

University of Waterloo
Faculty of Engineering
Department of Mechanical and Mechatronics Engineering

Intelligent Manufacturing Inspection Tool

University of Waterloo
200 University Avenue West
Waterloo, Ontario, N2L 3G1 CA

Prepared by
Herman Grewal 20556076
Lalit Lal 20572296
Mohamed Elshatshat 20576631
Oluwatoni Ogunmade 20574959
Rey Reza Wiyatno 20543168

3 December 2018

200 University Avenue West
Waterloo, Ontario, CA
N2L 3G1

3 December 2018

Professor William Melek, Director
Mechatronics Engineering
University of Waterloo
Waterloo, Ontario
N2L 3G1

Dear Professor William Melek,

This report, entitled "Intelligent Manufacturing Inspection Tool" was prepared as our MTE 481 final report to explain the full design process of the proposed manufacturing inspection tool. Currently, manufacturing inspection process for cosmetic defects is either very labour intensive or very customized to specific products and processes. We propose a machine learning based automated inspection tool which provides a versatile low-cost solution for small and medium size companies and generalizability.

We conducted a thorough needs assessment in consultation with several companies. After validating the problem, a patent search was completed to understand existing solutions. During the brainstorming phase, a morphological chart was developed from which four alternative designs were developed. Using weighted criteria in a decision matrix, a final proposed solution was decided. A detailed section was provided for analysis of the selected design. The sub-components include the enclosure, camera, lighting, robot arm, inspection algorithm, part identification, data communication and user interface. The bill of material and budget were also provided. Finally, a Gantt chart was developed to ensure a working deliverable at the end of MTE 482.

We would like to thank Kevin Eng (Ecobee), Aleksey Tsaplin (Ecobee), and Wim Deweer (Google Hardware) for their consultations and willingness to help with design decisions, as well as development of criteria and constraints. We would also like to thank both Ecobee and OpenROV for providing defective parts to validate our system. Finally, we would like to thank Professor Ayman El-Hag and Professor Jan Huissoon for their supervision and guidance throughout the term.

We hereby confirm that we have received no further help other than what is mentioned above in writing this report. We also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Sincerely,



Herman Grewal 20556076
Lalit Lal 20572296
Mohamed Elshatshat 20576631
Oluwatoni Ogunmade 20574959
Rey Reza Wiyatno 20543168

Summary

The purpose of this report is to explain the full design process for a low-cost solution for high quality manufacturing inspection. In this report, a design for low-cost intelligent manufacturing inspection tool is presented in order to fill this market gap. This system focuses on detecting visual defects which include scuffs, scratches, dents, and discolorations. The functional goals of the project include part placement (e.g. manual placement or using existing manufacturing lines), image collection, image analysis, database management, and user interface.

A metal enclosure with a robot arm that hangs at the top of the enclosure is the proposed solution presented. A camera is attached at the end effector of the robot arm, which allows the camera to move around and capture images of the inspected parts from various different positions without physically touching the parts. Once images are taken, they are transferred to a server computer where the data will be stored and processed with the vision algorithm. The analysis of each of the inspected parts can then be accessed by the operator and quality engineers through a user friendly interface.

The selected design was chosen using a decision matrix for each of the subsystems, and was also reaffirmed by the industry consultants in the validity of the design. Based on the design review, the selected design offers a solution that satisfies the specified constraints, criteria, as well as the objectives.

The use of instance segmentation algorithm such as the Mask R-CNN [1] is recommended to increase localization performance which will be beneficial to the overall system should the time permit. Investigation on meta-learning algorithms such as the Model Agnostic Meta Learning (MAML) [2] will also be interesting since it reduces the amount of data needed to train the model. Furthermore, the use of MAML allows one to fine-tune the model during test time quickly which can increase the performance of the model without having to spend significant amount of time to retrain the model.

Table of Contents

Summary	iii
List of Figures	v
List of Tables	vii
1 Introduction	1
1.1 Background	1
1.2 Needs Assessment	2
1.3 Patents and State of the Art	3
1.3.1 Patents	4
1.3.2 State of the Art Methods	5
2 Proposed Solution	11
2.1 Abstraction	11
2.2 Selected Proposed Solution	14
3 Detailed Designs	16
3.1 Design Proposed Solution	16
3.2 Design Analysis	17
3.2.1 Enclosure	17
3.2.2 Camera	21
3.2.3 Lighting	23
3.2.4 Robotic Arm	24
3.2.5 Inspection Algorithm	35
3.2.6 Part Identification	37
3.2.7 Data Communication	39
3.2.8 User Interface and Experience	39
3.3 Design Review	42
4 Schedule and Budget	43
4.1 Schedule	43
4.2 Budget	43
5 Conclusions and Recommendations	45
6 Teamwork Effort	46
Acknowledgement	47
References	48
Appendix A Detailed Part Drawings	52
Appendix B Bill of Material	62
Appendix C Sample of Defect Criteria	63

List of Figures

1	Illustration of the viewing zones [3].	3
2	Illustration of the cosmetic inspection system in US20140267691A1 [5].	4
3	Illustration of the visual inspection system described in US20180130197A1 [6].	5
4	Block diagram of the automatic defect classification system in US5544256A by IBM [7].	6
5	Hardware setup used in [8].	7
6	The classes of defects classified in [8].	7
7	Classification results in [8].	8
8	Hardware setup used in [10].	8
9	Example of defect detection using MobileNets-SSD [10].	9
10	Example of defect instance segmentation using Mask R-CNN [13].	9
11	Example of defect instance segmentation using Mask R-CNN generalized to an X-ray image of a jet turbine blade which was not included in the training dataset [13].	10
12	Sketch of the alternative design #1.	11
13	Sketch of the alternative design #2.	13
14	Sketch of the alternative design #3.	13
15	Sketch of the alternative design #4.	14
16	Sketch of the design proposed solution.	16
17	Illustration of the systems.	17
18	Bolted (top) and slotted (bottom) panel configurations.	19
19	Felt entry/exit design used in car washes [15].	19
20	FEA simulation on top plate loading.	21
21	Enclosure CAD model. Note that the bolts and felt are not shown.	21
22	Evaluation of programmatic focus adjustment on Logitech C525. Note how the focus can be adjusted using the slider shown at the bottom of Figure 22a and 22b. The software to implement this feature was done using Linux V4L2 API and OpenCV.	23
23	The result of camera mounted lighting testing. The images captured during the 360° scan of the part allows for all the defects to be visible. The green boxes outline scratches that would be easily classified by the inspection algorithm, whereas the red boxes outline scratches that would not be easily classified.	25
24	Sketch of the robotic arm with dimensions.	26
25	Calculation of the final torque required at each joint.	26
26	Illustration of the thread strip failure mode. The factor of 4 in the denominator represents the 4 bolts that are used to distribute the load, as dictated by the mount pattern of the actuators themselves.	27
27	Illustration of bolt shear failure mode.	29
28	Sketch of tear out failure mode.	30
29	Mock-up of robotic arm with camera attached.	31
30	High level illustration of the electrical architecture.	32
31	Illustration of the power distribution system.	33

32	The main functionalities of the software.	34
33	Examples of scratch detection on a plastic part. The model is based on MobileNets-SSD and trained on small amount of random pictures of scratched objects acquired from Google search.	37
34	Illustration of the data communication pipeline.	39
35	Illustration of the touch screen with the user interface.	40
36	Illustration of the database query interface.	41
37	Project timeline.	43

List of Tables

1	Viewing zone classifications on parts.	2
2	Summary of constraints and criteria.	3
3	The morphological chart.	12
4	Decision matrix for the alternative designs.	15
5	Comparison of different cameras considered in this project.	23
6	Masses of individual robotic arm components	28
7	Electrical components used in the robotic arm.	31
8	Comparison of various inspection algorithms considered in this project (“High” is the best).	36
9	Comparison of commercial embedded barcode scanners.	38
10	Bill of material.	62

1 Introduction

In this section, background to the problem is introduced in Section 1.1. Based on the current need, a thorough assessment is conducted to develop a list of objectives, constraints, and criteria which are presented in Section 1.2. Finally, a list of patents and state of the art methods are investigated in Section 1.3 to understand existing solutions and to evaluate the patentability of the proposed solution.

1.1 Background

Virtually every consumer device is mass manufactured in factories. Manufacturing processes have limitations that result in deviations and deformities. Subsequently, manufacturing inspection is very important to maintain a consistent quality, especially in consumer facing products. In this age of globalization, the manufacturing is commonly done in different parts of the world. This further complicates the challenge to maintain consistent quality.

Large companies have invested huge amounts of capital to develop visual inspection tools and procedures to ensure consistency. These tools are often very expensive and customized to the specific product. In many cases, inspectors are often employed to validate the quality of productions. Although inspectors adhere to a product quality inspection manual, there are often inconsistencies due to varying levels of experience among inspectors, on-the-job fatigue and distractions. Companies often create large manufacturing quality departments who audit suppliers, conduct statistical analysis and create control processes. All of these current solutions lack high levels of accuracy and are typically expensive. Furthermore, current solutions does not offer full traceability since inspectors are usually only tasked to determine whether a part is defective or not, without having to provide the explanations on why the part is defective.

As a result, smaller and medium size companies with limited resources and funds have difficulty conducting manufacturing inspection. This is a problem since consistency in product quality is pivotal to not only maintain profit margins and competition, but also to build the reputation, which is very crucial for smaller and medium size companies who are not well known yet. in the consumer electronics industry.

A low cost automated manufacturing visual inspection tool would enable smaller companies to improve their quality control processes and ensure a consistent level of quality for their consumers. Should the technology exist, smaller companies will be able to provide customers with high quality assurance and gain trust from the customers. In this report, a design for intelligent low-cost manufacturing inspection tool is proposed.

1.2 Needs Assessment

The formulated problem statement based on the background information presented in Section 1.1 is as follows: there is no low-cost automated solution for visual inspection of high end consumer facing products.

The functional goals of the system are broken down into five steps. The first is to load the part into the system to inspect. The next is to scan the part by taking images or videos of the part. The images would then be processed to classify and localize defects. The results of the image processing would be displayed for the user, and stored to be accessed in the future. Finally, the part would be unloaded from the system. As discussed in Section 3.1, the scope of the project is limited to designing the enclosure, image capturing mechanism, inspection algorithm, and post-processing. Note that this excludes any automated loading and unloading mechanisms.

The objectives of the system are determined based on the functional goals and are as follows. First, the system should have an accuracy and precision greater than 90%. The accuracy of the system is determined by the ability to correctly detect and classify defects. The precision of the system is a measure of the repeatability of the system in generating results. Next, the system must be able to localize and classify scuffs, scratches, dents, discolorations, which are the four main defects to be detected in the system. These types of defects were chosen based on the recommendations of industry consultants. The system must be able to identify minimum size of defects of 1mm. Defects smaller than 1mm across are hard to notice and are not typically an issue in manufacturing inspection [3]. Finally, the system must be able to inspect class A, B, and C viewing zones. These viewing zones are outlined and explained in the Table 1 and Figure 1.

Table 1: Viewing zone classifications on parts.

Viewing Zone	Explanation
Zone A	All areas that include the primary appearance and interface area, as the customer views or interacts with the product part. Refer to Figure 1, which identifies this area. This is the area that is most visible to the customer.
Zone B	Areas adjacent to A zone, but not readily visible in normal open and close positions.
Zone C	Areas that are visible only when special effort must be made to see a sizable defect.
Zone D	All areas that are not exposed once the unit is populated.

The constraints for the system were developed in order to validate the proposed solution. The first constraint is the cost, where the maximum manufacturing cost of the system must not exceed \$1000 CAD. Other companies in this space offer solutions that range from \$20000 USD to \$150000 USD [4] based on the scale and complexity of the defect detection systems. The third constraint is that the system must be able to process each surface in 10 seconds. This metric comes from the industry consultants and various guidelines for inspecting cosmetic defects [3]. The part is also

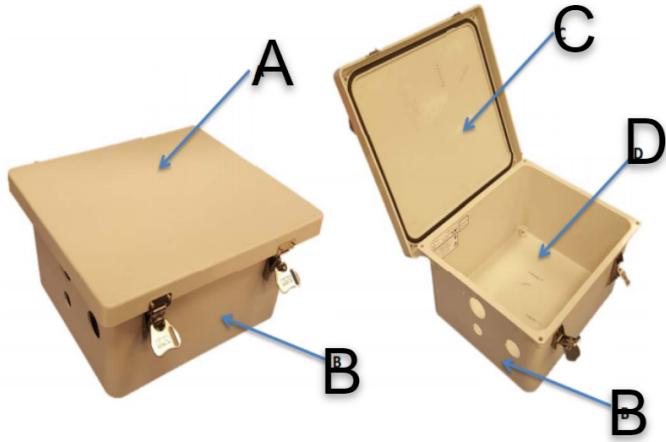


Figure 1: Illustration of the viewing zones [3].

being constrained to parts with flat bottom surface to avoid balancing issues during inspection. Finally, the system must not introduce new defects during inspection. This means that once the part has been loaded, there must be no physical manipulation of the part.

The criteria used for selecting the best alternative design were chosen with the problem statement in mind. The system should be as affordable as possible while still delivering the required performance. The processing speed should be maximized in order to improve the throughput of parts. The proposed solution should be easily used and interpreted. The system is meant to be automated, thus it is preferred to have a solution that requires as little maintenance as possible. Finally, the level of autonomy should be as high as possible to minimize human intervention. Table 2 summarizes the constraints and criteria.

Table 2: Summary of constraints and criteria.

Constraints	Criteria
Maximum cost \$1000 CAD	Cost
Maximum processing time per viewing zone 10 seconds	Processing speed
Must be able to inspect parts with flat surface	Maintenance
Must not introduce new defects	Degree of automation
	User experience

1.3 Patents and State of the Art

Several patents and methods from both the industry and academia were investigated. Section 1.3.1 and 1.3.2 discuss the patents and state of the art methods, respectively.

1.3.1 Patents

Several patents in automated manufacturing inspection technologies were investigated to evaluate whether or not the proposed design can be patented.

US20140267691A1 The first patent identified, US20140267691A1, is held by FedEx [5]. Filed in 2014, the patent describes a system of identifying cosmetic and operational features of an electronic device. The patent claims to be able to inspect the part from multiple views using a robotic arm as shown in Figure 2. The patent also claims that the system can identify the electronic components on the inspected part and includes an interface which allows the operator to order replacement parts if needed. Furthermore, the patent claims that the system is able to detect the defects by comparing the images to a baseline non-defected part. The user interface contains a database of models associated with each Original Equipment Manufacturer (OEM) and display the image of the part being inspected.

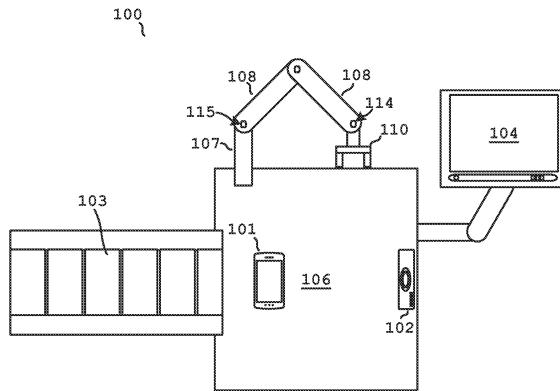


Figure 2: Illustration of the cosmetic inspection system in US20140267691A1 [5].

US20180130197A1 The second patent investigated was the US20180130197A1 [6] which claims an enclosure for inspecting parts using a downward facing camera as shown in Figure 3, and a dark-field and bright-field lighting modules. This patent was filed by Instrumental Incorporated. The patent claims an enclosure that can be installed along an assembly line, which utilizes a nest to hold the inspected part at a fix distance from the camera. The light sources inside the enclosure are directional and perpendicular to the surface. The patent also claims the inclusion of a weight sensor coupled to the nest and a set of weight sensors coupled to the corners of the nest module.

US5544256A The third patent investigated was a system for defect detection and classification. Filed by IBM in 1993 [7], the patent claims an adaptive detection system capable of labelling

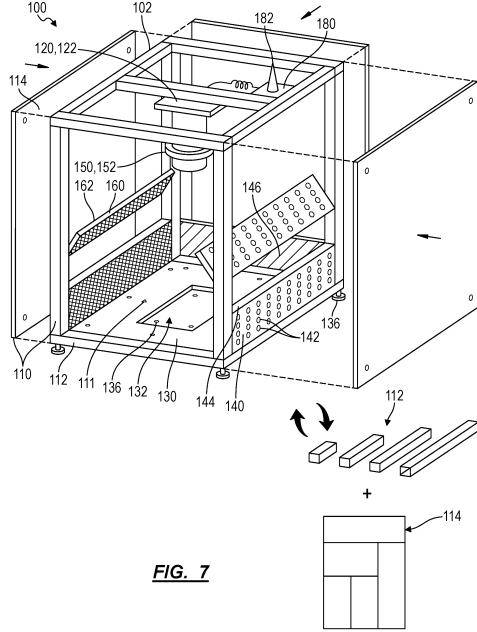


Figure 3: Illustration of the visual inspection system described in US20180130197A1 [6].

defects and a knowledge-based classification engine using fuzzy logic. The overall architecture of the system is shown in Figure 4. The system is primarily targeted at semiconductor and electronic package manufacturing.

In conclusion, based on the evaluation of existing patents, the proposed solution is determined to be patentable. The investigation into existing patents suggested that the proposed design in this report does not infringe on any existing patents. The existing patents often rely on a comparison to a baseline image or a *golden template* as it is sometimes referred to as in the industry. The proposed solution does not rely on a golden template because, which allows the proposed solution to be part agnostic. The proposed solution is also able to inspect at least class A and B viewing zones (see Table 1 and Figure 1), which offers more capability compared to the US20180130197A1.

1.3.2 State of the Art Methods

Several state of the art methods to perform defect detection in academia were investigated. These methods include machine learning based classification, object detection, and instance segmentation algorithms.

Classification Method A classification method by was proposed [8] by leveraging a neural network based classification model called ZF-Net [9] in order to classify various types of defects. In

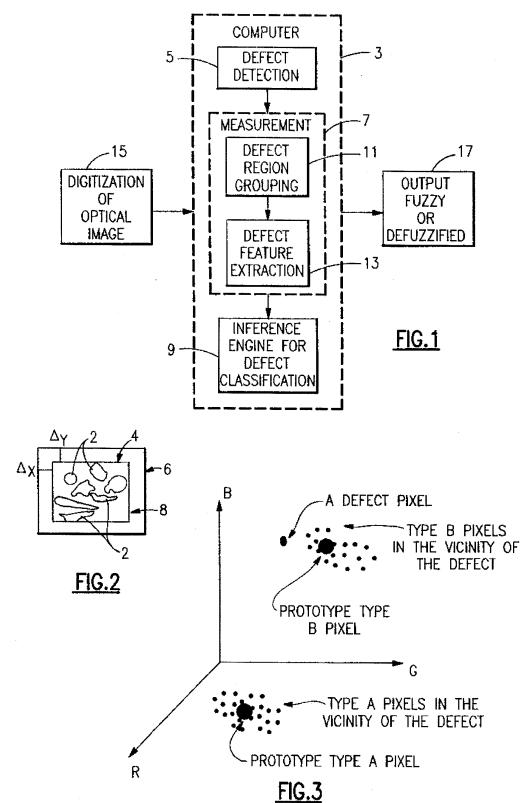


Figure 4: Block diagram of the automatic defect classification system in US5544256A by IBM [7].

their work, the defects are constrained to 11 types of defects on the surface of a touch panel glass as shown in Figure 6. The authors also developed the hardware of the inspected system as shown in Figure 5.

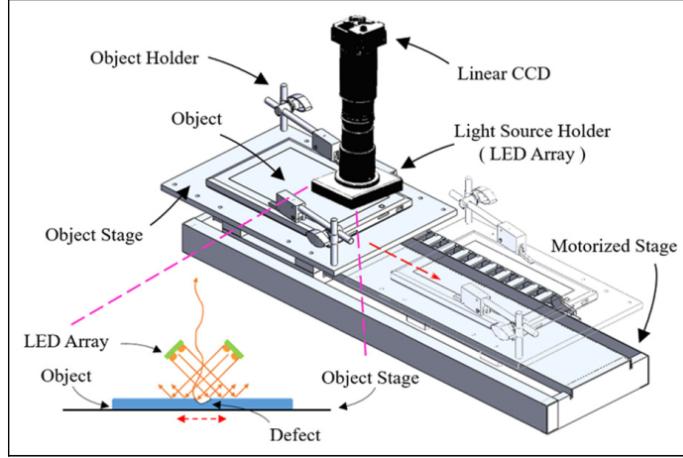


Figure 5: Hardware setup used in [8].

The hardware setup used a linear charge-coupled device (CCD) camera and an LED array. Furthermore, the system was setup in such a way that only the light scattered from the surface of the inspected part is received by the camera. As a result, the camera will produce a black image when the surface of the touch panel glass is smooth. In contrast, the image will show some white pixels if the part is defective. These black and white images were then processed by the algorithm to be classified.

