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

Inspeksi: 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

25 March 2019

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

25 March 2019

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

Dear Professor William Melek,

This report, entitled ”Inspeksi: Intelligent Manufacturing Inspection Tool” was prepared as our MTE 482 final report to explain the final design of the proposed inspection tool. Currently, manufacturing inspection processes for cosmetic defects is either very labour intensive or very customized to specific products and processes. In MTE 481, we investigated this issue and explored possible solutions. We proposed a machine learning based automated inspection tool which provides a versatile low-cost solution for small and medium size companies and generalizability.

This report outlines the process of creating this tool. There were several modifications and redesigns required. The team adhered to the planned schedule and budget. Each component of the product was manufactured and tested. These components include the enclosure, camera, lighting, robot arm, inspection algorithm, data communication and results illustration. The team then integrated all the components together. This required system level testing and performance evaluation. The final product was successful due to the hard work and detailed planning of the team.

We would like to thank Kevin Eng (Ecobee), Aleksey Tsaplin (Ecobee), Wim Deweerd (Google Hardware), Eric Stackpole (OpenROV), Brian Grau (OpenROV), and Adam Mack (Flex) for their consultations and willingness to help with design decisions, as well as development of criteria and constraints. Additionally, we thank Robert Wagner (University of Waterloo) for his assistance in the enclosure manufacturing. We would also like to thank Ecobee 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 final design for a low-cost solution for high quality manufacturing inspection, as well as how the design has changed from the proposed design in the MTE 481 report. The final system focuses on detecting surface level visual defects, namely scuffs and scratches. Similar to the proposed design, the functional goals of the inspection tool include part placement (e.g., manual placement or using existing manufacturing lines), image collection, image analysis, database management, and user interface. The construction and manufacturing of the system is also discussed in this report. Finally, the commissioning and testing of the whole system are presented.

The final design still includes a metal enclosure with a robot arm and a camera attached at the end effector of the robot arm. Instead of moving the arm automatically using a distance sensor, the arm can only be moved using the backdrivability feature, and thus the distance sensor is omitted. Furthermore, the design of the arm links were modified to reduce unaccounted interference, as well as to reduce the total weights. In case of the detection algorithm, the algorithm was changed from object detection model (i.e., Single Shot Multibox Detector (SSD) [1]) into an instance segmentation model called Mask R-CNN [2]. Finally, several non-critical features were removed due to the time constraint. The removed features include: servoing based on laser distance sensor, detection of more complex defects (e.g., dent, discoloration,etc.), part identification using barcode scanner, and UI/UX. Once the construction was completed, the design was tested and evaluated. Based on the evaluation, the design was concluded to meet all the specified constraints.

To further improve the design, the use of meta-learning algorithms such as the Model Agnostic Meta Learning (MAML) [3] is recommended. MAML 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. Due to the fact that collecting dataset from scratch is time-consuming, another interesting thing to experiment with is to train the model using only synthetic dataset (e.g, CAD models) and then perform transfer learning from simulation to the real world. In addition, redesigning the shoulder joint of the arm would improve the performance and prevent motor failure. The redesign should include replacing the shoulder motor responsible for raising and lowering the arm and a redesign of the plastic shoulder turntable piece. Furthermore, the accuracy of the detection would be improved with a better autofocusing system as most of the undetected scratches were missed because they were out of focus.

Table of Contents

Summary . . . iii
List of Figures . . . vi
List of Tables . . . vii
1 Introduction . . . 1
1.1 Background . . . 1
1.2 Description of Design from MTE 481 . . . 2
1.2.1 Enclosure . . . 3
1.2.2 Camera . . . 4
1.2.3 Lighting . . . 4
1.2.4 Robot Arm . . . 5
1.2.5 Inspection Algorithm . . . 6
1.2.6 Part Identification . . . 7
1.2.7 Data Communication . . . 7
1.2.8 User Interface and Experience . . . 8
2 Final Design . . . 9
2.1 Final Design Details . . . 9
2.2 Modifications/Deviation from Original Design . . . 9
2.2.1 Enclosure Modification . . . 10
2.2.2 Robotic Arm Modification . . . 10
2.2.3 Inspection Algorithm Modification . . . 14
2.2.4 Data Communication . . . 15
2.2.5 Removed Features . . . 15
2.3 Construction/Manufacturing . . . 15
2.3.1 Enclosure Manufacturing . . . 16
2.3.2 Robot Arm Manufacturing . . . 17
2.4 Commissioning . . . 17
2.4.1 Enclosure . . . 18
2.4.2 Robot Arm . . . 18
2.4.3 Inspection Algorithm . . . 18
2.4.4 Data Communication . . . 19
2.5 Testing and Performance . . . 19
3 Schedule and Budgeting . . . 21
3.1 Schedule . . . 21
3.2 Budget . . . 21
4 Conclusions and Recommendations . . . 23
5 Teamwork Effort . . . 24
Acknowledgement . . . 25
References . . . 26
Appendix A Detailed Part Drawings . . . 28

Appendix B Bill of Material 39

List of Figures

1 Illustration of the viewing zones [4]. 2
2 Sketch of the design proposed solution [4]. 3
3 Illustration of the systems [4]. 3
4 Enclosure CAD model [4]. Note that the bolts and felt are not shown. 4
5 The result of camera mounted lighting testing [4]. 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. 5
6 CAD rendering of the original arm [4]. 5
7 Calculation of the torque required at each joint [4]. 6
8 Examples of scratch detection using MobileNets-SSD [4]. 7
9 Illustration of the data communication pipeline [4]. 8
10 Proposed user interface [4]. 8
11 The final design fully assembled. Note that a side plating is removed to show the inside. 9
12 Notches in the frame beams. 10
13 CAD rendering of the new arm design. 11
14 Comparison between the original and the new arm design. 11
15 Initial electrical system diagram. 12
16 Final electrical system diagram. 13
17 Robotic Arm Software State Diagram 13
18 Preliminary results of using Mask R-CNN to detect scratches on a plastic eyeglass case, where different colors indicates different instance of scratches. The numbers indicate the confidence level of the model. 14
19 Before and after sanding of the enclosure. 16
20 Electrical hardware on top of enclosure. 17
21 Results of processing images captured by the robot arm using Mask-RCNN. 20
22 Estimated project timeline [4]. 22

List of Tables

1 Viewing zone classifications on parts [4]. 1
2 Summary of constraints and criteria [4]. 2
3 Comparison of different cameras [4]. 4
4 Comparison of various inspection algorithms (‘‘High’’ is the best) [4]. 6
5 Steady state robot arm joint orientation error. 19
6 Bill of material. 39

1 Introduction

In this section, the background to the problem is introduced in Section 1.1. The design that was developed in MTE 481 is also discussed in Section 1.2.

1.1 Background

Manufacturing processes have limitations that result in deviations and deformities. Thus, inspection technologies are needed to ensure the highest quality of manufactured products. Although various inspection tools exist in the market, these tools typically are expensive and very customized to inspect specific types of product or material. In fact, many companies still rely on human labors to perform manual inspection. Although generalizable, this solution is still expensive and prone to human errors. Furthermore, human labors do not typically provide the details on why a product is defective, since it is too expensive for factory owners to ask these labors to document all types of anomalies. Consequently, companies with limited resources and funds face difficulty conducting high quality manufacturing inspection, which may damage the reputation of the company.

The formulated problem statement is as follows: there is no low-cost automated solution for visual inspection of high end consumer facing products [4]. In [4], a set of objective, constraints, and criteria for the inspection tool were developed. The objectives of the system include: accuracy and precision greater than 90%, ability to localize and classify various defects (e.g., scuffs, scratches, dents, discolorations), ability to identify minimum defects size of 1 millimeter, and the ability to inspect class A, B, and C viewing zones (refer to Table 1 and Figure 1). The constraints and criteria that were developed in [4] are presented in Table 2.

Table 1: Viewing zone classifications on parts [4].

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 proposed solution is a closed-box solution known named Inspeksi. Inspeksi is a low-cost fully automated manufacturing visual inspection tool that would enable companies to improve their quality control processes and ensure a consistent level of quality for their consumers. This report is a continuation of [4], where a preliminary design for Inspeksi was proposed. In this report, a final

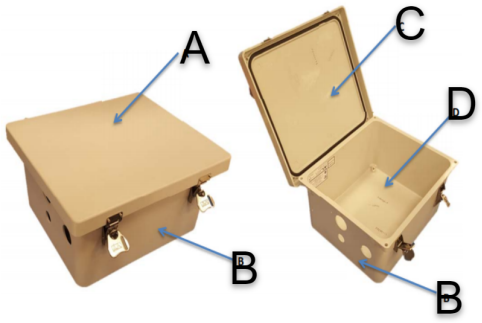


Figure 1: Illustration of the viewing zones [4].

Table 2: Summary of constraints and criteria [4].

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

design for intelligent low-cost manufacturing inspection tool is proposed. Furthermore, changes and deviations made from the original design are also discussed in Section 2.2.

1.2 Description of Design from MTE 481

A design for the inspection tool was proposed in MTE 481. The design includes an enclosure, which provides a controlled lighting environment to take pictures of the inspected parts. Inside this enclosure, there is a robotic arm hanging at the top of the enclosure with a camera attached at the end-effector. A barcode scanner was proposed at the base of the enclosure to scan the part numbers of the inspected item. The concept of the design proposed in MTE 481 is depicted in Figure 2.

The functional goals of the proposed design include part placement, image collection, image analysis, database management, and user interface. First, a part is placed inside an enclosure. The robot arm will then move the camera around the inspected part to capture images of the part from multiple angles. The data are then sent to a server computer where they will be processed and stored. The images are processed using a computer vision algorithm that performs defect classification and localization. The results are finally stored in a database that users (e.g., clients or contract manufacturers) can access remotely through an intuitive user interface. The proposed systems was divided into several independent subsystems as illustrated in Figure 3.

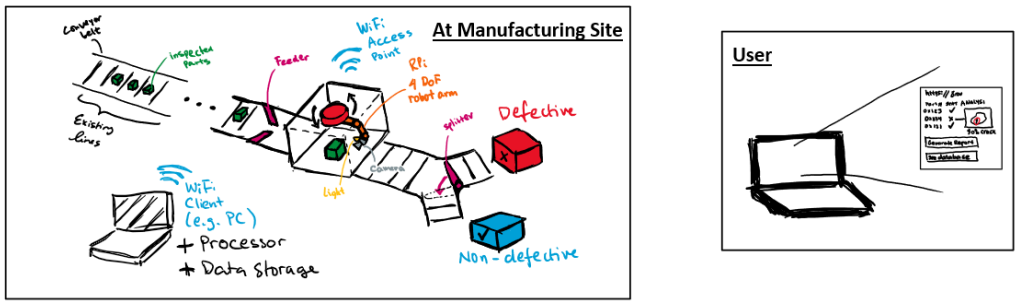


Figure 2: Sketch of the design proposed solution [4].

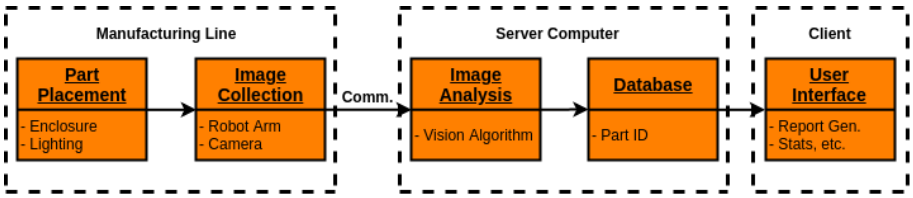


Figure 3: Illustration of the systems [4].

1.2.1 Enclosure

The enclosure is used to house the robotic arm hanging from the top plate of the enclosure, it moves to take pictures of the part. It is also used to block out ambient light and allow for part loading. The electrical hardware is also supported on top of the enclosure. The enclosure is designed to be used in either in-house or assembly line inspection.

The box is made of metal which is suitable for the factory environment and can be assembled using bolts. The modular design contains an aluminum frame, angle brackets, and steel plates, which allows for easy assembly and disassembly. The enclosure has dimensions of 60cm by 60cm with a 40cm height. It also has an opening for an assembly line which would pass through the enclosure. The openings were designed for a nominal assembly line of 40 cm width and 10cm height.

