AUTOMATED QUALITY CONTROL OF BISCUITS USING COMPUTER VISION

A Final Year Project Report

Presented to

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

Department of Mechanical Engineering

NUST

ISLAMABAD, PAKISTAN

In Partial Fulfilment
of the Requirements for the Degree of
Bachelor of Mechanical Engineering

by

Taimoor Shahzad

Muhammad Hamza Qaisar

Shaheer Ahmad Khan

Kalsoom Nazish

June 2022

EXAMINATION COMMITTEE

We hereby recommend that the final year project report prepared under our supervision by:

Taimoor Shahzad	265100
Muhammad Hamza Qaisar	249464
Shaheer Ahmad Khan	262224
Kalsoom Nazish	259005

Titled: "AUTOMATED QUALITY CONTROL OF BISCUITS USING COMPUTER VISION" be accepted in partial fulfilment of the requirements for the award of MECHANICAL ENGINEERING degree with grade ___

Supervisor: Dr. Muhammad Usman Bhutta, Assistant Professor	____________________ Dated:
Committee Member: Dr. Jawad Aslam, Assistant Professor	____________________ Dated:
Committee Member: Dr. Karam Dad, Assistant Professor	____________________ Dated:

______________________ (Head of Department)

___________ (Date)

COUNTERSIGNED

Dated: ________________

__________________________ (Dean / Principal)

<u>ABSTRACT</u>

This thesis is intended to describe the design and fabrication of a solution for Automated Quality Control of the Biscuits through Computer Vision. The project is aimed at providing an optimized solution for the quality control of biscuits which can be deployed at local industries in order to automate the process. The goal was to design, fabricate, test, and validate a working prototype of the solution.

Our target was to provide a well-integrated automated solution for the industries in order to localize this technology, to reduce human interaction in the production line and to ensure hygiene of the eatables. We also aim to reduce the labour cost and make the process cost effective. Its scope can be expanded to industries where quality is currently ensured via human vision.

To materialize a machine with such capabilities, we had to work in parallel to take right decisions regarding the hardware involved, and the software algorithm driving it. The complete setup should be able to differentiate between standard and sub-standard biscuits along with a responsive actuation mechanism driven by microprocessors.

Two prototypes were developed to achieve the set objective for the project. The first prototype V1 was to be developed as a proof of concept, and the insights gained from the first attempt assisted us in fabricating a much efficient, practical, and robust final prototype V2.

The objectives were successfully achieved with some minor shortcoming in the final prototype. Further work can be done on it to improve the colour resolution, robustness, and

serviceability. Machine learning can also be introduced in the algorithm for enhanced capabilities.

ACKNOWLEDGMENTS

First and foremost, we thank Allah Almighty for His blessings which allowed us to complete the project efficiently and effectively.

We would like to express our gratitude towards our supervisor **Dr Muhammad Usman Bhutta** and co-supervisor **Dr Muhammad Jawad Khan** for his invaluable assistance and insights to our project. Our sincere thanks also go to **Dr Khawaja Fahad Iqbal** for helping us and guiding us by providing critical analysis throughout our project. He has shared his valuable views and techniques to further improvise our project. We would also like to thank computer vision lab engineers at NICHE, NUST for their support. This project would not have been successful without our mentors' and professors' assistance.

Last but not the least, we would like to thank our parents and family for their support and motivations. It would not have been possible without their prayers and efforts.

ORIGINALITY REPORT

AUTOMATED QUALITY CONTROL OF BISCUITS USING COMPUTER VISION

ORIGINALITY REPORT

10%	9%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	pythonmana.com Internet Source	1%
2	Submitted to Higher Education Commission Pakistan Student Paper	1%
3	bbs.huaweicloud.com Internet Source	1%
4	labvolt.festo.com:443 Internet Source	<1%
5	github.com Internet Source	<1%
6	Submitted to Weber State University Student Paper	<1%
7	ndltd.ncl.edu.tw Internet Source	<1%
8	Submitted to University of Nottingham Student Paper	<1%
	blog.csdn.net	

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ROI	Region of Interest
CV	Computer Vision
IDE	Integrated Development Environment
QC	Quality Control
AC	Alternating Current
DC	Direct Current
TV	Television
ECH	Environment
PSQCA	Pakistan Standards & Quality Control Authority
ISO	International Organization for Standardization
EHS	Environmental Health and Safety
ARM	Advanced RISC Machine
RISE	Robotics and Intelligent Systems Engineering
V1	Version 1
V2	Version 2
FYP	Final Year Project
SMME	School of Mechanical and Manufacturing Engineering
NUST	National University of Sciences and Technology
SBC	Single Boards Computer

NOMENCLATURE

v	Linear Velocity output of encoder
ω	Angular Velocity output of encoder
r	Radius of inner wheel of encoder
n	No of slots in the internal wheel of encoder
t	time taken by each slot in the encoder across the laser gate
ɵ	Flap Angle
f	Oscillating frequency of flap
T	Time period required by flap oscillation
T_s	Settling time by flap

CHAPTER 1: INTRODUCTION

Food industry is the one of the biggest industries in Pakistan. In today's era of the innovation and technology most of the industries in developed countries have deployed the automated quality control solutions for biscuits while manual labour is employed in Pakistan. In Pakistan, biscuit manufacturing industry mainly comprises of machinery required for mixing and baking of the dough. Whereas inspection and rejection of the biscuits is carried out manually and is based entirely on human vision. This poses a critical challenge in quality assurance along with the personal health and safety of labour due to exposure to high temperature ovens.

In this project, we devised a solution for the industries which are willing to automate their process of quality control for biscuits. Its scope can be expanded by implementing this integrated solution in industries which require detection of visual discrepancies in their products.

1.1 Motivation of Work

Here is a detailed description for motivation of our project:

1.1.1 Hygiene Requirement

The major motivation and aim of the project is to minimize the human intervention in the production line. Human interaction poses a problem of hygiene which is a major concern in eatable industries. The target is to lower the rate of contaminants' transfer due to human intervention in order to improve the shelf life of the biscuits. Another vital concern is to meet ISO 22000:2005 [1] standard which specifies requirements for a food safety management system where an organization in the food chain needs to demonstrate its ability to control food safety hazards in order to ensure that food is safe at the time of human consumption along with the local Pakistan Standards & Quality Control Authority (PSQCA) standards that verifies through quarterly inspections of the production premises, laboratory, and collection of random product samples which are then tested in their own quality control labs. These two standards are necessary for the sales in national and international markets.

1.1.2 Introduce Automation

One of our primary motivations is to introduce automation in the process to replace the human invigilation and inspection process which tends to be slow. Computer Vision in comparison offers advantages of objective and consistent assessment. Although the setup cost would be high, but it would ultimately yield profit after the break-even point. Labour cost would be reduced which would ultimately result in lower running cost. Running time of the process will also be improved.

1.1.3 Lower Error Rate in Fault Detection

In most of the industries inspection and rejection continues to be primarily manual which mainly rely on the human vigilance. Substantial responsibility of the quality assurance by humans not only poses a problem to hygiene but is also tedious. Human intervention is prone to inconsistency and errors which can result in poor quality control. Human resource redundancy is required to ensure minimal down time which can increase running cost.

1.2 Problem Statement

Human inspection is prone to inconsistency and hygiene problems which is prime concern in biscuit industry. Therefore, to overcome this problem we devised a practical system for automated real time in-line quality control which will ensure the reliability and consistency in our production. This presents the following three critical challenges which are posed by human intervention in production line:

1. Quality assurance through human vision is susceptible to inconsistency which can result in customer unsatisfaction.
2. An EHS (Environmental Health and Safety) organization for the human workforce who work in close proximity of the machinery and ovens. [2]
3. Improve hygiene practices by reducing the interaction of humans in the production line.

1.3 Objective

The primary objective of this project is to devise a practical system for automated quality control of the biscuits using computer vision. The scope of the system can be expanded to industries which maintain quality assurance by human vision. Following are the main objectives of project:

- Design a fully integrated system.

- Instrumentation and data acquisition of the working prototype.
- Manufacturing and assembling of a prototype simulating under actual conditions.
- Automated real time quality control and inspection of the machine prototype.

CHAPTER 2: LITERATURE REVIEW

In biscuit industries, a large amount of biscuit quality control is carried out by manual inspection. This creates inconsistency, errors, and slows the process of the quality control. The main purpose of our literature review is to study:

- How Quality Control solution providers (local or international) ensure quality control of biscuits?
- Solutions provided by researchers.
- The challenges they faced and the concept they developed to overcome them.

An extensive literature review has been done to determine how quality control operations driven by computer vision are implemented and how can they be improved. This included studying resources such as:

- Research papers and articles.
- Videos of industrial implementation of such vision-based solutions.
- Brochures by quality control service providers.
- Visits to local biscuit manufacturers

The important quantifiable test parameters were found out to be:

- Colour of the biscuit (Perfectly cooked, over-cooked or under-cooked biscuit)
- Shape and Size (Major/Minor axis, parameter, area of biscuit)
- Embossed pattern (Embossed pattern on biscuit fully engraved or not)

The literature review is divided into four parts given below and each part will be discussed briefly:

1. Conveyor System,
2. Discarding Mechanism,
3. Camera and Sensors,
4. Software and Processing.

2.1 Conveyor System:

We require two conveyor belts deployed in series as a setup to test and train our solution. Important parameters for conveyor are **conveyor speed** which is directly associated with the algorithm and **belt width** which will decide how many biscuits can be simultaneously presented in a row. Our required belt width should suffice the space required for 3 biscuits in a row with the targeted speed of 20 cm/s (provided by Coronet Foods Pvt Limited). After thorough research, we came up with two plausible solutions, either to procure it or to fabricate it.

2.1.1 Fabrication:

For fabricating the conveyor belt, there are many options one can adopt. There are entire guides available online e.g., Kit Conveyor [3] which we can follow for manufacturing and assembling an in-house conveyor belt. The fabrication route can save us cost, but will require additional time and effort from team, which can better be utilized on the solution itself.

2.1.2 Procurement:

There are multiple options one can adopt for pre-built conveyor procurement. We can either buy it through online stores or local commercial vendors. The following table compares some of the available options for prebuilt conveyor procurement.

Table 1: Comparison of available conveyor belts in market

S.No.	Company	Description
1	Ali Express: Xinxhu Conveyor belt 1 [4]	• Dimensions 1500 mm × 400 mm × 800 mm (L x W x H). • Adjustable speed of 0-47 cm/sec. • Cost of PKR 35,142 • Material is stainless steel.
2	Ali Express: Xinxhu Conveyor belt 2 [5]	• Dimensions 1500 mm × 460 mm × 800 mm (L x W x H). • Adjustable speed (0-50 cm/sec) • Cost ranges from PKR 55,487 to PKR 57,373.
3	OLX: Sharp Technology [6]	• PKR 27000. • Used Item. Subpar

2.2 Discarding Mechanism:

The discarding mechanism is deployed in the gap between the two conveyor belts. This mechanism will discard the sub-standard biscuit while the standard biscuit will move forward to the next conveyor belt.

For Discarding mechanism, we have two options:

- Individual Flaps for each row.
- Compressed air ejection through nozzle.

2.2.1 Flap mechanism:

A flap mechanism is installed in the gap between the two conveyor belts. The flap moves downward to discard the sub-standard biscuit by sliding it, while maintains its position for the standard biscuit to move forward to the next conveyor. [7] A flap mechanism can be actuated using different options discussed below:

2.2.1.1 Linear actuators:

This actuation mechanism can be used to discard the biscuits using linear actuators. It works by moving the flaps with a linear actuator to reject the sub-standard biscuits as shown in figure 1.

Figure 1: Linear actuator controlling flap rotation [8]

In this case, we are concerned about two main factors which are taken into consideration:

1. The stroke which is the maximum length the plunger can travel from its start position.
2. The response time which is the critical factor to timely remove the sub-standard biscuit from the production line.

The following types of linear actuators are available for adoption in our solution.

2.2.1.1.1 Pneumatic Linear actuator:

Pneumatic actuator [9] can be used to move the flaps. It uses a two-way air supply which is used to move the piston up and down using the air coming from the air cylinder.

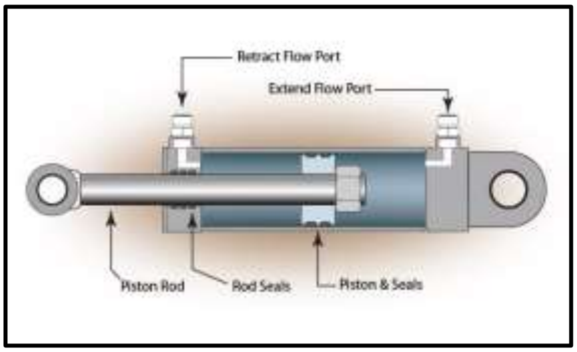

Figure 2: Pneumatic Linear Actuator [10]

They have the following characteristics:

- As pneumatic linear actuators provide high power output and have proof pressure of up to 200 - 250 Psi [11].
- Stroke Length of 150 mm to 250 mm.
- Requires maintaining a pressurized air supply.