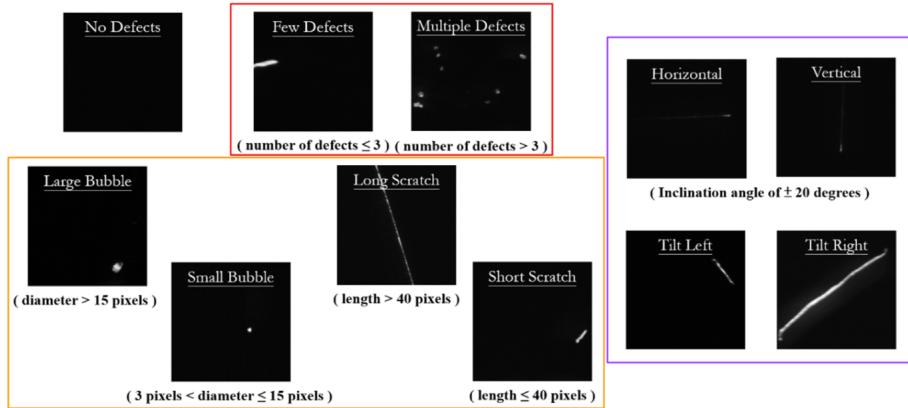


Figure 6: The classes of defects classified in [8].

The accuracy achieved using this method was found to be above 96% which shows promising direction to leverage on machine learning algorithms to perform defect classification. Some of the results are illustrated in Figure 7.

Test Image	Classification Results	Output	Test Image	Classification Results	Output
	No defects	89.61%	No defects	Multiple defects	33.20%
	Few defects	4.53%		Small Bubble	30.72%
	Small Bubble	4.11%		Large Bubble	28.41%
	Short Scratch	1.28%		Short Scratch	2.20%
	Tilt Left	0.22%		Tilt Left	1.51%
Test Image	Classification Results	Output	Test Image	Classification Results	Output
	Few defects	32.37%	Few defects Long Scratch Tilt Right	Few defects	24.14%
	Long Scratch	26.74%		Large Bubble	23.61%
	Tilt Right	25.58%		Short Scratch	22.72%
	Vertical	7.80%		Tilt Left	19.51%
	Short Scratch	6.61%		Vertical	5.17%

Figure 7: Classification results in [8].

Object Detection Method Object detection algorithm that not only classify defects on a part, but also localizing the defects in the image space using bounding boxes was also proposed for defect detection [10]. The detection model used was the Single Shot MultiBox Detector (SSD) [11]. Since the model is preferred to be lightweight, a model called MobileNets [12] was used as the backbone of the SSD model. The combination of SSD with MobileNets was shown to achieve a superior processing speed while not significantly sacrificing the classification and localization performances. Figure 9 shows some examples of how the proposed detection method was able to detect multiple defects in an image.

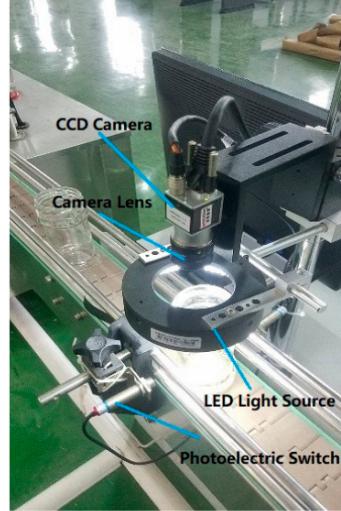


Figure 8: Hardware setup used in [10].

A hardware setup was also developed in order to evaluate the system as shown in Figure 8. Unlike in [8], the setup in [10] takes in a full RGB image as the input to the model. The authors also constrained the defect detection problem to only detect defects on a glass jar sealing surface. The defect types include breach, dent, burr, and abrasion.

The authors showed that the method was able to achieve average detection rate of 95%. The process-

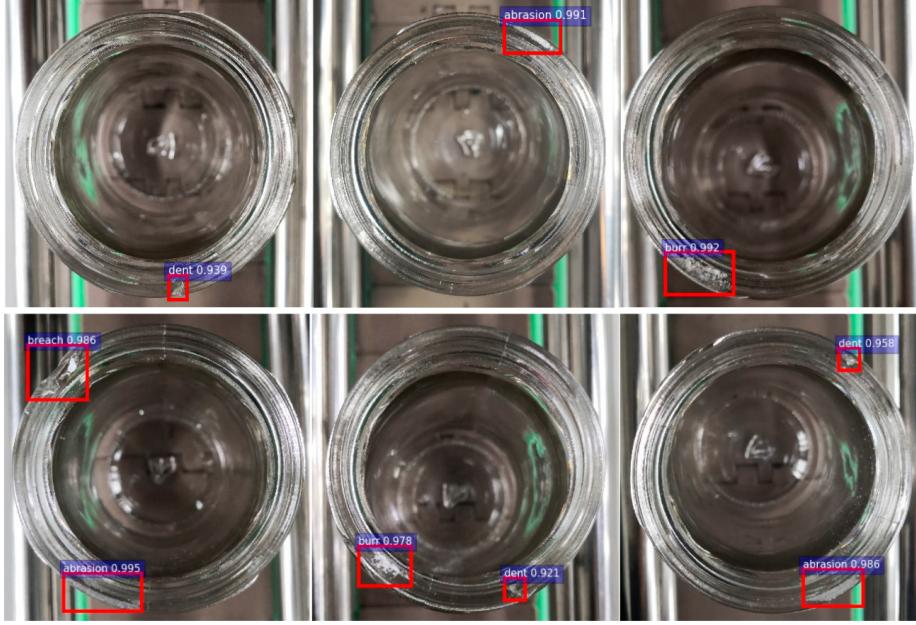


Figure 9: Example of defect detection using MobileNets-SSD [10].

ing time was 120ms per image using NVIDIA GeForce Titan X graphics processing unit (GPU). The results show promising direction on using the family of object detection algorithms for classifying and localizing defects. The localization component in particular is important since it provides more traceability from the users' perspective.

Instance Segmentation Method An instance segmentation based method using a model called Mask R-CNN [1] to perform defect detection was also proposed [10]. Similar to object detection methods, instance segmentation methods also predict the location of the defects in the image. Instance segmentation however localize the defects at the pixel level as opposed to drawing bounding boxes. As a result, instance segmentation method provides finer localization, which can be used to quantify the size of the defects more accurately. Figure 10 shows some examples of the segmentation results.

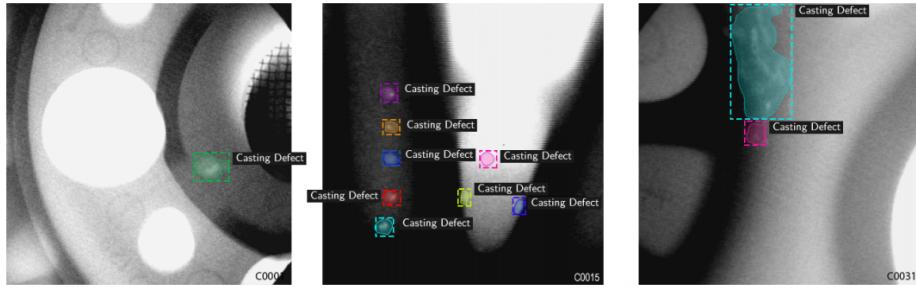


Figure 10: Example of defect instance segmentation using Mask R-CNN [13].

The model was trained using GRIMA X-ray (GDXray) dataset [14] to detect various casting and

welding defects. The performance of the model was quantified by measuring the mean average precision (mAP), and achieved highest mAP of 0.957 on GDXray dataset. The authors also demonstrated how the model was able to detect the same types of defects on different parts that are not included in the training dataset as shown in Figure 11. This shows how the model can be generalizable, which is an important feature for detecting defects since it does not require one to keep retraining the model for inspecting defects on different parts.

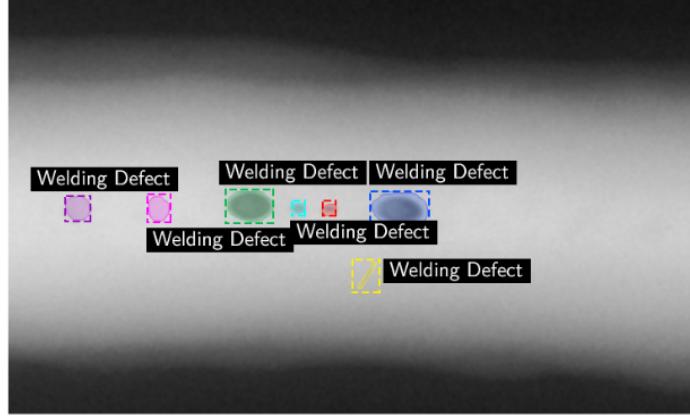


Figure 11: Example of defect instance segmentation using Mask R-CNN generalized to an X-ray image of a jet turbine blade which was not included in the training dataset [13].

2 Proposed Solution

2.1 Abstraction

Since the problem has very modular components and subsystems, a morphological chart was developed for the various options in each modular component. The morphological chart is presented in Table 3.

Using the morphological chart, four design alternatives were developed. The first design includes a robotic arm for picking and placing defective parts in and out of the inspection enclosure. This enclosure would be embedded with LEDs to provide lighting. The chosen camera was an RGB camera with a structured light sensor. The enclosure would be able to raise and lower in order to change the distance to the part. The camera can move around the rounded enclosure using magnetic manipulation. The selected algorithm was the deep learning based object detection method. The analysis would be connected to a cloud database that is accessible through a web interface. This proposed solution offers a high degree of automation but is very complex as there are many moving parts. This design is illustrated in Figure 12.

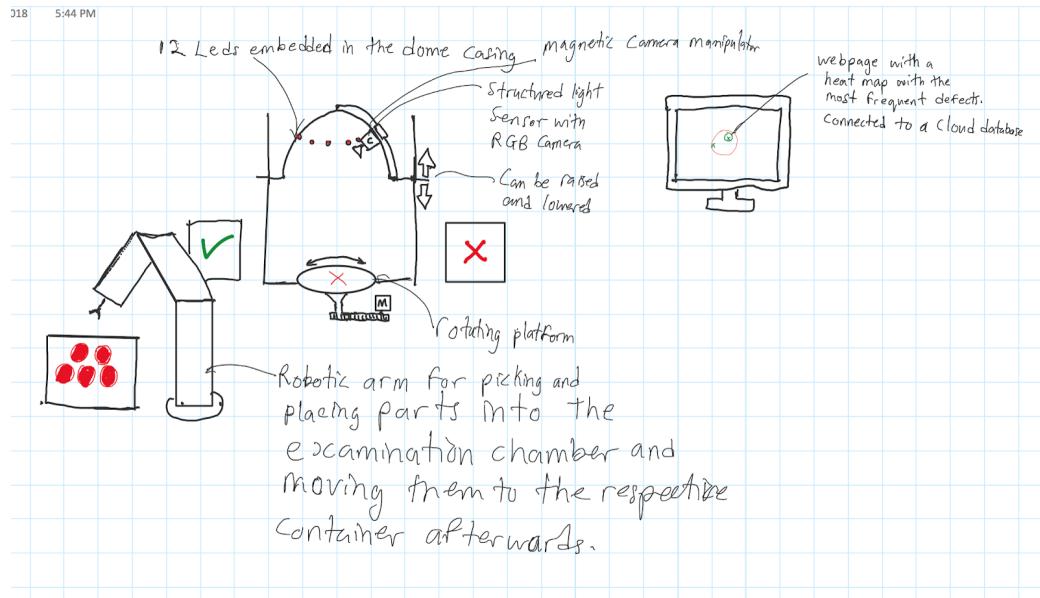


Figure 12: Sketch of the alternative design #1.

The second design is an enclosure that would sit on a conveyor belt. The parts to be inspected would be loaded into a hopper and sent individually to a dome-shaped enclosure. The camera would move around the enclosure using magnetic manipulation. The lighting would be mounted on the camera, and the chosen algorithm was the deep learning based image classification. The parts would then

Table 3: The morphological chart.

Part Loading	Lighting	Staging	Image Capturing	Scanning Mechanism	Image Processing	Interfacing
Robotic arm	LED strips	Light box	Rotating camera	Smartphone camera	Traditional classification algorithm	Report generation
Chute	Lamp	Wand	Rotating base	Depth camera	Machine learning based classification	Data visualization chart
Magnets	Sweeping light	Opaque box	Multi-cameras	Laser sensor	Object detection	Website
Manual insertion	Ambient	Magnetic		DSLR Camera	Machine based segmentation	Cloud database
Conveyor belt		Reflective container		Webcam	Traditional segmentation	Touch screen

be automatically sorted out to a good pile or bad pile. The results of the scans would be available on a tablet that would be mounted on the outside of the enclosure. This proposed solution offered a high throughput for parts, but required lots of maintenance due to the number of moving parts. This design is illustrated in Figure 13.

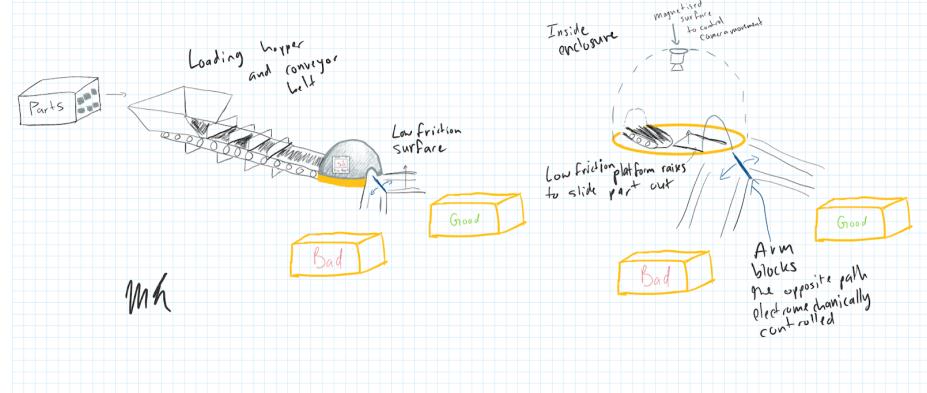


Figure 13: Sketch of the alternative design #2.

The third design also takes advantage of a robotic arm. The arm would be used to load and unload parts from the inspection, and it would also be used to manipulate the object which allows the camera to capture all surfaces of the part. The camera is stationary and placed on top of the enclosure, along with lighting. The base would be able to rotate in order to aid in capturing all surfaces of the part. The captured images would then be sent to a computer connected via USB and defects would be classified and localized using instance segmentation algorithm, and the results stored on a cloud database. This design offered a high degree of automation and a great user experience, but introduced the possibility of damaging the part during inspection. This design is illustrated in Figure 14.

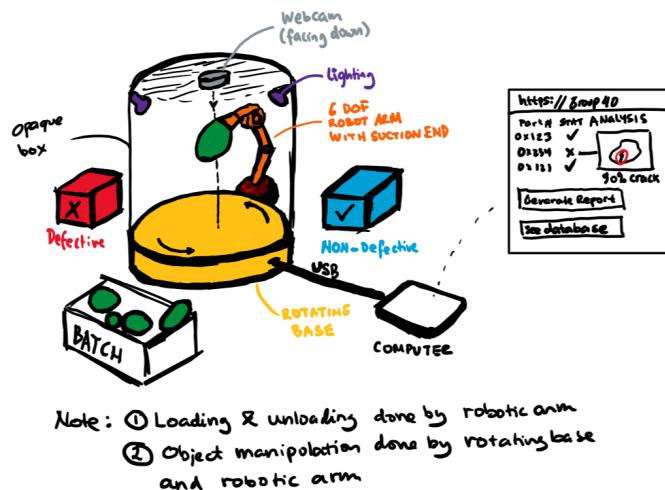


Figure 14: Sketch of the alternative design #3.

The fourth design is a wand-based scanner that would require a user to manually scan the part with a camera on a stick. The captured images would be offloaded to a computer that would classify and localize the defects using deep learning based image classification algorithm. This design is cost effective and small in size, but significantly lacks in degrees of automation. This design is illustrated in Figure 15.

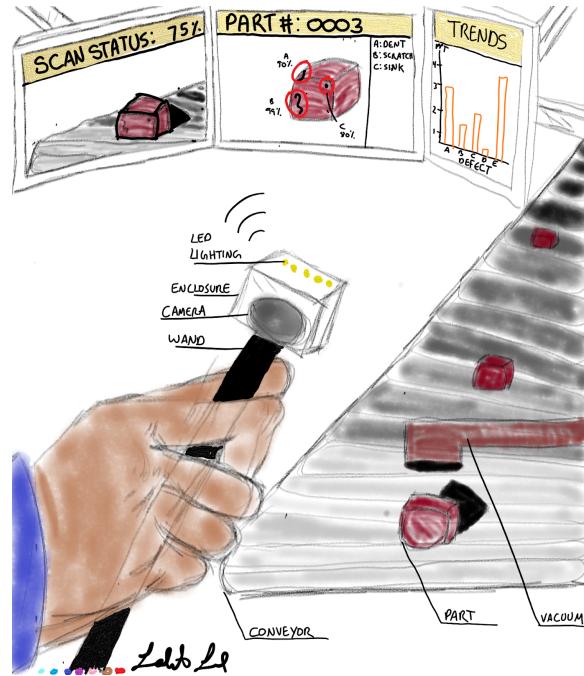


Figure 15: Sketch of the alternative design #4.

2.2 Selected Proposed Solution

The four alternative designs were evaluated using a decision matrix as provided in Table 4.

The criteria were assigned weights based on the importance of each criterion to the overall system. The most important criteria are cost, and degrees of automation. The system is meant to be a low-cost solution to match the problem statement identified in Section 1.2. The lower the cost, the better the solution as it will allow for more accessibility to smaller companies. The degrees of automation was also weighted heavily because the preferred solution will be as automated as possible. The system should be easy to set and forget, and should operate without interference to not introduce defects from operators. The other criteria - user experience, maintenance, and processing speed - were also considered as they impact the system, but to a lesser extent.

Table 4: Decision matrix for the alternative designs.

Criteria	Weightage	Design 1		Design 2		Design 3		Design 4	
		Rank	Calc.	Rank	Calc.	Rank	Calc.	Rank	Calc.
Cost	0.30	3	0.90	4	1.20	5	1.50	9	2.70
Processing Speed	0.20	8	1.60	7	1.40	7	1.40	7	1.40
User Experience	0.15	8	1.20	7	1.05	9	1.35	5	0.75
Maintenance	0.10	4	0.40	2	0.20	7	0.70	7	0.70
Degrees of Automation	0.25	9	2.25	8	2.00	9	2.25	2	0.50
		Total	6.35	Total	5.85	Total	7.20	Total	6.05

The selected proposed design is the alternative design #3. Design #3 was chosen because of the high degree of automation, and great user experience. The fully automated system allows for the operator to start up the machine, and let it run until it is finished. The data would be communicated effectively with the operators to a nearby computer that would have report generation, and give access to a database of information and previous information. Although this design will be costly with its 6 Degrees of Freedom (DoF) robot arm, this cost can be reduced by taking advantage of existing open source solutions.

3 Detailed Designs

The final design is presented in this section. First, the design proposed solution is presented and explained in Section 3.1. Each of the design subsystems were then analyzed in detail in Section 3.2.

3.1 Design Proposed Solution

Based on the preliminary designs presented in Section 2, a final design was developed. Some changes were made to the selected proposed solution (see Section 2.2) based on the discussions with Ecobee. Figure 16 illustrates the new design proposed solution.

One of the major suggestions was that the part should not be physically manipulated during inspection since it may introduce unwanted new defects during inspection process. Note that the proposed solution in Section 2.2 has a static camera and a robot arm that manipulates the pose of the object such that the camera inspect the object from multiple view angles. Instead, the new design let the inspected part to be static while having a moving camera inside the enclosure, which is possible by attaching a camera to the end of the robotic arm. The placement of the robot arm was also modified such that it hangs on the top plate of the enclosure instead of placing it side by side with the enclosure.

The other suggestion from Ecobee was to have the flexibility of incorporating the device into existing manufacturing lines, rather than constraining it only for in-house uses. Thus, the enclosure designed was changed such that one will be able to install it on top of an existing manufacturing lines (e.g., conveyor belt). Based on this, designing the loading and unloading mechanism is outside the scope of the project. This new suggestion must be considered while designing the enclosure structure with regards to modularity.

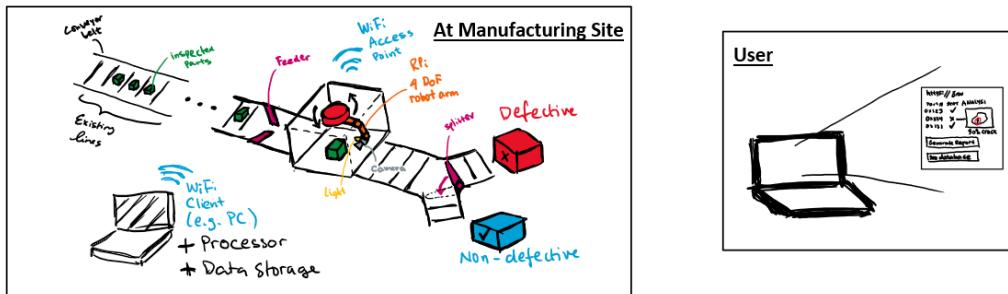


Figure 16: Sketch of the design proposed solution.