Black cloth is used to drape the openings to ensure ambient light is blocked. Standard sizes were chosen for the holes and bolts to reduce cost and ease the manufacturing and material sourcing process. Finite Element Analysis (FEA) was also conducted to validate that the enclosure is able to hold the weight of the hanging robot arm. The enclosure design is illustrated in Figure 4.

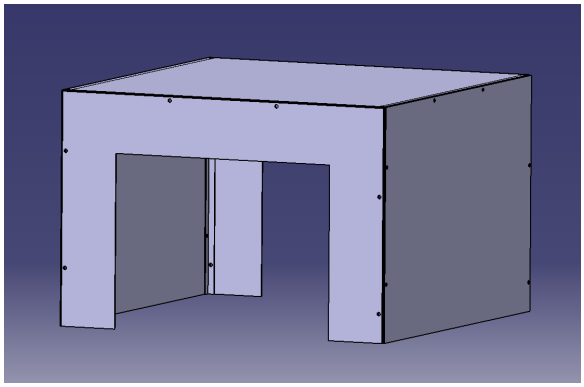


Figure 4: Enclosure CAD model [4]. Note that the bolts and felt are not shown.

1.2.2 Camera

Several criteria were developed in [4] to pick the camera. The criteria include price, weight, camera settings, resolution, and field of view. Various camera options were considered: Logitech C270, Logitech C525, Logitech C920, and RPi Camera. Table 3 shows the comparison between cameras.

Table 3: Comparison of different cameras [4].

	C270 [5]	**C525** [6]	**C920** [7]	**RPi** [8]
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°

The Logitech C525 was chosen since it is lighter and less expensive compared to the Logitech C920, while still meeting all the requirements needed to perform an inspection. Furthermore, an initial test using Linux V4L2 API and OpenCV [9] showed that the focus of this camera can be adjusted manually. Although the manual focus feature was not used in the final prototype, it is an important feature that enables the possibility of improving the inspection capability in the future.

1.2.3 Lighting

Lighting is an instrumental aspect of this design; proper lighting makes subtle defects visible to the camera. Two implementations were proposed for the lighting: a single light source mounted on the camera or multiple lights mounted within the enclosure. The criteria to choose the lighting were cost, complexity, and effectiveness of each method. A single light source mounted on the camera was chosen, mainly due to the fact that attaching light on camera does not cast shadows on the inspected part, while using multiple mounted lights within the enclosure may cast shadows when the arm is located in front of the lights. Furthermore, using single light source is cheaper.

The camera mounted lighting was tested empirically to verify that defects would be visible in images and videos. Based on the testing, it was confirmed that the defects are visible as long as the they are not facing the light source directly. The results are demonstrated in Figure 5.

(a) Image of part with camera mounted light.

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

Figure 5: The result of camera mounted lighting testing [4]. 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.

1.2.4 Robot Arm

The robotic arm is used to position the camera to acquire images of the inspected parts. The arm spans up to 50cm from the base to the end-effector. The arm comprises of 5 plastic structural pieces, 4 servos, the end-effector, and the wiring for power distribution and communication. The original model of the arm is shown in Figure 6.

Figure 6: CAD rendering of the original arm [4].

The structural pieces were 3D-printed due to the prototyping ease. The torque required at each of the joints were determined by computing the forces and moments at each of the links to determine if the motors would be capable of lifting the required load. The computations provided the numbers in Figure 7 below. With T1, T2 and T3 corresponding 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. For more complete design analysis of the arm, refer to [4].

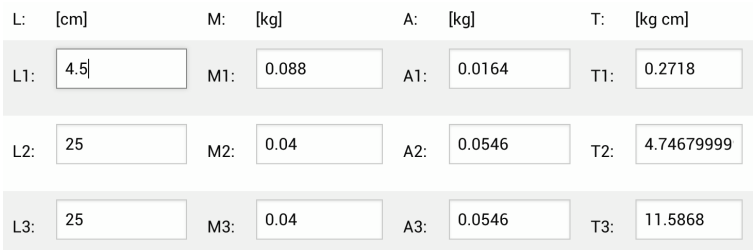


Figure 7: Calculation of the torque required at each joint [4].

1.2.5 Inspection Algorithm

Various learning-based computer vision algorithms to perform defect detection were considered during the design process due to its generalization capability to infer unseen data. This decision was made due to the fact deep learning [10, 11] based algorithms have shown state of the art performance in wide range of vision tasks such as in classification [12, 13, 14], object tracking [15, 16, 17], object detection [18, 19, 20], and image segmentation [21, 22, 23, 2]. The algorithms considered to perform defect detection include image classification, object detection, semantic segmentation, and instance segmentation algorithms. Comparison between these algorithms can be found in Table 4.

Table 4: Comparison of various inspection algorithms (‘‘High’’ is the best) [4].

	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 the comparisons, the object detection method was chosen. Given an input image, object detection model localizes defects in the image space in terms of bounding boxes. In addition to bounding boxes, the model also provides the confidence level of the model for each of the predicted bounding boxes. The specific model chosen was the combination of Single Shot Multibox Detector (SSD) [1] and MobileNets [24], which was already been applied in detecting defects [25]. This model was chosen mainly due to its lightweight and fast performance. A preliminary model was implemented using the Tensor-Flow library [26] and was shown to be capable to detect scratches.

Figure 8 demonstrates the results. However a suggestion to use instance segmentation was made in case the there is enough time to implement the Mask R-CNN model. This suggestion was eventually implemented, and will be discussed in Section 2.2.3.

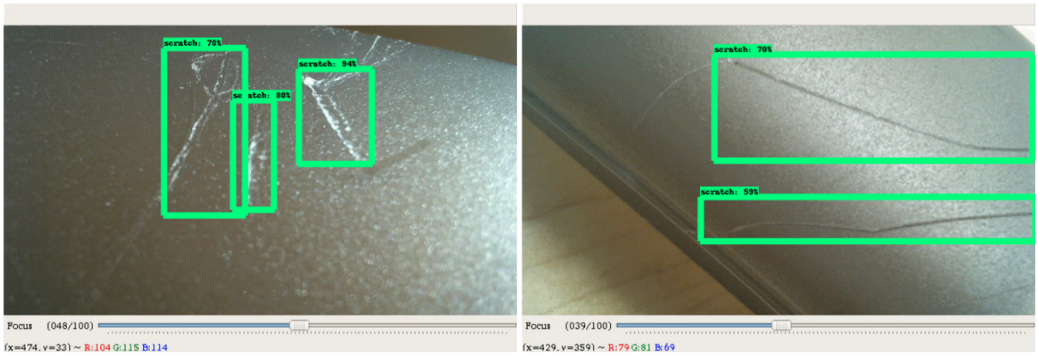


Figure 8: Examples of scratch detection using MobileNets-SSD [4].

1.2.6 Part Identification

The purpose of the part identification is to allow for the closed-box solution to integrate with existing manufacturing lines. The system would be able to scan barcodes on manufactured parts passing through its scanning environment and associate scan results to the part numbers for quick access and sorting of results. Two viable solutions for part identification involve camera-based detection or a keyboard-input laser scan.

The camera-based detection does not require additional hardware since there is already a camera attached to the robot arm. However, additional software is required to process and analyze the barcode or serial numbers. The keyboard input laser-based solution is simple, yet requires additional hardware of a keyboard input barcode scanner, with additional software to parse the inputs and activate the scanner during appropriate times. Due to the insignificant additional cost of a barcode scanner and simple software implementation of parsing the keyboard inputs of the scanner, the use of barcode scanner was proposed in the initial design. The Symcode MJ-2090-L model was chosen after comparing three laser barcode scanners.

1.2.7 Data Communication

Communication between the camera on the robot arm and a computer where the data is processed and stored was considered in [4]. Options considered to transfer the images were through USB ports, Wi-Fi, and bluetooth. Since not every image from the camera needs to be processed, one can instead focus on user friendliness by using Wi-Fi. Wi-Fi communication was concluded to be a more appropriate option due to its higher bit-rate. The Wi-Fi communication was proposed to be

implemented using Robot Operating System (ROS) [27] to simplify multithreading management and interprocess communication. Figure 9 illustrates the data communication pipeline.

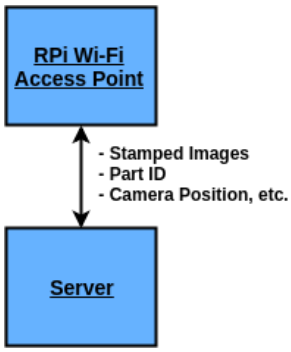
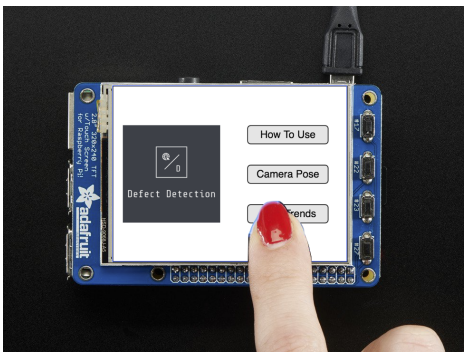


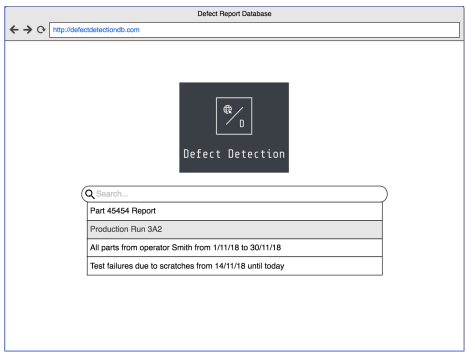
Figure 9: Illustration of the data communication pipeline [4].

1.2.8 User Interface and Experience

The user interfaces were designed to enable the operators on the factory floor and quality engineers to acquire and interpret quality assurance data. The interfaces include a touchscreen-based interface and a web-based interface. A touchscreen attached to the fixture allowed the operator to perform quality checks on the spot. Through the interface the operator can view the live feed of the camera, record motion sequences, play them back and debug the appliance. The touchscreen also allows the operator to learn how to use the device. The web interface allows the quality engineer to view the results of product quality inspection runs. The interface is similar to a search engine, which allows the engineer to query the database for trends in the data, reports from time ranges, and reports from individual parts. Figure 10 illustrates the proposed interface.



(a) Touch interface.



(b) Engineer interface.

Figure 10: Proposed user interface [4].

2 Final Design

The final design of the system is presented in this section. First, the detail of the final design is presented in Section 2.1. This is then followed by some discussions on why the modifications were made from the original design proposed in [4].

2.1 Final Design Details

The final design concept still follows the concept proposed in [4]. To recap from Section 1.2, this includes an enclosure with a camera attached to the end-effector of a 4 degrees-of-freedom robot arm. The robot arm is hung at the centre of the top plate of the enclosure, which allows it to rotate around the inspected object to take pictures. The enclosure was designed such that it can be installed on an existing manufacturing line.

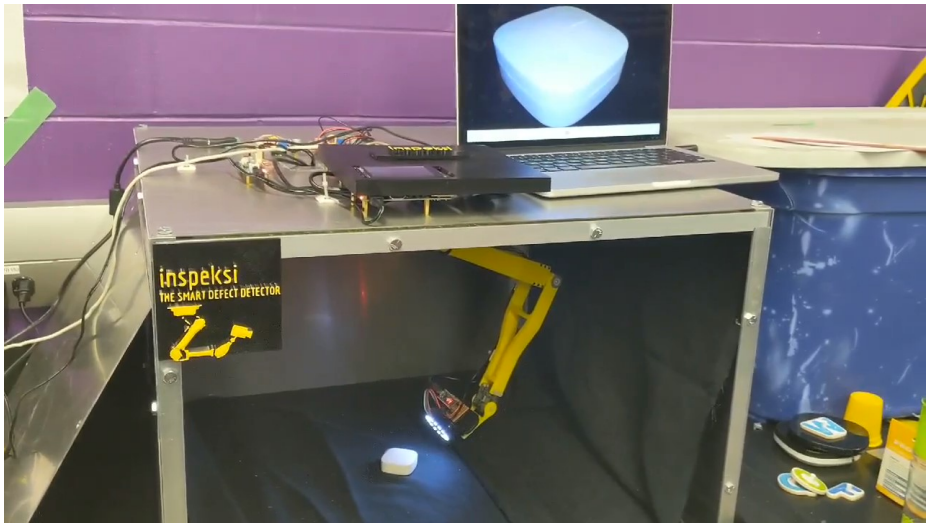


Figure 11: The final design fully assembled. Note that a side plating is removed to show the inside.