- Important selection parameters for the actuator includes its stroke, bore, force, proof pressure, operating pressure etc. Pneumatic linear actuators are primarily in heavy-duty applications.
- Cost of 3500 PKR to 7500 PKR.
- Pneumatic linear actuators are primarily in heavy-duty applications.

2.2.1.1.2 *Solenoid operated linear actuator*

Solenoid operated linear actuator [12] has a solenoid inside, which controls the shaft's linear movement.

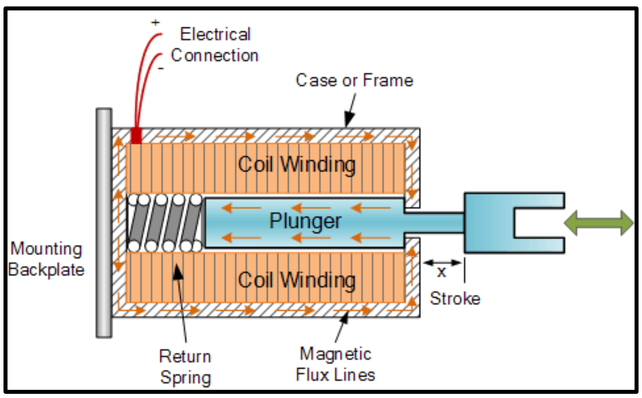

Figure 3: Solenoid operated linear actuator [13]

They have the following characteristics.

- Fast response time of 2 ms.
- High life expectancy of up to 300 M cycles.
- Stroke length of 30 mm.
- Reasonable cost.
- Low load carrying capacity.

2.2.1.1.3   DC Motor operated Linear Actuator

DC Motor operated Linear Actuator [14] has both piston and motor in a single casing.

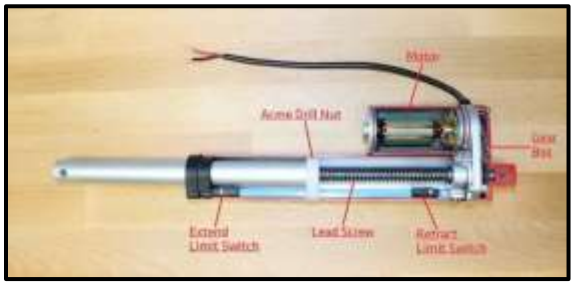

Figure 4: DC Motor operated Linear Actuator [15]

It has the following characteristics:

- Stroke length of 300 mm.
- Very slow response time of about 5-10 sec.
- Reasonable cost.

2.2.1.2   Angular Actuators

2.2.1.2.1   Servo motor:

Servo motors can perform controlled angular displacements, by using combination of a motor and a position sensor (potentiometer) for feedback. The flaps will be required to rotate to their end states at high frequency under loading condition.

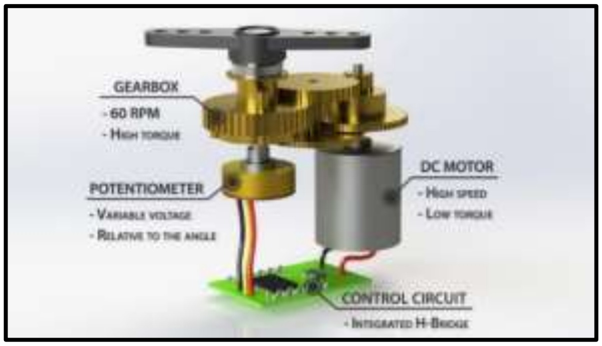

Figure 5: Servo motor [16]

Servo motors in the market can carry such loads while maintaining a fast response time. They have the following characteristics:

- High operating frequency (2 kHz +).
- Fast reaction times of about 60-200 ms.
- Provides high torques.

2.2.2 Compressed air ejection through nozzle:

Air nozzles [17] will be installed on top of the gap between two conveyors as shown in the figure 6.

Figure 6: Air nozzles [18]

Air nozzles will be normally closed, and their operation will be controlled using solenoid valves. Biscuits will have to jump from one conveyor to the next one. While travelling on the first conveyor belt, camera sensor will recognize the sub-standard biscuits. When these sub-standard biscuits pass through the gap air nozzle will eject pressurized air by which will deflect the sub-standard biscuit through the gap and only standard biscuits will be able to pass through.

- Response time of **30-100ms.**
- Requires maintenance of pressurized air reservoir.
- Increased number of parts in comparison to the rest options.

2.3 Camera and Environment:

2.3.1 Camera

For highspeed imaging applications, a high fps camera is preferred to track the objects movement accurately and acquire noise free images. High resolution camera can provide additional clarity for obtaining in-depth details of pattern on biscuit surface. We have the following options to choose a camera for our solution.

- Proprietary cameras for single-board computers
- Professional video cameras with a capture card
- Webcams

2.3.1.1 Proprietary cameras for single-board computers:

- Limited options
- Supported by Single Board Computers only with limited processing power
- Maximum output resolution ranges from 720p to 2160p. [19] [20]
- Framerate limited up to 60fps
- Best results and integration with SBCs due to their direct connection to GPU
- Price ranges from 15-50 USD
- For example, Pi Cam and Pine A64 etc.

2.3.1.2 Professional video cameras with a capture card:

- Vast variety
- Large sensor size, results in great image quality
- On-board video and image encoding
- Prices ranges from 600-1500USD for the camera [21]
- Requires a capture card (additional component) (additional 20-150USD).
- Provides greater control on acquired video's features.
- For example, Canon SL3 and 90D

2.3.2 Webcams

- Seamless integration with the computer and computer vision software.
- A 60-fps camera seems to be more in line with the application due to high-speed image capturing requirement.
- 720p/30 fps and 1080p/30fps webcams are easily available in the computer market.

  Two webcams on each end of the price range:

  - Logitech's C922 Pro can provide 720p 60 fps and 1080p 30 fps video feed costs around 15000 PKR. [22].
  - Logitech's C310 can provide 720p 30 fps and costs around 5000 PKR.

2.3.3 Enclosure:

An important aspect of computer vision-based quality control is regulating the same lighting conditions for all subjects to avoid any errors. For this purpose, usually image acquisition device is installed inside an enclosure that blocks external lights. Lights are deployed inside the enclosure to illuminate the subject. This technique regulates same lighting for all test subjects, due to no external light contamination. This results in consistent results for CV based applications. [23]

2.3.3.1 Light source:

For light sources there are many options. When selecting a light source, we need to check the following parameters:

- Power - Watts

- Brightness - Lumens
- Colour - Kelvins
- Colour and brightness deterioration with time - Chromaticity

Following are the various options:

2.3.3.1.1 LED Light Strips/TV Backlight Stripes:

- Inexpensive
- Easily Integrable
- High Chromaticity (do not deteriorate fast in colour or brightness over time)
- Brightness barely deteriorates with time

2.3.3.1.2 AC LED lights:

- Inexpensive
- Do not require a separate DC power supply
- Flickers at 50-60 Hz (AC frequency)
- Flicker can introduce black stripes in acquired video feed.

2.4 Software and Processing:

For Processing the feed coming from the webcam, we had two main options:

2.4.1 Using Single Board Computers:

Single Board Computers such as the Raspberry Pi or Asus tinker board S often can run a Linux based system with OpenCV and can relate to proprietary webcams. These boards also have input output pins which can be used to directly control the discarding mechanism. But these single board computers with ARM-based processors do not have much processing power thus might take more time to process and compute the results than a typical x64 processor would. Also, the Webcam support for these boards is almost close to none. They only support the proprietary webcams from their own companies which do not provide the best image quality and are hard to acquire.

2.4.2 Using Mainstream Desktop Computers or Laptops:

Mainstream Desktops are also a great candidate for performing image processing on the feed from the camera. Desktop computers can be connected with Arduino and communication can be established using serial bus over USB connection to perform input output tasks such as controlling the discarding mechanism or receiving signals from other sensors for feedback. To perform the image acquisition and processing, the following two software seem capable of carrying the task:

2.4.2.1 MATLAB:

MATLAB provides Computer Vision libraries and applications that can acquire, process, and compare the video feed coming from the webcam using a computer vision toolbox available in MATLAB.

2.4.2.2 Python (OpenCV):

Python is another feasible and mainstream option for performing computer vision. It can also acquire, process, and compare the video feed coming from the webcam. Both MATLAB and Python also allow machine learning which can also be applied for image comparison and decision taking. MATLAB is paid source while Python is an open source and used by most of the industries.

Processing techniques such as **greyscale conversion** which is a method of converting a colored image to an image which contains only shades of grey. The reason for doing this to it will provide less information for pixel which reduces the computation power. A study has been done by Hindawi Corporation [24] which compares different techniques of image processing. The study describes techniques for **crack detection** by converting a binary image into digital image. The main objective is to enhance the crack portion of the biscuit so to detect it more corrected. Several methods are given in the study. For **dimensionality** of the biscuit, an article written by Aashay Sathe [25], he writes "by changing the surroundings of the biscuit into black color while keeping the subject to white we can check its dimensions including size, area, perimeter, major-minor axis length etc. very easily". The **color identification** of the biscuits can be done using the same greyscale. The overcooked biscuit will fall in the darker portion of the color and undercooked will fall into the light portion of color. In this way we can set a threshold of color range. Our analysis says that color identification (baking extend) is easier if we do it in real colors of the biscuit.

CHAPTER 3: METHODOLOGY

Following the literature review, we set out to develop a solution from the thorough research on similar computer vision-based quality control systems. Our developed concept is inspired by the work done by researchers, implementation done in industries, and commercial quality control service providers etc. We have visited the local industry to gather technical data and to identify the quality control challenges faced by them. After gathering all the required data, we developed a concept (shown in flow chart below) to overcome those challenges. Two protypes were developed: V1 served as a learning opportunity, while prototype V2 was the final version of our solution.

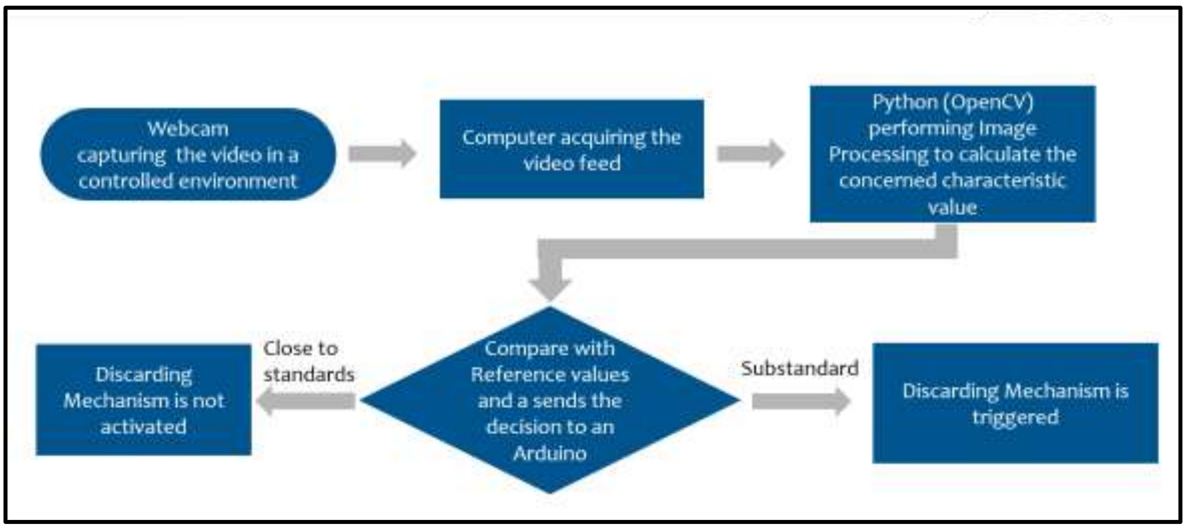

Figure 7: Flow Chart showing working of our solution

3.1 Solution Development:

From the literature review, we developed following working of our model as shown in the figure 8.

- A conveyor system consisting of two conveyor belts would be used.
- A discarding mechanism would be installed in-between the two conveyor belts.
- The detection of standard and sub-standard biscuit would take place on the first conveyor.
- Only the standard biscuits will travel to second conveyor belt, while sub-standard biscuits will be discarded when travelling from first to second conveyor.