The functional goals of the proposed design include part placement (e.g., manual placement or using

existing manufacturing lines), image collection, image analysis, database management, and user interface. The systems work as follow. First, a part is placed inside an enclosure where inspection occurs. Inside the enclosure, there is a robot arm attached to the top side of the enclosure with a camera attached at the end of the arm, as well as a barcode scanner at the base of the enclosure which is used to scan part number of the inspected part. The robot arm is used to move the camera around the inspected part, allowing the camera to capture images from multiple view angles around the object. Once the images are taken, the data (e.g., part number and images) are sent to a server where they will be processed and stored. The images are then processed using a computer vision algorithm that performs defect classification and localization. The image analyses are finally stored in a shared database that users (e.g., clients or contract manufacturers) can access anytime. Users will also have the option to automatically generate a report which contains useful information and statistics about the inspected parts through an intuitive user interface.

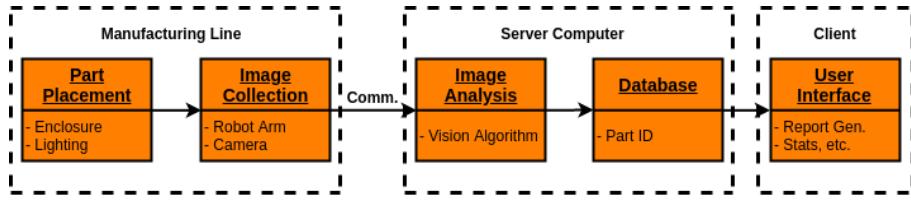


Figure 17: Illustration of the systems.

The design of the whole system is divided into several independent subsystems as shown in Figure 17. Engineering analysis of each of the subsystems are discussed in detail in Section 3.2.

3.2 Design Analysis

The design is divided into multiple independent subsystems based on the functional goals. In this section, analyses for each of the subsystems are presented. Based on the analyses, the best options for each of the subsystems were then selected.

3.2.1 Enclosure

The enclosure serves multiple purposes. Firstly, the enclosure supports the robot arm which hangs from the ceiling of the top surface. The enclosure also blocks out ambient light for consistent lighting conditions. This is important for the image capturing process and the subsequent localization algorithm. Additionally, the enclosure has openings to allow for the entry and exit of the inspected part.

When designing the enclosure, the environment and interfacing hardware must be considered. The tool may be used in-house at an engineering office or on the assembly line in a manufacturing

factory. Subsequently, the enclosure must be prepared to endure higher temperature and humidity conditions. Furthermore, the robot arm inside the enclosure requires space to move and capture all the images of the inspected part. This will constrain the dimensions of the box. Additionally, the robot arm introduces a loading condition on the enclosure, thus the enclosure must be strong enough to support this load. The enclosure was designed for either manual loading or assembly line driven loading. A nominal assembly line of 40 cm width and 10 cm height was considered. Also, as discussed in Section 3.2.4, a hook structure will be added to rest the robot arm when idle.

From the above considerations, the enclosure was designed to be 60 cm by 60 cm with a 40 cm height. The main configurations considered were a solid piece or a frame with panels. For modularity, the frame with panels configuration was preferred. One of the advantages is that it would be easier to manufacture individual pieces rather than a single solid enclosure. Additionally, if a specific component breaks, the entire enclosure would not have to be scraped.

For the assembly, the frame can be a solid piece or multiple beam elements assembled together. Manufacturing a solid frame using welding, rivets or glue is not preferred for the same reasons discussed previously with a solid enclosure design. It would also introduce significant complexity to the manufacturing process and decreased modularity. Instead, a bolted assembly was considered. This would provide complete modularity with regards to the frame and panels. If the frame needs to be longer, the beam elements can be replaced and the tool would still be valid. This will be beneficial for different assembly line dimensions. Moreover, a bolted assembly is easy to manufacture and assemble. For these reasons, a bolted frame design was selected.

Two of the configurations for the panels are using slots in the frame or bolting them to the frame. These configurations are illustrated in Figure 18. The bolted option for the panels is better for creating a stiffer enclosure structure. It also simplifies the manufacturing process since only bolt holes need to be drilled instead of slender slots. Also, in the slots option, the panels need a stopping block at the top to prevent the panel from sliding through. For these reasons, the bolted panel options was selected.

For loading the part, felt strips can be attached in an opening in two of the panels. It would be similar to the structure in car washes illustrated in Figure 19. This is beneficial since it still blocks out ambient light, does not damage the part, and works with either an assembly line or manual loading.

Given the design goals and considerations described previously, the material was determined. For the material, three options were considered for the final design: metal, wood and acrylic. Other materials like 3D printed ABS were initially considered but later disregarded for cost, manufacturing or structural reasons. Metal provides a very durable and strong material. It is also very common in industrial applications. There are abundant tools and manufacturing processes that can be used. Common structural metals like steel and aluminum are also easy to source. The standard raw mate-

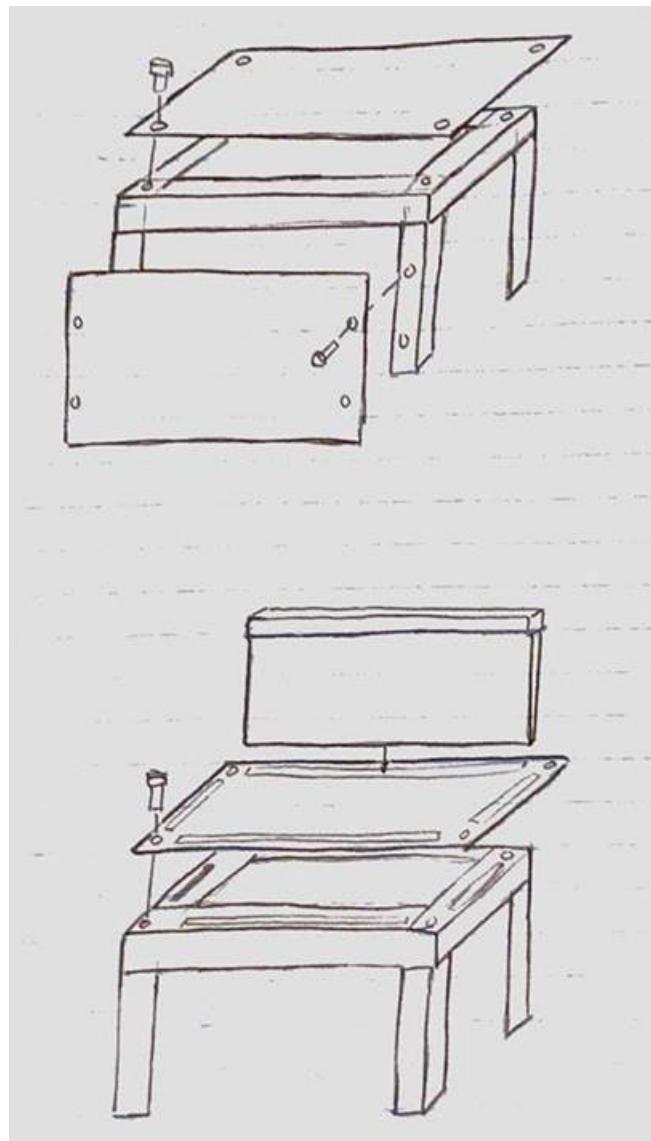


Figure 18: Bolted (top) and slotted (bottom) panel configurations.



Figure 19: Felt entry/exit design used in car washes [15].

rial comes in a variety of shapes and sizes which allows for more flexibility in the design and it is very cost effective.

Wood was another material considered. It is very easy to work with, given the abundantly available material and commonly available tools. The prototyping could be done quickly with the fast manufacturing techniques of wood. The main concern with wood is the suitability in an industrial setting. The higher temperature and humidity that may exist in manufacturing factories makes wood a poor choice for material selection. Additionally, the durability of wood overtime is also a concern, especially with a sustained load from the hanging robot arm.

The final material considered was acrylic. Acrylic is quite stiff and would work as a structural material. Acrylic typically comes in large sheets and lacks a variety of standard shapes and pieces like brackets and beams. As a result, acrylic would have to be cut from the sheet and assembled together. During prototyping, a full sheet must be bought since the available sizes and shapes are limited, even if only a small piece is required. Additionally, investigating the cost for the material indicates it would be more expensive than a metal structure. By considering the functional goals of the enclosure and factors like cost and manufacturing ease, metal was selected for the material.

For the metal design, some optimization can be applied. Angle beams in the frame were chosen to be aluminum to limit the weight. Steel was chosen for the panels for improved stiffness. The bolt and hole sizes were designed to be a 1/4 inch in diameter. This is a standard size which will limit the cost for the bolts. Additionally, a standard drill bit can be used which will simplify the manufacturing process. Moreover, the angle beam dimensions were chosen based on standard stock sizes, which will limit the manufacturing to simply cutting the beam to the correct sizes. The steel panels were also based on standard sizes and thicknesses.

During prototyping and testing of the algorithm, there may be some lights reflected from the metal enclosure. If it becomes an issue, to limit the reflected lights, a basic paint coating can be applied to the inside of the panels and frame elements. This is also beneficial for the assembly process since the paint will indicate the orientation of the components.

The design must be validated for the given loading conditions. The top panel experiences a loading from the hanging robot arm. Currently, the weight of the robot arm is 0.3182 kg. The weight introduces a relatively small force of 3.13 N especially given the material selection. Nevertheless, an Finite Element Analysis (FEA) simulation was conducted with a force of 500 N on the top plate. Since the frame is very stiff and well supported, the edges were considered to be supports. Figure 20 illustrates the analysis results. The maximum stress is 3.79 MPa and the maximum deflection is 0.013 mm. Based on these results, the plate sizes were validated for the given loading conditions. The current enclosure design can support heavier robot arm structures should it be required.

There is a sustained load on the top panel which may introduce long term fatigue. However, since

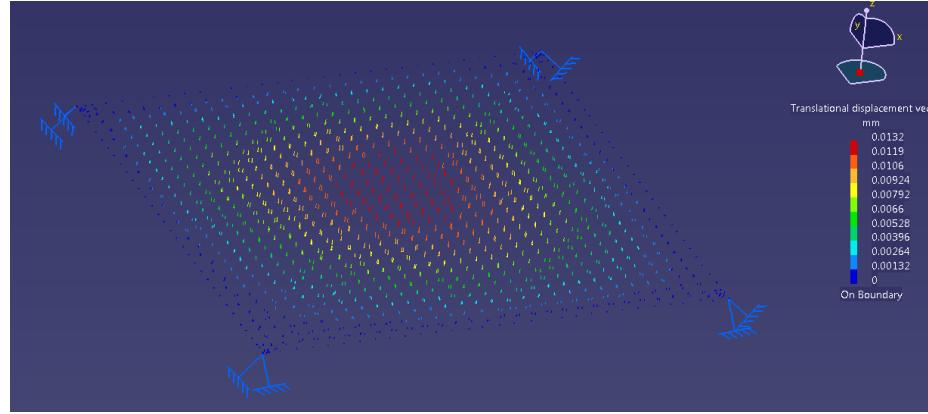


Figure 20: FEA simulation on top plate loading.

steel has an infinite life at small loadings, fatigue is not a concern. The failure modes of the bolts in the robot arm assembly are discussed in Section 3.2.4.

The final detail drawings are provided in the Appendix A. The bill of material and the cost estimation for the material are provided in Appendix B and Section 4.2, respectively. A CAD model is provided in Figure 21.

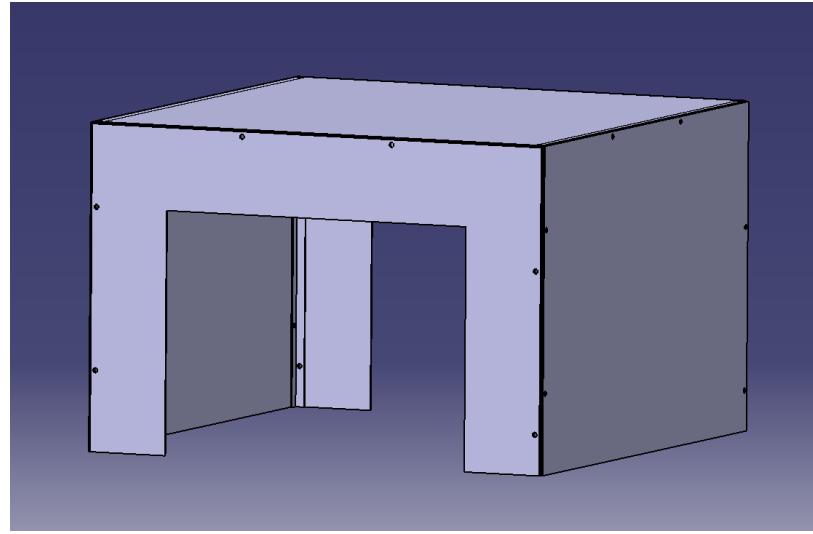


Figure 21: Enclosure CAD model. Note that the bolts and felt are not shown.

3.2.2 Camera

Several criteria were developed in order to pick the camera. The criteria include price, weight, camera settings, resolution, and field of view. Price is important because the camera is arguably one of the most expensive components in the whole system and the main goal of the project is to design a low-cost solution. Weight needs to be considered to ensure that it is not too heavy for the

robotic arm to hold (see Section 3.2.4) as it will affect the mechanical design of the support system as well as the controller design. Access to camera settings such as its exposure, white balance, and focus was also taken into consideration. Camera focus is especially important since it will affect the image quality and may constraint the placement of the camera if the camera has a fixed-focus lens (e.g. the distance between the inspected part and the camera needs to be fixed). Although many cameras offer autofocus capability, this feature is not reliable enough to perform defect detection since it does not know the exact features in the image that is of interest. Resolution of the camera is important to ensure the image quality. Finally, the field of view of the camera was also taken into consideration to ensure that the camera is able to obtain at least one of the inspected part's surfaces based on the given dimensional constraints. Various camera options were considered: Logitech C270, Logitech C525, Logitech C920, and RaspberryPi Camera.

Logitech C270 is sold at \$39.99 (CAD) retail price [16]. The weight of this camera is 75g, which is below the allowable weight (see Section 3.2.4). This camera can capture a video at 720p resolution and the field of view of the camera is at 60°. Access to camera focus is limited and has a fixed focus out of the box. The focus can be adjusted manually by opening the enclosure since the focus ring is exposed to the user. One can opt to design a system to control the focus ring with an additional small motor. This approach however is complex to implement and prone to error.

Logitech C525 is sold at \$79.99 (CAD) retail price [17]. The weight of this camera is 88g, which is below the allowable weight (see Section 3.2.4). This camera can capture a still image at 720p resolution and the field of view of the camera is at 69°. Unlike C270, this camera has autofocus feature which can be turned off and controlled electronically via software. An upgrade from C525 is the Logitech C920 which is sold at \$99.99 (CAD) retail price [18]. The weight of this camera is 160g, which may require more expensive motor if it exceeds the weight limit (see Section 3.2.4). This camera can capture a video at 1080p resolution and the field of view of the camera is at 78°. The advantage of this camera is that the access to camera focus is not limited and it comes with an autofocus feature, just like the C525. The focus can be adjusted programmatically, allowing one to change it based on the distance between the camera and the inspected part.

Finally, RaspberryPi camera can be purchased at \$32.95 (CAD) [19]. The weight of this camera is only 3g which is far below other options. This camera can capture a video at 1080p resolution with 62.2° field of view. Although the settings of this camera is more accessible compared to the Logitech C270, the focus of this camera is also fixed and can only manually be adjusted by rotating the focus ring. Similar to the Logitech C270, the focus ring can only be automatically adjusted if one designs a system to control the focus ring with an additional motor.

The summary of comparison between cameras can be found in Table 5. The main feature needed for the project is the ability to manually adjust the camera focus, which the C525 and the C920 have. This feature was implemented using Linux V4L2 API and OpenCV [20], and empirically

Table 5: Comparison of different cameras considered in this project.

	C270	C525	C920	RPi
Price (CAD)	\$39.99	\$79.99	\$99.99	\$32.95
Weight	75g	88g	160g	3g
Focus Adjustment	Not flexible	Flexible	Flexible	Not flexible
Resolution	720p	720p	1080p	1080p
Field of View	60°	69°	78°	62.2°

tested as shown in Figure 22. Although Logitech C920 is superior in terms of resolution and field of view, the C920 is more expensive and weigh twice compared to the C525. The weight is a major concern, since it constraints the design of the robot arm (e.g., the servo must be able to handle the weights), especially with the arm links. Higher resolution and high field of view are also not critical for this application. The resolution requirement was evaluated empirically on the 720p cameras: at 50cm distance between an object and camera, one pixel translates to 0.07cm in the real world measurement. This is far above the constraint of being able to detect minimum defects size of 1mm, as defined in Section 1.2. Thus, there is no strong reason to upgrade to the C920 from C525. Based on these considerations, the Logitech C525 was chosen.

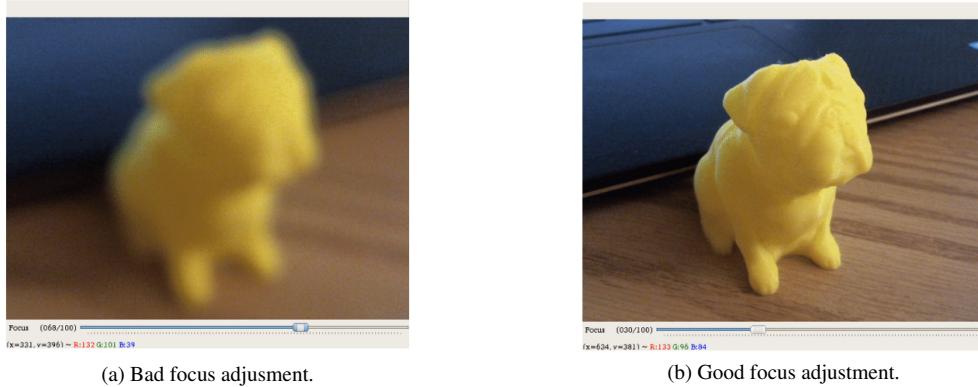


Figure 22: Evaluation of programmatic focus adjustment on Logitech C525. Note how the focus can be adjusted using the slider shown at the bottom of Figure 22a and 22b. The software to implement this feature was done using Linux V4L2 API and OpenCV.

3.2.3 Lighting

Lighting is crucial to make the defects visible to the camera. When light hits a smooth reflective surface, the light behaves predictably and travels orthogonally away from the incident ray. When the surface is not smooth, the incident ray is scattered in all directions and allows the camera to capture the defect. The placement of the lights relative to the position of the inspected part is important, since it will affect the image seen by the camera.

There are two proposed implementations for the lighting setup within the enclosure. The first is a

single light source mounted on the moving camera arm. This would work like the flash on a camera with the light source acting along the camera direction. The second proposed implementation is to have multiple lights mounted within the enclosure that are controlled by the system. The deciding factors in choosing the method for lighting within the enclosure were the cost, complexity, and effectiveness of each method.

For the lights mounted within the enclosure, multiple lights are required in the event that a defect is completely parallel to the light source which causes the defect to not be highlighted properly. The enclosure mounted lights has the advantage as they can illuminate the part from different angles. However, this option may pose a shadowing problem, especially when the arm is moving in front of the lights. As a result, one may have to perform post-processing to remove the shadows in the image. Another option to remove the shadows is by having a controller to turn off a particular light source if it detects the arm in front of the light source.

For the camera mounted lighting setup, the light would sit above the camera, and move along with it. The camera mounted lighting solution has the advantage of being cheaper because only one light source is required. The camera mounted lighting implementation is also less complex because it does not have the shadowing problem from the robot arm. As a result, this option does not require any image pre-processing or an extra controller for the light.

The camera mounted lighting was tested empirically to verify that defects would be visible in images and videos. This was performed using the selected camera in Section 3.2.2 and a small LED device would simulate the conditions of the enclosure. Based on the empirical testing, the camera mounted lighting is confirmed to still be able to make the defects visible to the camera, as long as the defects are not facing the light source directly. This is shown in Figure 23. The camera mounted lighting was chosen since it is more cost effective. Most importantly, unlike the multiple lights mounted within the enclosure, the camera mounted lighting does not produce shadowing problem.

Figure 23a and Figure 23b show the effectiveness of moving the camera around to capture defects, specifically scratches. Scratches that are parallel to the light source are difficult to see for an image processing algorithm because the reflection of the light is not very sharp. Scratches that are orthogonal to the light source refract light back into the camera and show up clearer and sharper. This makes it easier to create datasets, and also makes it easier for the algorithm to properly localize the defect.

3.2.4 Robotic Arm

Overview The selected design relies on a robotic arm to move the camera to various positions, which allows the part to be inspected from multiple view angles. The proposed arm design is a



(a) Image of part with camera mounted light.



(b) Image of part with camera mounted light with a 90° rotation.

Figure 23: The result of camera mounted lighting testing. The images captured during the 360° scan of the part allows for all the defects to be visible. The green boxes outline scratches that would be easily classified by the inspection algorithm, whereas the red boxes outline scratches that would not be easily classified.