2.2 Modifications/Deviation from Original Design

Several modifications were made from the original design. These include the design of enclosure, design of robotic arm, inspection algorithm, and data communication. Furthermore, some minor features like the part identification and UI/UX were also removed.

2.2.1 Enclosure Modification

There are several modifications made to the design of the enclosure. First, in order to save cost, the material was sourced from the University of Waterloo E3 Machine Shop instead of Home Depot. This saved approximately $200. This also allowed for the commissioning of the manufacturing to the Machine Shop instead of the team.

While discussing the proposed design with the physical material, it was discovered that the original angle brackets with a side length of 0.5 inches would not have enough space for the nuts. The design was changed for angle brackets with a side length of 0.75 inches. Consequently, the hole positions shifted on all of the parts. Revision A was issued to the Machine Shop as well as an exploded assembly drawing. These drawings are provided in Appendix A.

In the original design, there are some gaps between the plates and the frame, which would require spacers. To overcome this, notches were added to the top plate support beams to allow for a flush surface with the side plates. With this modification, spacers were no longer required. This is shown in Figure 12.

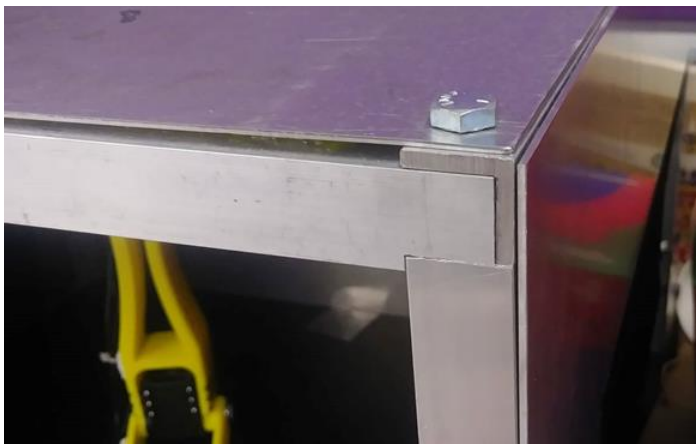


Figure 12: Notches in the frame beams.

Some other modifications that were not explicitly specified in the original design were also made. These include the holes in the top plate for standoffs to support electrical hardware, a slot for the robot arm wiring, and holes for motor mounting plate.

2.2.2 Robotic Arm Modification

The modifications in robot arm design consist of mechanical, electrical, and software changes. The new CAD model of the arm is shown in Figure 13.

In the mechanical modifications, the arm link design followed an iterative methodology. The it-

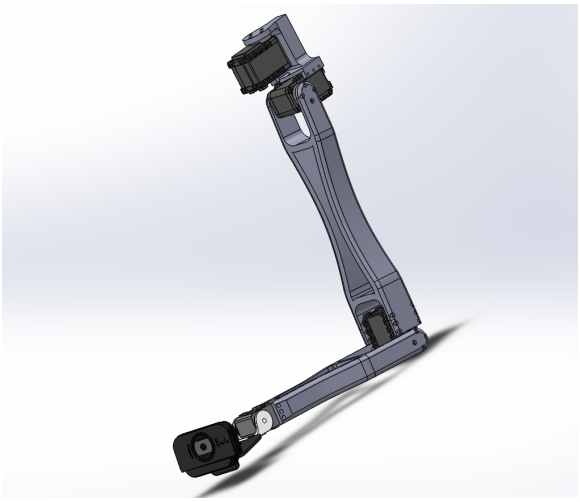


Figure 13: CAD rendering of the new arm design.

erations focused on achieving the range of motion goals, which was hindered by the wiring and connectors that were not taken into account during the initial design. The iterations also focused on reducing weight of the arm to allow for better control and smoother motion. The first iteration of the arm links were two solid body pieces that allowed for about 40 degrees of rotation at the base. This base rotation shortfall was due to the turntable piece. The second iteration included two changes that increased the range of motion to about 300 degrees of revolution at the base, and about 50% in weight reduction. The weight reduction was achieved by introducing cutouts to the longer pieces of the arm. The comparison between the first and second iteration of the arm are shown in Figure 14.

(a) Original arm design.

(b) New arm design.

Figure 14: Comparison between the original and the new arm design.

In the electrical modifications, the focus was to simplify development, increase interfacing flex-

ibility (e.g., via serial or bluetooth), and to reduce the cost of the entire system. The initial and final arm electronics included 4 Dynamixel motors (3 AX-12 series and 1 XL-320 OLLO series). Upon further investigation, it was determined that the previously sourced serial motor controller, the Dynamixel CM-5 was compatible with the AX series motors, but not the XL series motors [28]. Consequently, a new motor controller needed to be sourced to communicate with both the AX and XL series rather than sourcing two different controllers.

With some investigation, the OpenCM9.04 Rev C board was found to be compatible with both motors. In this domain, the second electrical modification was to utilize two OpenCM motors since the AX series motors operate at 11.1V and the XL series operate at 7.4V. Both of these rails were originally designed into account using a 12V power supply that was reduced using 2 tuneable buck converters. However, two unique controllers were needed since the OpenCM boards could only use one rail using a push-pull buffer.

An Arduino Nano module was also added to power and control the LED light strip using the RaspberryPi. The 5V buck converter and rail was removed from the electrical architecture, as it was not needed by any subsystems. The electrical modifications are reflected in the before and after figure shown in Figure 15 and 16, respectively.

Since the team agreed to focus the main functionality of the robotic arm to be a stable smooth motion along a user specified path, distance estimation for focus adjustment and part identification functionality were pushed to lower priority, with the likely possibility of foregone implementation if time did not permit. Specifically, for focus control, the team decided to rely on the Logitech C525 auto-focus functionality rather than to develop a custom laser-based distance measurement module that would require additional circuitry and logic in software. This decision helped to reduced cost, complexity, and development time. Furthermore, the part identification functionality was discarded, which also simplify the circuitry and development.

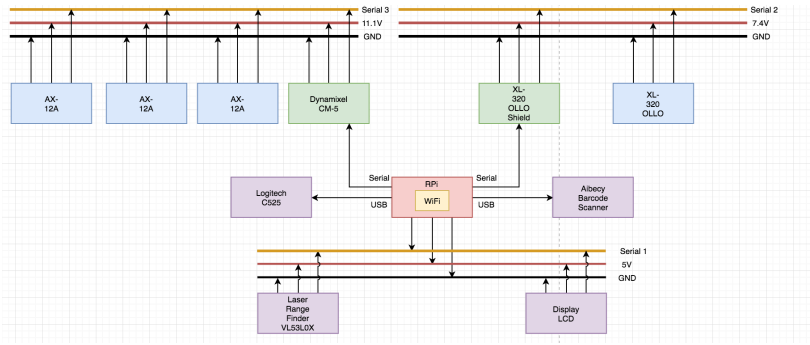


Figure 15: Initial electrical system diagram.

Finally, in the software domain, the major change was to move away from the inverse kinematics and complex path planning. This decision was justified as the unique functionality of this robotic arm is in its ability to save and playback any scanned path specified by the user. Since the initial

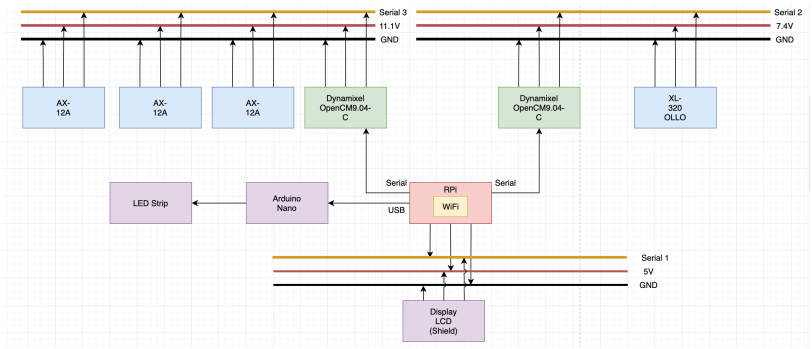


Figure 16: Final electrical system diagram.

system was designed to have a backdrivability feature which allow users to configure a unique path, the team realized that the inverse kinematics and path planning is no longer necessary. This simplified the software development and allowed the team to build a more polished product.

The interaction between a user and the robotic arm is defined in the state machine shown in Figure 17.

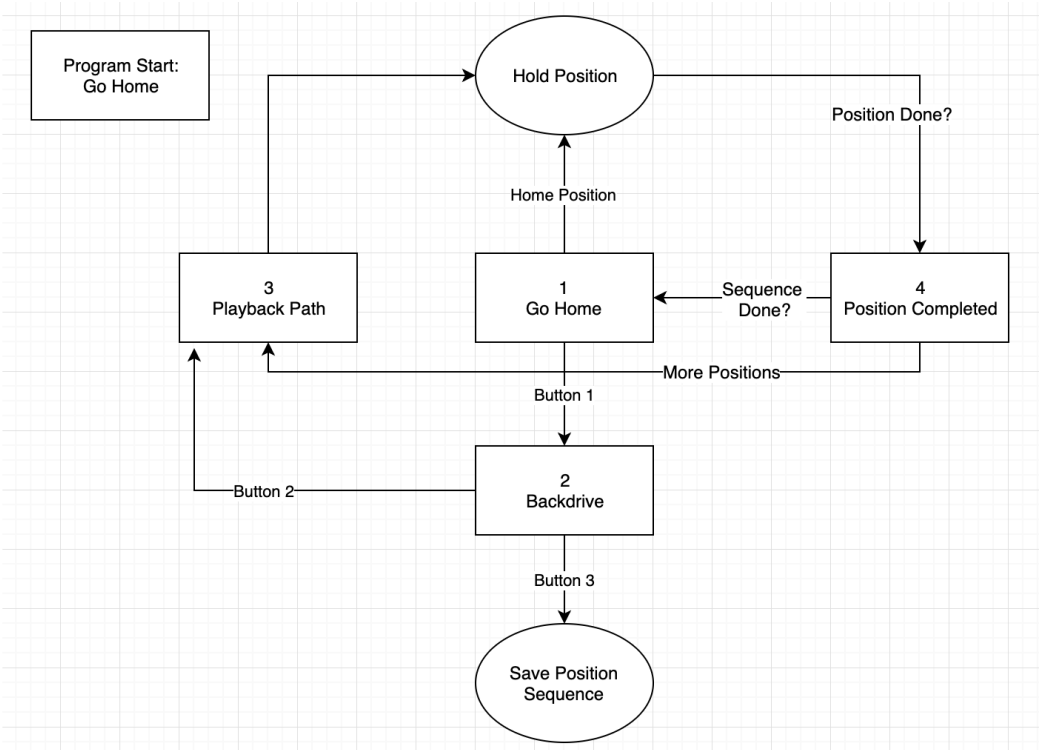


Figure 17: Robotic Arm Software State Diagram

2.2.3 Inspection Algorithm Modification

As discussed in Section 1.2.5, the initial proposed algorithm to perform detection of defects was the MobileNets-SSD. Single Shot Multibox Detector is an object detection model that predicts the location of the defects in the image in terms of bounding boxes. However, representing location of defects like scratches as bounding boxes may produce misleading results should one is interested to know about the size of the defects. For example, if a scratch appears diagonal from the camera's perspective, the bounding box will appear unnecessarily big since the box has to cover the whole scratch. In the application of defect detection, the size of the defects is often a very important metric to be measured.

In the final design, the algorithm was modified into an instance segmentation model called Mask R-CNN [2], following the suggested improvement made in [4]. Unlike object detection models, instance segmentation models, like the Mask R-CNN, are designed to localize object in an image on the pixel level. This is indeed desirable for the application of detecting defects to accurately measure the size of the defects.

