UNIVERSITY OF MORATUWA

Faculty of Engineering

Registered Module No: EN3992

INDUSTRIAL TRAINING REPORT

LE Robotics (Pvt.) Ltd.

From: 04/ 01 / 2022 To: 20 / 06 / 2022

Date of Submission:
20 / 06 / 2022

Thalagala B.P.

180631J

Department of: Electronic and Telecommunication Engineering

Preface

This report was composed as partial fulfillment of the requirements of Module EN3992 - Industrial Training, in the curriculum of B.Sc. in Engineering (Electronic and Telecommunication) at the University of Moratuwa, Sri Lanka. The experience and knowledge that I gained during the six months of my industrial training were used and were the inspiration to create this report.

Acknowledgment

I would like to gratefully acknowledge all of the people who helped me to make this six months of special industrial training period a massive success, starting from the point of applying to a company to the point I left the training organization at the completion of six months.

First of all, I would like to express my heartfelt gratitude to Professor Kapila Jayasinghe who was our supervisor for us throughout the six months of the internship period. The advice he provided us with regard to professional engineering practices and ethics was invaluable. In addition to that the directions he provided us to gain the required technical skills required for the allocated project were priceless. I believe the mindset that he built in us, towards working under minimum supervision, will be a massive support for us to thrive in the fast-moving industry.

Next, I would like to express my gratitude toward Miss. Laknie Jayasinghe who was the Engineer in charge of us in our training period. The support she provided us to improve our soft skills as well as technical skills as a professional engineers, is highly appreciated. The support she provided when composing the technical documentation of my allocated task by pointing out the mistakes and the areas of improvement was invaluable and must be mentioned at this point. Moreover, her support in validating the project deliveries at the end of the training period is highly appreciated.

Last but not least, I would like to express my heartiest gratitude to Mr Janka Kulathunga, the manager of the Lanka Electronics research and development section and Mr Aloka Perera, an electronic engineer in LE Robotics (Pvt.) Ltd for being available for us whenever we needed their support.

Contents

PDF is clickable

List of Abbreviations

CAN Controller Area Network

CNC Computer Numerical Control

CPLD Complex Programmable Logic Device

CV Computer Vision

EDM Electrical Discharge Machining

FOV Field of View

FPGA Field Programmable Gate Arrays

GUI Graphical User Interface

IDE Integrated Development Environment

IP Interllectual Property

IPC Inter-process communication

LAN Local Area Network

ML Machine Learning

NDA Non-Disclosure Agreement

OS Operating System

OSS Open-Source Software

PC Personal Computer

PCB Printed Circuits Boards

R&D Research and Development

ROI Region of Interest

RPi Raspberry Pi

SCARA Selective Compliance Articulated Robot Arm

SDLC Software Development Life Cycle

SE Software Engineering

SFTP SSH(Secure shell) File Transfer Protocol

SIFT Scale Invariant Feature Transform

SVM Support Vector Machine

UART Universal Asynchronous Receiver/Transmitter

List of Figures

List of Tables

Chapter 1

Description of the organization and business, its past, present and future

For my special industrial training, I got the opportunity to work as an engineering trainee at LE Robotics (Pvt.) Ltd. located at 100/4, Divulapitiya Road, Minuwangoda, Sri Lanka.It is a local Research and Development (R&D) facility where they work on industrial articulated robot arms and related technologies. This chapter provides an extensive description of the organization and business as well as information about its past, present and future.

1.1 Organization and Business

1.1.1 Introduction

LE robotics (Pvt.) Ltd. is a local R&D facility that has been established by Prof. J.A.K.S. Jayasinghe who is a senior professor in the Department of Electronic and Telecommunication Engineering at the University of Moratuwa in Sri Lanka.

LE robotics (Pvt.) Ltd. is the first in the Sri Lankan market to offer fully customisable robotics solutions 'Made in Sri Lanka' for various automation needs for an affordable price with expertise based in Sri Lanka. In the year 2005, the company designed and manufactured the first custom robotics solution in their affiliated company, Lanka Electronics (Pvt.) Ltd. Since then, they have been developing various robotics solutions and related technologies in the facility.

Figure 1.1: Logo of the LE Robotics (Pvt.) Ltd.

1.1.2 Services and Products

As mentioned previously LE Robotics Pvt. Ltd. provides services and tailor-made products for various automation needs of its customers. When it comes to the services they provide, the following key services can be highlighted.

1. Process Analysis to identify the automation needs and challenges of organizations
2. Product customization to provide the best tailor-made solution for requirements
3. Providing local expertise with ease of access
4. Providing lifetime support for the products

In addition to the services they provide, the following products are manufactured at LE Robotics Pvt. Ltd. R&D of the related technologies of those products is a key activity among the day-to-day activities in the facility.

1. **6 DOF Robots** - Robots with six degrees of freedom (Figure 1.2 depicts such a model)
   - Robotics solutions with the capability to mimic human arm operations
   - Offers the flexibility to provide you with fully customized robotic movements to suit your requirement
   - Around 1 m reach and 0.1 mm placement precision

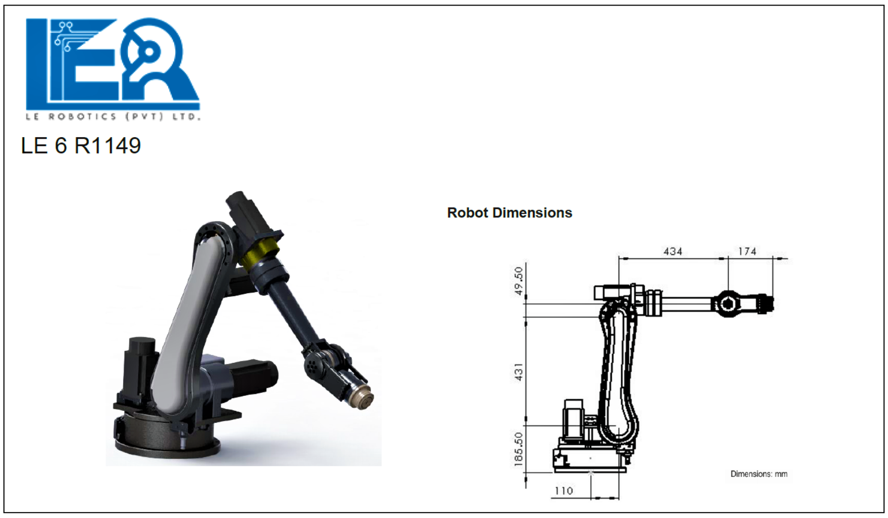


Figure 1.2: Model *LE 6 R1149* 6-DOF articulated robot arm[3]

2. **4 DOF Robots** - Robots with four degrees of freedom (Figure 1.3 depicts such a model)
   - Robotics solutions with superior high-speed performance, high rigidity and high accuracy
   - Array of options with a compact design
   - Around 500 mm reach and 0.1 mm placement precision

3. **Custom Made Robots** - Robots designed according to the customer's needs
   - Reach and Payload specifications are possible to be customized as per your process requirements
   - Robotics Solutions to carry out simple pick and place operations

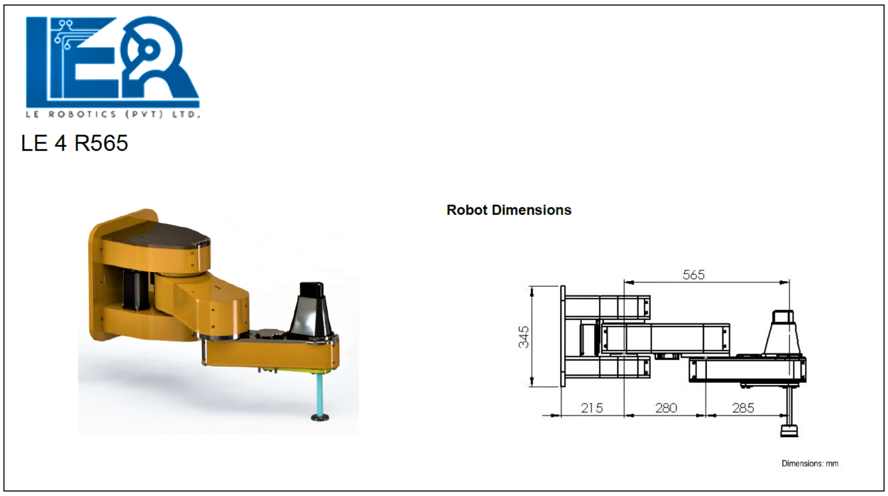


Figure 1.3: Model *LE 4 R565* 4-DOF SCARA robot arm[4]

1.2 Past, Present and Future

LE Robotics Pvt. Ltd. has been originally established as the R&D division of its parent organization, Lanka Electronics (Pvt) Ltd. Lanka Electronics (Pvt) Ltd has been founded in 1993 by a group of graduates of the University of Moratuwa, the leading technical college in Sri Lanka. In 1993 they established their first electronic product manufacturing facility in the city of Minuwangoda and currently they own another manufacturing facility in Anuradhapura city.