4 Degrees of Freedom (DoF) arm driven by servo motors based on the design of the Poppy Ergo Jr. [21]. The Poppy Ergo Jr. robotic arm is a 6 DoF arm used for education and art installations. It was chosen as a baseline design because the design is open-source, and due to the fact that the servos used are *backdriveable*. Backdrivability is a feature that allows the user to specify how the robot should move via demonstration. Two other arms considered include the Little Arm Big [22] and the Niryo [23] robotic arm. These arms are not open-source and the servomotors used in each design do not allow the arm to be backdriven. The backdrivability is important for the user experience of the device operator.

Mechanical The arm was modified to be a 4 DoF instead of a 6 DoF arm. Based on the design of the enclosure (see Section 3.2.1), it was determined that a 4 DoF arm is sufficient to reach all points of interest around the inspected part. The arm would be mounted on the top plate of the enclosure to simplify the path planning since the configuration space of the arm is symmetric from the top of the enclosure. The 4 DoF include a ball and socket joint at the mounting point on the enclosure, an elbow joint between two plastic links, and another elbow joint attached to the camera as shown in Figure 24.

Theoretically, a 2 DoF arm with 2 elbow joints and an end effector can reach any points in 2D space within the radius of its reach, where the radius is the sum of the arm links length. However, an additional elbow joint attached to the end effector allows more freedom to the camera to be oriented in any position. Finally, since the arm operates in 3D space and the analysis of the degrees of freedom has been within a 2D space, rotating the arm about the base elbow joint combines with the base socket joint to form a ball joint allows the arm to reach any point in 3D space.

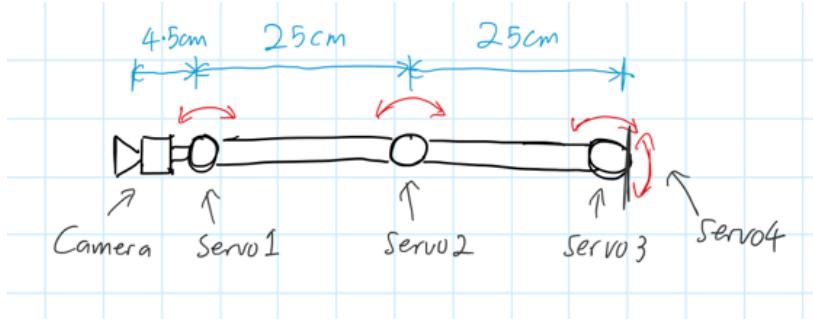


Figure 24: Sketch of the robotic arm with dimensions.

The lengths of the arm links, dubbed the *humerus* and the *radius*, were determined based on the maximum width of the part to be inspected. The maximum width of the inspected part was determined to be 10cm. The maximum height for the object to be inspected was 20cm. An additional 10cm buffer volume around the object was added to give extra room for the camera to adjust its focus. With a maximum object height and width of 30cm and 20cm, respectively, and the chosen height of the box to be at 40cm, the length of each of the arm links was determined to be 25cm.

Based on the lengths of the arm links, the motors to actuate the joints were then selected. To select the motors, the weights of the links, motors, and the camera were considered. The minimum torque required was then calculated using a torque calculation tool [24]. The calculation suggested that the servo motors in the Poppy Ergo Jr. arm are not sufficient to move the entire system. The calculation was done over different servomotor combinations based on the final torque requirements as shown in Figure 25. In Figure 25, T1, T2 and T3 corresponds to the torques in the shoulder, elbow, and wrist joint, respectively. Furthermore, A1, A2 and A3 correspond to the mass of the motors at each joint, while M2 and M3 correspond to the mass of the humerus and radius arm links. The lengths of each of the links are denoted by L1 and L2, respectively. The motors selected include 3 Dynamixel AX-12A Smart Serial Servos for the ball and elbow joints, and 1 Dynamixel XL-320 OLLO Smart Serial Servo for the wrist joint where the camera is attached.

L:	[cm]	M:	[kg]	A:	[kg]	T:	[kg cm]
L1:	4.5	M1:	0.088	A1:	0.0164	T1:	0.2718
L2:	25	M2:	0.04	A2:	0.0546	T2:	4.74679999
L3:	25	M3:	0.04	A3:	0.0546	T3:	11.5868

Figure 25: Calculation of the final torque required at each joint.

The most convenient choice for prototyping and validating the robotic arm is by 3D printing the parts. The considered 3D printing material for the links is the Polylactic Acid (PLA), which is commonly used in 3D printing. Next, the design of the robotic links and bolt selection were considered.

Some considerations were taken when designing the links and validating the mounting bolts. First, there was a concern for the mounting bolts to sustain the weight of the robotic arm when fixed to the top plate of the enclosure. This is because there is a potential for the weight of the robotic arm to be too heavy for the bolts which may cause a thread strip failure mode. The second concern was within the robotic links due to two possible areas of failure. One failure mode is shear failure in the bolts that mount the motors to the robotic arm. If the weight of the links and actuators are too heavy at the mount points, the bolts may experience shear failure. The second mode of failure is tear out stress at the mount points of the robotic links, which defines the minimum cross-sectional thickness of the mounts.

Since the robot arm will be mounted to the top plate of the enclosure, the bolts must be sufficiently strong to hold the weight of the robotic arm in both static and dynamic conditions. This is illustrated visually in Figure 26. The expected stress on the bolts in the longitudinal direction was calculated by

$$\sigma_{threadstrip} = \frac{F_{applied}}{4A_{perpendicular}} = \frac{M_{arm}g}{4\pi r_{bolt}^2}, \quad (1)$$

where $F_{applied}$ denotes the force due to the weight of the arm, $A_{perpendicular}$ denotes the area perpendicular to the tensile force direction, M_{arm} denotes the mass of the arm, g denotes gravity, and r_{bolt} denotes the radius of the bolts. There is a dividing factor of 4 since the load is distributed across 4 identical bolts.

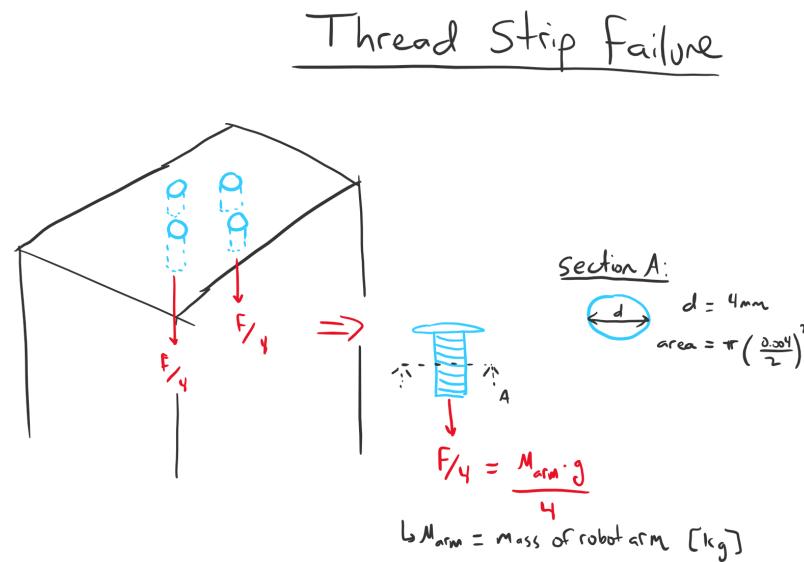


Figure 26: Illustration of the thread strip failure mode. The factor of 4 in the denominator represents the 4 bolts that are used to distribute the load, as dictated by the mount pattern of the actuators themselves.

The masses for all components except the links were determined using the manufacturer datasheets and are presented in Table 6. The mass of the links may change depending on the cross section area of the links, thus a conservative mass was chosen to determine the stress on the bolts and the mounts. This mass was determined based on the density of PLA which was found to be $1.25 \frac{g}{cm^3}$ [25]. Based on the calculated dimensions and assuming the 3D printing is done at 25% infill, a candidate model was developed in SolidWorks. The mass of each link was calculated by multiplying the sourced density with the volume of the designed part in SolidWorks.

Table 6: Masses of individual robotic arm components

Component	Mass [Kg]
Dynamixel AX-12A Smart Servo (x3)	0.0546
Dynamixel XL-320 OLLO Smart Servo (x1)	0.0164
Arm links (x3)	0.040
Camera	0.088
Laser range finder	0.010
Total mass	0.3182

$$M_{link} = 0.25\rho V_{link} = 0.25 * 1.25 \frac{g}{cm^3} * 150cm^3 = 37.5g. \quad (2)$$

Based on the calculated M_{link} , the mass of the links was conservatively assumed to be 40 grams to ensure that the motors can hold the links without reaching the stall torque loads.

The loading on the bolts on the Dynamixel AX-12A Smart Servo was also considered. The radius of the bolts used for the Dynamixel AX-12A Smart Servo are $2mm$, which are made of press hardened steel (PHS). However, the property of a regular steel was conservatively used as a safety measure, since PHS is stronger than regular steel. The final piece of information is the baseline strength of hardened steel to validate the requirements are well below the bolt capabilities. The tensile strength of the motor mounting bolts were found to be 350MPa [26]. The total strength required from the bolts mounted to the top plate of the enclosure were calculated by

$$\sigma_{ts} = \frac{M_{arm}g}{4\pi r_{bolt}^2} = \frac{0.3182 * 9.81}{4\pi 0.002^2} = 62.101kPa \leq 350MPa. \quad (3)$$

Therefore, the bolts that mount the robotic arm to the enclosure will not fail from thread strip.

The second area of concern, as previously mentioned, was the design of the mounting links, which introduce two failure modes at each link: bolt shear and mount tear out failure. Bolt shear occurs when the vertical load of the robotic arm along the cross-sectional area of the bolts is greater than the yield stress of the bolt. The worst case scenario was assessed to validate the design. The maximum stress is experienced at the joints where the bolts attach the base of the robot arm to the top plate of

the enclosure. Bolt shear failure is illustrated in Figure 27.

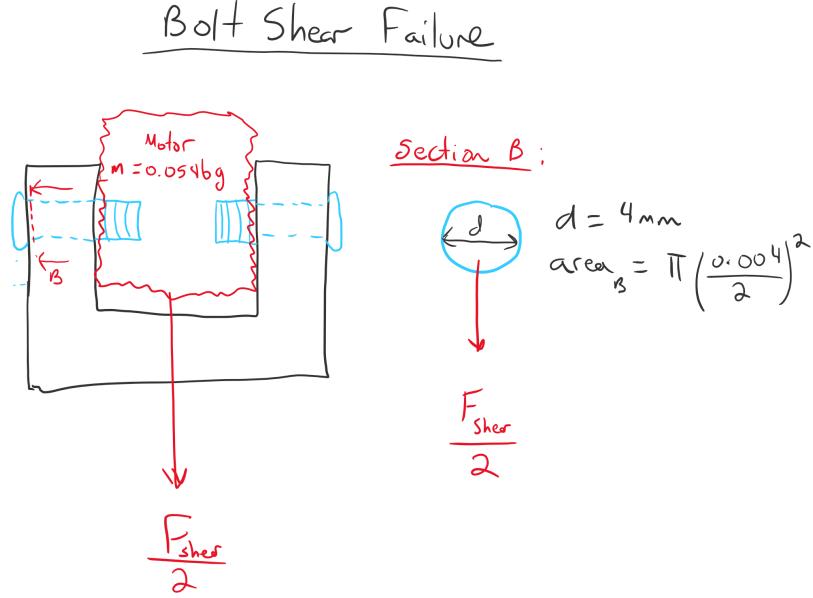


Figure 27: Illustration of bolt shear failure mode.

The governing equation for this mode of failure is not much different from the thread strip failure mode. In fact, the only change is the surface along which the load is acting. The force is applied along the cross-sectional area of the bolts rather than perpendicular to it. The second difference is the mount pattern. Since there are four bolts per side of the mount, the load is symmetrically divided in half, where each component of the force is distributed between four bolts. Therefore, the shear stress was calculated by

$$\tau_{\text{shear}} = \frac{F_{\text{Applied}}/2}{4A_{\text{shear}}} = \frac{M_{\text{arm}}g/2}{4\pi r_{\text{bolt}}^2}. \quad (4)$$

The expected shear force on the bolts was then determined to be

$$\tau_{\text{shear}} = \frac{M_{\text{arm}}g/2}{4\pi r_{\text{bolt}}^2} = \frac{0.3182 * 9.81/2}{4\pi 0.002^2} = 31.050\text{kPa} \leq 344.7\text{MPa}. \quad (5)$$

The expected shear is compared to the nominal shear stress of hardened steel (344.7MPa [27]). Thus the calculation shown in Equation 5 validates that there will be no shear failure of the bolts at the mount points.

Finally, the second mode of failure in the robotic link mounts is the tear out failure, which determines the design of the mount thickness. The tear out failure mode is illustrated in Figure 28 where t denotes the cross-sectional thickness of the mount and L denotes the length of the cross sectional

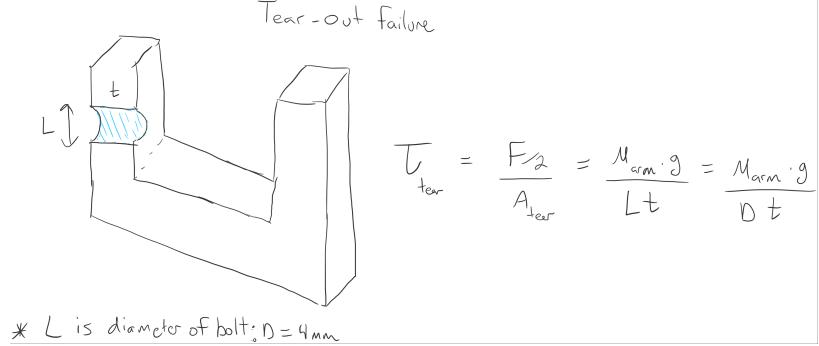


Figure 28: Sketch of tear out failure mode.

area, which is equivalent to the diameter of the bolt of 4mm. The tear out stress can be calculated by

$$\tau_{tear} = \frac{F_{applied}/2}{A_{tear}} = \frac{M_{arm}g/2}{Lt} \leq \tau_{PLA}. \quad (6)$$

Note that the force was considered over the cross sectional area of the mount rather than the bolt, since the bolts are no longer of concern in tear out failure mode.

Using the properties of PLA, Equation 6 can be rearranged to determine the minimum thickness of the mounts required to sustain the applied force. Since the yield stress of PLA is nominally known to be around 9MPa [25], the minimum thickness was determined to be

$$t \geq \frac{M_{arm}g/2}{D\tau_{PLA}} = \frac{M_{arm}g/2}{D\tau_{PLA}} = \frac{0.3182 * 9.81/2}{0.004 * 9 * 10^6} = 0.0433\text{mm}. \quad (7)$$

In other words, to be safe from tear out stress in a maximum loading scenario, the minimum mount thickness needs to be greater than 0.0433 mm. The current design considers a mount thickness of 10mm, which provides a tear out stress safety factor of $10/0.0433 = 230$.

These designs ensure that there will be no mechanical loading failure from the robotic arm, either from mounting it to the top plate of the enclosure, or the mount links themselves. Figure 29 illustrates the envisioned robotic arm mounted to a mock-up enclosure. Technically drawings for the robotic arm are provided in Appendix A.

Electrical This section entails the high level electrical design involved in the robotic arm. This includes a high-level system overview on the various components involved. Table 7 provides the electrical components used in the robotic arm.



Figure 29: Mock-up of robotic arm with camera attached.

Table 7: Electrical components used in the robotic arm.

Component	Name	Power [W]
Actuator Type 1 (x1)	Dynamixel XL-320 OLLO	4.44 (7.4V, 0.6A)
Actuator Type 2 (x3)	Dynamixel AX-12A	16.7 (11.1V, 1.5A)
Control Module	RaspberryPi 3 Model B+	6 (5V, 1.2A)
Servo Shield	Dynamixl CM-5, XL-320 OLLO	33 (11.1V, 3A)
Camera	Logitech C525	1.5
Laser Range Finder	ST Microelectronics VL53L0X	0.02
Buck Converter (X3)	SMAKN DC-DC 5A	1
LCD Display	PiTFT Plus	0.220
Power Supply	12V, 100W	N/A
Total:		98W

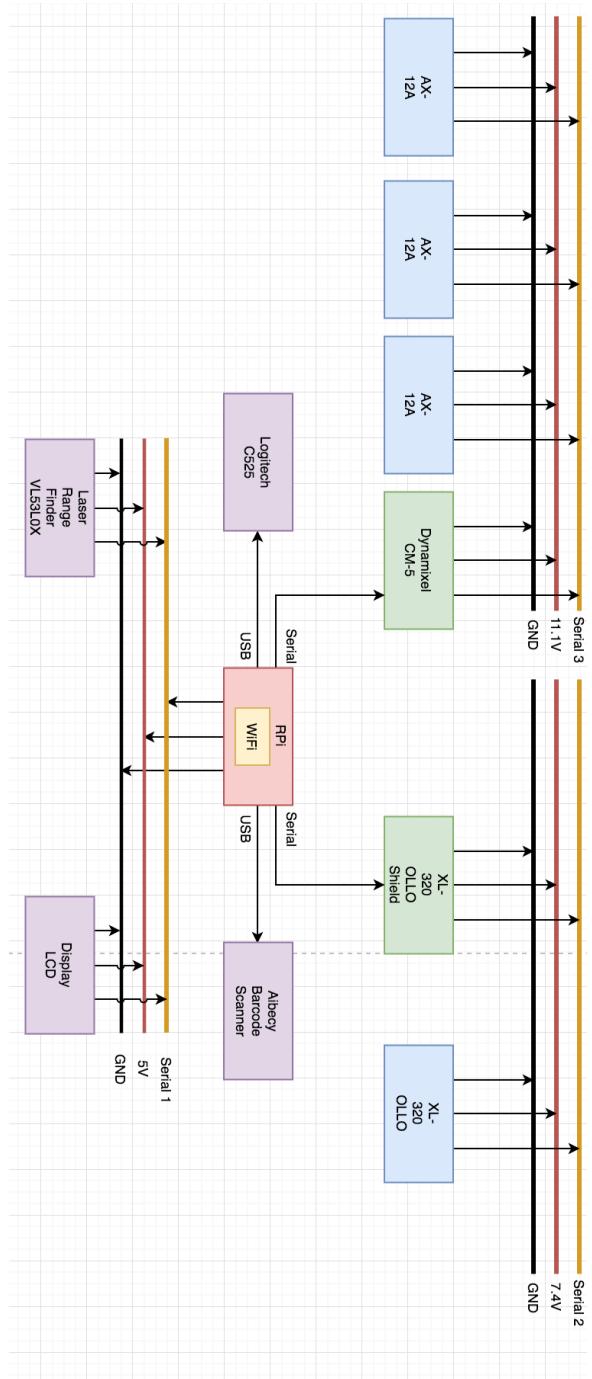


Figure 30: High level illustration of the electrical architecture.

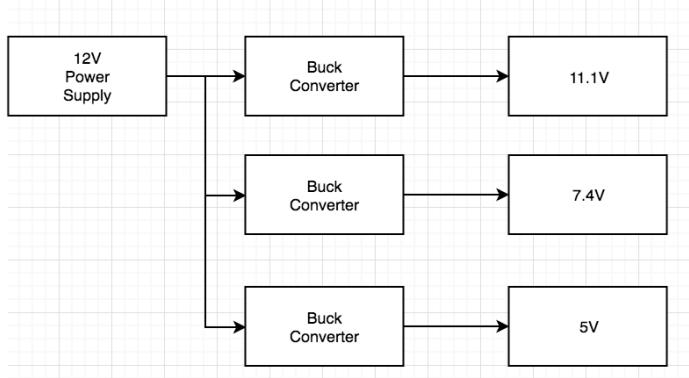


Figure 31: Illustration of the power distribution system.

This electrical system architecture shown in Figure 30 and Figure 31 allow for understanding of how the components communicate to one another and how the power is distributed. For the power distribution, a switching regulator (i.e., buck converter) was used instead of a linear regulator since it can output different voltages with a fixed input voltage. Furthermore, the chosen buck converter can handle the maximum current requirements of 3A.

Additionally, the linear regulator was not used in the power distribution since the resistors cannot tolerate the current requirements of the system (3A from the motors alone), resulting in them burning out from the excess heat and current draw. Finally, the linear switching performance degrades overtime as the resistors will vary in their values while operating at relatively extreme power requirements. For these reasons, the power distribution was designed using switching regulators.

The Dynamixel AX-12A series of motors was chosen for three out of the four motors to satisfy two main requirements. The first constraint is the nominal range of torque that the robotic arm needs to apply on the actuators, while the second constraint is the number of DoF which allows for a more flexible configuration space of the robotic arm to scan the part from various view angles. The other actuator, the Dynamixel XL-320 OLLO series, was chosen to be the actuator for rotating the camera, since there is a smaller torque requirement and not much rotation will be needed on the camera. This brand of servos was chosen because in addition to being renowned for its reliability, both products are smart serial servos. This means that the servos support back-driveability feature, daisy-chained communication to simplify design, and the ability to receive and transmit various parameters over the serial bus. For these reasons, the motor selections were justified within a reasonable price range.