Although this method has been used by [29] in a similar application, in order to test and validate the algorithm specifically to detect scratches and scuffs, a Mask R-CNN model was trained on a very small dataset that contains images of objects with scratches. Indeed, the algorithm was able to classify and localize the scratches in the image better than the object detection model. The preliminary results of using Mask R-CNN can be found in Figure 18. Based on the preliminary results, instance segmentation model was decided to be more appropriate to classify and localize defects in images.

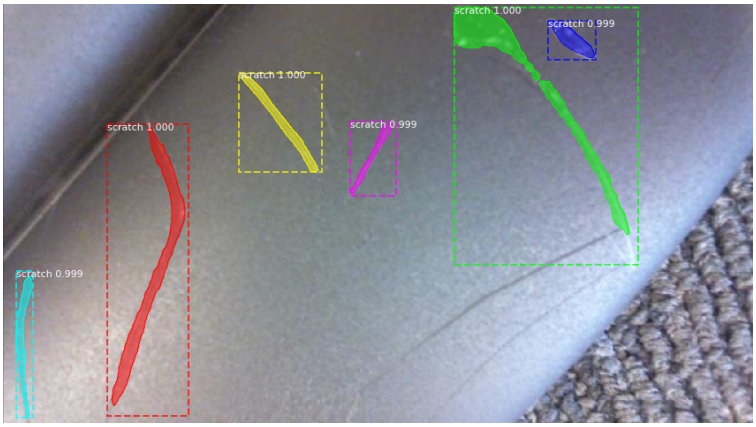


Figure 18: Preliminary results of using Mask R-CNN to detect scratches on a plastic eyeglass case, where different colors indicates different instance of scratches. The numbers indicate the confidence level of the model.

The results of running the Mask R-CNN model on the test parts from Ecobee are shown in Section 2.5.

2.2.4 Data Communication

Instead of connecting the camera into the RaspberryPi, and communicating the camera readings over some medium to the server computer, the camera was directly connected to the server computer. This was done to avoid sending possibly large size of image data through internet connection during crowded symposium day which may delay the data transmission.

Consequently, the server computer must know the state of the robot arm and when to command the camera to take a picture. To synchronize the robotic arm and RaspberryPi system with the server computer and camera, a simple HTTP web server was hosted on the RaspberryPi using built in Python modules called HTTPServer, allowing the server computer to poll the web server using HTTP GET requests. This communication method is modelled after a client-server setup, where the server computer is the client and the RaspberryPi is the server that updates the robot state and publishes it once prompted. There were three types of command in total that the RaspberryPi needs to send to the computer. The first command, '0', indicates that the robot arm is moving, and the computer should not do anything other than showing the camera stream to the user. The second command, '1', indicates that the arm has stopped moving, and a picture can be taken. Finally, the last command, '2', indicates that the arm has finished inspecting the part, and the computer should display the inspection results.

In order for the RaspberryPi to be available over the internet, it was setup to run a predefined module called Dataplicity SSH on bootup. With the Raspberry Pi hooked to the Dataplicity server over the internet, the RaspberryPi was made public to the internet via a URL that can be reached via HTTP GET requests.

2.2.5 Removed Features

In order to meet the deadline, some features had to be dropped. These features include part identification, auto-focusing using a laser distance scanner, as well as part of the intuitive user interface and experience. These features were decided to be dropped as they are not critical features to build the minimum viable product. Finally, to simplify the problem, the algorithm was only trained to detect scratches and scuffs.

2.3 Construction/Manufacturing

The system is comprised of two main sections that need manufacturing and construction. These include: the Robot Arm which is comprised of 3D printed PLA plastic, and the enclosure, a bolted metal assembly with felt lining to ensure a consistent environment when scanning parts.

2.3.1 Enclosure Manufacturing

As discussed in Section 2.2.1, the manufacturing and material sourcing were primarily done at the University of Waterloo E3 Machine Shop. During manufacturing process, there was constant communication between the mechanical lead, Herman Grewal and the machinist, Robert Wagner. Modifications were made to the design to resolve interference of the nut and eliminate the need for washers.

The angle brackets were first cut to the required lengths. In an assembled state, the holes were marked and then drilled. Next, the panels were cut and assembled to the frame. By slowly building the product as an assembly instead of individual parts, hole locations and fits were then resolved, which allowed for the use of less precise machinery to save machining time. The openings in the side panels were cut using a band saw. However, the band saw produced a very rough edge which required to be sanded afterwards.

There were few deviations from the intended design during the machining. The first deviation was the extra holes in the wrong position on the angled beams. This was then accepted because it did not affect the performance of the enclosure. Additionally, to ensure the assembly fit, holes in the panels were noticeably widened. After receiving the enclosure from the Machine Shop, a rotary tool was used to sand and buff the rough edges. This made the enclosure safer to handle without the need of gloves for subsequent integration. The black cloth for the openings was cut to size and taped to the inside face of the enclosure. During integration, the modular design was proven to be very useful, as it allowed for easy disassembly of the top panel for additional machining. The additional machining was done for the slot, the wiring, and the standoffs that support the electrical parts. Figure 19 shows the progress of the enclosure.

(a) During manufacturing. (b) After sanding rough edges.

Figure 19: Before and after sanding of the enclosure.

Finally, a 3D printed enclosure and logo was designed to encase the electrical hardware on top of the enclosure. Unfortunately, the 3D printer broke and the enclosure design was not fully manufactured, and only the logo and a smaller enclosure was used for the symposium. The enclosure and the logo can be seen in Figure 20.

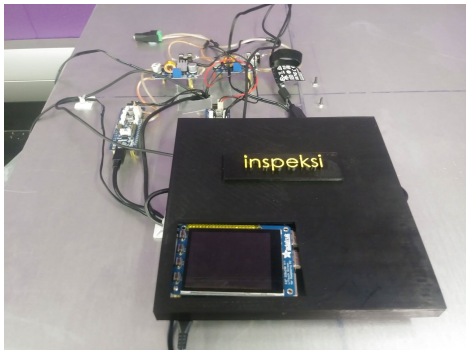


Figure 20: Electrical hardware on top of enclosure.

2.3.2 Robot Arm Manufacturing

The arm links and the base were all manufactured using a Fused Deposition Modeling (FDM) 3D printer. Specifically, the 3D printer used to print all the parts was the Prusa i3 MK2 [30]. The material used to print the parts was 1.75mm AMZ3D Polyactic Acid (PLA) [31], which was determined based on the weight and stress calculations during the design state of the robot arm.

The construction of the arm involves attaching the individual links to the motors, starting from the shoulder joint and working towards the wrist. The wrist motor utilized specialized rivets unlike the other motors that used standard bolts. The other 3 motors were mounted to the plastic links using socket cap M2 bolts and nuts. The camera was attached to the end effector appendage using Velcro, allowing for easy disassembling and assembling of the arm, which is important when reorganizing the wiring and replacing components.

Finally, the arm was assembled from scratch in less than an hour. The wires running to the devices along the arm were neatly bound together using zip-ties and sticky mounts at the end of the assembly process.

2.4 Commissioning

The system can be splitted into two main subsystems, the robot arm, and the enclosure. These two subsystems can be separately tested and verified, and then integrated and tested again as a whole.

2.4.1 Enclosure

The manufacturing of the enclosure was commissioned to the Machine Shop. While working with the machinist, the received product had to be tested for acceptability with respects to the design. The enclosure was measured to the drawing specifications and inspected for visual structural defects. The enclosure was also checked for sharp edges, as well as ease of assembly and disassembly. To test the sturdiness, the enclosure was shaken both before and after the robot arm and felt cloth were added. The enclosure was tested again for sturdiness while the robot arm was moving. The environment within the enclosure also needed to be tested in order to verify that the ambient light is blocked and the camera could see scratches on the inspected parts. The camera was manually positioned in order to get the part into the camera view. This was performed using a number of defective parts of different shapes, materials, and colours to simulate the conditions of the final system. The pictures were then evaluated qualitatively. Overall, the enclosure testing was successful and met the specification for the enclosure design.

2.4.2 Robot Arm

To commission the robotic arm, the motors were first evaluated. Each motor was run through a sweeping motion to verify its range of operation. Individual joints were then assembled to verify that they could move through the specified range of motion. The joints were connected and the daisy chained serial communication bus was tested. This test involved reading from each motor and writing to each motor. After this test was completed, the backdriveability and playback features were also tested, while verifying that the motors could handle the expected torque loads.

The robot arm is solely responsible for moving the camera into the specified positions so its ability to perform that function was evaluated. To ascertain the performance of the robotic arm, the arm was run through a few user selected motions and the position of the end effector was compared. Positions exercising the arm's joints were recorded by a user and they were played back to be evaluated. The joints were inspected visually to identify deviations from the specified pose, and the deviations were verified by comparing it to the encoder readings.

2.4.3 Inspection Algorithm

The inspection algorithm was tested independently before being integrated with the robot arm. This was done after the enclosure was fully manufactured, so that the pictures taken by the camera inside the enclosure reflect the actual testing conditions. In order to test the algorithm, a temperature sensor testing unit from Ecobee was placed inside the finished enclosure to be inspected. The camera was then placed manually in several different positions around the inspected part to take pictures. These

pictures were then processed using the trained Mask R-CNN model. Finally, the model's outputs (i.e., the segmentation map) were evaluated qualitatively by comparing the outputs and the actual defects on the test unit.

2.4.4 Data Communication

To test that the robot arm and the server can communicate properly without major delays, the communication system was tested by running a server on the RaspberryPi. A test Python script was developed and run on the server computer to act as an HTTP client that can send HTTP requests to the RaspberryPi. The RaspberryPi would reply to the server computer's requests using the three comands depending on which state it is in, as discussed in Section 2.2.4, periodically. As the RaspberryPi responds to HTTP requests with these commands, the messages received by the server computer was received, and the time required for the server computer to receive each command was recorded to ensure that the message can be sent and received in timely manner.

2.5 Testing and Performance

To appropriately test the performance, the system was evaluated using constraints and criteria spelt out in the previous report [4]: processing time (must be less than 20 seconds), lack of contact with the inspected part, and the accuracy of the defect detection. The system was tested to scan a defective Ecobee temperature sensor unit. Four scanning positions were recorded and the arm was played back four times.

Before evaluating the detection method, the performance of the robot arm was first evaluated, following the methodology outlined in Section 2.4.2. Table 5 summarizes the testing of the robot arm.

Table 5: Steady state robot arm joint orientation error.

Joint	**Error(Degrees)**
Wrist Joint	0.3
Elbow Joint	0.5
Shoulder Rotation	0.5
Shoulder Raise	3.5

The error in the shoulder raise joint manifested as a 1.5cm position error in the height of the camera when the arm was set in a horizontal position. The source of this error comes from the weight of the arm which deformed the plastic turntable piece in the shoulder joint. To counteract this error, an offset was added to the shoulder raise joint. This was deemed an acceptable solution because the joint error was consistent at different positions. After this adjustment was made, it was concluded

that the errors in joint orientations were small enough.

Once the arm was tested, the full systems was tested by running the arm through the user specified path while taking pictures along the way to be processed by the inspection algorithm. The data communication performance was also to ensure the commands from the robot arm were received correctly by the server computer in timely manner. From this test, it was concluded that the server computer never missed any messages sent from the robot, and the server computer received the message within 0.03 seconds after each command was sent by the robot.

The results of defect detection can be seen in Figure 21. In this figure, the identified scratches from a single run are highlighted. The four images cover the class A and class B surfaces of the part. The processing of all four images took 8.3 seconds, which is less than the 20 seconds required, and the arm did not collide with the part during scanning.

An average of 10.5 scratches were identified in the 4 scans taken from a total of 10 scratches on the aforementioned temperature sensor. The scratches can be seen in 21, not all scratches are present in one image as multiple angles were photographed. Furthermore there was a single false positive consistent to all the scans, where the detection algorithm mistook an embossment for a scratch. This resulted in 95.2% precision. The accuracy was consistent between tests, fluctuation between 100% and 95% and the precision remained a stable 95.2%. This consistency within the 4 scans speaks to the repeatability of the system.