Being a leading innovation company in Sri Lanka, Lanka Electronics (Pvt) Ltd has manufactured professional electronic devices for the Sri Lankan government and multinational companies such as *Variosystems (Pvt) Ltd*. In chronological order, some of the professional electronic products they have manufactured so far are shown in Table 1.1. In addition to those, the company manufactures several consumer electronic products as well. TV antennas and Boosters are the mainstream consumer electronic products.

TABLE 1.1
PROFESSIONAL ELECTRONIC PRODUCTS OF LANKA ELECTRONICS (PVT) LTD.

Year	**Product**
1993	Laboratory training panels: discontinued manufacturing in 2010
1996	Die-Sink Electrical Discharge Machining (EDM) Machine: the first EDM machine built in Sri Lanka using 100% locally developed technologies and local resources
1997	Motor Traffic Control Lights: the first traffic light system, that was built in Sri Lanka using local technologies and resources as per a request from the Sri Lankan government.
2009	Wire Winding Machines: manufactured for special request made by Variosystems (Pvt) Ltd.

In present, none of the products mentioned in the Table 1.1 are in production and the mainstream professional electronic product of the company is customisable industrial articulated robot arms.

In addition to that related technologies such as servo motors, servo motor controllers and motor encoders are researched and developed within the facility.

Chapter 2

Description of familiarization work carried out

2.1 Projects Assignment

Although the official internship period commenced on Tuesday 4^{th} of January in 2022, we had had our first meeting with the supervisor about a month prior to that. The meeting was conducted online through a video conferencing platform due to the Covid 19 situation in the country. Figure 2.1 shows a moment captured during the meeting.

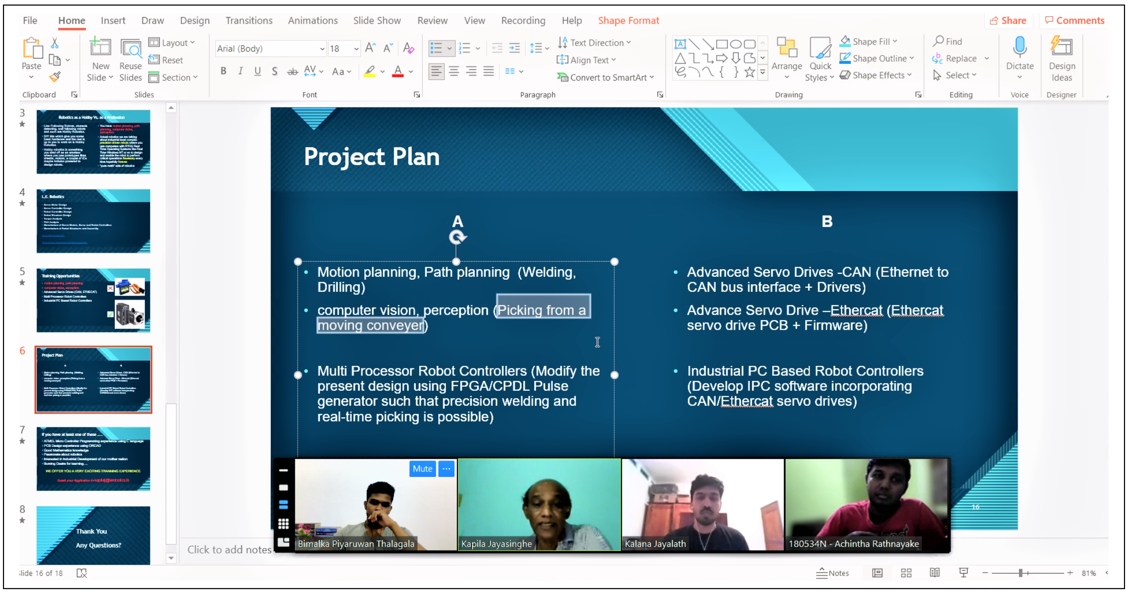


Figure 2.1: First meeting with the supervisor to choose a project plan

In that meeting, our supervisor introduced the available projects that we can work on in our training period. He offered us two separate paths we can select to work during our internship period as described in Table 2.1. Depending on our interests, we chose ‘Plan A’ to continue with and I selected the project related to CV. In summary, that project was about engineering a CV subsystem to integrate with an existing pick and place machine to pick objects from a moving conveyor belt.

TABLE 2.1
PLANS WE WERE OFFERED PRIOR TO BEGINNING OUR INDUSTRIAL TRAINING

Plan A	**Plan B**
Motion planning, Path planning (welding, drilling)	Advanced servo drivers - Controller Area Network (CAN) (Ethernet to CAN bus interface + Drivers)
CV, Perception (picking objects from a moving conveyor)	Advanced servo drive - Ethercat (Ethercat servo drive Printed Circuits Boards (PCB) + Firmware)
Multiprocessor robot controllers (modify the present design using Field Programmable Gate Arrays (FPGA)/ Complex Programmable Logic Device (CPLD) pulse generator such that precision welding and real-time picking is possible)	Industrial Personal Computer (PC) based robot controllers (Developed Inter-process communication (IPC) software incorporating CAN/ Ethercat servo drivers)

2.2 Facility Familiarization

LE Robotics Pvt. Ltd. is a fully featured R&D facility located in Minuwangoda, Sri Lanka. Facility consists of most of the machinery required for industrial robot arm designing, including a Computer Numerical Control (CNC) machine, a Lathe machine, a Milling machine and various other machines. In addition to that facility has the resources for single side PCB designing and testing. On the first day that we visited the facility, we were accompanied around it to see the various machines and electronic test instruments. Our supervisor gave brief descriptions about most of such tools and encouraged us to learn how to operate them in our free time.

2.3 Introduction to Industrial Robot Arm Designing

On the same day, Supervisor gave us an explanation about the evolution of already developed industrial robot arms that were described in the Section 1.1.2. The explanation included capabilities and drawbacks of each design as well as the advantages and disadvantages of incorporating third party technologies to manufacture them locally.

Then he explained the importance of manufacturing all the parts of an industrial robot arm within the facility rather than simply using the available parts. Because then the manufacturer has the full potential to customize any design to best fit to a given application. If the third party parts are used this customization capability is reduced drastically and we have to adjust our designs just to suit them rather than the application. Due to this reason, LE Robotics Pvt. Ltd. has taken a great initiative to manufacture different types of motors locally using the available materials and knowledge.

2.4 Non-Disclosure Agreement

Another important concept that the supervisor pointed out on the familiarization session was Non-Disclosure Agreement (NDA). It is a legal contract or part of a contract between at least two parties, that outlines confidential material, knowledge, or information that the parties wish to share with one another for certain purposes but wish to restrict access to[15]. An NDA document contains the details given below.

1. Definitions of the terms used in the document such as Confidential Information, Disclosing Party and Receiving Party.
2. Rules and regulations related to Use, Disclose and Reproduce the shared confidential information.
3. Actions to be taken in an event of breach of a term of the agreement mentioned in the previous point.
4. Non-Competition Clause which emphasizes the requirement of a written permission prior to direct or indirect attempt to register or use the Disclosing Party's confidential information, during the term of the agreement and the expiry of the agreement.
5. Governing Law which describes the way of interpreting the agreement.
6. Confidentiality Period/ Termination which specifies the duration of time that the agreement remains in force, if its is not terminated earlier in writing by mutual agreement.

Since we were temporary employees of the company we did not have access to any of the trade secrets of LE Robotics Pvt. Ltd., such as hardware blueprints, software source codes and etc. Therefore, the supervisor mentioned that there was no requirement to sign an NDA. However, he explained the importance of obeying an NDA as a professional engineer.

Chapter 3

Exposure to systems (HSE, Financial, Administration, Logistics, etc.)

3.1 Health, Safety and environment (HSE) Department

Health, safety and environment (HSE) refers to a branch, or department, within a company that is responsible for the observance and protection of occupational health and safety rules and regulations along with environmental protection.

3.2 Financial Department

3.3 Administration Department

3.4 Logistics Department

Chapter 4

Project Work

By title, the project that I was assigned, was ***Machine vision based Real-time Motion Planning for an Industrial Articulated Robot Arm***. In simple words, it was a project related to an automatic pick and place machine which can be used to pick objects placed on a conveyor belt and pass them to the next stage of processing.

