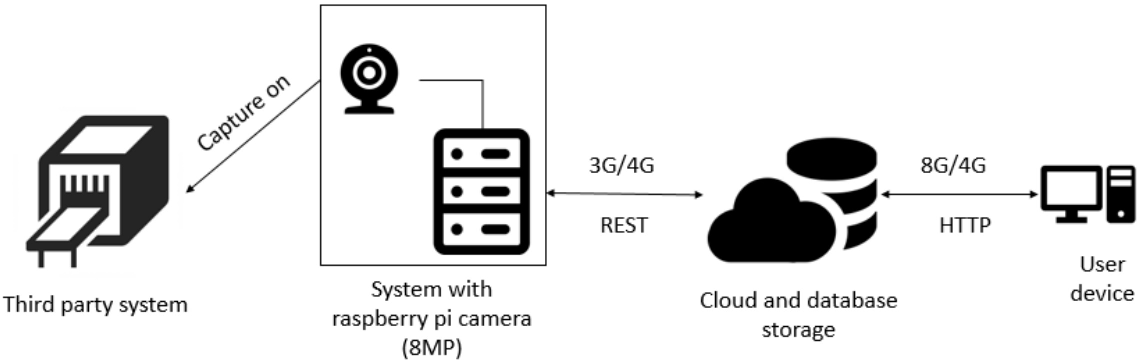
ORIGINALITY REPORT

2%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Bolton Institute of Higher Education Student Paper	1%
2	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
3	www.iaeng.org Internet Source	<1%
4	J.D. Rouse. "A pilot-plant study of a moving-bed biofilm reactor system using PVA gel as a biocarrier for removals of organic carbon and nitrogen", Water Science & Technology, 05/2007 Publication	<1%
5	www.agog.org.tw Internet Source	<1%
6	faziancomputers.blogspot.com Internet Source	<1%
7	bscpolicingnetwork.com Internet Source	

Appendix B System Architecture



Appendix C Supervisor's Comment on Originally Report

UTAR	**Universiti Tunku Abdul Rahman**
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)	

Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lai Suk Ling
ID Number(s)	1502541
Programme / Course	Computer Science
Title of Final Year Project	Optical Character Recognition (OCR) Solution for Capturing Data from Legacy Manufacturing Machines

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:________ % **Similarity by source** Internet Sources: ________________% Publications: _____________ % Student Papers: ________________ %	
Number of individual sources listed of more than 3% similarity: __________	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____

Signature of Supervisor

Name: _Dr. Ooi Boon Yaik___

Date: __08/04/2019__________

____

Signature of Co-Supervisor

Name: __________________________

Date: ___________________________

Appendix D Checklist for FYP1 Thesis Submission

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student Id	1502541
Student Name	Lai Suk Ling
Supervisor Name	Dr. Ooi Boon Yaik

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Title Page
	Signed form of the Declaration of Originality
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ______________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ______________________ (Signature of Supervisor) Date:

Appendix E Poster