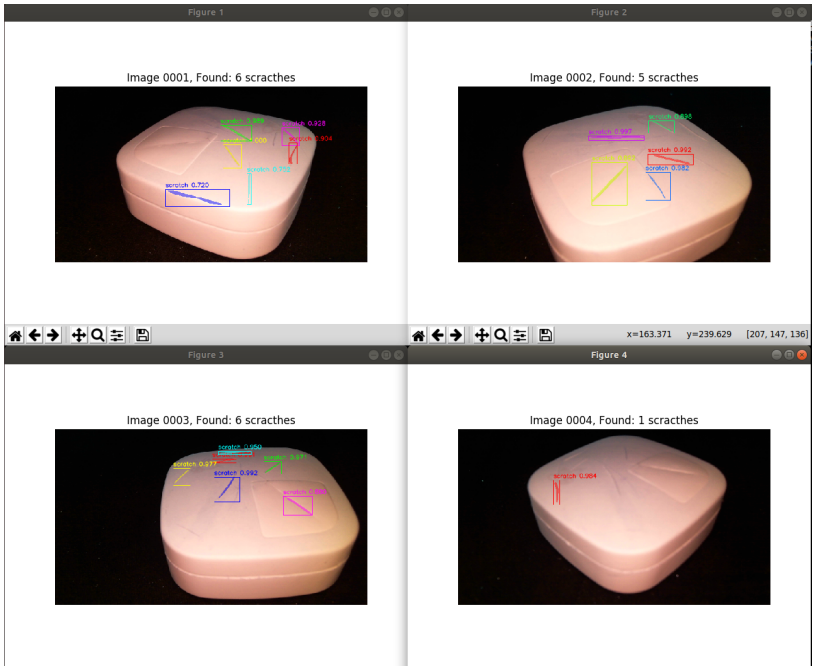


Figure 21: Results of processing images captured by the robot arm using Mask-RCNN.

3 Schedule and Budgeting

In this section, scheduling and budgeting of the project are discussed.

3.1 Schedule

The Gantt Chart in Figure 22 shows the proposed schedule estimation that was meant to be followed. The estimate did a good job of understanding the start times and the bulk of time spent on a specific task. The schedule was referred to during the team's weekly planning to give estimates of what needed to be done. The schedule did poorly when it came to accounting for the time required to make modifications or for repairing broken parts. For example, the robot arm needed to be reprint a few times to account for wiring complications, mechanical issues, and broken parts.

Starting in January, the team worked on the manufacturing of the enclosure and the robot arm. The enclosure was completed two weeks after a few redesigns and some assistance from the University of Waterloo E3 Machine Shop. The robot arm was 3D printed and assembled by the end of the month given that the main bottleneck was the arrival of the motors. The wiring of the electronics was worked on in parallel, and everything was fully assembled by the end of January.

The next step was to work on the software related items. A couple of weekends were spent on dataset collection required to train the Mask R-CNN model. Another important task completed in this stage was the development of the record-and-playback feature of the arm. This took the bulk of February, given that the team took a week off for the Winter Term Break.

March was dedicated to integration, testing, and any final additions to the system to improve it for the symposium. March was also used for poster design and symposium demo preparation, which included designing a full demo to show how the systems work.

Overall, the Gantt Chart in Figure 22 was a good estimate for the required time for completing the tasks. It was useful for planning when tasks were meant to be started, and how long they should take in order to complete the project by mid-March.

3.2 Budget

The detailed line-by-line budget is provided in the Bill of Material given in Appendix B.

The total spent budget was $880.05 CAD, which is almost $100 CAD less than the original estimated budget of $987 CAD [4]. The actual cost of the system is lower because of borrowing

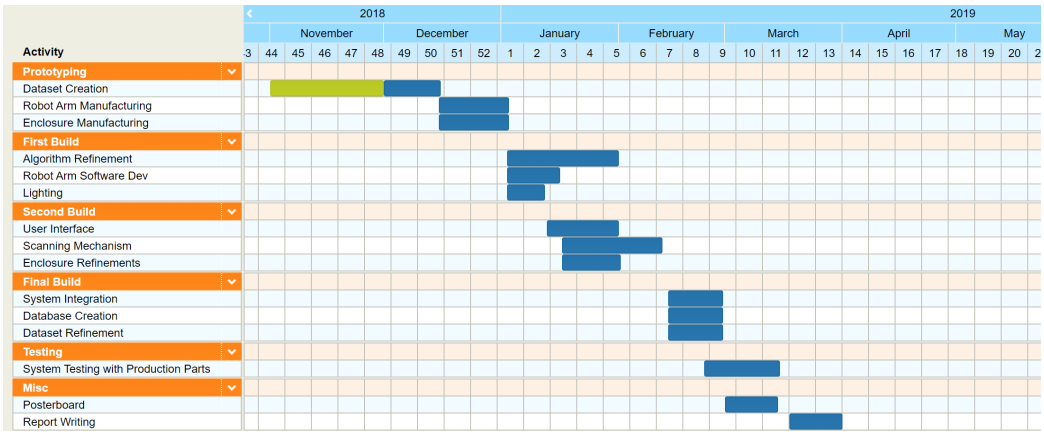


Figure 22: Estimated project timeline [4].

expensive hardware from the Mechatronics Equipment Surplus System (MESS) such as the RaspberryPi 3B+, and from taking advantage of the University of Waterloo E3 Machine Shop's student prices, and cheap materials. This budget also includes additional items purchased in order to create the dataset for the deep learning model.

The new total spent budget is below the constraint of $1000 CAD. The most expensive components are the enclosure, the motors, and the camera.

4 Conclusions and Recommendations

There is a need for low-cost, fully-automated, solution for high quality visual inspection of manufactured parts. In this report, the design of the low-cost automated visual inspection tool was finalized. Some deviations and modifications from the proposed design in [4] were reported. The results of the testing as well as the performance of the product were also discussed in this report.

Although some modifications were made from the initial design, these deviations were considered as minor changes, and the design concept and its functionality did not change. The design still involves an enclosure with a robot arm and a camera attached at the end-effector of the arm. Minor modifications were made to the enclosure design such as the change in dimension of the angle brackets, hole locations, and the use of notches to the top plate support beams (see Section 2.2.1). Some changes were made to the design of robot arm links to reduce interference and reduce overall weights as discussed in Section 2.2.2. Instead of sending image data through Wi-Fi, the camera was decided to be connected to the server computer directly, and the communication between the robot arm and the server computer was done using HTTP request. Finally, the inspection algorithm was modified from MobileNets-SSD [24, 32] to Mask-RCNN [2]. Overall, the whole systems worked as expected without any failures. The arm was able to move via backdriveability feature and communicate properly with the server computer to take pictures as instructed. Finally, the instance segmentation model was also able to detect scratches and scuffs with a reasonable performance.

Mask R-CNN [2] was implemented as suggested in [4]. However, there are still room for improvements and thus several suggestions were made. First, training Mask R-CNN with standard training algorithms (i.e., back-propagation [33] with gradient descent) requires fairly huge dataset, which may not always be available in the real world settings. Thus, meta-learning algorithms such as the Model Agnostic Meta Learning (MAML) [3] will be interesting to be investigated as it allows one to rapidly train and adapt the model to new data. Finally, due to the fact that collecting dataset from scratch is time-consuming, another suggestion is to train the model using only synthetic dataset and then adapt the synthetic model to work in the real world. An example of this type of transfer learning was proposed in [34] by using domain randomization. To improve the performance of the arm, the turntable piece should be redesigned to reduce the deflection currently experienced. This could be done by using a different materials to manufacture the part or redesigning the part to be able to prevent deflection. Furthermore, the accuracy of the defect detection algorithm can also be improved by improving upon the camera's autofocus feature, possibly with a laser distance sensor as initially recommended.

5 Teamwork Effort

This project required dedicated effort for component design, system integration and product validation. The team had several meetings to understand the high level design and decision making in each of the components. The team also used task management software, Trello, to keep track of the work available and ensure fair distribution of tasks. For efficiency, each individual on the team was responsible for different aspects of the design and manufacturing process. The team worked very well with each other with regular meetings. The team was also able resolve issues quickly as they arose. Every member demonstrated a willingness to support others to meet the scheduled deadlines. During some system level problems, the team scheduled meetings to come up with out-of-the-box solutions. Some examples include the communication system for the camera to know when to take a picture and the robot motion algorithm for smooth motion. Each member was pivotal to the completion and success of the project and the individual contributions are listed below.

Herman Grewal was primarily responsible for the enclosure design. Herman aided Oluwatoni Ogunmade and Lalit Lal in the redesign of the robot arm. He also assisted in the system level design of the robot path planning and communication. Herman also supported Mohamed El Shatshat with collecting parts and images for the training data set.

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 co-designed and implemented the robotic control software with Toni, as well as the communication system and integration with Toni and Rey.

Mohamed El Shatshat was responsible for product development, website, and poster design. He also helped with the enclosure manufacturing and testing, as well as sourcing supplies for dataset creation. He was pivotal in the final system integration and meeting the scheduled milestones.

Oluwatoni Ogunmade was responsible for the robot arm redesign and testing. He also created the architecture for controlling the arm with Lalit, and worked on the final system integration including the communication to the camera for taking pictures.

Rey Reza Wiyatno was responsible for implementing, testing, and validating various state of the art methods for defect detection, as well as data communication. Rey created the dataset needed to train both the object detection and instance segmentation models. He was also responsible with 3D printing of the robot arm and other components.

Acknowledgement

The authors thank Kevin Eng (Ecobee), Aleksey Tsaplin (Ecobee), Eric Stackpole (OpenROV), Brian Grau (OpenROV), Adam Mack (Flex), and Wim Deweerd (Google Hardware) for validating the problem, as well as identification of constraints and criteria. The authors also thank Ecobee for providing production parts with defects to validate the proposed system. Additionally, Robert Wagner's assistance in the enclosure manufacturing is also acknowledged. Finally, the authors thank Professor Ayman El-Hag and Professor Jan Huissoon for the supervision and guidance throughout the term.

References

[1] 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.

[2] Kaiming He et al. ‘‘Mask R-CNN’’. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017.

[3] 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.

[4] Herman Grewal et al. ‘‘Intelligent Manufacturing Inspection Tool’’. In: *MTE 481 Report* (2018).

[5] Logitech. *Logitech C270 HD Webcam*. 2018. URL: https://www.logitech.com/en-ca/product/hd-webcam-c270 (visited on 11/27/2018).

[6] Logitech. *Logitech C525 HD Webcam*. 2018. URL: https://www.logitech.com/en-ca/product/hd-webcam-c525 (visited on 11/27/2018).

[7] Logitech. *Logitech C920 HD PRO Webcam*. 2018. URL: https://www.logitech.com/en-ca/product/hd-pro-webcam-c920 (visited on 11/27/2018).

[8] Raspberry Pi Foundation. *Camera Module V2*. 2018. URL: https://www.raspberrypi.org/products/camera-module-v2/ (visited on 11/27/2018).

[9] G. Bradski. ‘‘The OpenCV Library’’. In: *Dr. Dobb's Journal of Software Tools* (2000).

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. ‘‘Deep learning’’. In: *Nature* 521 (2015), 436 EP –. URL: http://dx.doi.org/10.1038/nature14539.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[12] 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.

[13] 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.

[14] 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.

[15] David Held, Sebastian Thrun, and Silvio Savarese. ‘‘Learning to Track at 100 FPS with Deep Regression Networks’’. In: *European Conference Computer Vision (ECCV)*. 2016.

[16] Luca Bertinetto et al. ‘‘Fully-Convolutional Siamese Networks for Object Tracking’’. In: *ECCV 2016 Workshops*. 2016, pp. 850–865.

[17] 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.

[18] 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.

[19] Ross Girshick. ‘‘Fast R-CNN’’. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2015.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[27] Morgan Quigley et al. ‘‘ROS: an open-source Robot Operating System’’. In: *ICRA Workshop on Open Source Software*. 2009.

[28] Robotis. *Controller Compatibility*. 2019. URL: http://emanual.robotis.com/docs/en/parts/controller/controller_compatibility/ (visited on 03/17/2019).

[29] 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.

[30] Prusa Research. *Original Prusa i3 MK2S kit*. 2019. URL: https://www.unipunch.com/support/charts/material-specifications/ (visited on 03/17/2019).