Since the servos are smart servos, they require serial communication as opposed to a simple PWM signal for controlling standard servos. Therefore, it is common for smart servos to require some form of proprietary serial communication wrapped around a basic serial protocol such as RS-485 to convert the logic to a digital signal. Consequently, most smart servos series have a range of compatible shields that are capable of converting a serial command from a control module (e.g., RaspberryPi) to an appropriately packaged, proprietary serial command for the on-board controller

in the smart servos. For this reason, two Dynamixel shields were used to communicate with the AX-12A and XL-320 OLLO series motors. Two shields were required since the two types of motors operate at different voltage levels (11.1V and 7.4V).

The laser scanner and LCD display were chosen based on similar criteria: size, weight, cost, reliability, ease of interfacing with the RaspberryPi module, and power consumption. Both entities are able to communicate with the control module (i.e., RaspberryPi) on the same serial communication bus as seen in Figure 30.

As outlined in Table 7, the maximum power consumption of all electrical components in the design was determined to be 100W at 12V. Therefore, the power supply was specified to be capable of handling 12V and 100W. A power supply capable of delivering 12V at 8.5A was sourced to allow for a safety margin. It is expected that the system only requires a constant power consumption in the range of 40-80 watts. Thus, the sourced power supply should be well capable of providing this.

Software The software comprises of three functionalities as illustrated in Figure 32. The goal of the software is mainly to control the movement of the robot arm. This includes obtaining goal configuration for camera, planning a feasible path using inverse kinematic, and tracking the patch by controlling the joint angles.



Figure 32: The main functionalities of the software.

The arm operates in two modes: goal position tracking mode and camera pose acquisition mode. Goal position tracking mode is the mode during normal operation where the arm is moving the camera to different positions to acquire images of the part. In camera pose acquisition mode, the arm can be moved by the operator to view a feature of interest on a part. The configuration of the camera, its position and orientation in space, is recorded and added to a queue of camera configurations that are iterated through during a scan. This mode relies on the servos in the arm being backdriveable.

To keep track of where camera should be moved to, a queue of goal configurations is stored. These goal configurations specify where the camera should be. Software on-board the RaspberryPi takes the goal points from the queue and plans a path from the arm's current position to the goal position while avoiding obstacles in the environment. To perform this planning, a kinematic model of the arm will be developed and an inverse kinematic technique will be used to determine the joint angles required to put the camera into the specified configuration. When the trajectory has been obtained for each joint, it is output to each joint and software on the servo is used to track the trajectory.

3.2.5 Inspection Algorithm

Various computer vision algorithms to perform defect detection were considered during the design process. In particular, machine learning based vision algorithms were particularly of interest due to its generalization capability (i.e., the ability to perform inference on unseen data), which is essential in designing a part-agnostic manufacturing defect detection system. Furthermore, machine learning based computer vision algorithms, particularly those leveraging deep learning [28, 29] models, have shown state of the art performance in various vision tasks such as classification [30, 31, 32], object tracking [33, 34, 35], object detection [36, 37, 38], image segmentation [39, 40, 41, 1], and many more. The algorithms considered to perform defect detection include image classification, object detection, semantic segmentation, and instance segmentation algorithms. Meta learning algorithms, class of machine learning algorithms which allow one to train machine learning models with few examples, were also considered as an additional feature. The use of meta learning algorithms may be beneficial due to the fact that the amount of defective parts available to train the model may be limited in real world scenario or expensive to obtain.

Image classification is a task where a model is responsible to assign a class to an image of interest with some probability values. In the case of deep learning models, classification can be learned by training a model with fixed output classes on available datasets such as the ImageNet dataset [42] using backpropagation [43]. Classification-based method has been shown to be applicable in detecting manufacturing defects [8] as discussed in Section 1.3.2. Although the implementation of classification models is relatively simple, it has a limitation in localizing important features in the input (e.g., pixels in an input image) that drives the models' decision. Several methods have been developed in order to extend classification models' capability in performing object localization such as Class Activation Mapping (CAM) [44] and Gradient-weighted Class Activation Mapping (Grad-CAM) [45]. Grad-CAM, for example, is possible since deep learning models are typically differentiable, which allows one to calculate the input Jacobian of the model by performing backpropagation (i.e., by calculating the gradients of each output node with respect to the input). Albeit the localization performance can be used to indicate the rough location of the object in the image, it is not precise enough to quantify the size of the defects. Furthermore, calculating the input Jacobian via backpropagation are computationally expensive which will negatively impact the overall processing speed.

Object detection is a task where a model is required to classify and localize as many objects as possible within an image. In object detection, the location of the objects are represented as bounding boxes. In other words, given an input image, the model outputs several predictions of object classes with their corresponding locations, where each bounding box is represented by its top-left and bottom-right corner coordinates. Deep learning methods such as R-CNN [36], Fast R-CNN [37], Faster R-CNN [38], You Only Look Once (YOLO) [46, 47, 48], and Single Shot Multibox Detector (SSD) [11] are currently the state of the art methods in performing object detection task. YOLO

variants and SSD in particular, have shown promising detection performance at incredible processing speed which allows them to be used in real time applications, thanks to the advancement in lightweight neural networks models like the MobileNets [12]. Furthermore, variant of SSD powered by MobileNet has been successfully applied in defect detection application [10] as discussed in Section 1.3.2 (see Figure 9).

Semantic segmentation is a vision task to classify every pixel in the input image. Segmentation produces more refined object localization since it operates at the pixel-level detail. However semantic segmentation model is not capable of detecting different instances of objects that come from the same class. For example, if there are two persons in an image, semantic segmentation model will label all the pixels that belong to both persons as “person”, but the model does not know which of these pixels belong to the first or second person. As a result, semantic segmentation methods are not ideal to perform defect detection since it does not allow one to measure the size of each defect in case of multiple defects. On the other hand, instance segmentation methods are not only required to classify each pixel, but also to determine every object instance in the image. One may think of instance segmentation as a combination between object detection and semantic segmentation, which makes it ideal to perform defect detection. Furthermore, an instance segmentation model, Mask R-CNN [1], has been shown to be capable of detecting manufacturing defects [13] as discussed in Section 1.3.2 (see Figure 10). Creation of segmentation dataset required to train the model however is expensive to obtain since one needs to label every single pixel in the image. One may be able to compensate for this by creating simulated dataset or by using meta-learning approach, which a machine learning method to learn from few examples. However, using simulated dataset does not guarantee the model to work well when detecting real world defects since the statistics of the simulated and real world images may be different. Meta-learning methods such as the variants of Model Agnostic Meta Learning (MAML) [2] may be beneficial, but the application of MAML in segmentation models is still in its infancy.

Table 8: Comparison of various inspection algorithms considered in this project (“High” is the best).

	Classification	Object Detection	Semantic Seg.	Instance Seg.
Accuracy	High	High	High	High
Localization	Low	Medium	Medium	High
Speed	Low	High	High	High
Data requirement	High	High	Low	Low

Based on these analysis, the algorithm to perform defect detection was chosen. The comparison between methods is summarized in Table 8. The criteria used to determine the ideal algorithm for defect detection task were accuracy, defect localization capability, processing speed, and data requirement. Although instance segmentation should theoretically produce best performance, the cost of obtaining the training dataset to achieve the best performance is high and time consuming. On the other hand, the dataset to train object detection models is easy to obtain. Although the

localization performance of object detection is less superior, the overall performance would still be acceptable. Therefore, object detection model (i.e. SSD + MobileNets) was chosen to build the minimum viable product. If time permits, instance segmentation model should be implemented.

A preliminary object detection model based on SSD and MobileNets was implemented to evaluate the feasibility of the chosen algorithm. This was implemented in Python using the TensorFlow library [49]. Following the work in [11], the model was trained to optimize for both localization loss (e.g., smooth L_1 loss [37]) and classification loss (e.g., cross-entropy loss). Although far from perfect, the preliminary model was concluded to be able to detect scratches as shown in Figure 33. This was expected due to fact that the dataset used to train the model was small and does not reflect the actual condition of how the inspection will be done (e.g., inside the enclosure). However, the performance of the algorithm is expected to improve significantly as the dataset gets larger and reflects the actual testing condition.

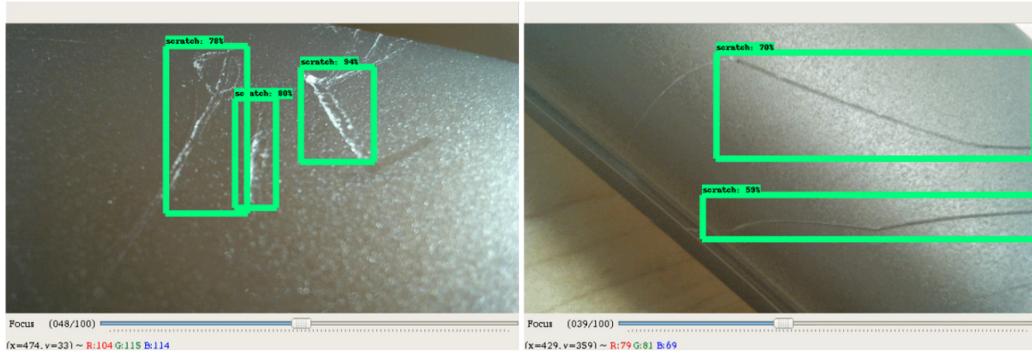


Figure 33: Examples of scratch detection on a plastic part. The model is based on MobileNets-SSD and trained on small amount of random pictures of scratched objects acquired from Google search.

3.2.6 Part Identification

A function of the proposed solution involves the ability to identify unique parts that are passed through it for visual inspection. Particularly, this function should be easily integrated with existing manufacturing processes for part identification. Through consultation with industry partners, it was noticed that typical forms of part identification involves a QR code, barcode, a serial number, or some combination of these three options. Two viable solutions for part identification involve camera-based detection or a keyboard-input laser scan.

Typical keyboard-input laser scanners are accurate and capable of scanning barcodes and QR codes alike. Additionally, since the scanner is form independent (QR code or barcode), there can be support for multiple identifications on each part because of the accuracy of a laser, and a back-end software can be used to parse each input for the appropriate ID. This solution is simple, yet requires additional hardware of a keyboard input barcode scanner, with the additional overhead of software

to parse the inputs and activate the scanner during appropriate times. A potential hurdle with this solution is limited scanning range, depending on where the barcode is placed. Also, if it is fixed or movable, the capabilities and complexity of this implementation will vary. Since most parts have their ID labels on the bottom, a stationary barcode scanner on the underside is a reasonable assumption.

The other option includes using image processing and computer vision techniques on the obtained pictures to parse barcodes and QR codes. The benefits of this approach include no additional hardware (only the camera that is already in the solution) which reduces the hardware costs. However, there is added complexity of software that is needed to process and analyze images for identification.

Another solution used in practice involve using communication protocols to communicate with the on-board electronics of the device and extract its manufacturing information (serial number, state, part type, etc). This alternative is not relevant for the proposed solution since the visual defect tool is not intended to attach to electronics in order to support part independence.

Due to the insignificant additional cost of a barcode scanner yet and simple software implementation of parsing the keyboard inputs of the scanner, the barcode scanner method will be pursued. The remainder of this section describes the specifications of the barcode scanner.

To maintain a low cost, a low form factor, and some degree of modularity, off the shelf commercial barcode scanners will be considered in this solution rather than any custom hardware. Commercial barcode scanners are available in a variety of form-factors, the most common being a hand-held device. Many solutions differ in how they integrate with systems which includes bluetooth, wifi, or serial (USB). For simplicity and reliability factors, a USB solution will be considered. The following portion outlines the selection process of USB barcode scanners. The main factors involve cost, size, accuracy, scan method, interface method, and scanning flexibility (auto, manual). Table 9 provides the comparison between various commercial embedded barcode scanners. Based on these comparisons, Symcode MJ-2090-L was selected due to its small form factor and low price.

Table 9: Comparison of commercial embedded barcode scanners.

	Aibecy EP-3000	Symcode MJ-2090-L	Aibecy RYS9880091261517UB
Cost	High	Low	Medium
Size	Big	Small	Small
Scan Method	CCD	CCD	CCD
Interface	USB/Serial	USB/RS232	USB/Serial
Control	Man/Auto	Man/Auto	Man/Auto

3.2.7 Data Communication

Communication between the sensors on the robot arm (e.g., camera and laser scanner) and a computer where the data is processed and stored was considered. Data communication between the robot arm and the server computer is needed since the data processing and management are to be done by the server.

As discussed in Section 3.2.4, the robot arm is powered by a RaspberryPi 3 Model B+ which has USB ports, Wi-Fi, and bluetooth. The criteria to choose between these three options were bit-rate, and user interface. The bit-rate was considered to ensure that the image can be transferred and processed within the specified time frame. The user interface was considered to make the entire system easier to operate.

Since the images do not need to be processed in realtime, the data transmission speed requirement is more relaxed. Instead one can shift the focus from processing performance to user interface and experience. Due to this consideration, the use of USB as a means to transmit data is not required, while the use of wireless options such as Wi-Fi and bluetooth are preferable. Wi-Fi was concluded to be a more appropriate option due to its higher bit-rate compared to bluetooth considering the data needs to be transferred include camera images. Finally, this pipeline will be implemented using Robot Operating System (ROS) [50] to simplify multithreading management and interprocess communication. Figure 34 illustrates the data communication pipeline.

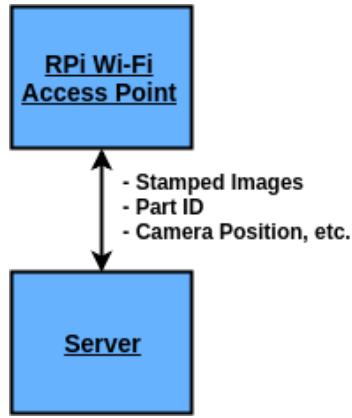


Figure 34: Illustration of the data communication pipeline.

3.2.8 User Interface and Experience

The user interfaces are designed to enable the target users, the device operator working in the factory and the product quality engineer, to get relevant information about product quality and the device operation. The interface facing the operator provides instructions on how to operate the device, communicates daily defect trends, and allows the user to view the live camera feed from the

manufacturing site. The interface on the engineers' side allows for individual device reports to be fetched, provides summarized reports for each production run, and plots trends in defects based on given criteria. The operator can interact with the device through a physical touchscreen connected to the device, while the quality engineers can interact through a web interface remotely.

Operator Interface Two options were investigated on the operator interface. These options include: a touchscreen connected to the device and a web interface. The criteria used in making this decision were cost and ease of use. Here, cost refers to the monetary cost of the option while ease of use refers to how easy it would be for the operator to interact with the device in a factory setting. Ease of use was ranked higher compared to cost.

Although the touch screen would cost significantly more than a free webpage, touch screen was selected since it is easier to use on the factory floor. Since the touch screen is attached to the inspection device, it eliminates the chance for the operator to connect to the wrong device, especially in situations where there are multiple inspection devices on the same network. The PiTFT Plus Assembled 320x240 2.8 inch Resistive Touchscreen [51] was chosen as shown with a mockup of the interface in Figure 35. This is due to the fact that it is the most affordable Raspberry Pi compatible touchscreen and due to its small form factor.

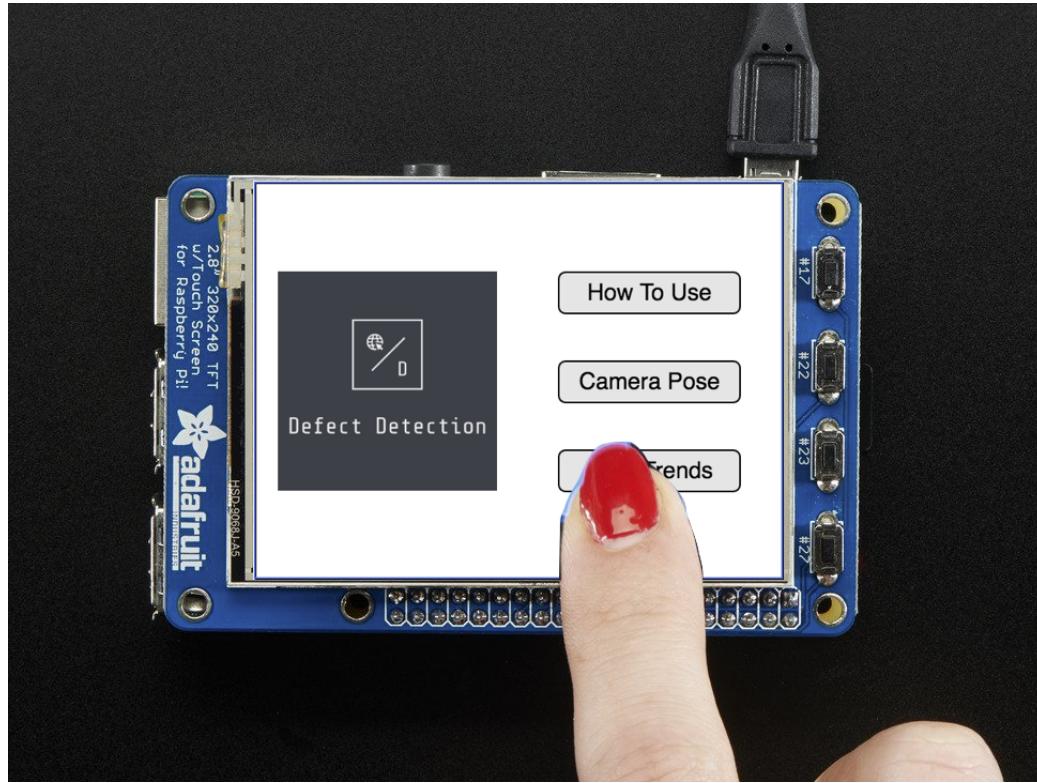


Figure 35: Illustration of the touch screen with the user interface.

Through the touch screen based user interface, a device operator is also able to learn how to op-

erate the device, acquires the camera poses for the inspection, and view trends of defects during production. The camera pose acquisition mode is used to specify which part of the inspected part should be inspected if necessary. In this mode, the operator will be allowed to move the robot arm to a desired position, see the camera output at that position, and record the robot movement to be reproduced. This is where backdrivability comes into play using the selected motors. The low effort required to backdrive the servos is made possible due to the internal friction within the gearbox in the servomotors, and the moment generated due to the lengths of the humerus and the radius. The screen also shows what percentage of inspected parts have passed or failed the tests in each day, as well as the most frequent defects.

Quality Engineer Interface A web-based interface for the quality engineers was chosen, since it allows them to see the results remotely without having to travel to the manufacturing plant. Through the provided interface, the engineer can find individual reports containing similar metrics as shown in Appendix C. This can be done using natural search queries as shown in Figure 36. With the provided information (e.g. trends plot, inspection report), the engineer can be more efficient in spotting issues in production runs, and can narrow them down to specific station, operator, or factory where certain defects are more prevalent.

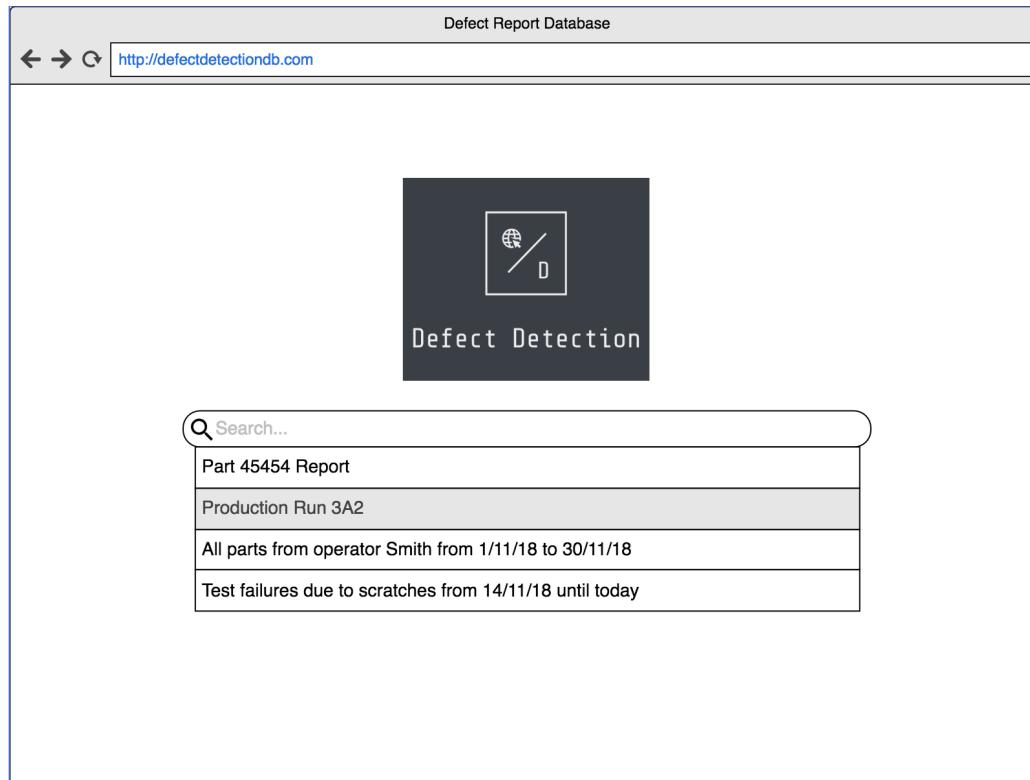


Figure 36: Illustration of the database query interface.