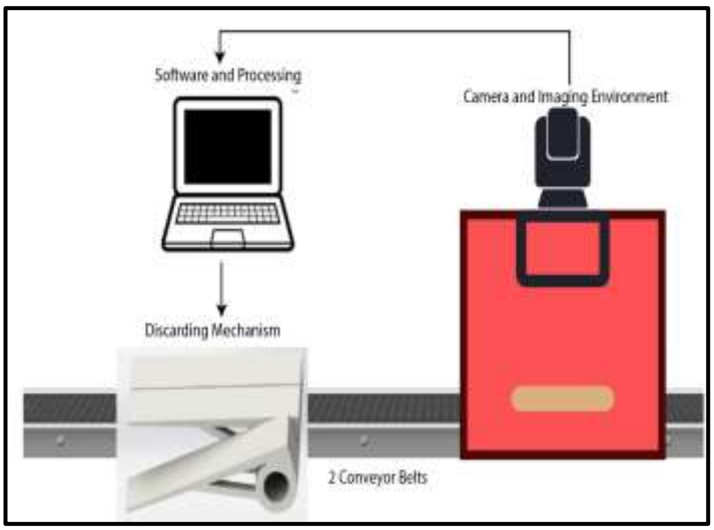

Figure 8: Pictorial representation of the solution's working model

We made following design decisions to implement the working model:

- The conveyor belts available in the RISE Lab of SMME were decided to be used.

- The discarding mechanism of flaps actuated by Servo motors was decided to be used.
- A Logitech C310 (720p/30 fps) webcam was selected.
- A controlled environment with DC LED light strips was sustained.
- Python was used for differentiating between standard and sub-standard biscuits and for the decision-making process. Arduino IDE was used for actuating the servo motor.

The reasoning for all the above decisions is explained below:

3.1.1 Conveyor Belt

Conveyor belt is the basic platform required for our project where we can deploy, test, and refine our QC system. As the manufacturing of the conveyor belt was outside the domain of this project, and the procurement was a costly alternative, we decided to use the conveyor belt system available in our school. Access to LabVolt Series 5210 Conveyor belts available in RISE Lab SMME was taken from the lab in charge, along with Dr Fahad's permission. The LabVolt Series 5210 Conveyor belt has the following description and specifications [26]:

3.1.1.1 General Description

The Belt Conveyor is used in material-handling experiments. It has a self-contained power supply and interface electronics. The control panel has inputs for interfacing with the Robot Controller. The inputs enable the conveyor to be remotely controlled. It can also be used as a stand-alone unit. The Belt Conveyor comes with a movable limit switch to detect the presence of parts on the belt [26].

3.1.1.2 Specifications

Table 2: Belt Conveyor Specifications

Belt Conveyor, LabVolt Series (FESTO)	
Parameter	**Value**
Motor	
Type	DC servo motor with closed-loop control
Encoder	Optical
Conveyor Belt	
Length	1880 mm (74 in)
Width	127 mm (5 in)

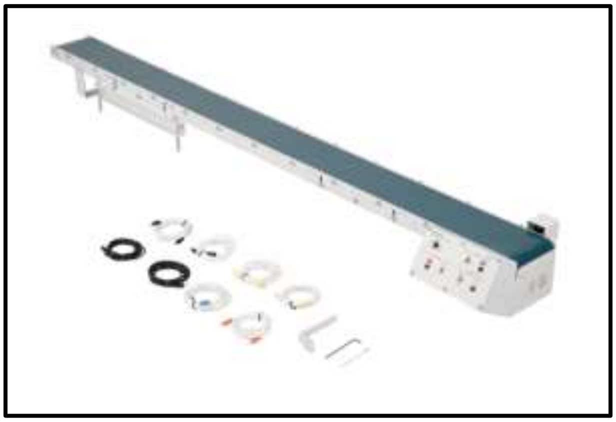

Figure 9: Belt Conveyor of RISE Lab

3.1.1.3 Speed Encoder Assembly

Knowing the speed of the conveyor belt was of paramount importance since it would be required to calculate the time between taking a decision and the discarding action. An encoder was 3D modelled in SolidWorks as shown in figure 10. The encoder was fabricated using acrylic sheet, off-the-shelf shaft, tyres, and speed sensor module as shown in figure 11.

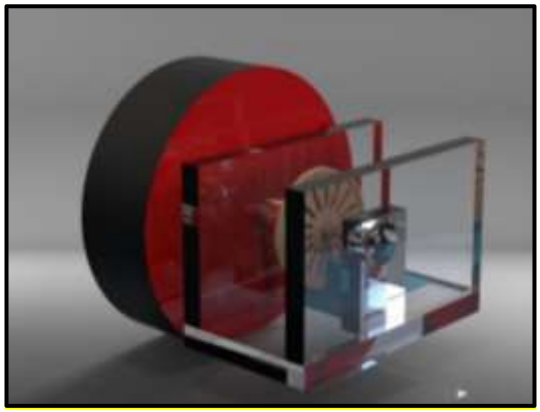

Figure 10: Speed Encoder Render

Figure 11: Speed Encoder Fabricated (isometric view)

3.1.1.3.1 Speed Sensor Module

We used an off-the-shelf LM393 Speed Sensor Module which is basically an Infrared Light Sensor integrated with LM393 Voltage Comparator IC. This module has a slot which counts the number of times IR light has been blocked. Using this, and the number of slots in the tire, we can calculate the RPM of the conveyor belt. Then, using the following formula we can calculate linear speed of conveyor:

$$v = r\omega$$

Where,

v	=	Linear speed of conveyor belt
r	=	Radius of the outer wheel
ω	=	Angular speed of conveyor belt in rad/sec.

3.1.1.3.2 Working

Angular speed of conveyor is calculated using the number of slots in the inner slotted wheel and time taken across each slot using the formula below:

$$\omega = \frac{2\pi}{n.t}$$

Where,

n	=	No of slots in the internal wheel of encoder = 20
t	=	time taken across each slot.

Following steps were involved in working of the speed encoder assembly:

- An outer wheel would rotate with moving conveyor belt.
- Since the outer and inner slotted wheel has a common shaft, the inner wheel would correspondingly cut the IR lights rays of the speed sensor module.

- This blocking of IR lights would be used to calculate speed of conveyor mentioned above.

3.1.2 Discarding Mechanism

The discarding mechanism is deployed in-between the conveyors to discard the sub-standard biscuits and transfer the standard biscuits to the next conveyor. There were many considerations when selecting a suitable option such as:

- Response time of the mechanism
- Cost effectiveness
- Availability of a compressed air on deployment site

3.1.2.1 Linear Actuators

The linear actuators (Pneumatic, Solenoid Operated, DC Motor Operated) we reviewed, either had much higher weightlifting capacity than required in our project (Pneumatic) or had insufficient stroke length (Solenoid Operated) or slow response times (DC Motor Operated). Thus, none of the linear actuators were considered.

3.1.2.2 Compressed Air Ejection through nozzle

Compressed air ejection through nozzles was not considered as a viable option because of the following reasons:

- It required use of compressed air, which although is commonly available in industries for where this project is targeted, but for testing and refinement purposes in lab environment, a pneumatic solution comprising of comparatively greater number of number of parts as compared to a servo motor assembly is not easy to setup and procuring it would have been a costly option [29], thus mechanisms employing compressed air were not considered as a suitable option.

- Another reason was that although closely related, focus on this option would have diverted us from the main scope of the project thus, we did not opt for pneumatic solutions.

3.1.2.3 Servo Motor with flap mechanism

At last, servo motor with flaps was the best suited option according to our design and budget constraints. The TowerPro SG90 servo motor was finalized as it fulfilled the torque and frequency requirements of our discarding mechanism. A motion analysis of a flap made of ABS plastic in SolidWorks was performed.

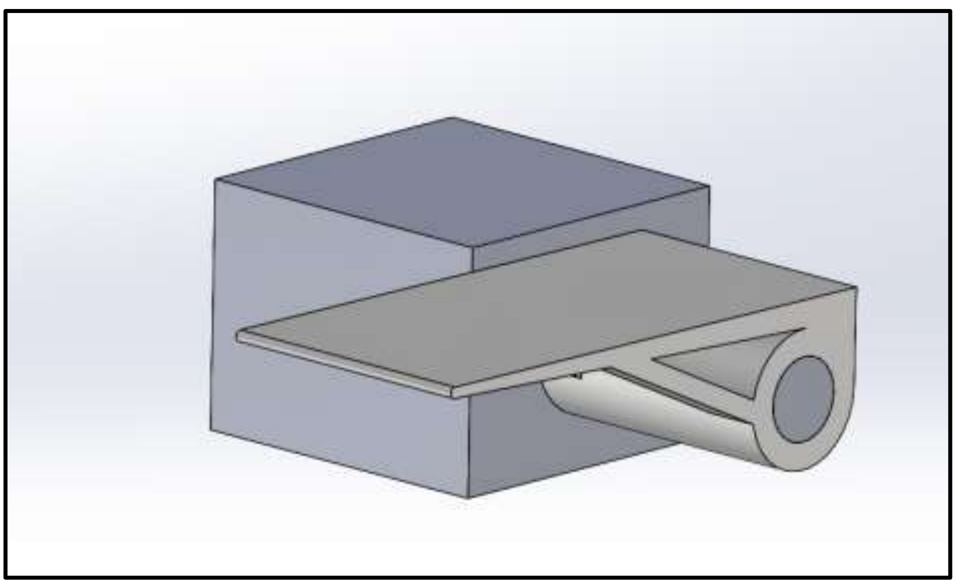

Figure 12: 3D Model of a flap

Table 3: Flap Motion Analysis Parameters

Motion	
Parameter	**Value**
Angle	45 degrees
Frequency	4 Hz
Forces	
Gravity	9.8 ms^{-2}
Weight	20 grams (average weight of biscuit) on flaps' edge (worst case scenario)