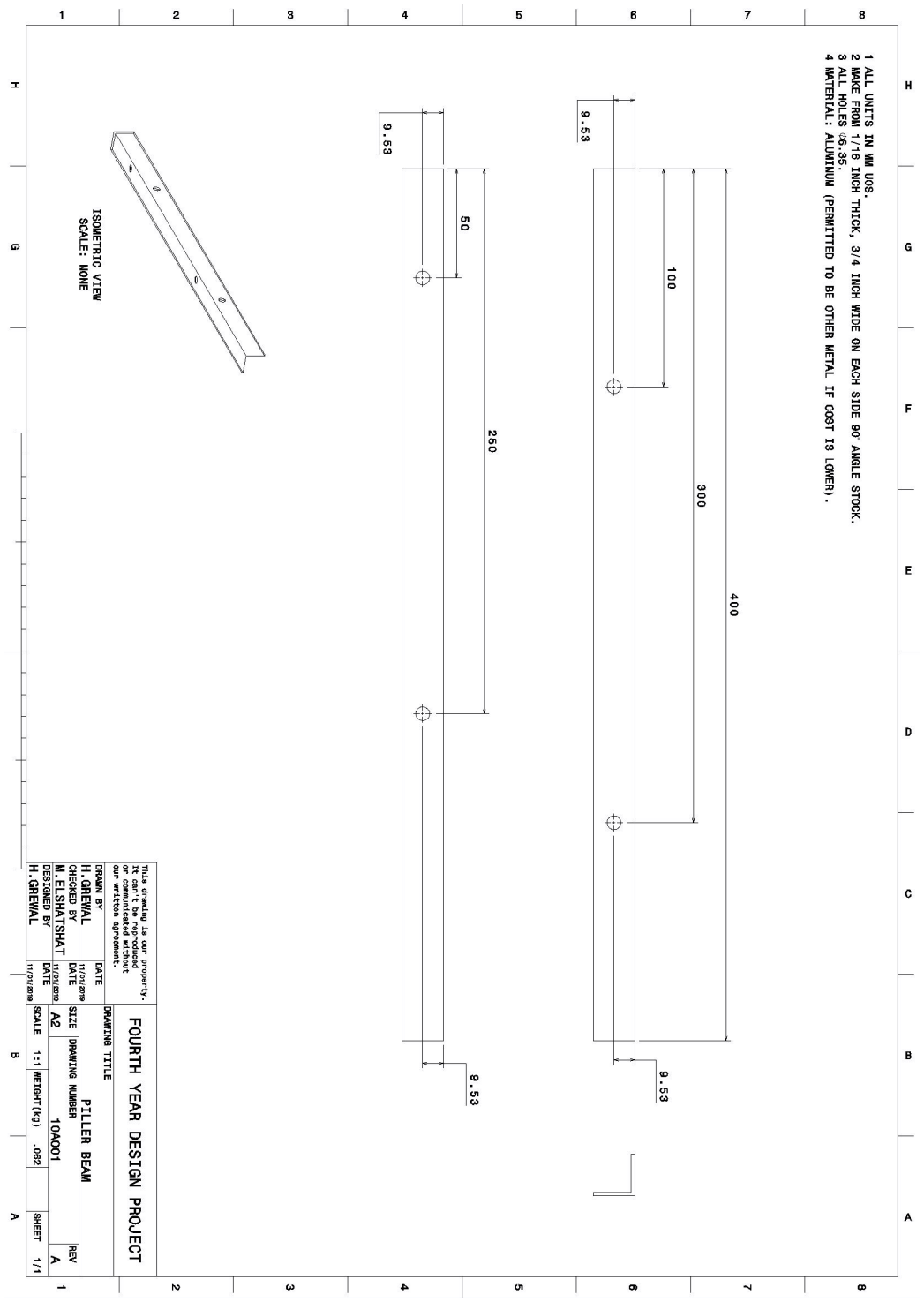
[31] Amazon. *AMZ3D 1.75mm PLA 3D Printer Filament, Black, 1 Kg spool (2.2 lbs)*. 2019. URL: https://www.amazon.ca/AMZ3D-1-75mm-Printer-Filament-Black/dp/B01BZ5ND8O/ref=sr_1_1?keywords=amz3d&qid=1552801563&s=gateway&sr=8-1 (visited on 03/17/2019).

[32] Wei Liu et al. ‘‘SSD: Single Shot MultiBox Detector’’. In: To appear. 2016. URL: http://arxiv.org/abs/1512.02325.

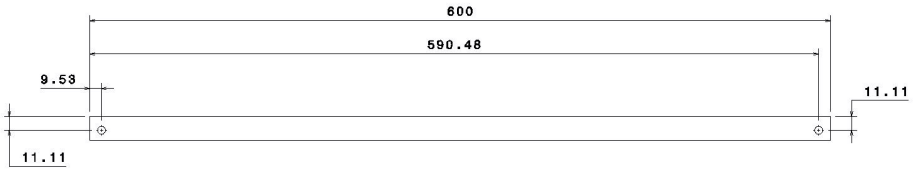
[33] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. ‘‘Learning representations by back-propagating errors’’. In: *Nature* 323 (1986), 533 EP –. URL: http://dx.doi.org/10.1038/323533a0.

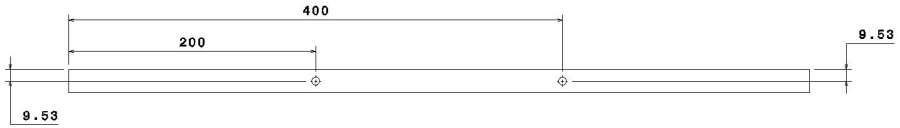
[34] Jonathan Tremblay et al. ‘‘Training Deep Networks With Synthetic Data: Bridging the Reality Gap by Domain Randomization’’. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2018.

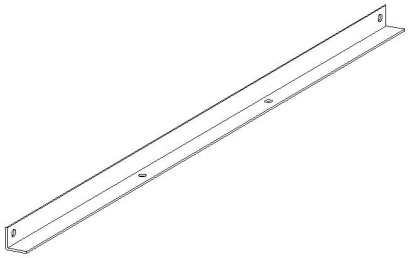
Appendix A Detailed Part Drawings



1 ALL UNITS IN MM UOS.
2 MAKE FROM 1/16 INCH THICK, 3/4 INCH WIDE ON EACH SIDE 90° ANGLE STOCK.
3 ALL HOLES Ø6.35.
4 MATERIAL: ALUMINUM (PERMITTED TO BE OTHER METAL IF COST IS LOWER).







ISOMETRIC VIEW
SCALE: NONE

DRAWN BY	H.GREWAL	DATE 11/01/2019
CHECKED BY	M.ELSHATSHAT	DATE 11/01/2019
DESIGNED BY	H.GREWAL	DATE 11/01/2019

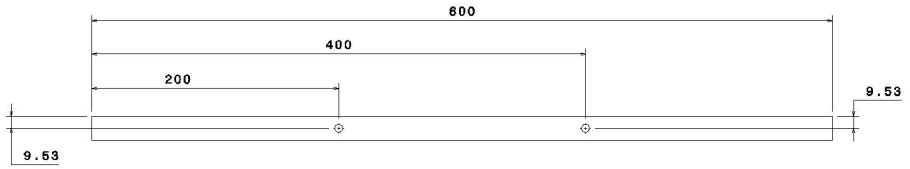
FOURTH YEAR DESIGN PROJECT

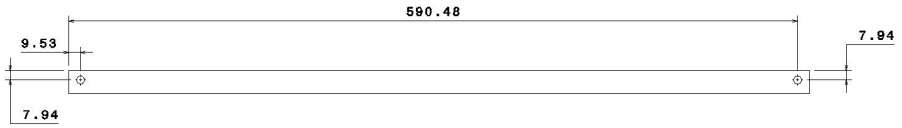
DRAWING TITLE: TOP SUPPORT BEAM

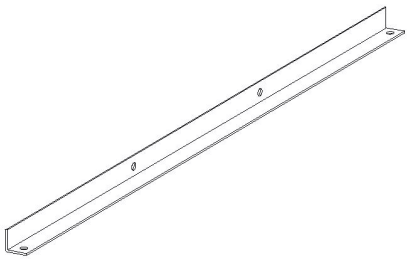
SIZE	DRAWING NUMBER	REV
A2	10A002	A

SCALE 1:2 | WEIGHT(kg) .084 | SHEET 1/1

1 ALL UNITS IN MM UOS.
2 MAKE FROM 1/16 INCH THICK, 3/4 INCH WIDE ON EACH SIDE 90° ANGLE STOCK.
3 ALL HOLES Ø6.35.
4 MATERIAL: ALUMINUM (PERMITTED TO BE OTHER METAL IF COST IS LOWER).







ISOMETRIC VIEW
SCALE: NONE

DRAWN BY	H.GREWAL	DATE 11/01/2019
CHECKED BY	M.ELSHATSHAT	DATE 11/01/2019
DESIGNED BY	H.GREWAL	DATE 11/01/2019

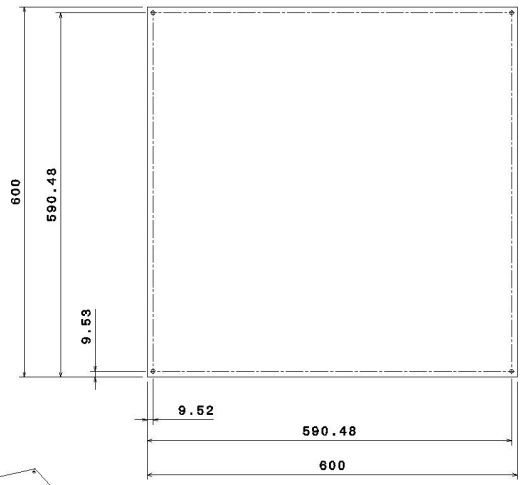
FOURTH YEAR DESIGN PROJECT

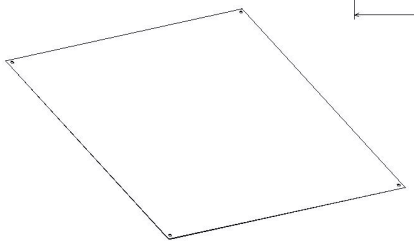
DRAWING TITLE: BOTTOM SUPPORT BEAM

SIZE	DRAWING NUMBER	REV
A2	10A003	A

SCALE	WEIGHT(kg)	SHEET
1:2	.084	1/1

1 ALL UNITS IN MM UOS.
2 MAKE FROM 22G STOCK (THICKER STOCK PERMITTED WITH VERBAL ACCEPTANCE OF DESIGNER).
3 ALL HOLES Ø6.35.
4 MATERIAL: STEEL





ISOMETRIC VIEW
SCALE: NONE

DRAWN BY	H.GREWAL	DATE 11/01/2019
CHECKED BY	M.ELSHATSHAT	DATE 11/01/2019
DESIGNED BY	H.GREWAL	DATE 11/01/2019

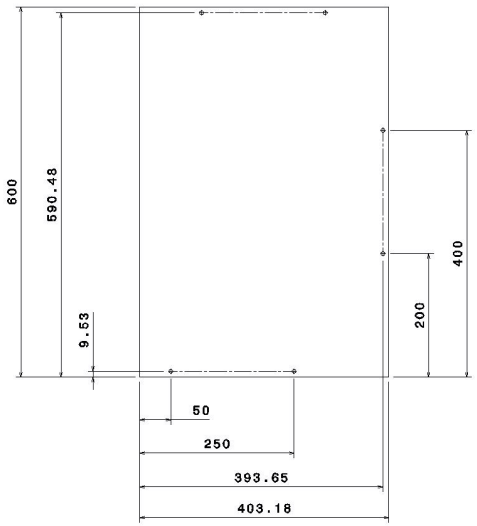
FOURTH YEAR DESIGN PROJECT

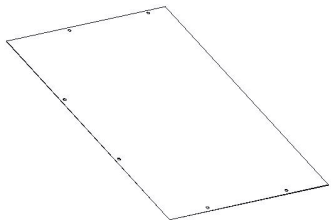
DRAWING TITLE: TOP PLATE

SIZE	DRAWING NUMBER	REV
A2	10A004	A

SCALE 1:4 | WEIGHT(kg) 2.245 | SHEET 1/1

1 ALL UNITS IN MM UOS.
2 MAKE FROM 22G STOCK (THICKER STOCK PERMITTED WITH VERBAL ACCEPTANCE OF DESIGNER).
3 ALL HOLES Ø6.35.
4 MATERIAL: STEEL (PERMITTED TO BE OTHER METAL IF COST IS LOWER).