My contribution to that project was to develop the following three aspects of the system.

I. Development of an object detection framework

II. Development of an application for camera calibration (referred as *Camera Calibrator GUI* in this document)

III. Development of an application to train an object classification model (referred as *Classifier Trainer GUI* in this document)

Subsequent sections explain the mentioned sub-projects that were undertaken by me during the industrial training period. Implementation details are not exposed in this report as it does not comply with professional engineering ethics.

4.1 Development of an Object Detection Framework

The first project, that I was assigned as a trainee electronic engineer was related to the CV field. In this project, an Object Detection Framework was developed to be deployed in an Automatic Pick and Place Machine. In a summary, the framework is capable of identifying Region of Interest (ROI), detecting and classifying objects, and determining the location and orientation of objects with respect to a real-world coordinate system for grasping (picking). This section explains everything related to the development of that framework.

4.1.1 Problem Definition

Most of the robotic arms used in industrial environments operate in a pre-programmed cycle. When comes to the way a human does the same task is much different as the path planning for picking an object may change from cycle to cycle because of the perception obtained through human vision. CV is the technology and methods incorporated to mimic the human vision in order to gain insights into the operating environment of the robotics system.

When it comes to real-time object detection using CV, there is an inevitable trade-off between the accuracy and the speed of the operation. This depends entirely on the used CV algorithms and the computational power of the available hardware. If the robotic system/ arm in interest is not controlled through a dedicated industrial PC with adequate computational resources, the amount of resources that can be allocated to the CV unit becomes limited. This will eventually result in great delays (which is not desirable when it comes to real-time operations) to produce the required outputs by processing the acquired images through the associated camera.

Due to the limitations in the resources, the initial problem of real-time object detection on a moving conveyor was subdivided into three problems.

- **Case 1:** Detection of objects placed in a ***static environment*** (no conveyor belt is associated). Picking and placing is carried out by the robot in the usual way.
- **Case 2:** Detection of objects placed on a ***non-continuous conveyor belt***. The conveyor belt can be paused when objects are detected and picking and placing is carried out thereafter.
- **Case 3:** Detection of objects placed on a ***continuously moving conveyor belt***. Picking and placing happen while the conveyor is moving.

4.1.2 Solution

As the initial stage of the project, case 1 mentioned in Section 4.1.1 was developed. It consisted of developing a CV subsystem for detecting ***stationary objects*** placed in a ***static environment***. In this case, no conveyor belt is used and objects are placed in the Field of View (FOV) of the camera.

Figure 4.1 shows the interconnection between inputs and outputs of the developed system. Once the video stream from a camera is fed into the system, it is capable of,

1. Determining the grasping location of an object in a real-world coordinate system. The centroid of the 2D view of the object was considered as the grasping location.
2. Determining the gripper's angle. Orientation of object with respect to the positive x direction of the same real-world coordinate system was used for this.
3. Determining the class/ type of the detected objects using a Scale Invariant Feature Transform (SIFT) and Support Vector Machine (SVM) based classification algorithm.

and providing them to the main controller of the pick and place machine as requested by it.

Figure 4.2 depicts the interconnections between the existing robot controller and the newly developed computer vision subsystem. Outputs shown in Figure 4.1 are provided to the robot controller through a serial communication link built using an Universal Asynchronous Receiver/Transmitter (UART) circuitry. For transferring data to the Raspberry Pi from the working PC, SSH(Secure shell) File Transfer Protocol (SFTP) over Wi-Fi technology was used. For the implementation of the prototype, third-party tools given below are used.

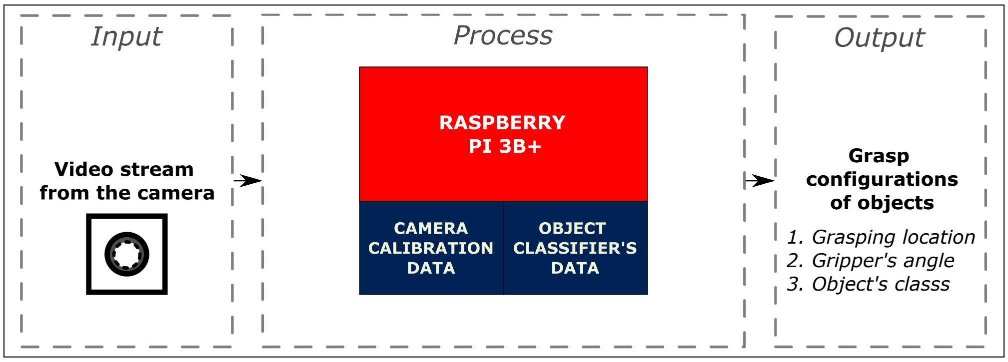


Figure 4.1: Interconnection between inputs and outputs of the CV subsystem

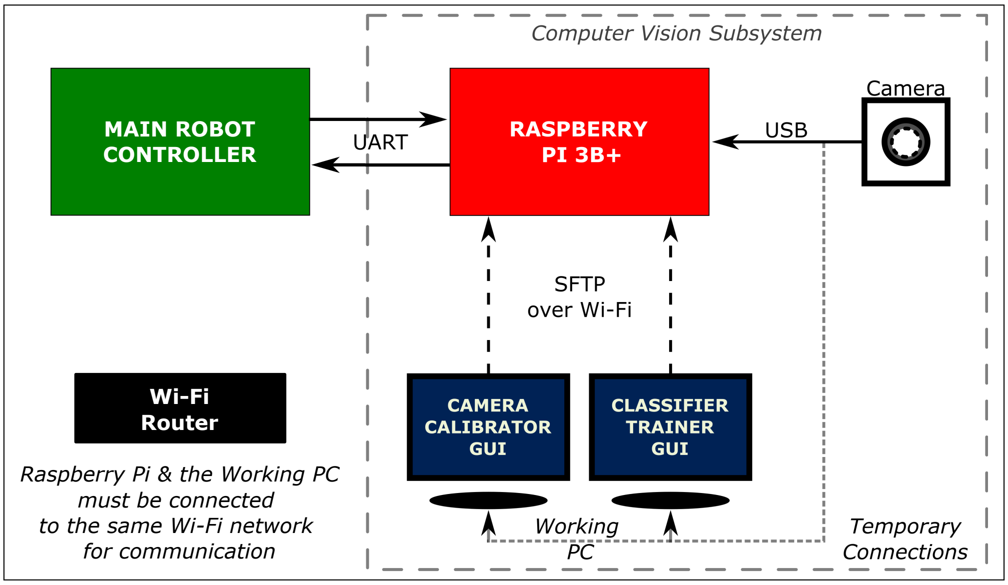


Figure 4.2: Interconnections between the existing robot controller and the CV subsystem

1. **Raspberry Pi (RPi) 3 B+ Single Board Computer (SBC)**
   Product brief: Online available at `https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf`.

2. **Logitech C310 HD Webcam, 720p Video**
   Technical specifications: Online available at `https://support.logi.com/hc/en-us/articles/360023464573-Logitech-HD-Webcam-C310-Technical-Specifications`.

3. **OpenCV open source computer vision library stable release 4.4.0**[8]
   Stable and development releases: Online available at `https://github.com/opencv/opencv`; A pre-compiled version of this library, which is optimized for CV and Deep Learning on Raspberry Pi was used.

4. **Gordon's Arduino wiring-like WiringPi Library for the Raspberry Pi**[12]
   Unofficial Mirror for WiringPi bindings: Online available at `https://github.com/WiringPi/WiringPi`.

During the operation of the pick and place robot, the developed object detection algorithm runs only inside the RPi. However, for the operation of that algorithm, it requires the data shown in Table 4.1. A Windows GUI was developed to generate those data and the 'Description' column of Table 4.1 provides only a simple overview of that data . Subsequent sections explain the functionality and the working principles of the mentioned GUI, in detail.

TABLE 4.1
DATA FILES REQUIRED TO RUN THE OBJECT DETECTION FRAMEWORK

File Name	Description
`calibration_data.xml`	This file includes camera calibration data. That is, intrinsic and extrinsic parameters of the associated camera.
`objects_data.xml`	This file includes object classification model's data. That is, the data required for the object classification (class names, trained support vector machines and etc.).

4.2 Development of an Application for Camera Calibration

The second project, that I was assigned as a trainee electronic engineer was related to the Software Engineering (SE) field. In this project, a Windows GUI was developed to calibrate a given camera. In a summary, the application is capable of generating the required data, to remove the distortions of captured images and to transform 2D image points back to a given 3D real-world coordinate system with an accuracy of $\pm 0.5\ mm$.