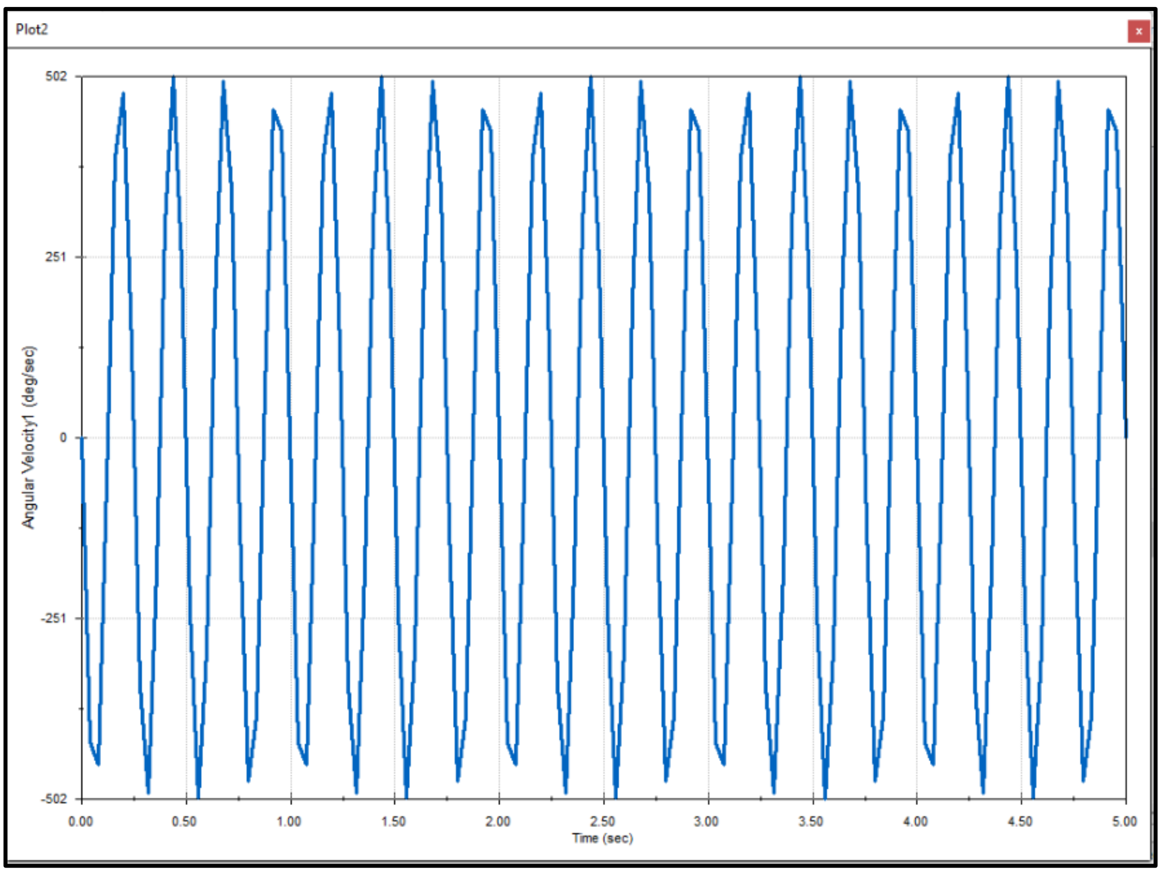

Figure 13: Angular Velocity Vs Time Graph

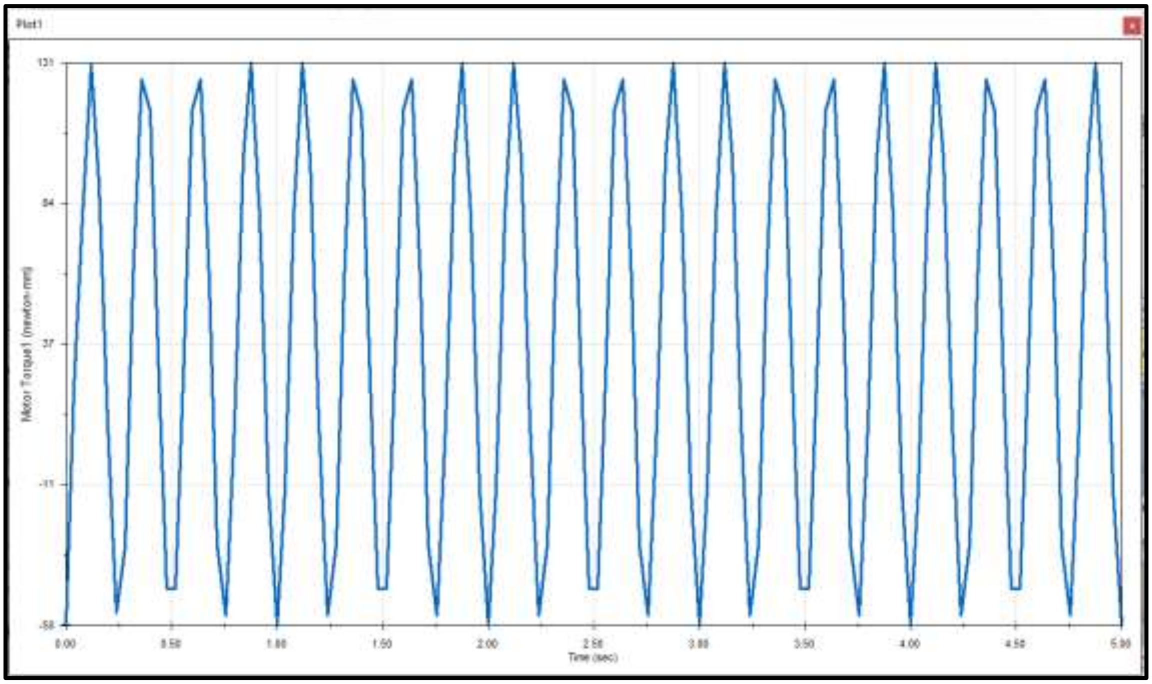

Figure 14: Motor Torque Vs Time Graph

Table 4: Motor Requirements

Motor Requirements	
Parameter	**Value**
Peak Velocity Required (at 0 Required torque)	502 deg/sec
Peak Torque Required (at 0 angular velocity)	131 N-mm

Table 5: SG-90 Motor Specifications

SG-90 Motor Specs	
Parameter	**Value**
No Load Velocity	600 deg/sec
Stall Torque	176.52 N-mm

As both the available stall torque and no-load velocity values are greater than the requirement, hence Tower Pro SG-90 servo motor can sustain the loads for our application.

3.1.2.3.1 Flap Assembly

For controlling the discarding mechanism through servo motor, a flap assembly employing a four-bar mechanism is a feasible option. Servo motor was used as an input to this four-bar and flap movement was the output. The flap assembly consists of following parts:

- Flap
- Servo Motor
- Supporting structure for flap and servo motor

This flap assembly would be installed between two conveyor belts.

3.1.2.3.1.1 <u>Prototype Version 1 Flap Assembly</u>

A prototype V1 of flap and its mounting assembly was designed in SolidWorks; the 3D model of flap assembly is shown in figure 15. The flap design was manufactured using additive manufacturing facilities available in DME. The model was 3d printed in Resin and cured for 4-5 hours in sunlight, while its mounting assembly was fabricated by using 5 mm thick Acrylic Sheet. The fabricated prototype V1 assembly is shown in figure 16. This prototype V1 flap assembly was made because of following reasons:

- To check its validity for use as discarding mechanism for biscuits
- To check its compatibility and communication with rest of the solution
- To check responsiveness of a servo motor four-bar flap mechanism

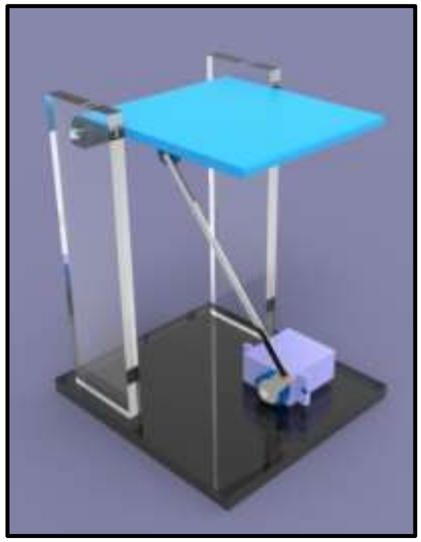

Figure 15: Prototype V1 Flap Assembly 3D Render

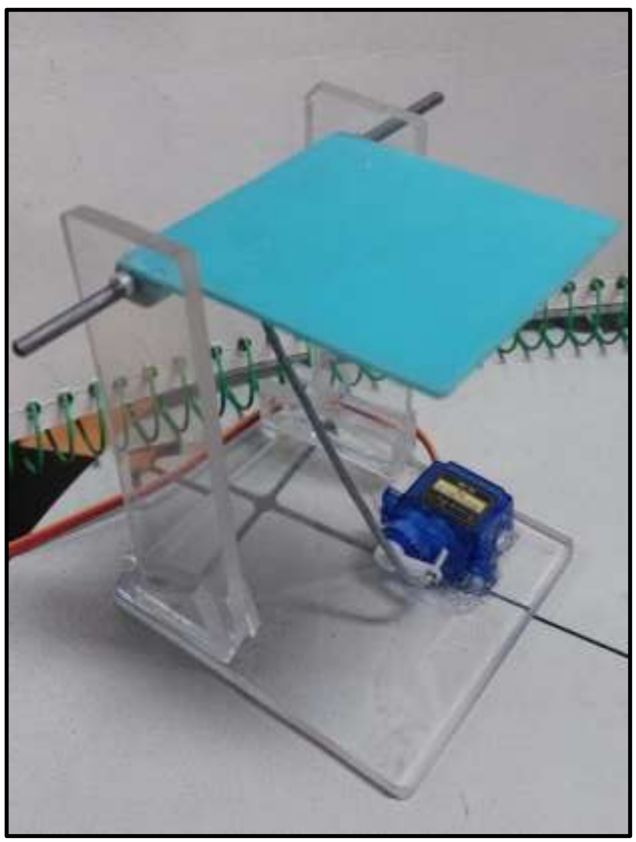

Figure 16: Prototype V1 Flap Assembly Fabricated

3.1.2.3.1.1.1 Four Bar Mechanism

We used four bar mechanism to actuate our flap which can be seen clearly in detail in figure 17.

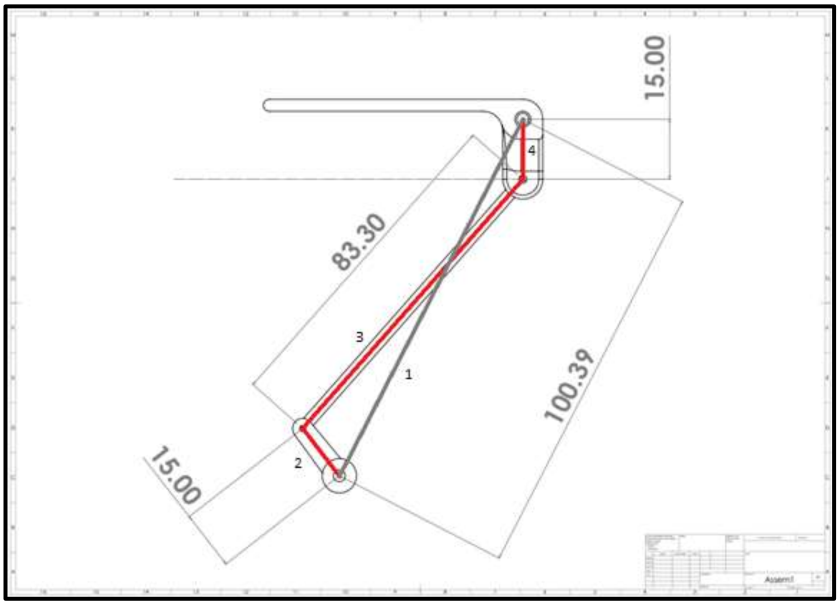

Figure 17: Picture showing four-bar mechanism details

Following were the lengths of links:

Table 6: Four-bar Mechanism Details

Sr No.	Link	Link Length (mm)
2	S	15
3	P	83.3
4	Q	15
1	L	100.39

Since S + L > P + Q and shortest link is the input link, we have the triple rocker condition, which is a non-Grashof condition.

Also, to study the movement of link S (15mm) and link Q (15mm) and the relation between input and output links each other a motion analysis study was performed in SolidWorks. The angular motion of Link S, Link Q with time and With each other is graphed below, to setup a relation between the input and output displacement.