3.3 Design Review

In order to qualify the proposed design, the design was compared to the project constraints and criteria. The constraints and criteria are summarized in Table 2.

The first constraint is with regards to the cost. The expected total cost for the project is \$987.02 CAD which meets the maximum cost of \$1,000 CAD (see Section 4.2). The bill of material is presented in Appendix B.

The scanning system consists of a robotic arm with a mounted camera. The arm will be able to rotate and position itself around the part and capture various angles of the part surfaces without physical part manipulation. This satisfy the constraint of being able to inspect at least class A and B viewing zones. Additionally, the proposed design also work for inspecting parts with a bottom flat surface.

The chosen algorithm (i.e., MobileNets-SSD) is expected to meet the minimum detection rate and processing speed constraints. This analysis is supported by the results shown in [10], which used the same algorithm and achieved 95% detection rate on glass jar sealing surface defects and processing speed of 120ms per image on NVIDIA GeForce Titan X GPU.

The design also offers a superior user experience with an intuitive user interface which considers both the operator and the quality engineer. This includes a back-driveable robotic arm that allows users to calibrate the scanning points, increasing the flexibility of the system to scan parts of various shapes. Finally, the interface also allows the users to generate report which contains the inspection results and trends related to the inspection process, which increases efficiency to spot issues in production runs.

The proposed design provides low maintenance. The maintenance criteria takes into consideration the number of moving parts, the requirement of part replacement, and the cost of these replacements. The various incorporated off-the-shelf parts are highly supported in the electronics community through developers and manufacturers. This makes the troubleshooting of the components easier.

Finally, the selected design has a high degree of automation. The 4 DoF robotic arm can scan various angles of the part on an automated cycle. The hardware and software combination requires minimal human intervention, apart from calibration of the scan points on the first run. After the initial assembly and calibration, the system is intended to operate autonomously.

Overall, the proposed design meets the specified constraints and is optimized to excel with regards to the criteria.

4 Schedule and Budget

4.1 Schedule

A simplified Gantt chart in Figure 37 depicts the expected dates for various high level tasks. The tasks are grouped into five subsections that depict the overall goal of a time frame. These include prototyping, first build, second build, final build, and testing. A miscellaneous group is also created for things that are not related to a component of the system, such as report writing and presentation preparation for the symposium.

The prototyping stage of the schedule has already commenced with dataset creation. The manufacturing of the robot arm and enclosure is scheduled to start in late December. Once the robot arm is completed, the first build of the system will commence. This constitutes the development of the software to control the robot arm to scan parts. The lighting conditions will need to be validated and implemented, and because of this the defect detection algorithm will need some refinement in order to match the data received by the camera. The second build has a focus on the user interface, the scanning mechanism, and the enclosure. In this build the end goal is to have a working user interface in place, and to have a working prototype of the robot arm mounted in the enclosure. Once this is in place, the final build will work towards integrating all the components together to match the block diagram in Figure 17.

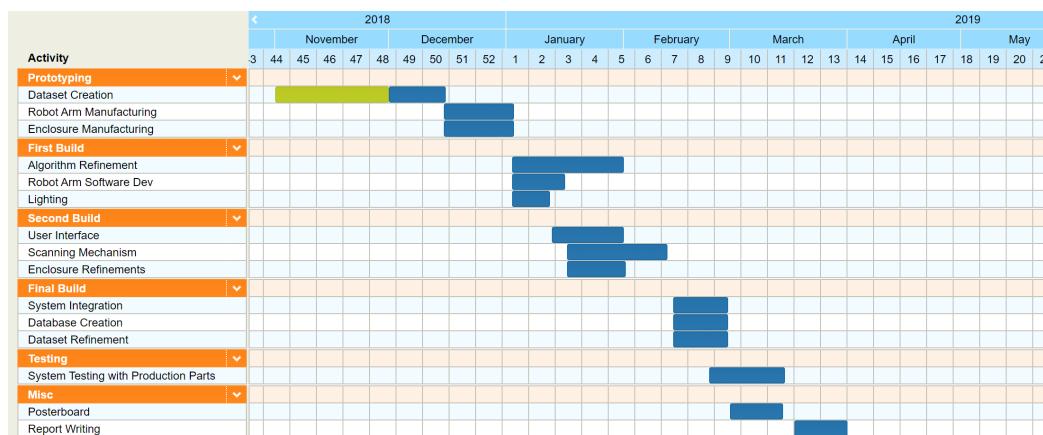


Figure 37: Project timeline.

4.2 Budget

The detailed line-by-line budget is provided in the Bill of Material given in Appendix B. The total budget stands at \$987 CAD, just under the constraint of \$1000 CAD. The two biggest expenses come from the enclosure, budgeted at \$250 CAD, and the robot arm, budgeted at \$450 CAD. The

other hardware costs come from the camera, barcode scanner and circuitry components.

The budget was developed conservatively using retail values. Efforts will be made to find cost effective sources which will reduce the overall cost. If significant cost savings can be achieved with a similar component, a change order will be completed. As part of the change order, substation will be conducted to ensure the new component meets the requirements for the system.

5 Conclusions and Recommendations

There is a need for low-cost solution for high quality manufacturing inspection, especially for small and medium sized companies who have limited resources. In this report, a design for low-cost intelligent manufacturing inspection tool was presented in order to fill this market gap. This project focuses on detecting visual defects which includes scuffs, scratches, dents, and discolorations. The functional goals of the project include part placement (e.g. manual placement or using existing manufacturing lines), image collection, image analysis, database management, and user interface.

An enclosure with a robot arm that hangs at the top of the enclosure was presented. A camera is attached at the end of the robot arm together with lighting, which allows the camera to move around and capture images of the inspected parts from different positions without physically touching the inspected parts. Once images are taken, they are transferred to a server computer where the data will be stored and processed with the vision algorithm. The analysis of each of the inspected parts can then be accessed by the users through a user friendly interface. Based on the analysis and design review (see Section 3.3), the proposed design was determined to meet the specified constraints and criteria.

Should time permit, the use of instance segmentation algorithm such as the Mask R-CNN [1] is recommended to increase localization performance which will be beneficial to the overall system. Investigation on meta-learning algorithms such as the Model Agnostic Meta Learning (MAML) [2] will also be interesting since it reduces the amount of data needed to train the model. Furthermore, the use of MAML allows for rapid fine-tuning of the model during testing which can increase the performance of the model without having to spend significant time to retrain the model.

6 Teamwork Effort

Each individual on the team was responsible for different aspects of the design and analysis process. The team worked very well with each other through regular meetings. The team was able to divide the work and resolve issues quickly as they arose. Every member demonstrated a willingness to support others to meet the scheduled deadlines.

Herman Grewal was responsible for designing the enclosure structure, drafting engineering drawings, developing a bill of material and finalizing the budget. Herman aided Lalit Lal to investigate failure modes in the mechanical design of the robotic arm. Herman also supported Mohamed El Shatshat with the needs analysis and project scheduling.

Lalit Lal was responsible for designing and specifying the electrical system architecture, sourcing electronics, and co-designing the mechanical components of the robotic arm with Oluwatoni Ogunmade. Lalit was also responsible to specify the part needed for the part identification functionality of the system, as well as its integration.

Mohamed El Shatshat was responsible for testing the lighting conditions of the system, and assisting Herman Grewal with the design of the enclosure. Mohamed was also in charge of developing the constraints and criteria, and outlining the schedule for the upcoming work.

Oluwatoni Ogunmade was responsible for outlining the design of the user interface, reviewing patents for similar solutions and reviewing the electrical design of the system. Oluwatoni and Lalit Lal worked on designing and analyzing the robotic arm used to actuate the camera.

Rey Reza Wiyatno was responsible for investigating state of the art methods for defect detection, camera selection and testing, inspection algorithm and its implementation, as well as data communication.

Acknowledgement

The authors thank Kevin Eng (Ecobee), Aleksey Tsaplin (Ecobee), and Wim Deweerdt (Google Hardware) for validating the problem, as well as identification of constraints and criteria. The authors also thank Ecobee and OpenROV for providing production parts with defects to validate the proposed system. Finally, the authors thank Professor Ayman El-Hag and Professor Jan Huissoon for the supervision and guidance throughout the term.

References

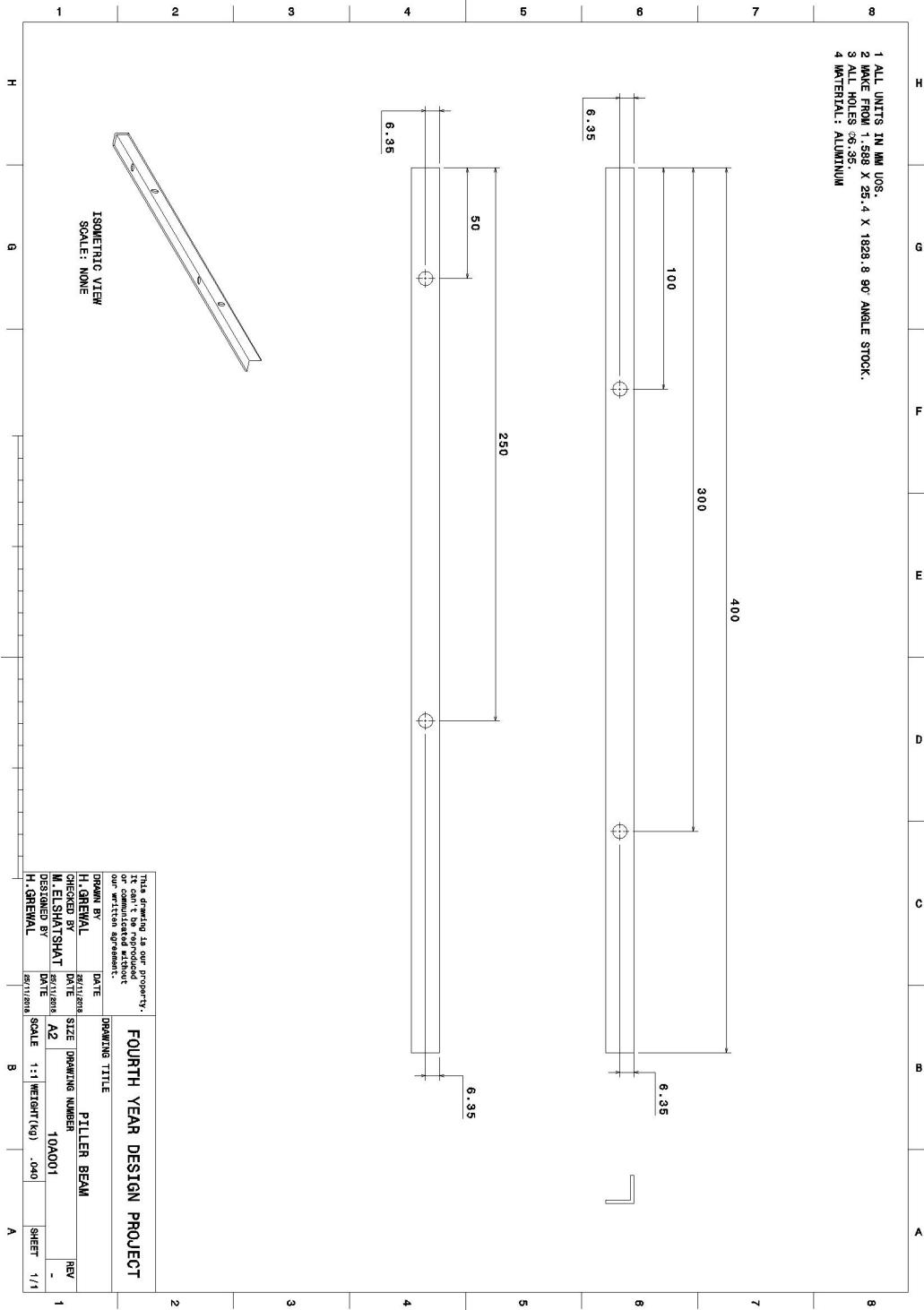
- [1] Kaiming He et al. “Mask R-CNN”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017.
- [2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 1126–1135. URL: <http://proceedings.mlr.press/v70/finn17a.html>.
- [3] GE Energy Connection. *Cosmetic Inspection Guidelines for Mechanical Components*. 2018. URL: <https://www.geenergyconnections.com/sites/geem/files/GA-SRC-0002%20Cosmetic%20Inspection%20Guidelines%20for%20Mechanical%20Components%20%28rev%201.0%29.pdf> (visited on 11/25/2018).
- [4] J. V. Earwage. private communication. Oct. 2018.
- [5] Clark Humphrey and Brian Morris. “System and method for automated cosmetic inspection”. Pat. 2014.
- [6] Samuel Bruce Weiss and Anna-Katrina Sheldetsky. “Modular optical inspection station”. Pat. 2016.
- [7] Ravishankar Rao Stuart L. Riley Martin C. Sturzenbecker Virginia H. BrecherChou Paul B.-L Robert W. HallDebra M. Parisi. “Automated defect classification system”. Pat. 1993.
- [8] Ruifang Ye et al. “Intelligent defect classification system based on deep learning”. In: *Advances in Mechanical Engineering* 10.3 (2018), p. 1687814018766682. DOI: 10.1177/1687814018766682. eprint: <https://doi.org/10.1177/1687814018766682>. URL: <https://doi.org/10.1177/1687814018766682>.
- [9] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision -- ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 818–833. ISBN: 978-3-319-10590-1.
- [10] Yiting Li et al. “Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD”. In: *Applied Sciences* 8.9 (2018). ISSN: 2076-3417. DOI: 10.3390/app8091678. URL: <http://www.mdpi.com/2076-3417/8/9/1678>.
- [11] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *Computer Vision -- ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 21–37. ISBN: 978-3-319-46448-0.
- [12] Andrew G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *CoRR* abs/1704.04861 (2017). arXiv: 1704.04861. URL: <http://arxiv.org/abs/1704.04861>.
- [13] Max Ferguson et al. “Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning”. In: *CoRR* abs/1808.02518 (2018). arXiv: 1808.02518. URL: <http://arxiv.org/abs/1808.02518>.
- [14] Domingo Mery et al. “GDXray: The Database of X-ray Images for Nondestructive Testing”. In: *Journal of Nondestructive Evaluation* 34.4 (2015), pp. 1–12. URL: <http://dmery.sitios.ing.uc.cl/Prints/ISI-Journals/2015-JNDE-GDXray.pdf>.
- [15] Tom Harrison. *Tesla Model 3 LLCar Wash Mode Needed*. 2018. URL: <https://tomharrisonjr.com/tesla-model-3-car-wash-mode-needed-4f59a7da76c2> (visited on 11/25/2018).
- [16] Logitech. *Logitech C270 HD Webcam*. 2018. URL: <https://www.logitech.com/en-ca/product/hd-webcam-c270> (visited on 11/27/2018).

- [17] Logitech. *Logitech C525 HD Webcam*. 2018. URL: <https://www.logitech.com/en-ca/product/hd-webcam-c525> (visited on 11/27/2018).
- [18] Logitech. *Logitech C920 HD PRO Webcam*. 2018. URL: <https://www.logitech.com/en-ca/product/hd-pro-webcam-c920> (visited on 11/27/2018).
- [19] Raspberry Pi Foundation. *Camera Module V2*. 2018. URL: <https://www.raspberrypi.org/products/camera-module-v2/> (visited on 11/27/2018).
- [20] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).
- [21] Poppy. *ERGO JR: A LOW COST ROBOT FOR EDUCATION*. 2018. URL: <https://www.poppy-project.org/en/robots/poppy-ergo-jr> (visited on 11/26/2018).
- [22] LittleArm. *LittleArm Big*. 2018. URL: <https://www.littlearmrobot.com/littlearmbig.html> (visited on 11/27/2018).
- [23] Niryo. *Niryo - Smart Collaborative Robot For You*. 2018. URL: <https://niryo.com/> (visited on 11/27/2018).
- [24] Robotshop. *Robot Arm Torque Calculator*. 2018. URL: <https://www.robotshop.com/community/tutorials/show/robot-arm-torque-calculator> (visited on 11/26/2018).
- [25] Prospector. *Polylactic Acid (PLA) Typical Properties*. 2018. URL: <https://plastics.ulprospector.com/generics/34/c/t/polylactic-acid-pla-properties-processing> (visited on 11/27/2018).
- [26] MatWeb. *Steel, General Properties*. 2018. URL: http://www.matweb.com/search/datasheet.aspx?bassnum=MS0001&fbclid=IwAR1WC-S4dGQyyM_iDTLQtw8U40tIv0DF_0DBDAZXYup-103jAMQ20eSYLc0&ckck=1 (visited on 11/27/2018).
- [27] Unipunch. *Shear Strength Metal Specifications*. 2018. URL: <https://www.unipunch.com/support/charts/material-specifications/> (visited on 11/27/2018).
- [28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521 (2015), 436 EP –. URL: <http://dx.doi.org/10.1038/nature14539>.
- [29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. URL: <http://dl.acm.org/citation.cfm?id=2999134.2999257>.
- [31] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.
- [32] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993 (2016). arXiv: 1608.06993. URL: <http://arxiv.org/abs/1608.06993>.
- [33] David Held, Sebastian Thrun, and Silvio Savarese. “Learning to Track at 100 FPS with Deep Regression Networks”. In: *European Conference Computer Vision (ECCV)*. 2016.
- [34] Luca Bertinetto et al. “Fully-Convolutional Siamese Networks for Object Tracking”. In: *ECCV 2016 Workshops*. 2016, pp. 850–865.
- [35] Jack Valmadre et al. “End-To-End Representation Learning for Correlation Filter Based Tracking”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- [36] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.
- [37] Ross Girshick. “Fast R-CNN”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2015.

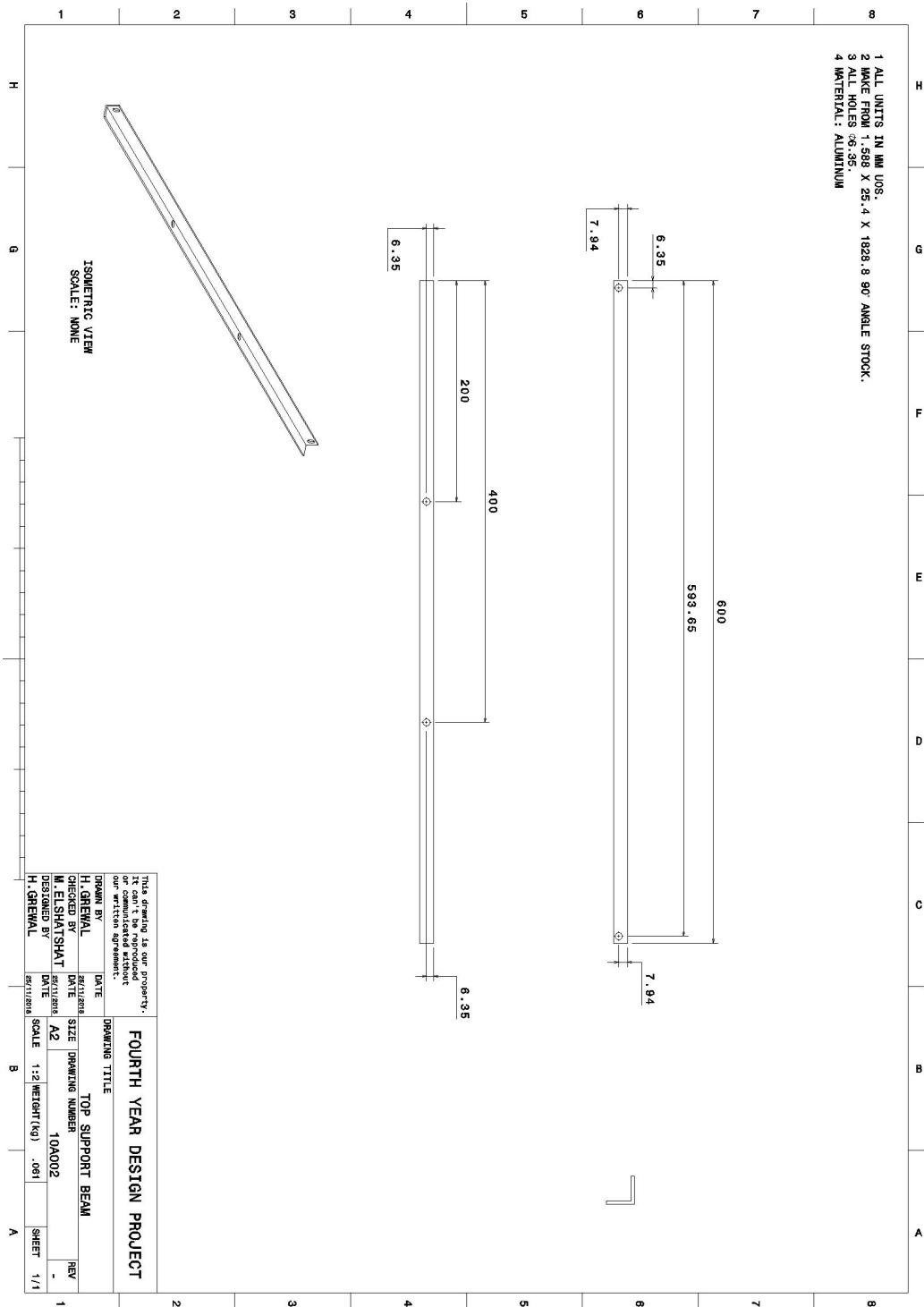
- [38] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 91–99. URL: <http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>.
- [39] Evan Shelhamer, Jonathan Long, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.4 (Apr. 2017), pp. 640–651. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2572683. URL: <https://doi.org/10.1109/TPAMI.2016.2572683>.
- [40] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Vol. 9351. LNCS. (available on arXiv:1505.04597 [cs.CV]). Springer, 2015, pp. 234–241. URL: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>.
- [41] Simon Jégou et al. “The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation”. In: *CoRR* abs/1611.09326 (2016). arXiv: 1611.09326. URL: <http://arxiv.org/abs/1611.09326>.
- [42] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009.
- [43] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), 533 EP –. URL: <http://dx.doi.org/10.1038/323533a0>.
- [44] B. Zhou et al. “Learning Deep Features for Discriminative Localization.” In: *CVPR* (2016).
- [45] R. R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.
- [46] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91. URL: <https://doi.org/10.1109/CVPR.2016.91>.
- [47] Joseph Redmon and Ali Farhadi. “YOLO9000: Better, Faster, Stronger”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690. URL: <https://doi.org/10.1109/CVPR.2017.690>.
- [48] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: <http://arxiv.org/abs/1804.02767>.
- [49] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [50] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software*. 2009.
- [51] Adafruit. *PiTFT Plus Assembled 320x240 2.8” TFT + Resistive Touchscreen*. 2018. URL: <https://www.adafruit.com/product/2298> (visited on 11/27/2018).
- [52] Home Depot. *Paulin 1/16X1X6 ft. Aluminum Angle*. 2018. URL: https://www.homedepot.ca/en/home/p.116x1x6-ft-aluminum-angle.1000861546.html?fbclid=IwAR3fcRnKYKKgSF6EbY82MsBvc9VG6Cn2LfD01DtvX_TbVzeYLt_fckILZI (visited on 11/27/2018).
- [53] Home Depot. *Paulin 24X48 22Ga Steel Sheet*. 2018. URL: https://www.homedepot.ca/en/home/p.24x48-22ga-steel-sheet.1000861562.html?fbclid=IwAR1n0piTwTBfGGnp_MgNj7K6S1m1i2TxvtSEJ2Awlr0sT37aI4rGS6kWy7k (visited on 11/27/2018).