ISOMETRIC VIEW
SCALE: NONE

DRAWN BY H.GREWAL	DATE 11/01/2019	
CHECKED BY M.ELSHATSHAT	DATE 11/01/2019	
DESIGNED BY H.GREWAL	DATE 11/01/2019	

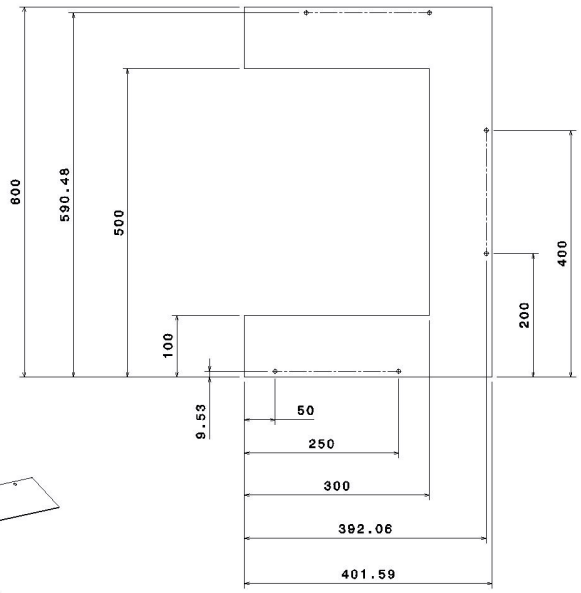
FOURTH YEAR DESIGN PROJECT

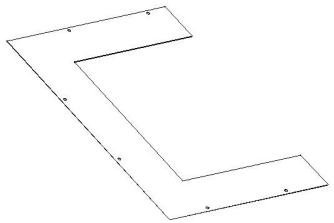
DRAWING TITLE: SIDE PLATE

SIZE	DRAWING NUMBER	REV
A2	10A005	A

SCALE 1:4 | WEIGHT(kg) 1.508 | SHEET 1/1

1 ALL UNITS IN MM UOS.
2 MAKE FROM 22G STOCK (THICKER STOCK PERMITTED WITH VERBAL ACCEPTANCE OF DESIGNER).
3 ALL HOLES Ø6.35.
4 MATERIAL: STEEL (PERMITTED TO BE OTHER METAL IF COST IS LOWER).





ISOMETRIC VIEW
SCALE: NONE

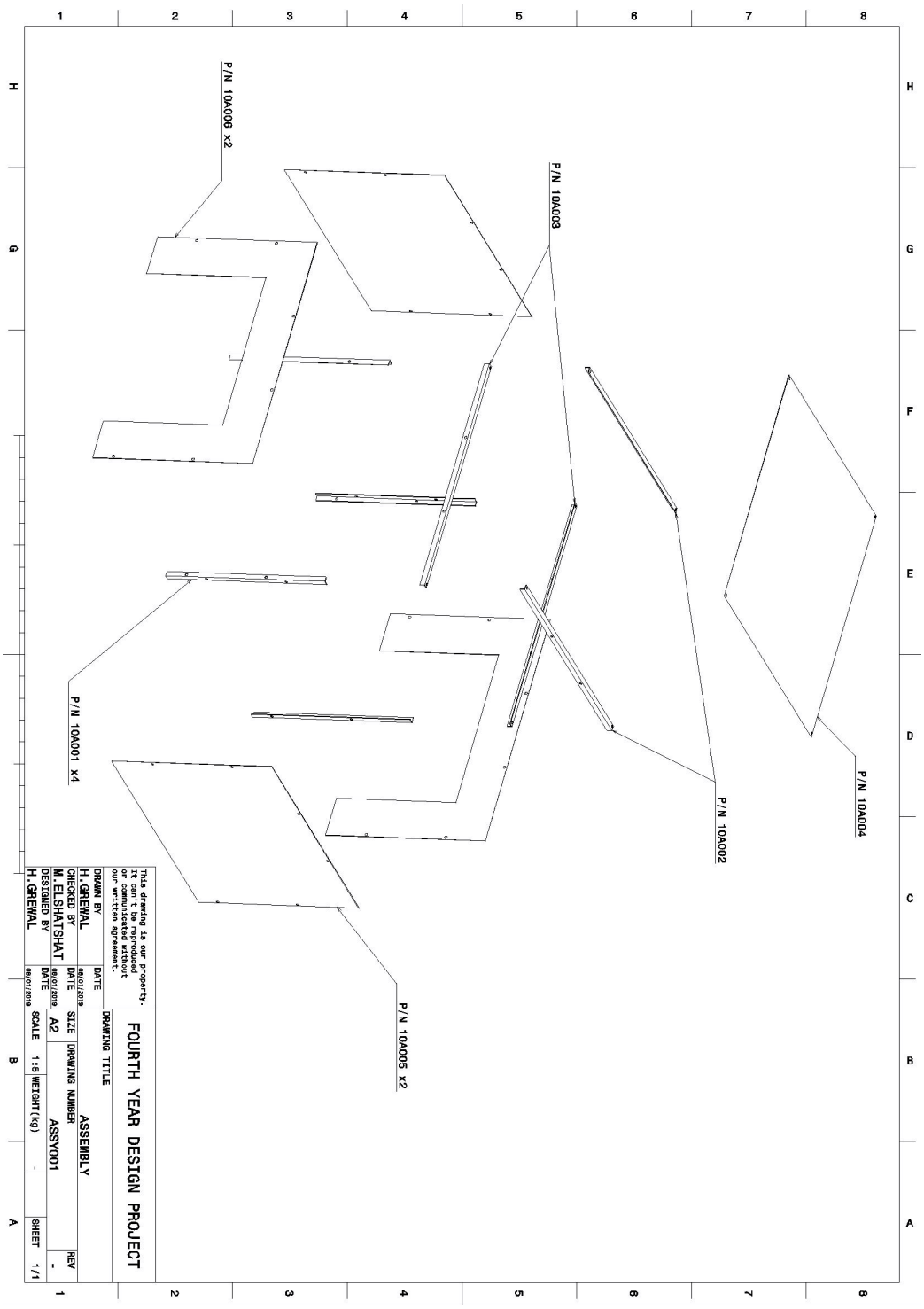
DRAWN BY	H.GREWAL	DATE 11/01/2019
CHECKED BY	M.ELSHATSHAT	DATE 11/01/2019
DESIGNED BY	H.GREWAL	DATE 11/01/2019