4.2.1 Problem Definition

The end goal of the CV subsystem mentioned in the Section 4.1.2 is to provide the location and orientation of the identified objects, with respect to a known real-world coordinate system. However, when an image is captured, the details about the depth of the object relative to the camera are lost due to the real world to image plane transformation (the forward projection [11]) happens inside the camera. This operation is non-invertible.

There are several methods used in CV literature to back-project image plane coordinates to the real-world coordinate system. In this project, a method which uses a single calibrated camera and analytic geometry was implemented. It was observed that the back-projected coordinates are accurate at least to $\pm 0.5\ mm$. As mentioned for this back-projection process, a *calibrated camera* is required and the rest is engineering mathematics. Therefore, the actual problem to be solved was to calibrate the camera associated with the pick and place machine.

4.2.2 Solution

Camera Calibration refers to the extraction of various parameters related to the camera that will be used to capture the video stream/ images. These parameters are required when the back-projection is done. For the camera calibration, the method that uses an asymmetric chess board was used[5]. Because that method was really simple and could be implemented easily using the resources that were available in the facility. The process of camera calibration is given in the Appendix A.

Once the camera calibration is done, we will be able to extract the intrinsic and extrinsic parameters of a given camera. Those parameters can then be used for back-projection which was my problem at hand. Extracted intrinsic parameters which are specific to the used camera, do not depend on the image. Among those parameters, *Camera Matrix* is what has the greatest important. *Radial/ Tangential Distortions* have a minimal importance and therefore can be neglected. Extracted extrinsic parameters of the image shown in Figure A.3 are specific for that image. If in any case, orientation or the place of the camera is altered, these must be recalculated in order to be used in the back-projection algorithm.

As mentioned previously ultimate goal of this sub project was to develop a GUI application for camera calibration. The process mentioned above was therefore implemented in `C#` programming language using '*Visual Studio 2019*' software. For the implementation, the *Emgu CV* library which is a cross platform *.Net* wrapper to the OpenCV image processing library was also used. It allows OpenCV functions to be called from .NET compatible languages like `C#` and therefore make the software development process efficient. The developed GUI is shown in the Figure 4.3.

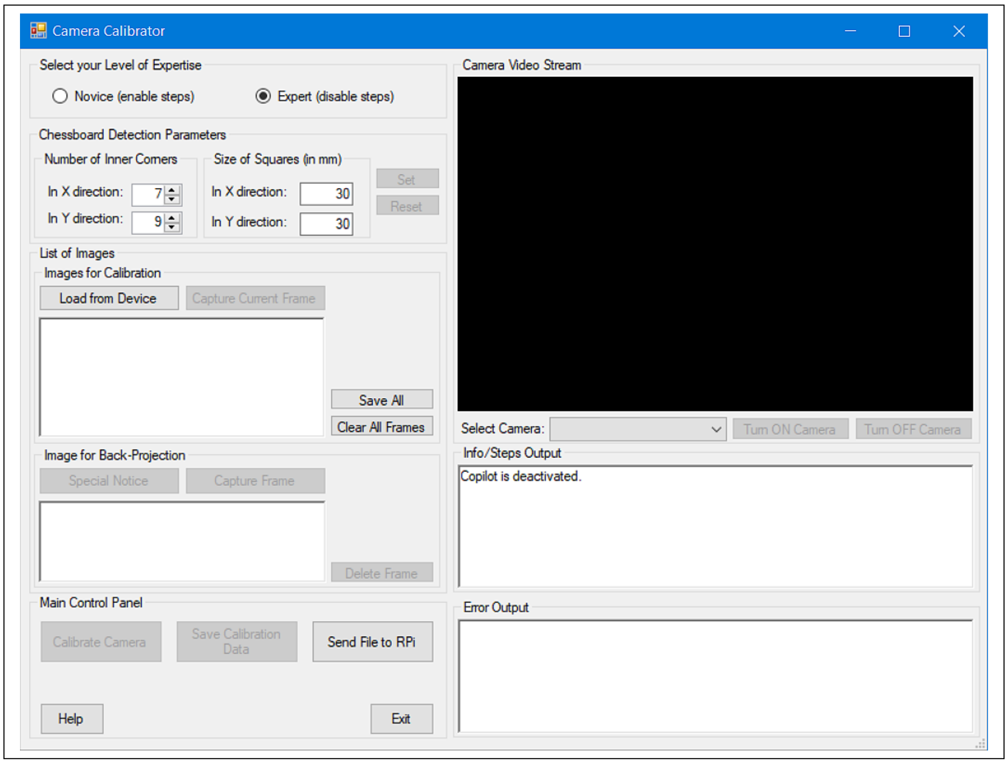


Figure 4.3: Camera calibrator GUI for Windows

Once the camera calibration is done, the software can also be used to send the file that includes the necessary data to the RPi. For that the RPi and the working PC where the software is run, must be connected to the same Local Area Network (LAN) over WiFi.

4.3 Development of an Application to Train an Object Classification Model

The third project, that I was assigned as a trainee electronic engineer was also related to the SE field. In this project, a Windows GUI was developed to train a Machine Learning (ML) model which can be used classify set of pre-defined objects. In a summary, the application is capable of generating the required data, to classify identified objects by the object detection framework explained in the Section 4.1. Figure 4.4 shows how the object detection framework classifies objects with the help of object classification model.

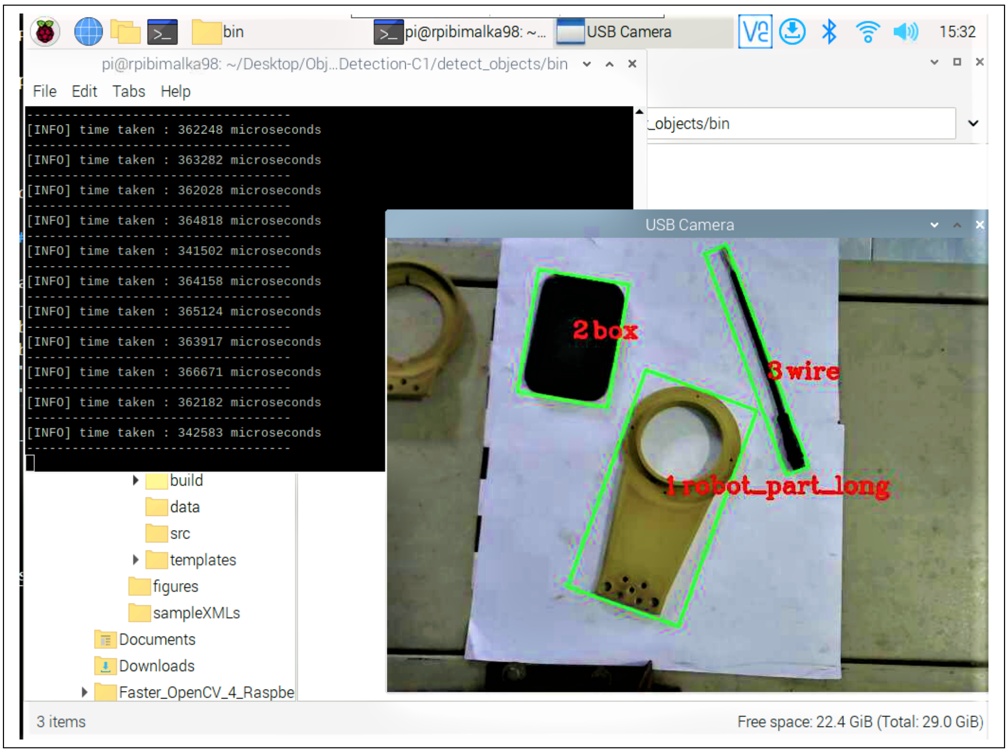


Figure 4.4: Output of the object detection framework with object classification capability

4.3.1 Problem Definition

One of the key aspects of an object detection framework is to classify the detected objects. Because, in order to effectively make use of the information of the detected objects, it is necessary to know the type/class of them. In CV literature there are various methods to determine the class of a detected object. Accuracy as well as the speed of the predictions was a major concern when selecting a method, as the end product was supposed to use in an industrial environment. By considering all of such aspects it was decided to use an ML model which is based on SIFT features and SVMs.

4.3.2 Solution

Prior to deploy any ML tool for predictions in any system, they must be trained for the target task by feeding it the necessary data. In literature this task is known as *training an ML* *model*. Similarly, in our case the object classification model must be trained prior to couple it with the object detection framework explained in the Section 4.1. This training phase of the classification model is carried out using a Windows GUI application developed using Emgu CV (a cross platform .Net wrapper to the OpenCV image processing library) package[1]. Figure 4.5 depicts the layout of the designed Windows GUI.