- [54] Canada Bolts. *Hex Bolt, Full Thread, Stainless Steel, 1/4"-20x1 1/4"*. 2018. URL: <https://canadabolts.ca/collections/imperial-hex-bolt-full-thread-stainless-steel/products/is1441-1721> (visited on 11/27/2018).
- [55] Canada Bolts. *Flange Nuts Serrated, Stainless Steel, 1/4"-20*. 2018. URL: <https://canadabolts.ca/collections/nuts-imperial/products/flange-nuts-serrated-stainless-steel-1-4-20> (visited on 11/27/2018).
- [56] Michaels. *Creatology Felt Sheet Pack*. 2018. URL: <https://www.michaels.com/creatology-felt-sheet-pack/10224639.html#start=3> (visited on 11/27/2018).
- [57] Pololu. *VL53L0X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 200cm Max*. 2018. URL: <https://www.pololu.com/product/2490> (visited on 11/27/2018).
- [58] Gearbest. *Symcode MJ2090 USB Auto Sense Laser 1D Barcode Scanner - BLACK*. 2018. URL: https://www.gearbest.com/scanners/pp_557446.html (visited on 11/27/2018).
- [59] Amazon. *AMZ3D 1.75mm PLA 3D Printer Filament, Black, 1 Kg spool (2.2 lbs)*. 2018. URL: <https://www.amazon.ca/AMZ3D-1-75mm-Printer-Filament-Black/dp/B01BZ5ND80/> (visited on 11/27/2018).
- [60] RobotShop. *Dynamixel XL-320 OLLO Smart Serial Servo*. 2018. URL: <https://www.robotshop.com/ca/en/dynamixel-xl-320-ollo-smart-serial-servo.html> (visited on 11/27/2018).
- [61] RobotShop. *Dynamixel AX-12A Smart Serial Servo*. 2018. URL: <https://www.robotshop.com/ca/en/dynamixel-ax-12a-smart-servo-serial.html> (visited on 11/27/2018).
- [62] Robotis. *CM-5*. 2018. URL: <http://www.robotis.us/cm-5/> (visited on 11/27/2018).
- [63] Adafruit. *Raspberry Pi 3 - Model B+ - 1.4GHz Cortex-A53 with 1GB RAM*. 2018. URL: <https://www.adafruit.com/product/3775> (visited on 11/27/2018).
- [64] Amazon. *SMAKN DC-DC 5A Automatic Boost Buck Converter 1.25-20V Adjustable Car Regulated Power Supply*. 2018. URL: <https://www.amazon.com/Automatic-Converter-1-25-20V-Adjustable-Regulated/dp/B00K6ARMX2> (visited on 11/27/2018).
- [65] Robotis. *S1 Bolt PHS M2x6 (400 pcs)*. 2018. URL: <http://www.robotis.us/s1-bolt-phs-m2x6-400-pcs/> (visited on 11/27/2018).
- [66] Bearing Wholesale Lots. *6mm Inner Diameter Bearings*. 2018. URL: <https://www.bearingwholesalelots.com/category-s/2346.htm> (visited on 11/27/2018).
- [67] Amazon. *AmpFlow S-100-12 100W, 8.5A, 12V DC Power Supply*. 2018. URL: <https://www.amazon.com/AmpFlow-S-100-12-100W-Power-Supply/dp/B00AL3II60> (visited on 11/27/2018).
- [68] Amazon. *NeoPixel 16 LED Ring with Integrated driver*. 2018. URL: https://www.amazon.ca/NeoPixel-LED-Ring-Integrated-driver/dp/B00UW5X7PS/ref=sr_1_4?ie=UTF8&qid=1543431407&sr=8-4&keywords=neopixel (visited on 11/27/2018).

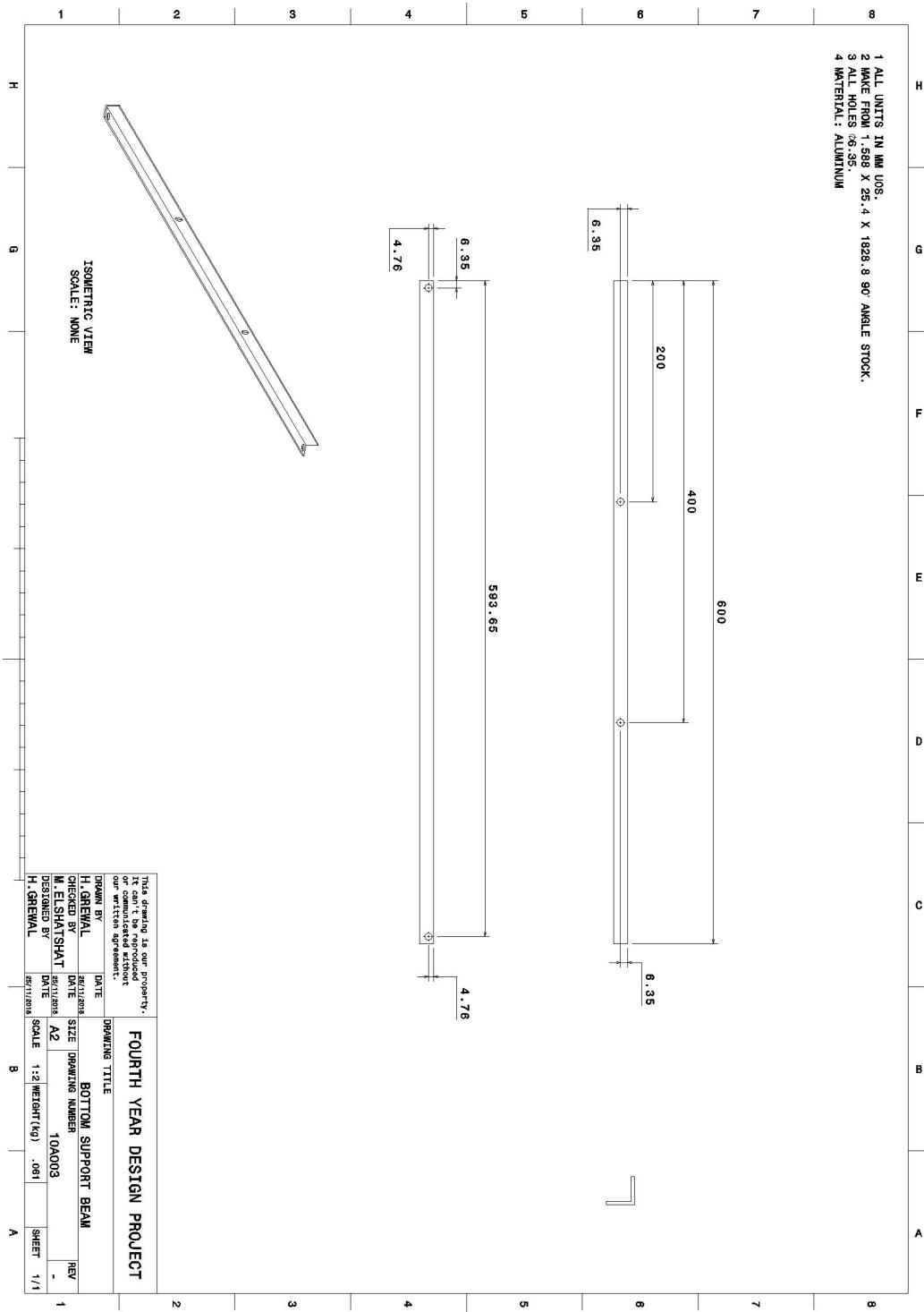
Appendix A Detailed Part Drawings



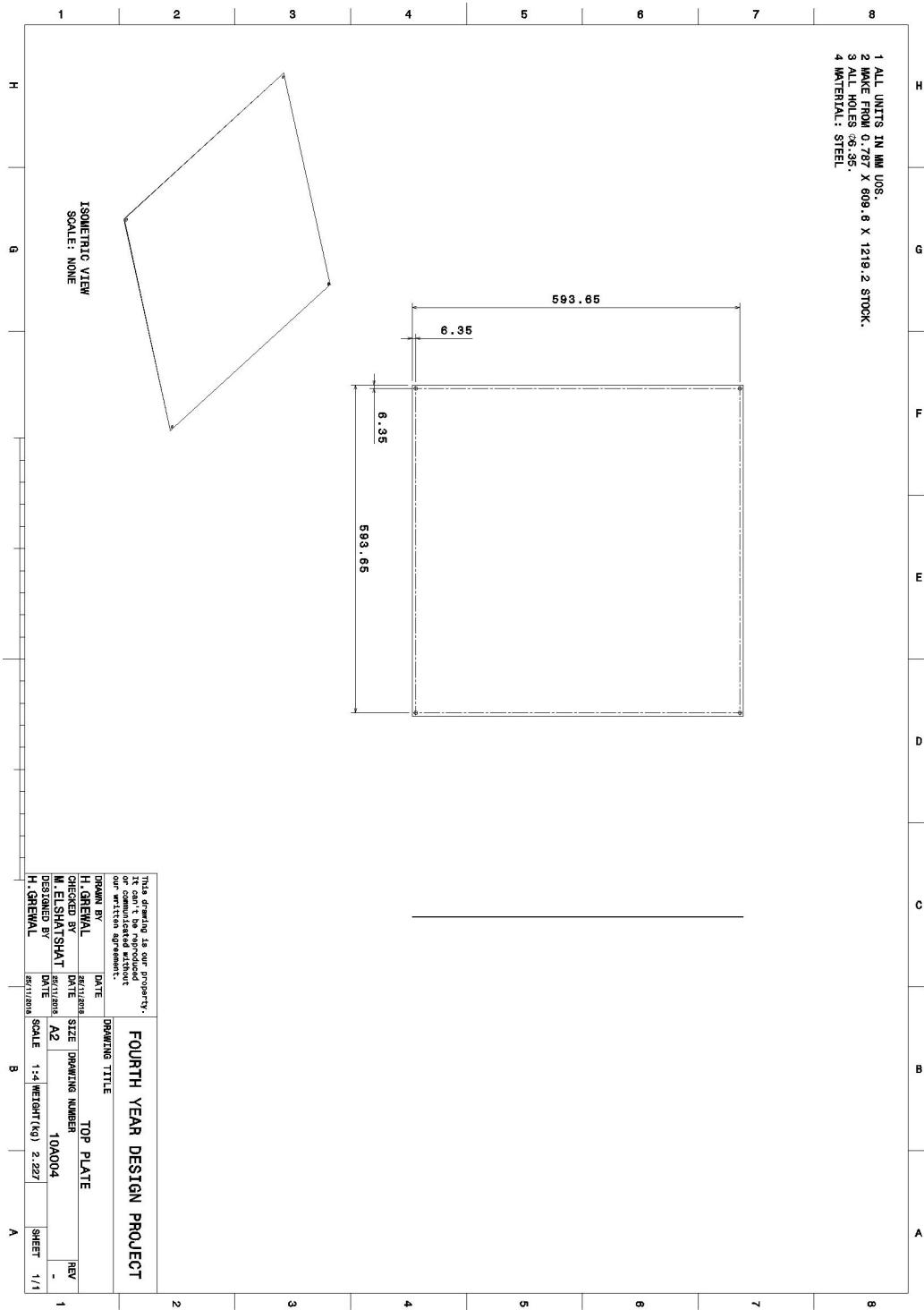
1 ALL UNITS IN MM UDS.
 2 MADE FROM 1.568 X 25.4 X 1828.8 90° ANGLE STOCK.
 3 ALL HOLES Ø6.35
 4 MATERIAL: ALUMINUM