FOURTH YEAR DESIGN PROJECT

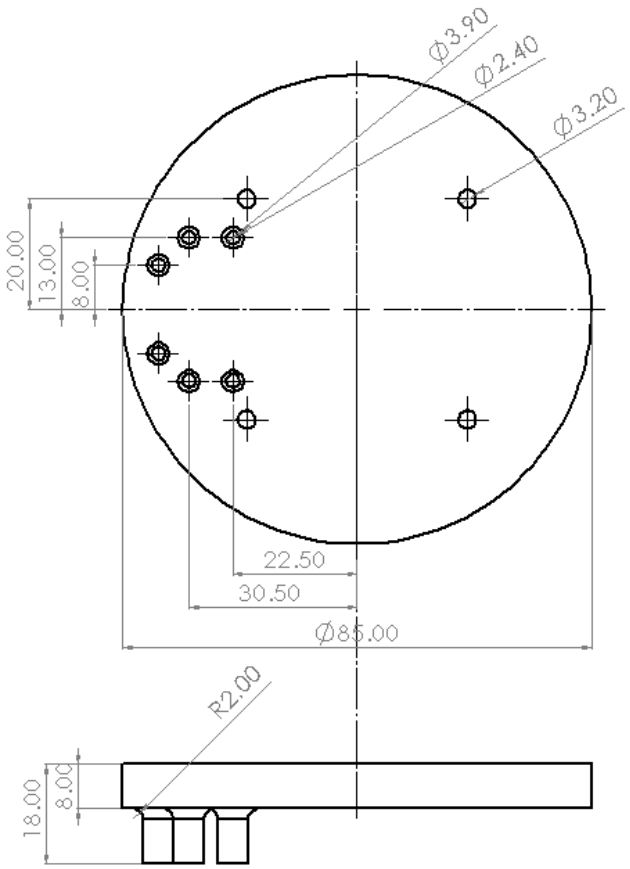
DRAWING TITLE: ENTER EXIT PLATE

SIZE	DRAWING NUMBER	REV
A2	10A006	A

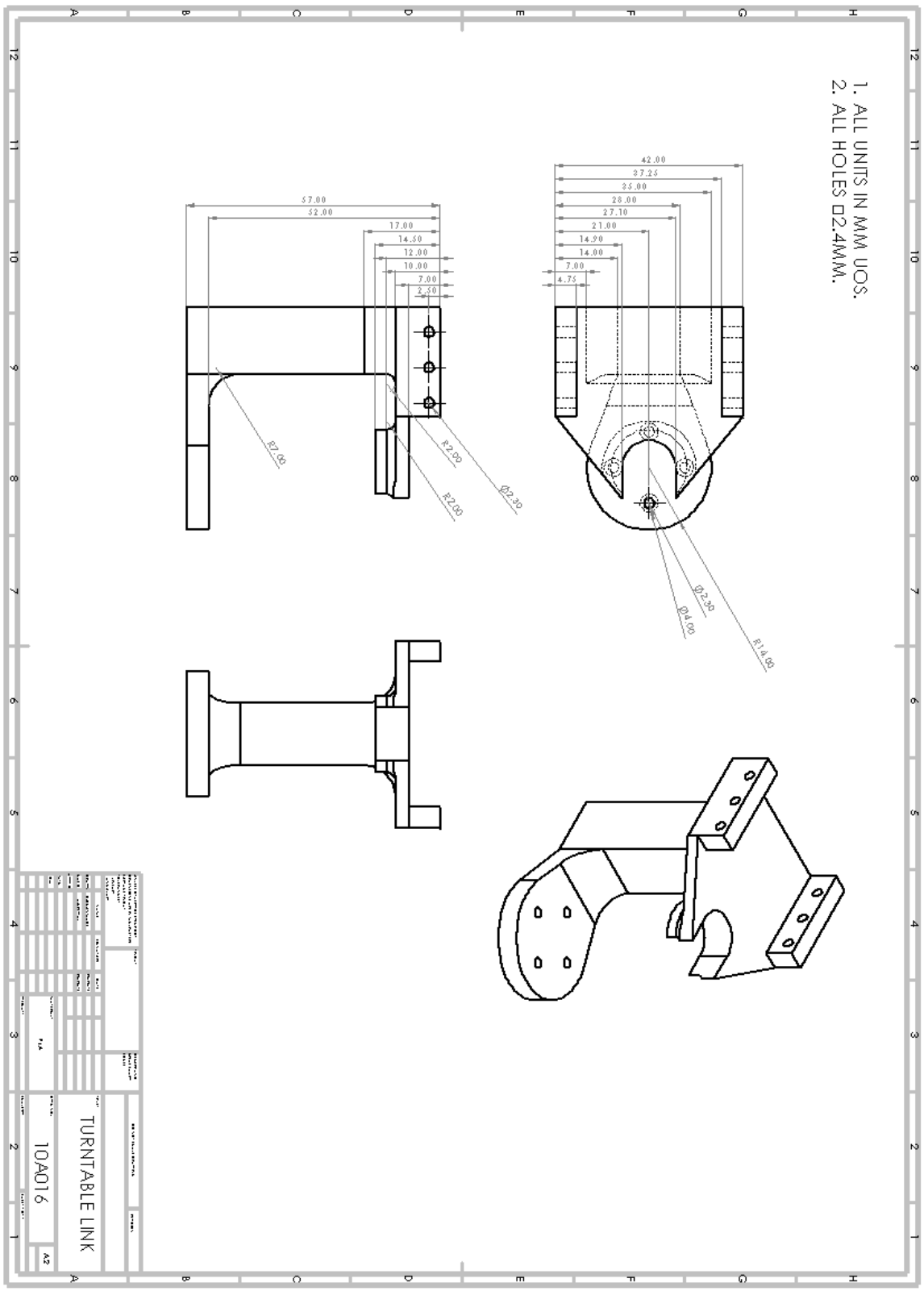
SCALE 1:4 | WEIGHT(kg) .753 | SHEET 1/1

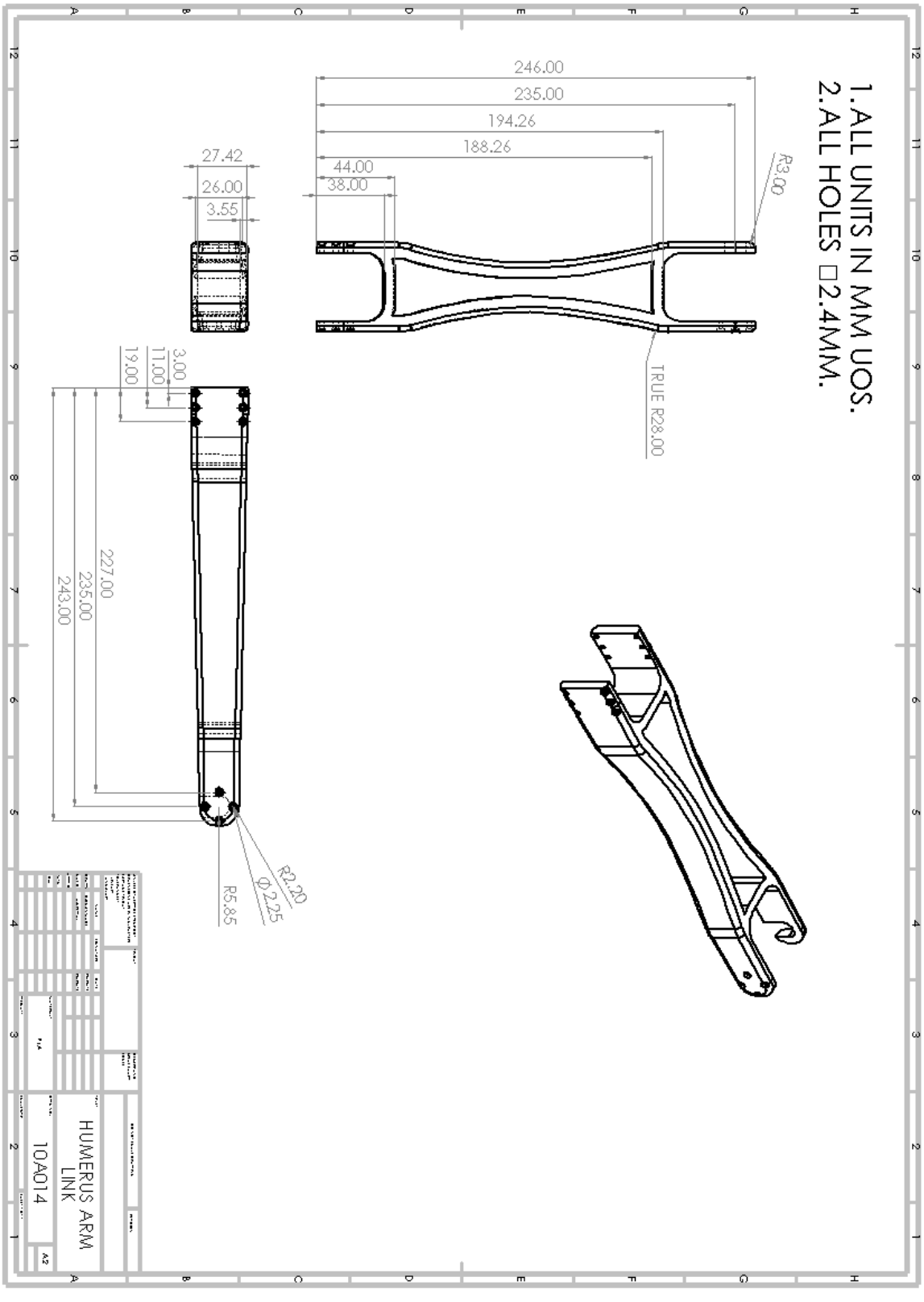

P/N 10A006 x2
P/N 10A003
P/N 10A004
P/N 10A001 x4
P/N 10A002
P/N 10A005 x2
This drawing is our property. It can't be reproduced or communicated without our written agreement.
DRAWN BY
H.GREWAL
DATE
09/01/2019
CHECKED BY
M.ELSHATSHAT
DATE
09/01/2019
DESIGNED BY
H.GREWAL
DATE
09/01/2019
FOURTH YEAR DESIGN PROJECT
DRAWING TITLE
ASSEMBLY
SIZE
A2
DRAWING NUMBER
ASSY001
REV
-
SCALE
1:5
WEIGHT(kg)
-
SHEET
1/1

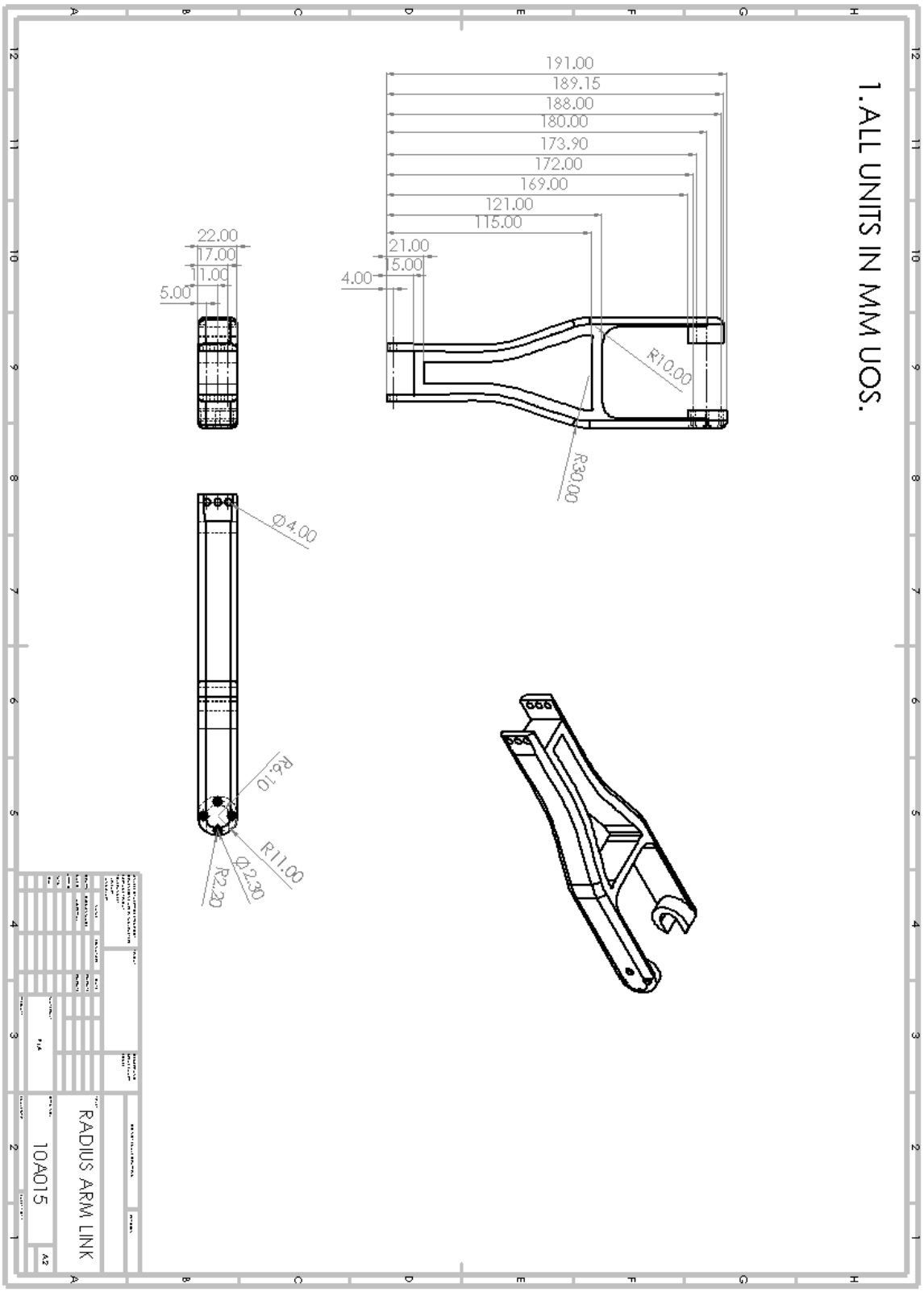
1.ALL UNITS IN MM UOS.



Arm Base Plate	
10A013	A2


1. ALL UNITS IN MM UOS.
2. ALL HOLES ⌑2.4MM.
57.00
52.00
17.00
14.50
12.00
10.00
7.00
2.50
R7.00
R2.00
R2.00
Ø2.30
42.00
37.25
35.00
28.00
27.10
21.00
14.90
14.00
7.00
4.75
Ø2.30
Ø4.00
R14.00
TURNTABLE LINK
10A016
A2


1. ALL UNITS IN MM UOS.
2. ALL HOLES □2.4MM.
246.00
235.00
194.26
188.26
44.00
38.00
R3.00
TRUE R28.00
27.42
26.00
3.55
3.00
11.00
19.00
227.00
235.00
243.00
R2.20
Ø2.25
R5.85
HUMERUS ARM LINK
10A014
A2


1.ALL UNITS IN MM UOS.
191.00
189.15
188.00
180.00
173.90
172.00
169.00
121.00
115.00
21.00
15.00
4.00
R10.00
R30.00
22.00
17.00
11.00
5.00
Ø4.00
R6.10
R11.00
Ø2.30
R2.20
RADIUS ARM LINK
10A015
A2

Appendix B Bill of Material

The bill of material is provided in Table 6. Note that prices are given in CAD.

Table 6: Bill of material.

Part Number	Description	Subsection	Quantity	Price/Unit	Full Price
Part Number	**Decription**	**Subsection**	**Quantity**	**Price/Unit**	**Total Price**
10A001	Enclosure Machining	Enclosure	1	208.49	208.49
10A002	Curtain	Enclosure	1	13.42	13.42
10A003	Playdough	Training	1	3.26	3.26
10A004	Phone Case	Training	1	3.00	3.00
10A005	Alphabet	Training	1	12.94	12.94
10A006	Dinosaur Bones	Training	1	1.50	1.50
10A007	Bowling Set	Training	1	4.00	4.00
10A008	Logitech C525	Camera	1	52.47	52.47
10A009	NeoPixel 16 LED Ring	Lighting	1	28.25	28.25
10A010	3D Printing Material	Arm	1	25.99	25.99
10A011	SMAKN DC-DC 5A Automatic Boost Buck Converter	Arm	3	19.68	59.04
10A012	M2 bolts and nuts	Arm	1		0.00
10A013	Servo bearing	Arm	3	2.23	6.68
10A014	XL320 OLLO Rivet Set	Arm	1	5.99	5.99
10A015	Dynamixel OPENCM9.04C	Electrical	2	19.9	39.80
10A016	Dyamixel XL320 OLLO Motor	Electrical	1	29.2	29.20
10A017	Dynamixel AX-12 Motor	Electrical	3	62.39	187.17
10A018	Molex Female	Electrical	1	1.99	1.99
10A019	Molex Male	Electrical	1	2.95	2.95
10A020	Cable Ties	Electrical	1	3.95	3.95
10A021	Cable Tie Mounts	Electrical	1	3.1	3.10
10A022	Fuse 5A	Electrical	1	2.5	2.50
10A023	Fuse 1A	Electrical	1	2.4	2.40
10A024	Fuse Block	Electrical	4	1	4.00
10A025	AC/DC Power Supply 100W	Electrical	1	23.98	23.98
10A026	PiTFT Plus Touch Screen Display	User Interface	1	86.99	86.99
10A027	Raspberry Pi 3 B+	Control	1	66.99	66.99
				Total	**880.05**