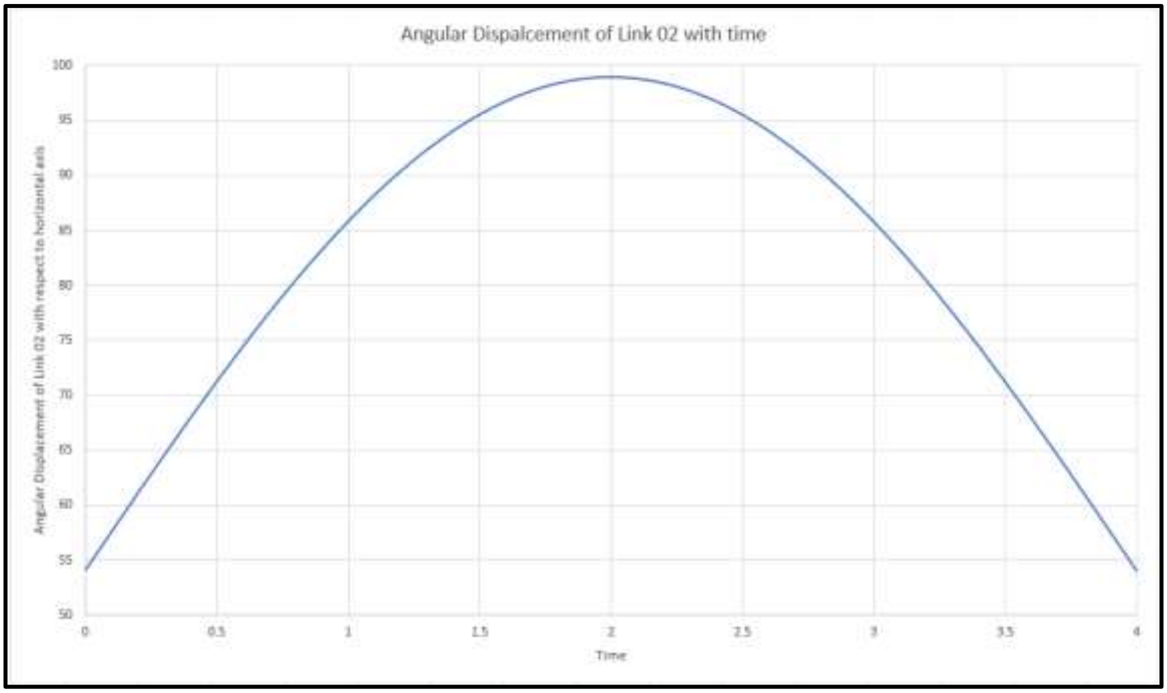

Figure 18: Angular displacement of link S (Input) vs time

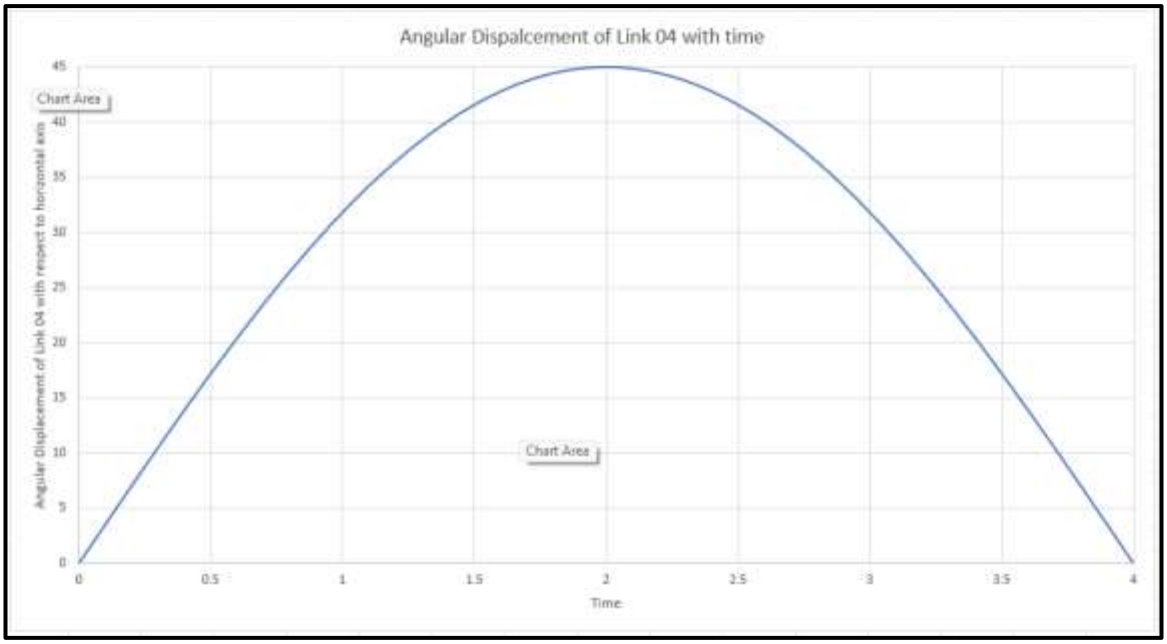

Figure 19: Angular displacement of link Q (Output) vs time

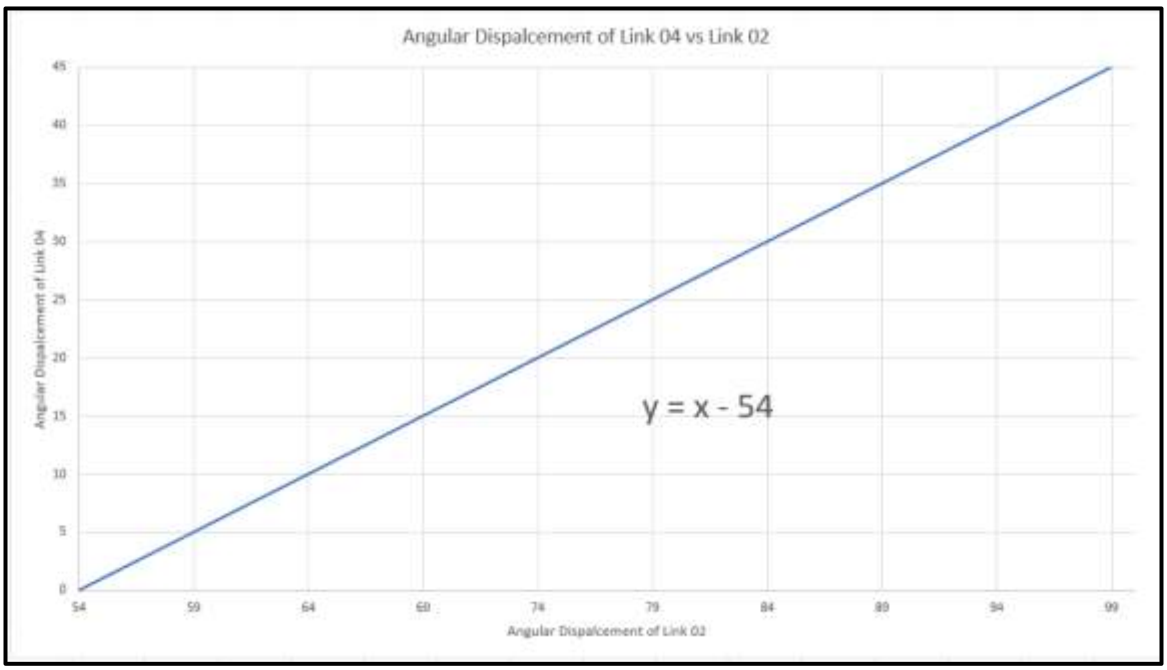

Figure 20: Angular displacement of Link Q vs S (Output Link vs Input Link)

The above graph was interpolated to get a mathematical relation between the output and input links angular displacement. Interpolation function was found using Excel.

Interpolated Function:

$$y = x - 54$$

Where: y is output Angular Displacement (Link Q)

x is input Angular Displacement (Link S)

3.1.2.3.1.2 Prototype Version 2 Flap Assembly

Since the prototype V1 flap assembly was designed and fabricated before it was finalized that RISE Lab conveyor would be used, an update to the initial design was needed to cater to the design requirements of the available conveyor. Also, from the working of prototype V1 flap assembly many design blind spots were identified such as the biscuit would not fall directly on the flap etc. These were catered for in the prototype V2 design. Following considerations were considered when designing this prototype V2 flap assembly:

- Width of conveyor belt
- Height of conveyor belt from the table
- Mounting design for flap assembly to the conveyor
- Accommodation for conveyer curvature so that biscuit falls directly on flap

A single unit assembly has been designed which will contain 2 flaps side by side (since the conveyor width can accommodate only 2 biscuits in a row) with a mounting position for motor, axle for flaps and flaps. It will also connect the flaps and motors with a four-bar linkage in between. The prototype V2 flap assembly design will take into consideration the curvature of the belt at the end of the conveyer. It will also be a bolt on addition to the conveyor hence the conveyor will not be damaged when the flap assembly is attached with it. The design for this prototype V2 flap assembly is shown in figures 21 and 22.

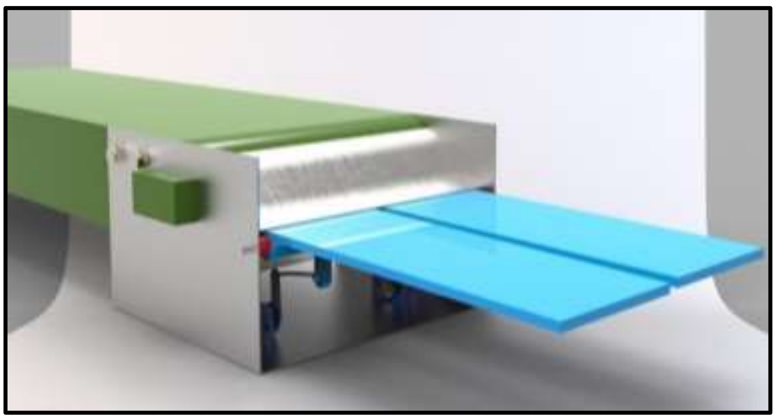

Figure 21: Prototype V2 Flap Assembly integration with conveyor belt (Render)

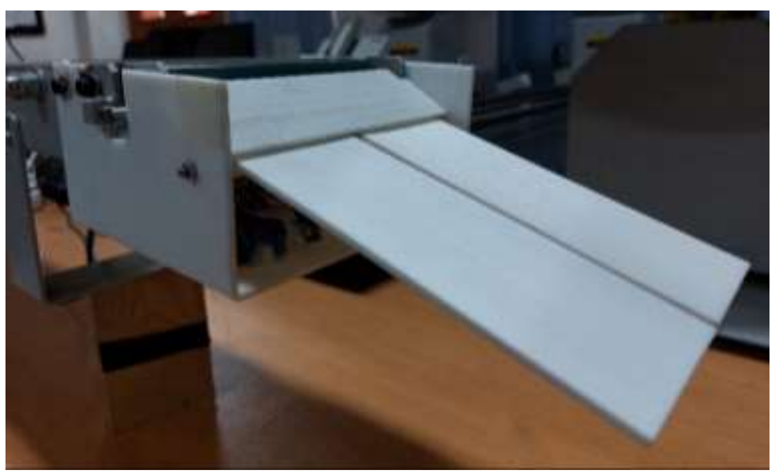

Figure 22: Prototype V2 Flap Assembly integration with conveyor belt (Image)

3.1.2.3.2 Servo Motor's integration with system

The flap mechanism was integrated with the rest of the system by using an Arduino as a bridge between the desktop and the motor. A signal wire containing the information about the angle the servo motor is connected with any of digital PWM pins on the Arduino board. Power was also provided to the single SG90 motor with the Arduino board as it complied with the maximum power Arduino can deliver. Communication between the Arduino and Servo was established with a USB connection. Data transfer from the computer regarding flap's status was communicated to the Arduino using the serial bus. UTF-8 encoded signals were transferred for the two flap states (0 degrees and 45 Degrees). When the corresponding encoded signal for the particular state is detected in the serial bus by the Arduino, it triggers the flap to update its state accordingly.

3.1.3 Webcam

For our computer vision application, with the level of details required and dynamic environment of our operation, Logitech C310 Webcam (720p/30 fps) was our choice. The decision criteria were "best quality, low cost, easily integrable camera".

We checked the minimum resolution and frames per second (fps) required for our application. Using the camera of Samsung Galaxy A52 smartphone, it was determined by trial and error that 720p/30fps was adequate enough for our project as it gave enough details for our algorithm to differentiate between standard and sub-standard biscuits. Then since most webcams offer plug-n-play solutions, we narrowed ourselves to webcams out of possible options described in literature review. For webcam, Logitech was the company offering good quality and good reviewed webcams. At last, Logitech C310 fulfilled our design requirements and thus was selected. It has following detailed specifications:

Table 7: Logitech C310 Webcam Specifications

Specifications	
Parameter	**Value**
Max Resolution	720p
Max FPS	30 fps
Camera Megapixel	1.2
Diagonal Field of View	60^o

3.1.4 Controlled Environment

A controlled environment was necessary to limit the effect of surrounding lighting conditions and to give reliable results in every condition. A Carboard box was used in prototype V1 as an enclosure with LED strips attached on the top side facing downwards. A hole in the centre was made as a viewport for the camera. Upon confirmation of the design functionality, it was then manufactured in sheet metal with compatibility LabVolt Series 5210 conveyor belt in mind. It is designed to be a bolt on addition to the LabVolt Series 5210 conveyor.

3.1.4.1 Carboard Enclosure

Initially for testing purposes, a carboard box was used as an enclosure as shown in figure 23. LED strip lights with self-fabricated heat sinks were installed on inside of roof of enclosure, while webcam was installed on a hole on top of enclosure.

Figure 23: Cardboard Enclosure installed on Lab Conveyor Belt

3.1.4.1.1   LED Lights

Samsung 4014-144 LED strip lights were used as shown in figure 24.

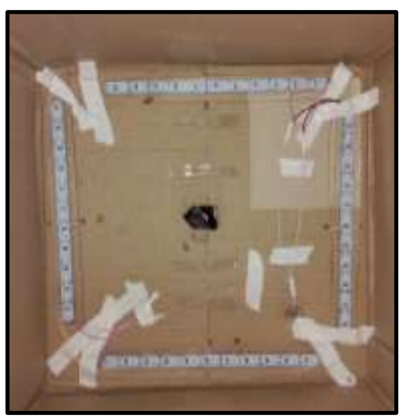

Figure 24: LED Strip Lights installed on roof of cardboard enclosure

3.1.4.1.2 Heat Sink

Since LED lights need to dissipate heat, and cardboard is not a heat conductor. heat sinks were necessary to increase their heat dissipation capacity. These heat sinks were made from 5 mm Acrylic Sheets as shown in figure 25. LED Strips were glued onto Acrylic and Acrylic was then taped to the enclosure.

Figure 25: Heat sink for LED strip lights

3.1.4.2 Sheet Metal Enclosure

Following the testing with cardboard enclosure, an enclosure was 3D modelled on SolidWorks considering the RISE Lab conveyor as shown in figure 26.

Figure 26: Sheet metal enclosure for lighting and camera deployment (Render)

3.1.4.2.1 LED Lights

Samsung 4014-144 LED strip lights were used.

3.1.4.2.2 Heat Sink

The LED Lights were mounted on the sheet metal enclosure using screws and nuts to ensure adequate contact pressure, which reduces the thermal contact resistance. This design uses the metal enclosure itself as a heat sink.

Figure 27: Sheet metal enclosure for lighting and camera deployment (Image)

3.1.5 Software

The algorithm to differentiate between standard and sub-standard biscuits was developed in Python programming language using the open-source OpenCV library.

We had two options to choose from: MATLAB and Python. In MATLAB the algorithm can be developed a lot faster, since many functions for image processing are already written, but it is a licensed software and run time may increase computation cost, whereas Python is free and faster, but development of algorithm may be a bit slower, since we need to write some functions in order for image processing.

Since lower computation cost and a free software was offered by Python, we opted for it.

Since, as Mechanical Engineers we didn't have much exposure to the image processing techniques and syntax, free courses on Coursera and YouTube were used to bridge our knowledge gap.

For developing the algorithm, there are many editors, and we opted for Visual Studio Code as it is free, fast and open source. Python was installed in Visual Studio Code, with OpenCV library imported (for computer vision application) and NumPy library (for numerical calculations).

3.1.5.1 Working in Python

The algorithm for differentiating between standard and sub-standard biscuits based on colour was written in Python. We are focusing on colour because industries face the challenge of discarding over cooked and undercooked biscuits in the largest magnitude.