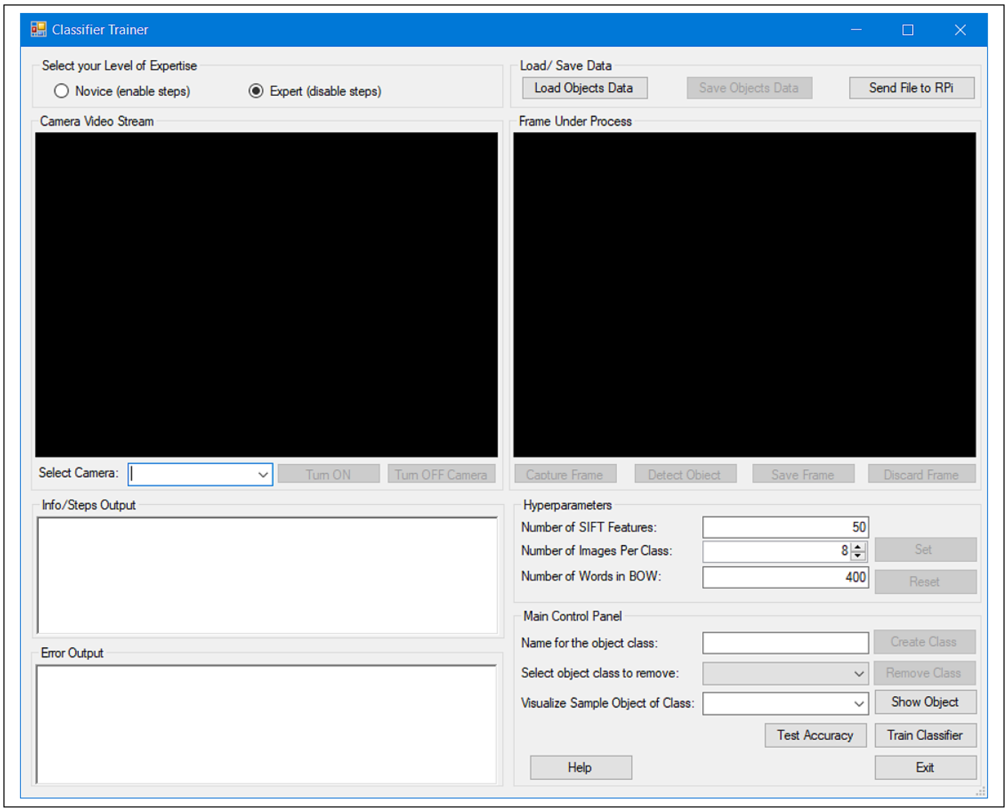


Figure 4.5: Classifier trainer GUI for Windows

Once the training is done, the software can also be used to test the accuracy of the prediction model to fine tune its performance. Ultimately, the file that includes the necessary data can also be sent to the RPi using the same software. For that the RPi and the working PC where the software is run, must be connected to the same LAN over WiFi.

When the model is trained on a set of object classes, we can keep using the model for classification as long as we do not introduce new objects classes to our object detection framework.In addition to that, SVMs were trained using the one-vs-all approach. Therefore, each object class has its own SVM which decides whether an object belongs to that class or not (a binary classification problem). Therefore, the calculated histogram of features, of a given ROI (an object) is passed through all of SVMs when deciding the class of an object. This improves the accuracy of predictions but sacrificing the time. Therefore, time it takes to run a single object detection routine will depend on two factors.

1. Number of non overlapping objects placed in the FOV of the camera: *Higher the number of objects, longer the time object detection takes.*
2. Number of classes of objects: *Lesser the number of classes shorter the time object detection takes.*

Chapter 5

Hands-on experiences

The entire six months period of my industrial training was full of hands-on experiences. It was an ideal combination of learning, unlearning and relearning. LE robotics Pvt. Ltd. had no experts in the CV field during the time I was working there. Therefore, I had to learn most of the things related to my assigned projects by actually doing them. This made an ideal opportunity for me to learn various technologies really fast and with minimum supervision.

5.1 Resources for Self-Learning

As I made a lot of design decisions on my own while doing the project, I had to extensively refer to various knowledge resources throughout my training period. This taught me the right skill set to identify reliable information sources among a lot of garbage on the internet. Following is a list of information sources that I used during my training.

- Google Search
- Stack Overflow
- OpenCV Documentations
- EmguCV Documentations
- Research Publications

5.1.1 Google Search

Website: `https://www.google.com/`
Without any doubt, the starting point of exploring any information on internet was the *Google Search*. It is the search engine provided by Google. The power of modern google search is pretty amazing and it can quickly adapt to your search patterns. This lets it bring the most relevant information to the user. Therefore, using the correct terms during a search matters and the quality of retrieved information extensively depends on that.

5.1.2 Stack Overflow

Website: https://stackoverflow.com/
Stack Overflow is a question and answer website for professional and enthusiast programmers. It features questions and answers on a wide range of topics in computer programming[17]. If you are not doing something really new that no one has ever done, there's a high probability of finding an answer to any question which you may come across. Stack Overflow is that rich of various solutions to fit exactly to your problem at hand.

Additionally, the number of *Scores* an answer has obtained for a particular question is an ideal measure to estimate the the reliability of the answer. Moreover, answers can be easily linked in your documentations and source codes using their associated *Share* links. Figure 5.1 depicts an answer to a question asked on the Stack Overflow platform. The actual answer is available at https://stackoverflow.com/a/10230489/15939357.

Figure 5.1: A sample answer on Stack Overflow and its associated shareable link

5.1.3 OpenCV Documentations

Website: https://docs.opencv.org/4.x/
OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source library that includes several hundreds of computer vision algorithms. It also includes high quality documentation as well, for each module of the library. These documentation contains theoretical concepts and a vast amount of example code snippets. Such code snippets can be used to understand the way of using a particular function in a given application.

5.1.4 EmguCV Documentations

Website: https://www.emgu.com/wiki/index.php/Documentation
EmguCV is a cross platform .Net wrapper for the OpenCV image-processing library. It allows OpenCV functions to be called from .NET compatible programming languages like C#. The wrapper can run on Windows, Android, iOS, Mac OS and Linux. EmguCV also includes a rich documentation which explains usage of various functions and their interfaces. This

documentation comes handy when it comes to Windows GUI development in `C#` using Visual Studio software.

5.1.5 Research Publications

Research publications are various documents published by academic institutes, research laboratories and other renowned companies such as Google, Microsoft and etc in the world. Therefore, the information on those publications is well documented and validated which makes them an ideal information source. In addition to that, their reliability is also very high as the authors are professionals in their respective fields. If you are working in a specific field and you want the most up-to-date knowledge in that field research publications are the tool to use. There is a vast number of websites which publish such articles and therefore they are not mentioned here.

5.2 Usage of Open Source Software

5.2.1 Introduction

Open-Source Software (OSS) is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose[16]. The purpose may be either commercial or private use. Most of the time such OSS are developed in a collaborative manner through online platforms such as *GitHub* (*`https://github.com/`*) and general public has the access to examine the source codes. Additionally, any potential developer can contribute to such open source technologies and number of contributors has no limit.

5.2.2 Advantages

During our internship, we were encouraged to use open source technologies as much as possible inside our project implementations. Because it has a lot of advantages when it comes to R&D field. Few of them are listed below.

- Most of the open source software are free of charge and there is no need of purchasing a license to use the source codes.
- Since OSS are developed in collaborative manner through publicly available online platforms, software are highly optimized due to the diverse perspectives of their developers.
- There are well established community driven online platforms to sort out different problems that may encounter during project development. Most of the time answers to such problems are readily available.
- Due to the indefinite number of contributors, there is a rich set of documentations related to the software. This makes the production really efficient as there is no need of blindly experimenting with the software.

The EmguCV docs that were introduced in the Section 5.1.4 contain much description about functions in its library. However, EmguCV docs does not contain code examples.

This was a challenge that I faced during the development of the software which were explained in the Sections 4.2 and 4.3. This is where the power of open source technologies can be used. EmguCV's source codes are hosted and developed at `https://github.com/emgucv/emgucv`. Therefore, they can be accessed by anyone without any charge and can be used to understand the usage of available functions in various applications. Situation is the same for OpenCV and its source codes are hosted and developed at `https://github.com/opencv/opencv`.

In addition to that, each such repository contains a section named as *Issues* with various modifications requests and questions raised by the community. This issues section can be a life saver when working with latest versions of the software, as most of the problems that may encounter when using the latest versions may not be answered anywhere else on the internet. Figure 5.2 depicts the issues section of the OpenCV project repository.