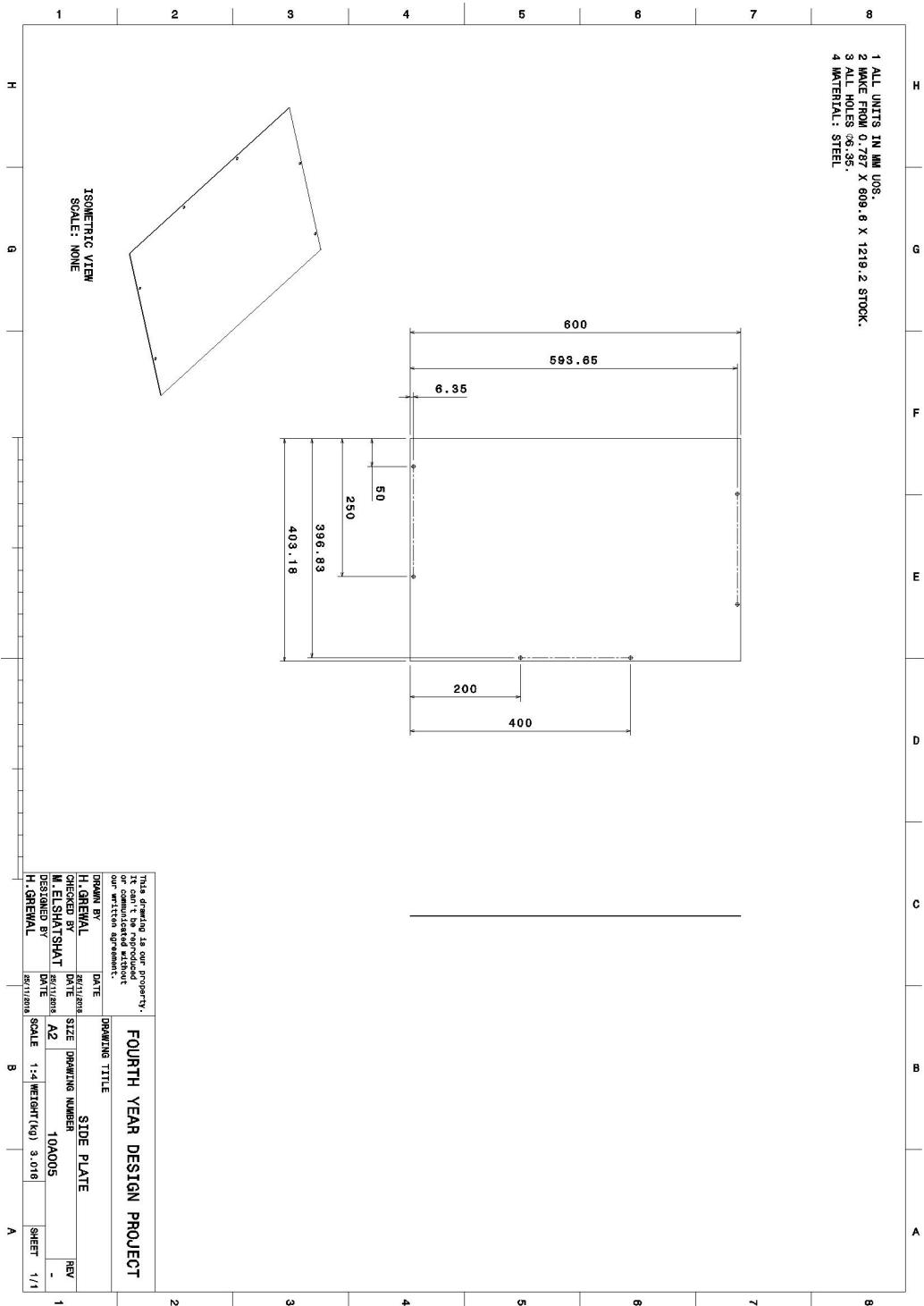


1 ALL UNITS IN MM UDS.
 2 MADE FROM 1.568 X 25.4 X 1828.8 90° ANGLE STOCK.
 3 ALL HOLES Ø8.35
 4 MATERIAL: ALUMINUM

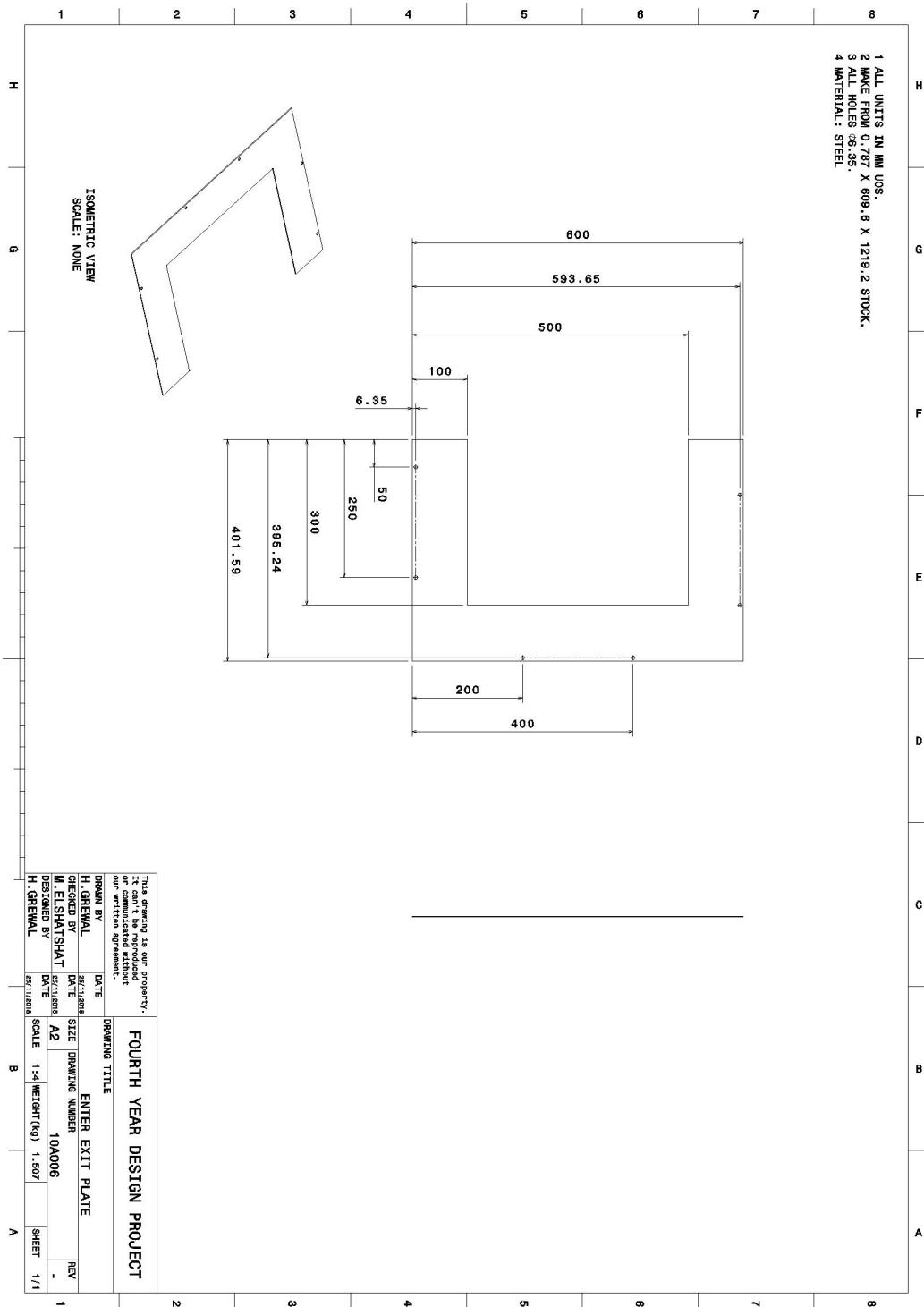


1 ALL UNITS IN MM UDS.
 2 MADE FROM 0.767 X 609.6 X 1219.2 STOCK.
 3 ALL HOLES 36.35.
 4 MATERIAL: STEEL

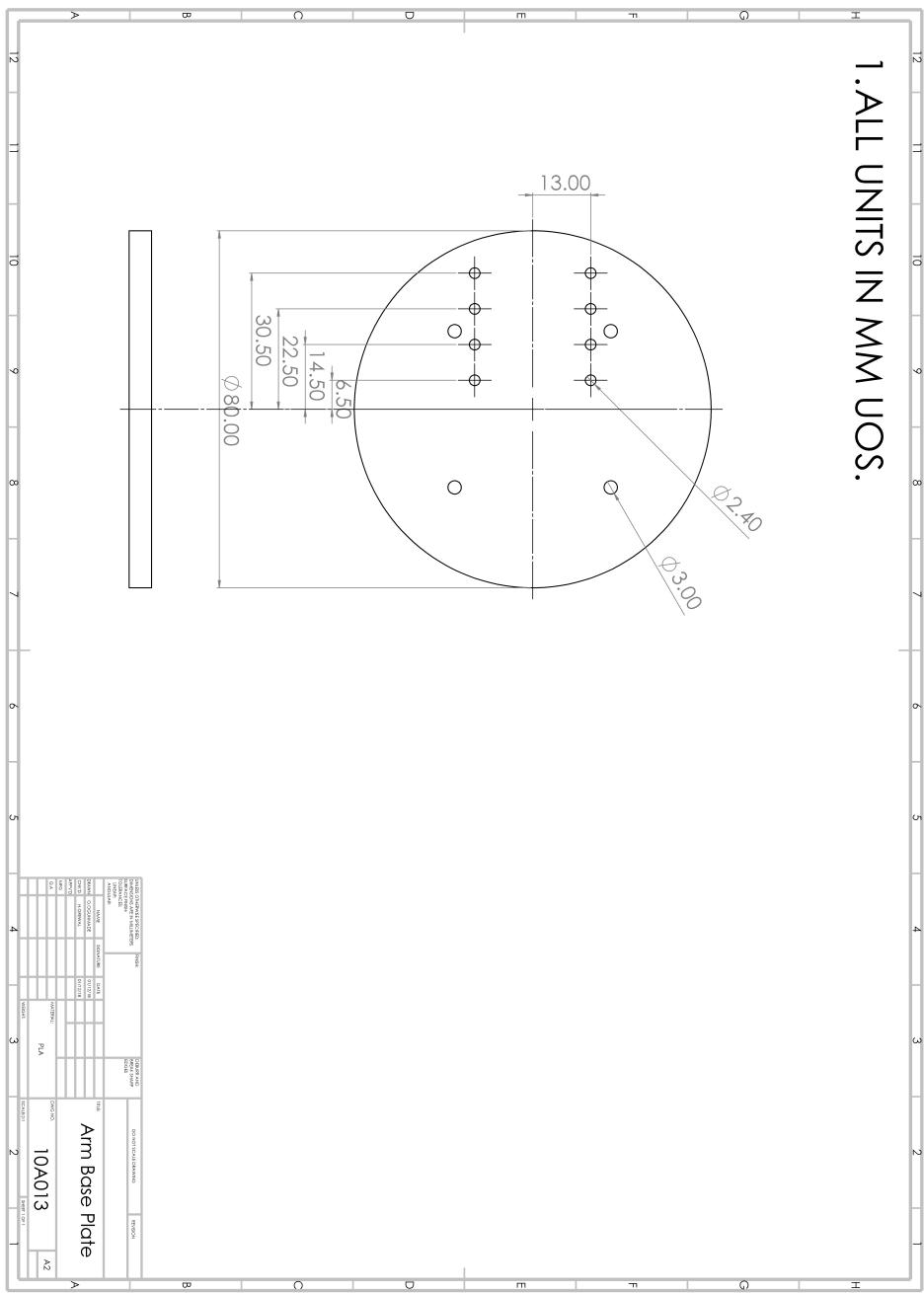


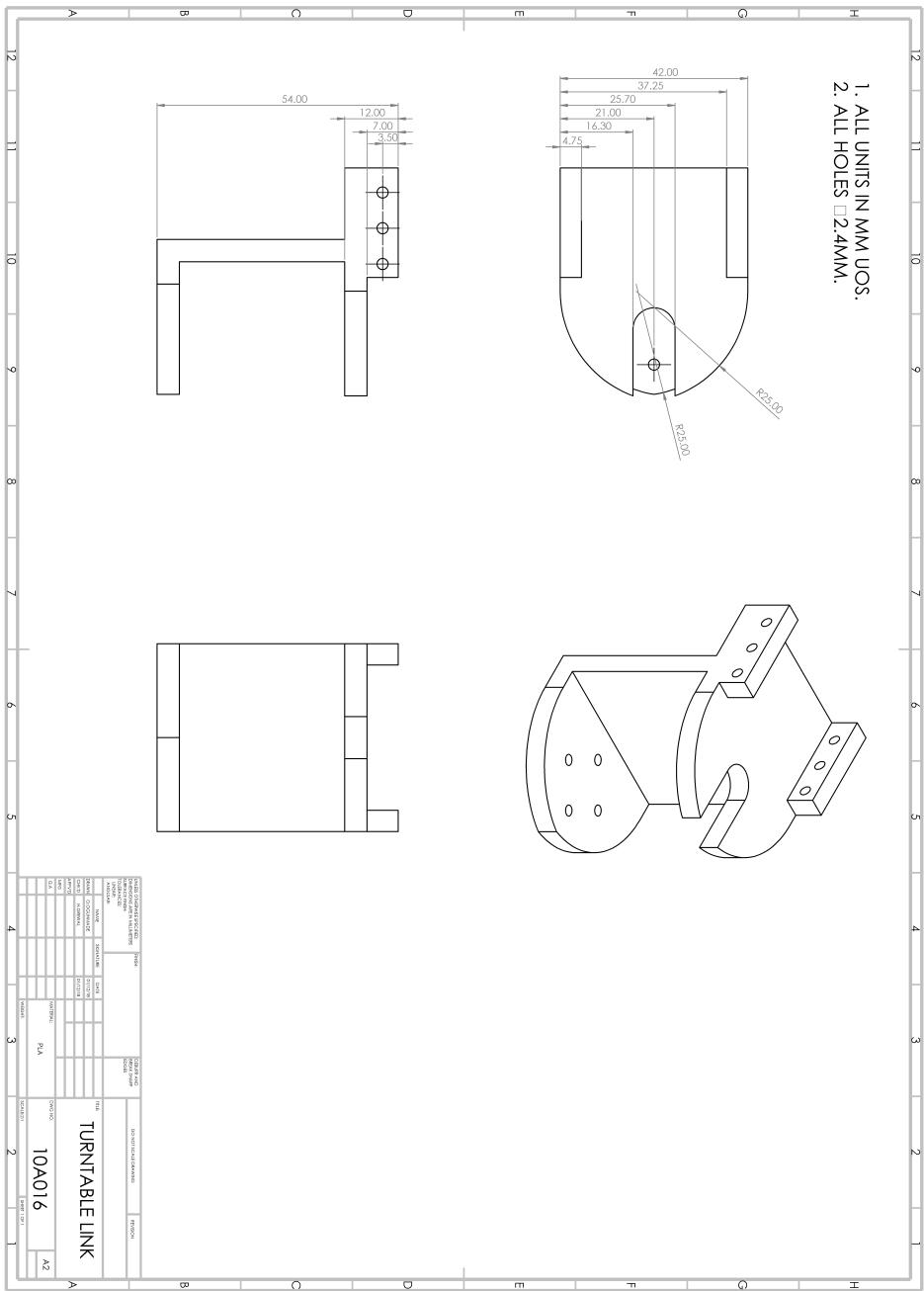


1 ALL UNITS IN MM UDS.
 2 MADE FROM 0.767 X 609.6 X 1219.2 STOCK.
 3 ALL HOLES Ø6.35.
 4 MATERIAL: STEEL

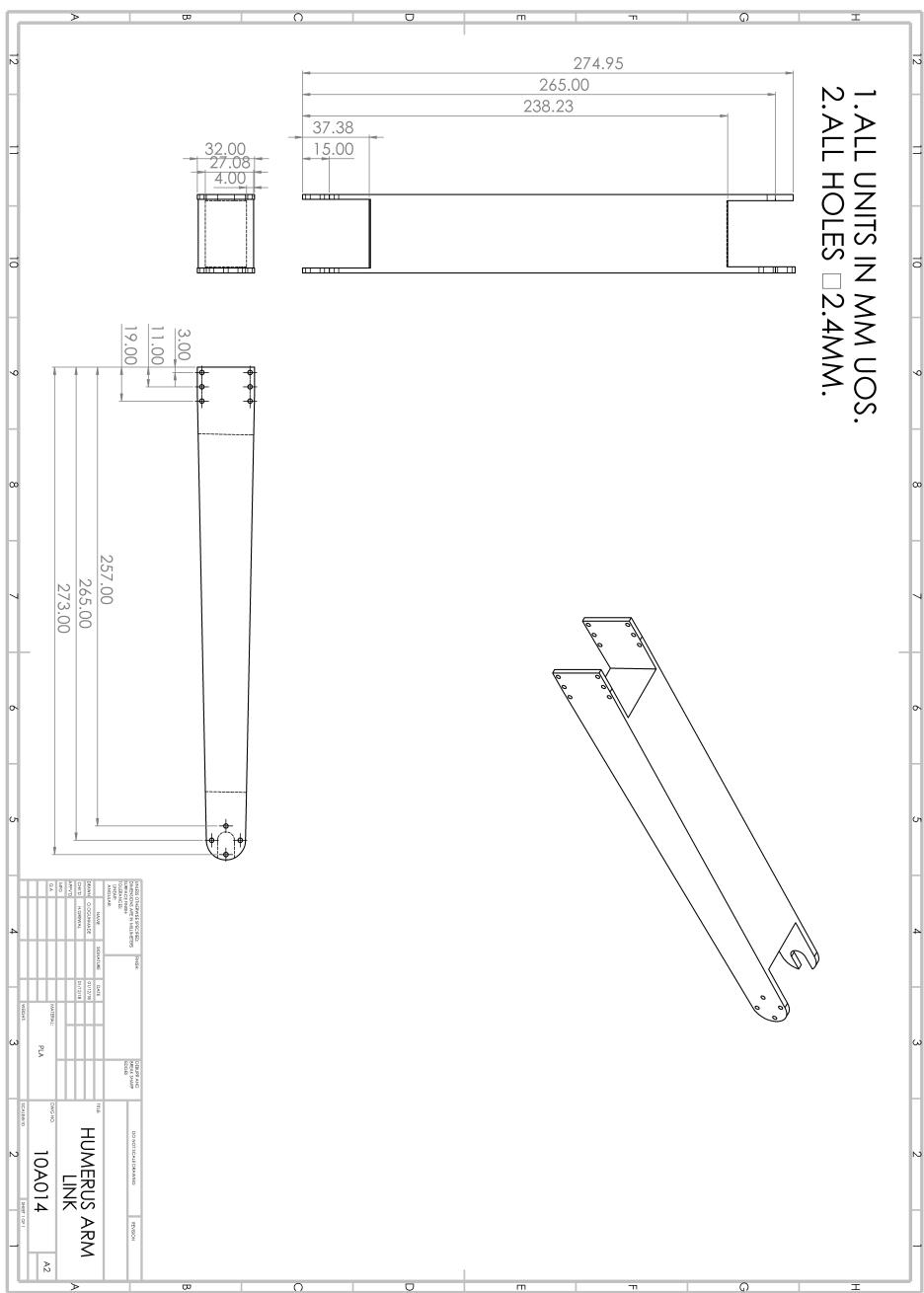


1.ALL UNITS IN MM UOS.

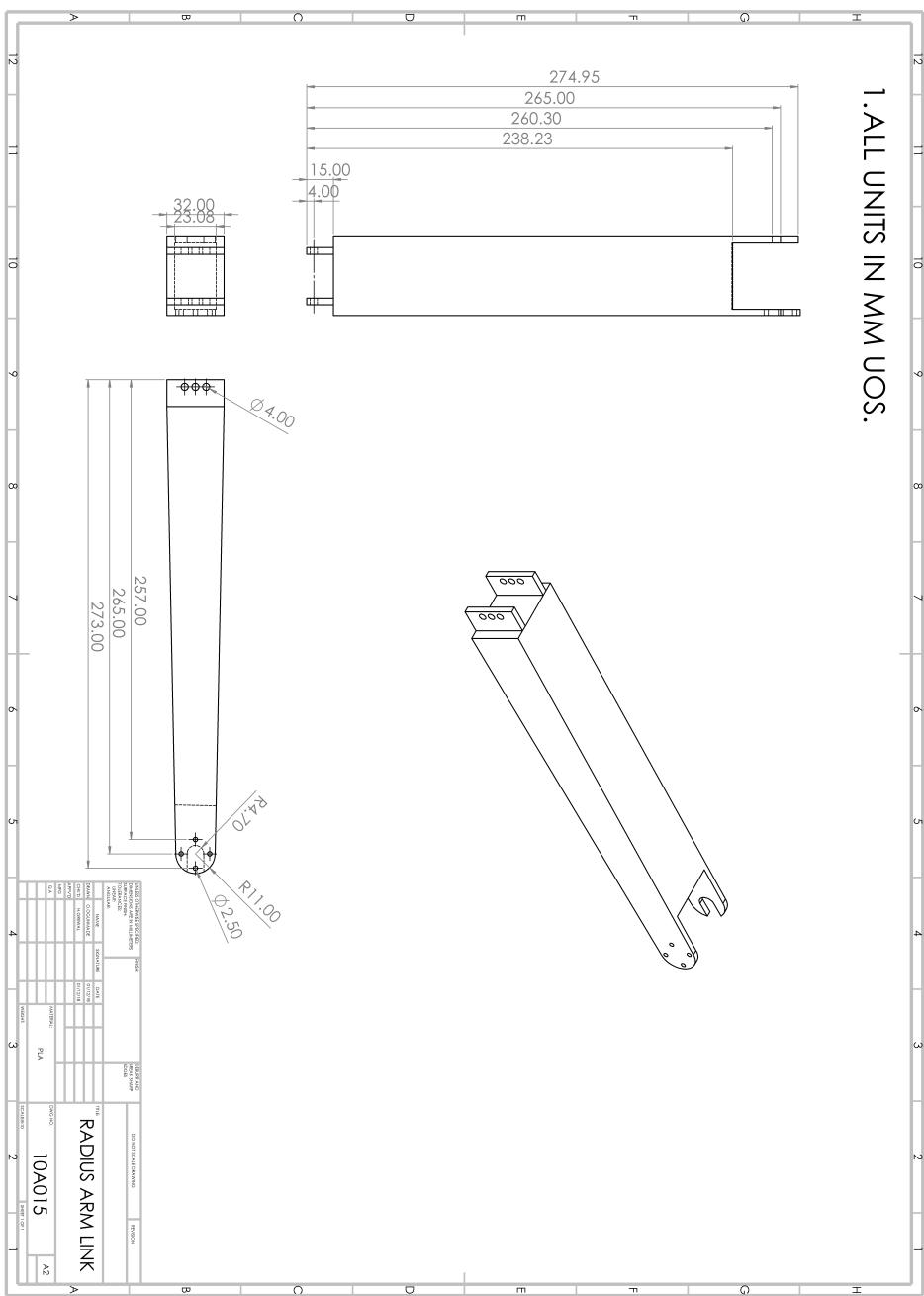




1.ALL UNITS IN MM UOS.
 2.ALL HOLES □2.4MM.



1.ALL UNITS IN MM UOS.



Appendix B Bill of Material

The bill of material is provided in Table 10. Note that prices are given in CAD.

Table 10: Bill of material.

Part Number	Description	Subsection	Quantity	Price/Unit	Full Price
10A001	Piller Beam	Enclosure	4	10.76	43.03 [52]
10A002	Top Support Beam	Enclosure	2	21.52	43.03 [52]
10A003	Bottom Support Beam	Enclosure	2	21.52	43.03 [52]
10A004	Top Plate	Enclosure	1	14.51	14.51 [53]
10A005	Side Plate	Enclosure	2	14.51	29.03 [52]
10A006	Enter Exit Plate	Enclosure	2	14.51	29.03 [52]
10A007	1/4" Bolt	Enclosure	20	1.45	28.93 [54]
10A008	1/4" Nut	Enclosure	20	0.36	7.23 [55]
10A009	Felt	Enclosure	1	9.61	9.61 [56]
10A010	Logitech C525	Camera	1	90.39	79.99 [17]
10A011	VL53L0X	Camera	1	11.24	11.24 [57]
10A012	Barcode Scanner	Part ID	1	57.90	57.90 [58]
10A013	Arm Base Plate	Arm	1	2.45	2.45 [59]
10A014	Humerus Arm Link	Arm	1	2.45	2.45 [59]
10A015	Radius Arm Link	Arm	1	2.45	2.45 [59]
10A016	Turntable Link	Arm	1	2.45	2.45 [59]
10A017	Camera Attachment	Arm	1	2.45	2.45 [59]
10A018	Dynamixel XL-320	Arm	1	32.57	32.57 [60]
	OLLO Smart Serial Servo				
10A019	Dynamixel AX-12A	Arm	3	69.57	208.72 [61]
	Smart Serial Servo				
10A020	Dynamixel CM-5	Arm	1	78.99	78.99 [62]
10A021	Raspberry Pi 3 model B+	Arm	1	51.43	51.43 [63]
10A022	SMAKN DC-DC 5A Automatic Boost Buck Converter	Arm	3	24.48	73.43 [64]
10A023	M2 x 6	Arm	50	0.01	0.73 [65]
10A024	Servo bearing	Arm	3	2.23	6.68 [66]
10A025	AC/DC Power Supply 100W	Electrical	1	43.63	43.63 [67]
10A026	PiTFT Plus Touch Screen Display	User Interface	1	52.56	52.56 [51]
10A027	NeoPixel 16 LED Ring	Lighting	1	31.92	31.92 [68]
				Total	987.02

Appendix C Sample of Defect Criteria

GE Energy Connection
Quality Management System

Cosmetic Inspection Guidelines for Mechanical Components

Cosmetic Reference Standard Table

Defect	Zone	Part's entire Area Less than 400 in ² per side		Part's entire Area Greater than 400 in ² per side	
		Max Defect Size Allowed	Max Number Allowed per 100 in ²	Max Size Allowed	Max Number Allowed per 300 in ²
Applicable to all Parts					
Fracture, Split, Crack	N/A			Defect not Allowed	
Incomplete Fill/ Cold Shot in Cast Metals	N/A			Defect not Allowed	
Corrosion, Oxidation, Rust	N/A			Defect not Allowed	
Short Shot injection molded plastic	N/A			Defect not Allowed	
Burrs and Sharp Edges	N/A			Defect not Allowed	
Scuff, Abrasion, Mark (light)	A	None	0	1" long	1
	B	0.25"	2	1.5" long	2
	C	0.5"	2	1.5" long	4
Note: Must not catch fingernail	D	1.0"	8	Acceptable	8
Scratch (catches fingernail)	A	None	0	0.25" Long	1
	B	0.125"	1	0.5" Long	2
	C	0.25"	2	1.0" Long	4
Note: No exposed metal ; 0.015" width max	D	0.5"	8	Acceptable	8
Pits	A	None	0	0.03"	3
	B	0.04" dia x 0.04" deep	2	0.03"	6
	C	0.06" dia x 0.04" deep	4	0.045"	6
	D	0.10" dia x 0.04" deep	8	Acceptable	Any
Gouge	A	None	0	None	0
	B	None	0	0.03"x0.06"	2
	C	0.075"x0.25"	2	0.075"x0.25"	2
Note: Maximum depth of gouge is 0.04"	D	0.125"x 0.5"	8	Acceptable	Any
Dent, Ding, Nick	A	None	0	None	0
	B	0.100" dia	1	0.25" dia	1
	C	0.125" dia	3	0.50" dia	1
Note: No exposed metal ; 0.040" depth max	D	0.250" dia	5	1.00" dia	Any

GE Proprietary



Uncontrolled when Printed
or
Transmitted Electronically

Page 6 Of 26

GA-SRC-0002 Rev: 1.0