The over-cooked and under-cooked biscuits can be differentiated from a perfectly cooked biscuit using colour. The algorithm works in following way:

3.1.5.1.1 Video Acquisition

- Webcam video is obtained.
- For consistency in our results, most of the webcam parameters are controlled and specified in algorithm. The parameters we control include brightness, contrast, saturation, hue, gain, and exposure.
- The video is obtained with the lowest possible exposure so that the video can be as sharp as possible. This low exposure is made possible by a faster shutter speed, which in turn allow for a sharp image i.e., the image is not blurry.

3.1.5.1.2 Frame Processing

- Python captures video in BGR (Blue Green Red) colour space, and this video is converted to YCrCb colour space. Y represents Luminance; Cr represents red-difference chroma components while Cb represents blue-difference chroma components.
- Conversion to YCrCb colour space allowed for a greater control on thresholding for differentiating between standard and sub-standard biscuits as compared to other colour spaces (such as RGB, L*a*b, HSV etc.) and thus we opted for it.
- A mask is created using the converted video with the upper and lower bounds (in YCrCb Colour space) given using self-created trackbars. A mask is a binary image consisting of zero- and non-zero values. [28] The upper and lower bounds for the

mask are decided so in-between those bounds only the region of interest (ROI) remains (i.e., only biscuits remain, and all other background is blackened).

- Gaussian blur is applied on the mask, which is further dilated. Blurring and dilation allow for the noise in the image to be reduced.

3.1.5.1.3 Contour Development

- A "getContours" function is defined in the main body of code. Further processing of image takes place here.
- The dilated image is fed into this function where the ROI's contour and the area enclosed within those contours are found.
- The contours are developed only in a small window in the longitudinal axis (the axis along which the conveyor travels). This window is defined such that only one biscuit per column could be detected. This allowed us for greater control on the discarding decision for the biscuits.
- Red contours with black bounding box is made for sub-standard biscuit, while green contours with cyan bounding box is made for standard biscuits.

3.1.5.1.4 Decision Formulation

- The areas of standard and sub-standard biscuits will be different because in region of interest (ROI) the over-cooked biscuits will have more zero-pixel values (thus smaller area) while under-cooked biscuits will have lesser zero-pixel values (thus larger area).

- Thus, by defining a range of area in which only the perfectly cooked biscuit will be, we were able to differentiate between standard and sub-standard biscuits.

3.1.5.1.5 Discarding Decision:

- The standard and sub-standard biscuits are sorted using their x-coordinate and stored in a matrix. The decision for actuation is stored in the same matrix. For standard biscuits "0" is stored, while for sub-standard biscuits "1" is stored.
- This "0" and "1" decisions are encoded with UTF-8 and dumped onto the COM port after accounting for time delay.
- If there is no biscuit, "2" is stored in the decision matrix to exclude false positives.
- The algorithm also accounts for the position of biscuit in x-axis and thus ensuring that decisions for biscuits is communicated to both the servo motors according to biscuit position.

3.1.5.1.6 Scanner And Actuator Time Synchronization

- Since the biscuit is not discarded immediately where a decision of discarding is taken, the timing delay needs to be accounted for so that only sub-standard biscuit gets discarded.
- The timing delay must be tuned and adjusted to account for communication delays introduced by electronics (Computer, Arduino), the response time of servo motor and the speed of the conveyor belt so that only the sub-standard biscuit is discarded.
- This can be adjusted either in Python – the decision can be communicated with a delay to Arduino or in Arduino IDE – servo motor can be actuated with a delay.

- Since during our testing Arduino IDE could not provide delay of more than 15 seconds, we integrated the delay in Python.
- The detection time + time delay is stored against each biscuit, and only when this combined time is equal to the current operating time of code, the decision is communicated with Arduino for actuation.

3.1.5.1.7 Code Output

- Code is also written to show the original and output image in a single window. The output image shows the contours formed on biscuits according to the decision criteria i.e., if they are standard or sub-standard.
- Real-time and Cumulative statistics are also shown on the output image. Statistics include number of standard and sub-standard biscuits and total biscuits passed in a minute.

3.1.5.2 Working in Arduino IDE

- In Arduino IDE code is written to actuate the servo motor.
- Arduino looks for any input to the COM port for commands.
- Code is written so that Arduino actuates servo motor when it receives a different decision than the current one i.e., it will only actuate when it receives “1” when its current state is according to the “0” command from python and vice versa, otherwise it will not actuate.

3.2 Parts Shortlisting

Many parts were shortlisted for procurement or fabrication as shown in summarized in the table below with detail Bill of Materials attached in appendix.

Table 8: Shortlisted Parts and material along with their uses

Sr No.	Part/ Material	Use
1	Logitech C310 Webcam	As a camera for CV application
2	DC Lights	For controlled environment in enclosure
3	Sheet Metal	For fabricating enclosure
4	Acrylic Sheet	For fabricating flap assembly prototype V1 and speed encoder
5	Miscellaneous (Bearings, Plugs, Fuses etc.)	Used in various places

CHAPTER 4: RESULTS AND DISCUSSIONS

Overall, two protypes were designed and developed in order to achieve the set objectives. Both prototypes were tested in order to examine their performance, efficiency, effectiveness, and practicality.

The prototype V1 was developed using readily available and easy to work with materials. A light enclosure made of cardboard, flap and encoder assembly was made of acrylic and a 3D printed flap, was assembled. The solution was integrated with the available LabVolts conveyor belt available in RISE Lab, SMME.

The working on V1 was tested as a proof of concept. Its successful demonstration along with rigorous testing helped us get better insights on our challenge and how we could better equip our next prototype for achieving the targeted deliverables.

A more sophisticated, final prototype V2 of the solution was then designed and developed keeping in mind the shortcomings of the initial version.

4.1 Testing and Results

A number of tests were performed on both prototypes to attest the working and quantify the performance of our developed solution.

a) Distinguishing substandard and standard biscuits

b) Ensuring both scanning and actuation were in the required synchronization

c) Maximum biscuits per minute sustainable

d) Minimum distance between biscuits required

e) Repeatability in decision-making

f) Effect from change in external lighting

g) Settling Time of actuating flap

h) Tribological Testing of flap surface (Angle required for a biscuit to slide)

i) Compatibility with the available conveyor belt

4.1.1 Distinguishing substandard and standard biscuits

Our developed algorithm was actively tested on the LabVolt Series 5210 conveyor belt for performance evaluation, loopholes, detecting false positives and algorithms shortcomings in the development phase. A single algorithm with 50+ revisions was carried from the initial Prototype V1 and was refined to be used in our final prototype V2 as well.

Biscuits were distinguished on the basis of their colour, to reject over or under cooked biscuits from passing onto the next conveyor. The algorithm checked the number of pixels within of set range of colour values in each frame.

If the amount exceeded a set value, which was almost equal to the area of a biscuit in pixels, it was declared as an optimum biscuit. When the amount of pixels within colour values prescribed, is lower than the area of the biscuit, it translates that some area of the subject biscuit is either overcooked(darker), or undercooked(lighter).

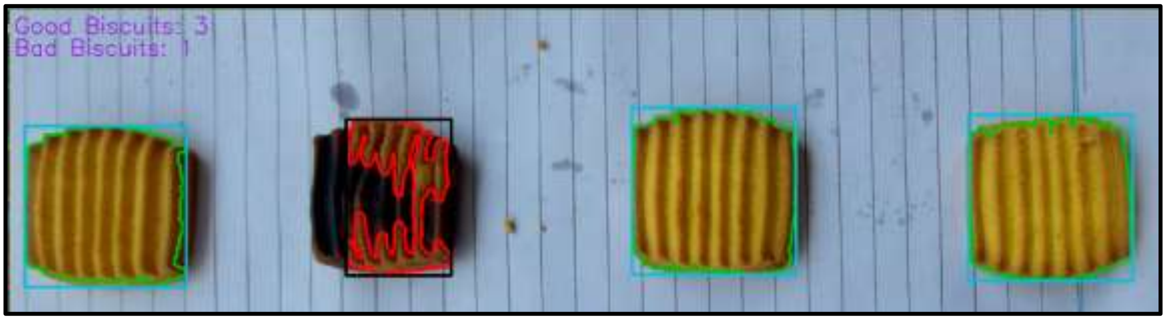

Figure 28: Decision Making on Sooper Biscuit

4.1.2 Ensuring both scanning and actuation were in the required synchronization

For successful rejection of biscuits at the actuation end, it was to be ensured that the same biscuits that are categorized as sub-standard at the scanning end are rejected at the actuation end. Knowing the time it takes for each biscuit from the scanning end to the actuation end, a delay was added before the decision was forwarded to the Arduino to actuate the flap.

4.1.3 Minimum distance between biscuits required

Our developed machine is capable of differentiating incoming biscuits according to the set input parameters with some physical limitations. The response time of the motors and the length of flaps used dictate the minimum distance between biscuits that is required for correct actuation in response to the decision taken by the algorithm. After multiple trials, this distance was found out to be 1.5 cm between the outer circumferences of each of the biscuit.

Figure 29: Finding minimum distance required between biscuits

4.1.4 Maximum biscuits per minute sustainable

With the value for minimum distance between each biscuit acquired, maximum biscuits per minute that our machine is capable of handling had to be calculated. Another parameter was required for this purpose i.e., maximum speed of the conveyor belt sustainable by the setup. The conveyor belt used maxes out at a speed of 6.4 cm/sec, at which the rest of the system is able to cope up with inter-biscuit distance of 1.5 cm and above. Keeping in mind that the current setup can fit 2 biscuits in each row, the maximum biscuits per minute sustainable for the machine was found out to be **2.13 biscuits per second or 128 biscuits**

per minute. (64 biscuits per minute per column). This number is easily scalable by increasing the number of biscuits in a row using a wider conveyor belt.

4.1.5 Repeatability in decision making

To determine the repeatability offered by the solution, the results for same set of biscuits were evaluated 5 times. Due to consistent lighting, fixed camera control parameters and the binary nature of results (Standard / Sub-Standard), 100% repeatability was recorded even with change in external lighting.

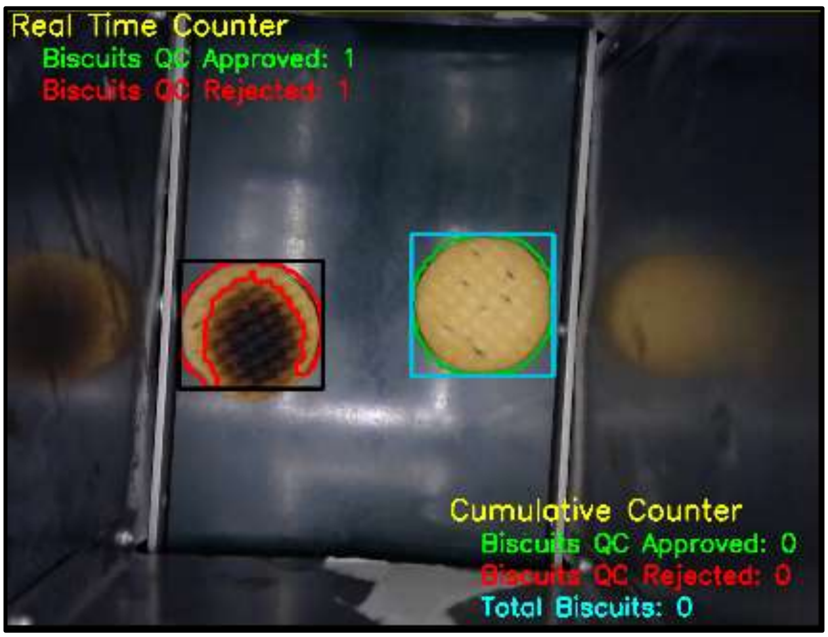

Figure 30: Repeatability in decision making

4.1.6 Effect from change in external lighting

An enclosure is required to block light contamination from the environment, which can affect our results. To test the performance of our enclosures, tests were conducted where the room's lights were turned OFF and ON and the algorithm results in both scenarios came out similar.