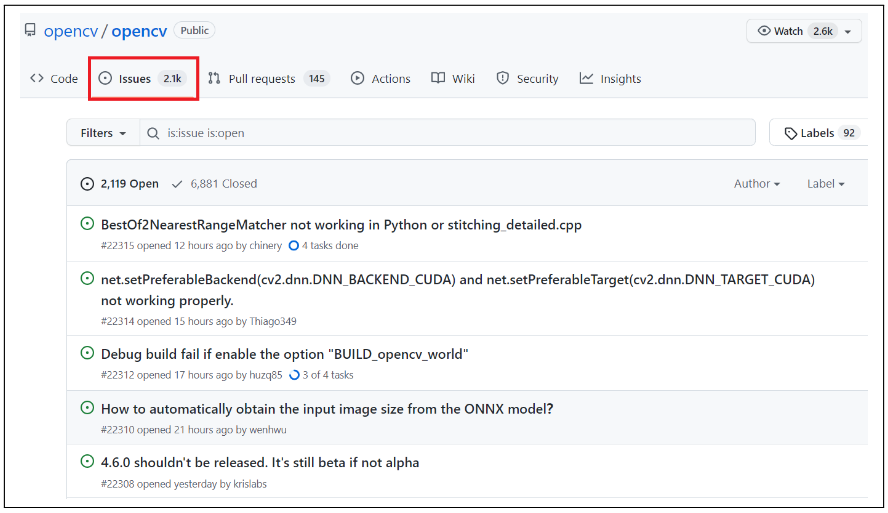


Figure 5.2: Issues section of the OpenCV project repository

5.3 Usage of Modern Tools

As described in the Section 4, throughout my internship period I had to work with technologies related to CV and SE. Therefore, I gained a lot experience of using various tools related to Software Development Life Cycle (SDLC). The tools I used can be classified into two categories depending on the area I used them. That classification is shown in the Table 5.1.

TABLE 5.1
MODERN TOOLS USED DURING THE DEVELOPMENT OF THE SOFTWARE

Windows GUI Development	**Object Detection Framework Development**
Visual Studio 2019 Git	Visual Studio Code Git CMake

5.3.1 Visual Studio 2019

Visual Studio 2019 is simply an IDE owns and developed by Microsoft. It provides necessary facilities for computer programmers to develop software. It consists of a source code editor, build automation tools, debugger and many more features that an IDE should contain [14]. Figure 5.3 shows an instance of the application.

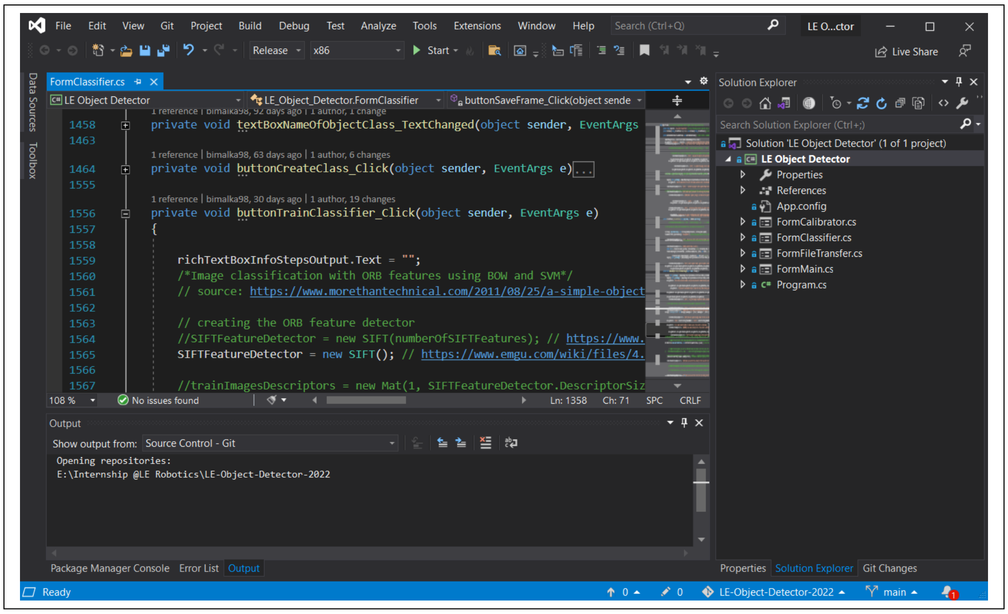


Figure 5.3: “Visual Studio 2019” IDE

Visual Studio uses various Microsoft software development platforms to develop software. In my project I used what is known as *Windows Forms* platform. It is a free and open-source GUI class library included as a part of Microsoft .NET Framework, providing a platform to write client applications for desktop, laptop, and tablet PCs. An application developed using the Windows Forms platform is called as a *Windows Forms Application*. These applications are event-driven applications which means they spends most of their time simply waiting for the user to do something, such as fill in a text box or click a button[18]. The code for the applications developed by me were written in `C#` language.

5.3.2 Visual Studio Code

Visual Studio Code is a lightweight OSS, which can be used to write/edit computer programs (source codes) efficiently. It is being developed by Microsoft with the help of thousands of other developers all over the world. It can run on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for a few programming languages and has a rich ecosystem of extensions for other languages as well to customize it according to the needs of the developer. In short, this cool, powerful application is called as *VS code* by the developers and Figure 5.4 shows an instance of the application.

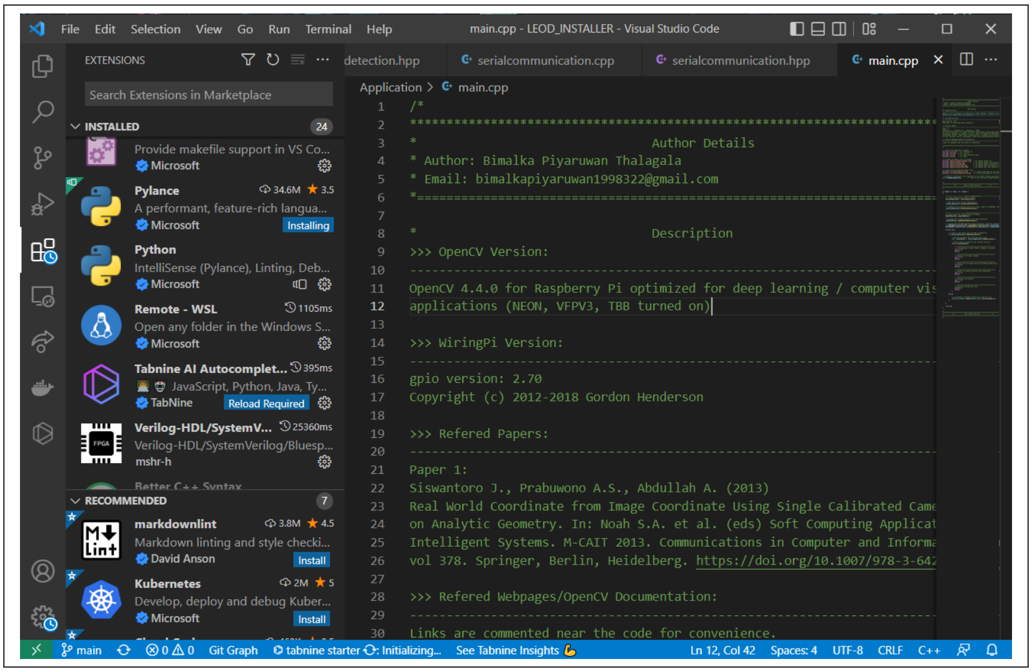


Figure 5.4: “Visual Studio Code” source code editor

During my internship, I used VS code extensively for the development of the object detection framework described in Section 4.1. I developed that algorithm using C++ language and the VS code extension for C/C++ was absolutely a game changer during the implementation. Some of the fascinating features of the VS code is given below[7].

- *IntelliSense*, which provides smart completions based on variable types, function definitions, and imported modules.
- Built-in debugger with breakpoints, call stacks, and interactive console features.
- Built-in version controlling services to keep track of every line of your code.
- Rich ecosystem of extensions for customization.

5.3.3 Git

If the project at hand is related to software development, *Git* is a must-have tool that you should be aware of. The reason can be explained as follows. Imagine an application is being developed by you, and you have made it to a working and stable state. Now your client wants you to add a new feature to the existing application. However, you are uncertain whether the application would work after the addition of the requested feature. Therefore, you tend to make a copy of your stable source code and start working on that copy. Now all the changes are done related to the new feature and you have made it works somehow.