Figure 31: Light contamination test - Prototype V1

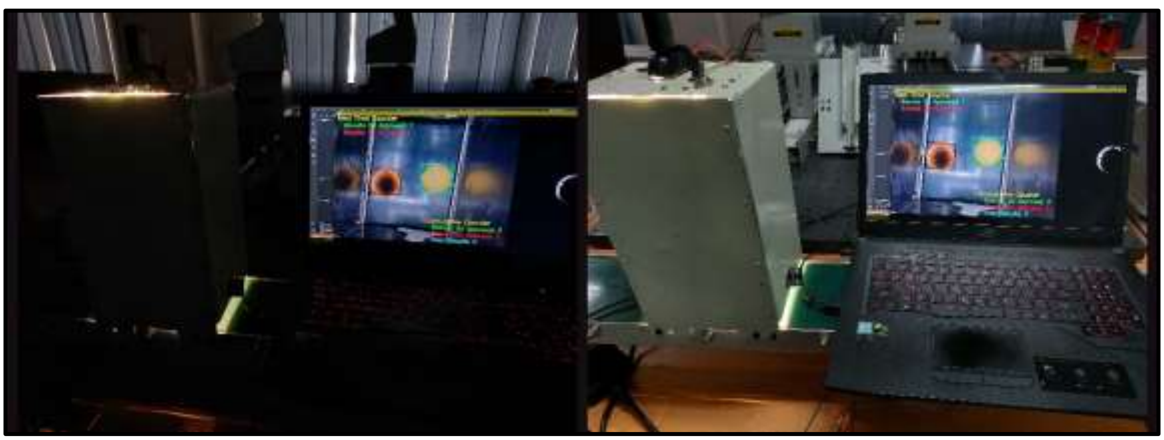

Figure 32: Light contamination test - Prototype V2

4.1.7 Settling time of actuating flap

The flap mechanism will require quick actuation that can be achieved with fast settling time.

The flap mechanism consisted of an SG90 servo motor rotating the flap with a four-bar mechanism. To test the stiffness and responsive of our mechanism, an experiment was performed to study the maximum frequency achievable by the flap to find out the settling time for flap movement. When vibrated with an amplitude of 50 degrees, following maximum frequencies were achieved.

Table 9: Maximum frequency Achieved by flap mechanism prototypes

Prototype	Cycles counted	Frequency (s)
Prototype V1	44	4.4
	46	4.6
	42	4.2
Prototype V2	52	5.2
	48	4.8
	50	5

4.1.7.1 Calculation for Prototype V1

Angular Displacement θ_1 = 50 deg

Average maximum frequency f = 4.4 Hz.

Time period T = 1/f = 0.23 s

Settling Time T_s = T/2 = 0.114 s or 114 ms

4.1.7.2 Calculation for Prototype V2

Angular Displacement θ_1	=	50 deg
Average maximum frequency f	=	5 Hz.
Time period T = 1/f	=	0.2 s
Settling Time T_s = T/2	=	0.1 s or 100 ms
Response Time	=	63 ms (Time to complete 63% of complete motion)

4.1.8 Tribological Testing

Tribological testing for flap was required to make sure substandard biscuits are successfully able to slide off the surface of the flap. For this purpose, a series of experiments were performed to find out the static coefficient of friction between the manufactured flap and a typical biscuit.

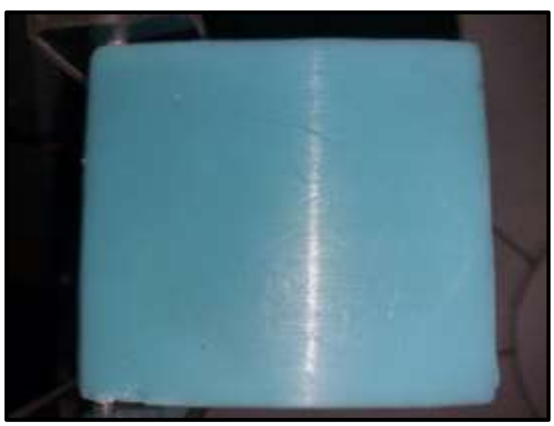

Figure 33: Resin 3d printed Flap Surface (Prototype V1)

Figure 34: Resin 3d printed Flap Surface (Prototype V2)

A biscuit (Sooper Biscuit, by Peek Freans) was placed on the flap, and the angle of flap was logged. The flap angle was increased with small increments of 1 degree and the angle at which the biscuit started sliding was noted down.

Table 10: Slip angle determination for biscuit on flap surface

Prototype	Sr No.	Flap Angle θ_2 (Degrees)	Static Frictional Coefficient ($\mu_s = \tan(\theta)$)
V1	1	29	0.55
	2	30	0.57
	3	25	0.48
V2	1	28	0.532
	2	31	0.6
	3	29	0.55

4.1.9 Compatibility with available conveyor

4.1.9.1 For Prototype V1

The initially manufactured flap was not compatible with the lab conveyor belt at all as it was designed and manufactured before getting access to the conveyor belt.

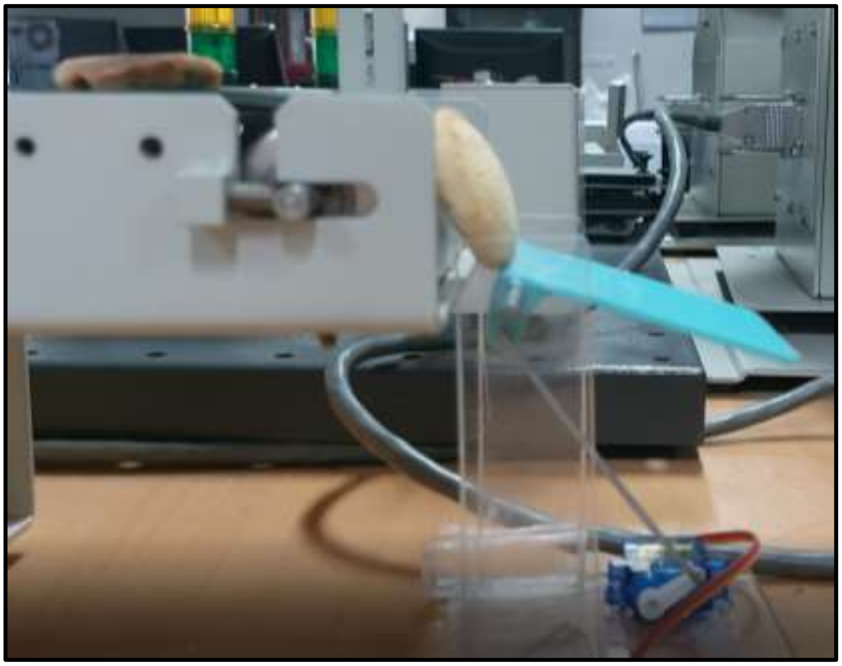

Figure 35: Lack of compatibility of solution with conveyor belt Prototype V1

4.1.9.2 For Prototype V2

Both the scanner and actuator end were designed to be compatible with the available conveyor belt. Both assemblies were designed to be installed as bolt-on standalone assemblies requiring just set of screws and a screwdriver to mount it on the subject conveyor.

The actuator end also took care of seamless transition of biscuit from one conveyor belt to another. A picture of the developed solution is attached below.

Figure 36: Compatibility of solution with conveyor belt Prototype V2

4.2 Shortcomings

4.2.1 Distinguishing between lighter-coloured biscuits

Our algorithm was also tested using the dataset provided by Coronet Foods (Pvt) Limited.

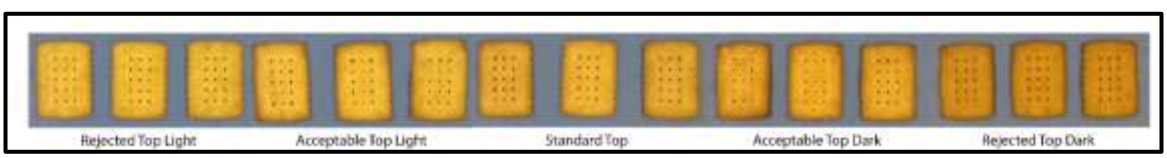

Figure 37: Compiled pictures of biscuits provided by Coronet Foods

We faced the major issue of not being able to make our algorithm distinguish between lighter-coloured biscuits.

The developed algorithm was successfully able to distinguish the difference between "Acceptable Top Dark" and "Rejected Top Dark" biscuits in the figure 38. But it is facing issues in differentiating between the "Acceptable Top Light" and "Rejected Top Light" biscuits due to the minor change in their colour as shown below.

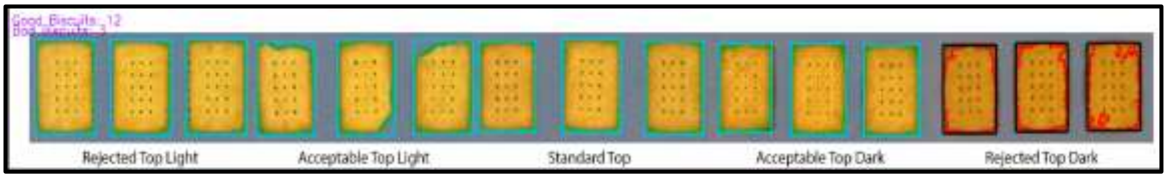

Figure 38: Code Output for the dataset provided by the Coronet Foods

Different strategies such as mentioned below were employed to solve this issue:

- Detecting the white level in grayscale converted images
- Detecting average colour in the contour developed on the biscuit
- Performing colour thresholding in different colour zones (RGB, YCrCb etc.)

4.2.2 Light Source Chromaticity and Luminosity Decay

Lack of any supporting documents along the available LED lights across the globe has left us in doubt regarding the performance of procured LED light strip's performance in degrading their colour and brightness over time. The colour of white light sources is defined in Kelvins. A product offered by any company falls in a particular colour range in which all the produced units must lie.

- Most lighting vendors do not have any documents regarding the range or tolerances the manufacturer had maintained while manufacturing the specific product.

- Nor do the manufacturers provide any detailed information about the light's colour or luminosity range.
- The information about the colour and luminosity decay is even scarcer.

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Quality inspection based on visual deviations from a manufacturer's set standard is a repetitive and tedious task if appointed to human labour. It requires maintaining a flow of experienced workers in 2 to 3 shifts. In today's modern era, where most repetitive industrial processes are being replaced by machines, our developed quality control solution removes the requirement for active supervision of manufactured items by human labour.

The scope of the developed solution is not limited to the biscuit or food industry only. An automated QC solution powered by computer vision can also be implemented in a number of other industries where human labour is used to diagnose visual discrepancies. A slight adaptation in the developed algorithm and modification at the actuator end can assist in deployment of the automated QC machine in several manufacturing industries.

Consistent results, 24/7 operation, low running cost of the designed solution make it a viable option for industrial usage. With machine led quality control, information like number of defective items can also be documented. The effect of changes in the raw material supplier, manufacturing equipment can easily be reflected with the data collected by the QC machine. It can assist in drawing comparison between raw material suppliers and manufacturing equipment providers as well.

5.2 Recommendation

We recommend the following changes for further work on this project.

5.2.1 PLCs instead of Arduino

PLCs are widely adapted in industrial environment due to their rugged design, ability to handle higher voltages and currents, and their modular layout. Most of our decision taking is done on a desktop computer for the solution, where Arduino just handles the communication between the computer and relays for actuating devices. The simple code for Arduino can simply be forwarded to PLC's where programming is mostly done through ladder logic.

5.2.2 SBCs for Processing

Single board computers are much lighter, smaller, and efficient than desktop computers or servers. Transferring the processing to SBCs while maintaining the same performance and accuracy, can reduce the risk of failure. It can also eliminate the use of a device for communication with relays, as most SBC come equipped with their own I/O pins for this functionality.

5.2.3 Design accommodates serviceability and ease of assembly

Design concept should consider the serviceability requirements of such an industrial solution, which might undergo 24/7 operation. Such continuous loads can lead to wear and tear easily. Hence, serviceable parts should have easy access and should be easily available for trouble free maintenance of the equipment.

5.2.4 System components failure detection and report

The QC machine can have the ability to detect failure the setup itself, such as detecting light brightness or colour deterioration. Or Detecting a failing motor, stuck flap or failing bearing etc. Detection of such failures and the ability to alert and report them to the plant manager can be a great add-on to this solution.

5.2.5 Components service life testing and with longer lives

Performing service life study and tests on all involved components, and ensuring the usage of long lasting, quality components can ensure the continuous working of the solution without breakdown 24 hours, 7 days in a week. Testing the lights for their colour, luminosity, and decay overtime is also necessary. These tests can also help in producing a maintenance schedule for the machine.

5.2.6 Better Camera

A camera with higher colour depth, greater frame rate and crispier resolution can result in a more robust and capable machine. Cameras providing extensive control parameters such as the camera’s gain, contrast, saturation, hue, brightness, exposure, white balance, shutter speed, aperture can be more beneficial for performing high speed QC.

5.2.7 Isolation from electromagnetic interference for an industrial environment

Cables carrying video and binary decision signals can require electromagnetic isolation in an industrial environment. Use of a metal sheath around such wires or using optic fibre for communication can help remove such issues.

5.2.8 One button recalibration for updated setup (lights or environment)

Light sources deteriorate their brightness and colour levels over time. An algorithm that takes care of such phenomenon and can recalibrate itself in a non-operational break period can be implemented to recalibrate the algorithm for the current light brightness and colour values. Deterioration past some points after which accurate performance is at stake can be reported to the plant manager for lights replacement.

REFRENCES

[1] ISO 22000:2005. (2016, July 6). ISO. https://www.iso.org/standard/35466.html

[2] Sathe, A., & Deshpande, A. (2017, December). Automated visual quality inspection and sorting. In 2017 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-6). IEEE.