What if the client wants to make a few changes and test another new feature? You have to repeat the same cycle by copying the latest stable version of your application. Imagine, what would happen if your client requested you to update the application for 10 more features. You will end up having a bunch of copies of the previous versions and it becomes an absolute mess. As you have to remember what each copy does. This is where the amazing tool ‘git’ comes into the picture. Its logo is shown in the Figure 5.5.

Figure 5.5: Logo for Git (introduced in 2012)[10]

Git records changes to a file or set of files over time so that you can recall specific versions later. Therefore, you only need to keep one project directory and there is no need of keeping several copies of the same application (folder). This can be seen as *time travel through the history of a specific project/ file*. In this way, git allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more[2].

5.3.4 CMake

In my project, the software development of the object detection framework described in Section 4.1 was carried out on a Windows PC. However, the final target platform to run that application was a RPi, which has a Linux based Operating System (OS) named as *Debian*. A software that is compiled for a Windows OS, can not run on a Linux system. Therefore, a method to bridge this gap as well as to automate the software build process inside the RPi was required. This is where, *CMake* came into the picture. Its logo is shown in the Figure 5.6.

Figure 5.6: Logo for CMake[9]

CMake is a cross-platform, free and OSS for build automation, testing, packaging and installation of software by using a compiler-independent method. That is, CMake doesn't actually execute the build process; instead, it generates build files for other systems such as Linux or Windows. It supports directory hierarchies and applications that depend on multiple libraries[13]. In my projects CMake was used as a build automation tool for generating build files of the objects detection framework for RPi's OS.

Chapter 6

Soft Skills Development

When it comes to the present corporate world, it is obvious that technical skills alone can not take anyone to greater heights in his career. A person must be a perfect combination of both technical and soft skills to grow in whatever field he is in. Learning a new technical skill is not a big deal with the advancements in technology as everything is available at the fingertip. However, mastering a soft skill is much harder and takes time as we have to deal with human beings to do so.

LE Robotics Pvt. Ltd. was absolutely an ideal place to improve existing soft skills as well as to learn a new set of soft skills. Given below is a few of them. Please note that they are not in any specific order.

- Problem-solving
- Adaptability
- Interpersonal skills
- Project management
- Time management
- Emotional intelligence
- Professional work ethic
- Communication

Few of the above listed soft skills and their relation to the industrial training that I have received from LE Robotics R&D facility is explained in the upcoming sections of this chapter.

6.1 Problem solving

Solving problems is an art that needs to be mastered by every engineer regardless of his field. Problem-solving skills that I have gained in the university as an engineering undergraduate were a massive help for me to tackle the project work explained in Section 4. During my internship, I could sharpen those skills to suit to industrial level with the guidance of my supervisor.

Solving a problem includes several stages such as identifying and formulating the problem, researching the literature to find out available solutions and analyzing them, designing components of the system and validating their functionalities and etc. Before joining LE robotics Pvt. Ltd. I did not have much knowledge about researching the literature to find out potential solutions. All I used to do at the university was googling to find solutions from here and there and combine them to make one solution fit the problem at hand.

However, I could realise, that method fails when it comes to the industry. This is where my supervisor guided me towards conducting substantial literature research based on '*Research Publications*' which exposed me to the world of '**Research**'. Research publications are published by academic institutes, research laboratories and other renowned companies such as Google, Microsoft and etc. Therefore, the information on those publications is well documented and validated which makes them an ideal information source. In addition to that, their reliability is also very high as the authors are professionals in their respective fields.

6.2 Adaptability

Companies and working environments are different from one another. To get the most out of where you are, you need to effectively adapt to the current environment and its people. Inability to do so can lead to various problems within the work environment and you can lose interest in working there.

One of the main challenges that I faced during the internship was not having an expert in the field I worked in, to get any useful advice. Because my supervisor was only capable of setting an end goal and providing a high-level solution which can have tons of solutions in the CV literature. Moreover, he had no idea about the complexity of some of the assigned tasks. Due to the gap between what we learn at university and practical problems that arise when it comes to an industrial-level implementation, I was having a difficult time there in the first few weeks.

To receive some advice regarding this situation, I contacted one of the lecturers at our university. He provided me with some potential tricks to get out of the situation while keeping everyone happy. The best of those was "work, for yourself", not for anyone else, not for the company but for you. Because if you get mad and do nothing due to not receiving enough support, you are the one who is going to be adversely affected by it in the end. Therefore, I worked for myself and could complete my assigned task very well with the provided resources at the end.

6.3 Time management

There are several actions that can be taken to manage one's time effectively. Proper planning and organization of the assigned tasks depending on their priority are crucial when it comes to proper time management. Because, if you can not plan and organize the assigned tasks you can be easily overwhelmed by them. During my internship, time management was not a problem for me due to the experience I have gained at university regarding the proper management of the available time. Moreover, the knowledge of modern tools that I had, to get various jobs done was also a plus point. Because, lack of knowledge of the correct tool to use for a particular purpose, can waste your time.

6.4 Professional work ethic

Respecting professional engineering work ethics is a must-have soft skill when it comes to the cooperative world. You may be a high performer but if you can not be trusted, it will adversely affect the long run in the industry. Therefore, to be successful in the industry you must have good technical performance as well as the best professional work ethic. Our supervisor taught us more professional ethics than technical skills. Because with the advancement of today's technology gaining a technical skill is more than easy. However, becoming a man with professional ethics takes time and experience. Some of the ethics I have learned during my internship are mentioned below.

- Punctuality
- Trustworthiness and obeying the NDAs of the organization.
- Perform services only in the areas of competence.
- Having a mindset to complete the assigned tasks without limiting to working hours/ days.
- Respect the Interllectual Property (IP) of others.
- Paying special attention to the terms and conditions of the Licenses of OSS when using them in applications.

6.5 Communication

Communication in an industrial environment can be defined as conveying or sharing ideas effectively and listening actively. Communication can happen in verbal, written or as a combination of both formats. During my internship, I could gain much experience in effective communication through project demonstrations, various presentations and technical documentation preparations.

Project demonstrations included demonstrations of the projects explained in Section 4. There I had to measure the performance of the developed CV subsystem according to the criteria that were specified by the supervisor and present them to him for receiving feedback for further development of the system. Technical documentation preparation included preparing three technical documents which are explained below.

1. A 'System Installation Guide' for the CV subsystem depicted in the Figure 4.2 and explained in the Section 4.
2. A 'Further Development Guide' for the object detection framework explained in Section 4.1.
3. A 'User Manual' which explains how to use the software explained in Sections 4.2 and 4.3.

Through these three technical documentation preparation, I could gain a strong knowledge about documenting project works. It improved my technical writing capabilities as well as professional document editing using LaTeX which is a software system for scientific/technical document preparation.

Chapter 7

SWOT Analysis of the organization and self

7.1 SWOT Analysis of the Organization

7.2 SWOT Analysis of self

Chapter 8

Conclusion

The special industrial training was a whole new challenge which demanded a different set of technical skills as well as soft skills. LE Robotics (Pvt.) Ltd. did a great job in injecting much of those during the period of six months. One of the best achievements of the training is the mindset that our supervisor built in us, towards working under minimum supervision. This gave me an opportunity to experience my full potential as an engineering student and revealed my true calibre. This definitely, will be massive support for me to thrive in the fast-moving industry rather than surviving there. Because what I could realize during the training period was that there will be no one to spoon-feed you in the industry and you have to learn things fast on your own.

As a trainee, I was assigned a project to develop a Computer Vision subsystem for a pick and place machine. Since there were no experts in the Computer Vision field in the facility, the entire prototype development process was done individually by myself. Therefore, I could gain a very strong knowledge about turning an idea into a viable prototype using the available resources in the facility. During that process, I went over the stages such as idea generation, research, planning and prototyping. I believe, the process could be accelerated by a large factor if there was an expert in the Computer Vision field.

In the last month of the training period I had the opportunity to work with our supervisor and the other employees of the facility for integrating the developed Computer Vision subsystem with the existing six degree of freedom articulated robot arm. This taught me the importance of being a team player and completing the assigned tasks well in advanced, to be successful in the final stages of the product development. Because the subsystems/ modules that are developed by you must be compatible with the the subsystem/ modules developed by the other employees for the system to function as one single unit.

Appendices

Appendix A

Camera Calibration

The important input data needed for calibration of the camera is a set of 3D real world points and the corresponding 2D coordinates of these points in the image. 3D points are called object points and 2D image points are called image points.

I. **2D Image Points:** These points can be easily found from the images. These image points are the locations where two black squares touch each other in chess board. Functions in OpenCV is used to extract those points[5].

II. **Object Points:** For simplicity, the chess board was kept stationary at XY plane, (so Z=0 always) and camera was moved accordingly. This consideration helps us to find only X,Y values. Now for X,Y values, we can simply pass the points as (0,0,0), (1,0,0), (2,0,0), ... which denotes the location of points. In this case, the results we get will be in the scale of size of chess board square[5]. But if we know the square size, (for our implementation square size of the chess board was 30 mm), we can pass the values as (0,0,0), (30,0,0), (60,0,0).

The process of camera calibration is given below.

1. Keep the chess board stationary in the XY plane of the real world coordinate system as shown in the Figure A.1 and do not let it move due to any reason.

2. Capture about 10-15 images (higher the better) as shown in the Figure A.2 using the camera that will be used for object detection, in several distances and orientations.

   The actual number of required images is around 6. However, in some scenarios OpenCV is unable to detect the chess board pattern properly in every image. Two of such reasons are given below.

   - **Only a part of the chessboard is visible in the captured image**: For the program to work, whole chessboard must be visible in the captured image.
   - **No proper lighting conditions**: if the environment that the chessboard is placed, does not have proper lighting conditions application will not work as expected. Therefore make sure that the chessboard pattern receives enough light.

   Hence, it's better to capture images more than the required amount so that we can filter out damaged/ useless images later.

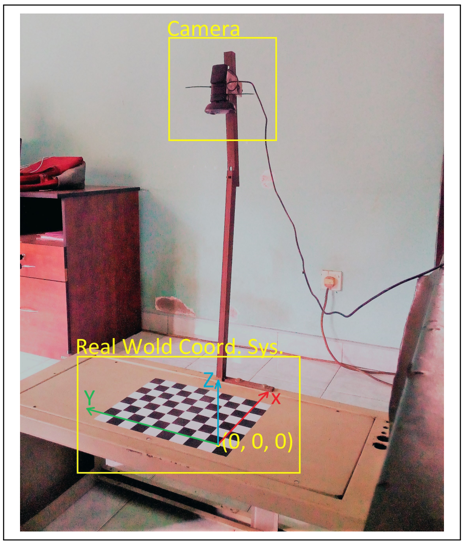


Figure A.1: Real world coordinate system and the position of the camera with respect to it.

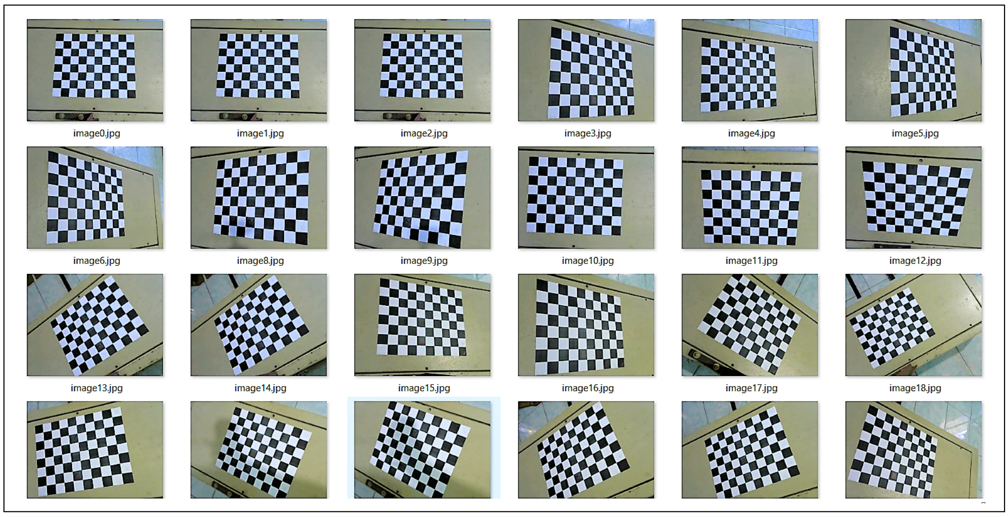
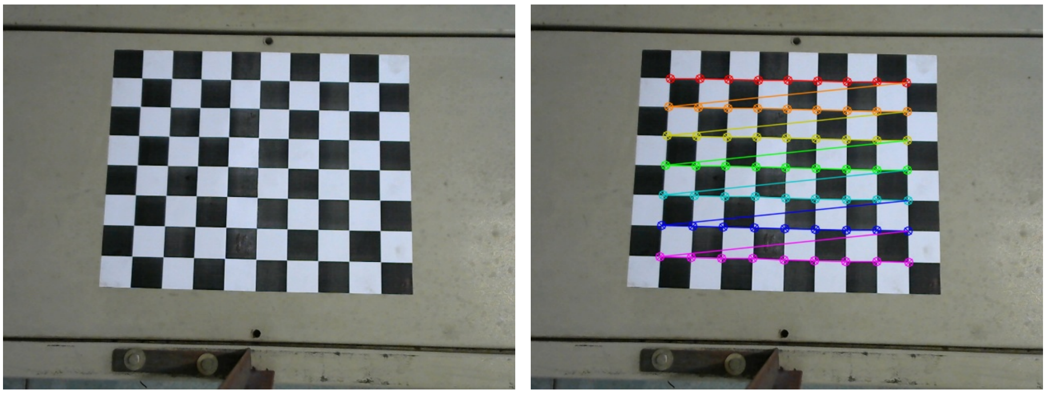


Figure A.2: Raw images captured by placing the camera in various distance and orientations

3. Capture one additional special image, placing the camera in the place that it will be mounted when the actual object detection algorithm runs. Extrinsic parameters[11] corresponding to this image will be directly used in our Back-projection algorithm.

   Extrinsic parameters contains a translation vector and a rotation matrix. These two together brings the calibration pattern from the object coordinate space (in which object points are specified) to the camera coordinate space. In more technical terms, the rotation matrix and translation vector performs a change of basis from object coordinate space to camera coordinate space. Due to its duality, it is equivalent to the position of the calibration pattern with respect to the camera coordinate space [6].

4. Use all of the captured images to calibrate the camera. The necessary documentation of the used functions are explained in [6]. After carrying out those functions you will be



(a) Raw image (b) Processed image to identify the pattern

Figure A.3: Captured special image to extract the extrinsic parameters that are useful for back-projection

able to extract the extrinsic and intrinsic parameters related to the special image and the camera respectively.

References

[1] Emgu CV: a cross platform .Net wrapper to the OpenCV. https://www.emgu.com/wiki/index.php/Version_History#Emgu.CV-4.5.1.

[2] Git - About Version Control. https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control.

[3] LE Robotics - Articulated Robots. http://www.lerobotics.lk/articulated-robots.php. publisher: LE Robotics Pvt. Ltd.

[4] LE Robotics - SCARA Robots. http://www.lerobotics.lk/scara-robots.php. publisher: LE Robotics Pvt. Ltd.

[5] OpenCV: Camera Calibration. https://docs.opencv.org/4.4.0/dc/dbb/tutorial_py_calibration.html.

[6] OpenCV: Camera Calibration and 3D Reconstruction. https://docs.opencv.org/4.4.0/d9/d0c/group__calib3d.html.

[7] Visual Studio Code - Code Editing. Redefined. https://code.visualstudio.com/.

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

[9] Wikimedia Commons. File:cmake.svg — wikimedia commons, the free media repository. https://commons.wikimedia.org/w/index.php?title=File:Cmake.svg&oldid=611355701, 2021. [Online; accessed 29-July-2022].

[10] Wikimedia Commons. File:git-logo.svg — wikimedia commons, the free media repository. https://commons.wikimedia.org/w/index.php?title=File:Git-logo.svg&oldid=629775061, 2022. [Online; accessed 29-July-2022].

[11] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[12] G Henderson. Wiring Pi: GPIO Interface library for the Raspberry Pi. http://wiringpi.com/.

[13] Wikipedia contributors. Cmake — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=CMake&oldid=1099974037, 2022. [Online; accessed 29-July-2022].

[14] Wikipedia contributors. Microsoft visual studio — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio&oldid=1098834529, 2022. [Online; accessed 28-July-2022].

[15] Wikipedia contributors. Non-disclosure agreement — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Non-disclosure_agreement&oldid=1096849528, 2022. [Online; accessed 29-July-2022].

[16] Wikipedia contributors. Open-source software — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=1096922686, 2022. [Online; accessed 28-July-2022].

[17] Wikipedia contributors. Stack overflow — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Stack_Overflow&oldid=1100396097, 2022. [Online; accessed 27-July-2022].

[18] Wikipedia contributors. Windows forms — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Windows_Forms&oldid=1094827942, 2022. [Online; accessed 28-July-2022].