[3] Kits to make aluminium conveyors. (2018). Kit Conveyors. Retrieved from: http://www.kit-conveyors.co.uk/how-to-build-a-conveyor.html

[4] Food conveyor belt. (n.d.). AliExpress. Retrieved from: https://www.aliexpress.com/item/1005002994078753.html?spm=a2g0o.productlist.0.0.4e2e7a42f3w5HI&algo_pvid=5bef7a06-4d94-4b74-b26a-5b05a839c152&algo_exp_id=5bef7a06-4d94-4b74-b26a-5b05a839c152-33&pdp_ext_f=%7B"sku_id"%3A"12000023129240197"%7D

[5] Inkjet printer conveyor belt. (n.d.). AliExpress. Retrieved from: https://www.aliexpress.com/item/32961057737.html

[6] Industrial conveyor belt. (n.d.). OLX. Retrieved from: https://www.olx.com.pk/item/industrial-conveyors-batch-expiry-price-printers-printing-iid-1033532274

[7] EyePro System. (2018, August 30). EyePro system - corporate video [Video]. YouTube. https://www.youtube.com/watch?v=pBnLY0gWEjI&t=94s

[8] Igor Lutsevich, I. [Igor Lutsevich]. (2014, May 22). Linear actuator door opener [Video]. YouTube. https://www.youtube.com/watch?v=myQBIrYzvtU

[9] TAILONZ PNEUMATIC 1/4"NPT Solenoid Valve 4V210-08 12V/24V/110V/220V Single Coil Pilot-Operated Electric 2 Position 5 Way Connection Type. (2019). Amazon. Retrieved from: https://www.amazon.com/Tailonz-Pneumatic-Solenoid-4V210-08-Pilot-Operated/dp/B07SJFV7S3/ref=sr_1_2

[10] CPI control products, inc. (n.d.). What is a Linear Actuator? Retrieved January 24, 2022, from https://www.cpi-nj.com/what-is-a-linear-actuator/

[11] Control pneumatic cylinder with arduino. (2019, December 30). [Video]. YouTube. Retrieved from: https://www.youtube.com/watch?app=desktop&v=iXNNjtK5hp8

[12] Geeplus. (2019). Linear solenoids. Retrieved from:_https://www.geeplus.com/linear-solenoids-2/

[13] Electronics Tutorials. (n.d.). Linear solenoid actuator. Retrieved January 24, 2022, from https://www.electronics-tutorials.ws/io/io_6.html

[14] Imported Electric Linear actuator 100mm Stroke linear motor controller dc 12V 24V 100/200/300/500/600/750/800/900/1100/1300/1500N product in Pakistan. (n.d.). Pkbazaar. Retrieved from: https://www.pkbazaar.pk/car-electrical-appliances/electric-linear-actuator-100mm-stroke-linear-motor-in-pakistan/32688359779.html

[15] Bong, N. (n.d.). Components of an electric linear actuator. Progressive Automations. Retrieved January 24, 2022, from https://www.progressiveautomations.com/blogs/products/inside-an-electric-linear-actuator

[16] howtomechatronics. (n.d.). How servo motor works & how to control servos using Arduino. How To Mechatronics. Retrieved January 24, 2022, from https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/

[17] Simac Masic bv. (2017, January 28). *Biscuit Inspection & Rejection System running on full speed* [Video]. YouTube. Retrieved from: https://www.youtube.com/watch?app=desktop&v=8jLqPuJahqg

[18] Simac Masic by. (2017b, January 28). Biscuit inspection & rejection system running on full speed [Video]. YouTube. https://www.youtube.com/watch?v=8jLqPuJahqg

[19] RaspberryPi. (n.d.). Introducing the raspberry pi cameras. Retrieved from: https://www.raspberrypi.com/documentation/accessories/camera.html

[20] pineStore. (n.d.). PINE A64 5 MEGAPIXEL CAMERA MODULE. Pine64. Retrieved January 16, 2022, from https://pine64.com/product/pine64-5-megapixel-camera-module/

[21] PHOTO4LESS. (2019, February 1). Amazon.com: Canon EOS rebel SL3 digital SLR camera (black): Electronics. Amazon. Retrieved January 26, 2022, from

https://www.amazon.com/Canon-EOS-Rebel-SL3-Black/dp/B07QLNH43L?dchild=1&keywords=canon+sl3&qid=1634571623&sr=8-3&linkCode=sl1&tag=zulfiqarali00-20&linkId=f014584602519d66adc69e5d6a07a967&language=en_US&ref_=as_li_ss_tl

[22] Logitech. (n.d.). C922 PRO HD STREAM WEBCAM. Retrieved January 16, 2022, from https://www.logitech.com/en-us/products/webcams/c922-pro-stream-webcam.960-001087.html

[23] Brosnan, T., & Sun, D. W. (2004). Improving quality inspection of food products by computer vision—a review. Journal of food engineering, 61(1), 3-16.

[24] Srivastava, S., Boyat, S., & Sadistap, S. (2014). A robust machine vision algorithm development for quality parameters extraction of circular biscuits and cookies digital images. Journal of Food Processing, 2014.

[25] Sathe, A., & Deshpande, A. (2017, December). Automated visual quality inspection and sorting. In 2017 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-6). IEEE.

[26] Didactic, Festo. *Belt Conveyor 587777 (5210-05) datasheet.* 01 2022. https://www.labvolt.com/downloads/datasheet_587777(80-5210-5)_en_220V_60Hz.pdf (accessed 2022)

[27] Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart. *Grayscale Images.* 2003. https://homepages.inf.ed.ac.uk/rbf/HIPR2/gryimage.htm (accessed 2022)

[28] Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart. Masking. 2003. https://homepages.inf.ed.ac.uk/rbf/HIPR2/mask.htm (accessed 2022)

[29] *OLX*. n.d. https://www.olx.com.pk/item/air-compressor-box-pack-oil-free-iid-1044847668 (accessed Jan 2022).

[30] Why do I get dark horizontal lines (stripes or bands) in bad lighting with my Fuji X-T1? (2015, March 4). Photography Stack Exchange. Retrieved January 24, 2022, from https://photo.stackexchange.com/questions/60701/why-do-i-get-dark-horizontal-lines-stripes-or-bands-in-bad-lighting-with-my-fu

APPENDIX I: ARDUINO CODE FOR FLAP ACTUATION

```
#include <Servo.h>
int servo1Pin = 3;
int servo2Pin = 5;
Servo Servo1;
Servo Servo2;
int servoPosition =23;
int a;
void setup () {
  Serial.begin(9600);
  Servo1.attach (servo1Pin); //attach the servo pin to the servo
  Servo1.write(27); //start servo at position 0
  Servo2.attach (servo2Pin); //attach the servo pin to the servo
  Servo2.write(30); //start servo at position 0
}
void loop() {
  if (Serial.available())
  {
    a = Serial.read();
    if (a=='L')
    {
      Servo1.write(51.5);
    }
    if (a=='H')
    {
      Servo1.write(27);
    }
    if (a=='P')
    {
      Servo2.write(10);
    }
    if (a=='Q')
    {
      Servo2.write(30); }
  }
```

APPENDIX II: PYTHON CODE FOR BISCUIT DIFFERENTIATION

```
import os
import cv2
from cv2 import transpose
import numpy as np
import serial
import time
from datetime import datetime, timedelta


Arduino_Serial = serial.Serial('COM3', 9600) # to communicate with Arduino
os.system("cls")
Time_delay = 120.5
StartupTime = round(time.time(), ndigits= 2)
totalbiscuits = 0
goodcount = 0
badcount = 0
c = 0
dl=5
gb=15
row1good = np.zeros([1,1])
row1bad  = np.zeros([1,1])
row2good = np.zeros([1,1])
row2bad  = np.zeros([1,1])
cha2=np.zeros([3,1])
cha1=np.zeros([2,1])
y1 = np.zeros([1,2])
brightness_default = 128
contrast_default = 32
saturation_default = 32
hue_default = 32
gain_default = 0
exp_default  = 9
whitebalance_default = 4150
ch=1

# "stackImages" definition code was copied from the internet. Nothing was
modified in this code.
# Code is written to show the original and output image in a single
window.
```

```
# The output image gives us feedback of how many standard and sub-standard
biscuits are present.



def stackImages(scale, imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1]
    height = imgArray[0][0].shape[0]
    if rowsAvailable:
        for x in range(0, rows):
            for y in range(0, cols):
                if imgArray[x][y].shape[:2] == imgArray[0][0].shape[:2]:
                    imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0),
None, scale, scale)
                else:
                    imgArray[x][y] = cv2.resize(imgArray[x][y],
(imgArray[0][0].shape[1], imgArray[0][0].shape[0]),
                                                None, scale, scale)
                if len(imgArray[x][y].shape) == 2: imgArray[x][y] =
cv2.cvtColor(imgArray[x][y], cv2.COLOR_GRAY2BGR)
        imageBlank = np.zeros((height, width, 3), np.uint8)
        hor = [imageBlank] * rows
        hor_con = [imageBlank] * rows
        for x in range(0, rows):
            hor[x] = np.hstack(imgArray[x])
        ver = np.vstack(hor)
    else:
        for x in range(0, rows):
            if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
                imgArray[x] = cv2.resize([x], (0, 0), None, scale, scale)
            else:
                imgArray[x] = cv2.resize(imgArray[x],
(imgArray[0].shape[1], imgArray[0].shape[0]), None, scale, scale)
            if len(imgArray[x].shape) == 2: imgArray[x] =
cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
        hor = np.hstack(imgArray)
        ver = hor
    return ver
```

```
# "getContours" function is defined in the main body of code.
# Further processing of image takes place here.
# The dilated image is fed into this function where the ROI’s contour and
# the area enclosed within those contours are found.

def getContours(img, imgContour, a):
    area_threshold = 38000
    area_noise = 5000
    contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
    good = 0
    bad = 0
    i = 1
    rang = 450
    currentTime = round(time.time(), ndigits= 2)
    global totalbiscuits, goodcount, badcount, StartupTime, row1good,
row1bad, row2good, row2bad, c, cha1, cha2

    for cnt in contours:
        area = cv2.contourArea(cnt)
        peri = cv2.arcLength(cnt, True)
        approx = cv2.approxPolyDP(cnt, 0.001 * peri, True)
        x, y, w, h = cv2.boundingRect(approx)
        c = c + 1
        ca = np.ones([1,2])
        # print(area)

        if area >= area_threshold and 650>(y+h/2)>400: #If condition
defined for good biscuits.
                                                       # we limit
detection of biscuits to middle cropped frame of our video
            cv2.drawContours(imgContour, cnt, -1, (0, 191, 0), 5)
            cv2.rectangle(imgContour, (x, y), (x + w, y + h), (222, 191,
0), 5)
            r = np.array([x, 0])                       # make a matrix of
x-coordinates and decision (0) for each good biscuit
            a = np.vstack([a, r])
            cr = np.array([y, 0])
            ca = np.vstack([ca, cr])
            good = good + 1
```

```python
        if area <= area_threshold and area >= area_noise and 
650>(y+h/2)>400: # Same is done for bad biscuits as done for good shown 
above.
            cv2.drawContours(imgContour, cnt, -1, (0, 0, 255), 5)
            cv2.rectangle(imgContour, (x, y), (x + w, y + h), (0, 0, 0), 
5)
            r = np.array([x, 1])
            a = np.vstack([a, r])
            cr = np.array([y, 1])
            ca = np.vstack([ca, cr])
            bad = bad + 1


        if len(ca)==3:
            ca=np.append(ca,cha2, axis =1)
            print("ca after", ca)
            cha2[:,0]=ca[:,0]
            cha1=np.zeros([2,1])

        if len(ca)==2:
            ca=np.append(ca,cha1, axis =1)
            # print(ca, "ca after")
            cha1[:,0] = ca[:,0]
            cha2=np.zeros([3,1])

        while i<len(ca):
            if len(ca)>1:
                if (ca[i, 2]>=rang>ca[i, 0]):
                    print ("add one")
                    totalbiscuits = totalbiscuits + 1
            i = i + 1


    text0 = "Real Time Counter"
    cv2.putText(imgContour, text0, (10, 35),  cv2.FONT_HERSHEY_SIMPLEX, 
1.5, (0, 255, 255), 3)
    text = "Biscuits QC Approved: " + str(good)
    cv2.putText(imgContour, text, (60, 85),  cv2.FONT_HERSHEY_SIMPLEX, 
1.3, (0, 255, 0), 3)
    text1 = "Biscuits QC Rejected: " + str(bad)
    cv2.putText(imgContour, text1, (60, 135),  cv2.FONT_HERSHEY_SIMPLEX, 
1.3, (0, 0, 255), 3)
```

```
    text5 = "Cumulative Counter"
    cv2.putText(imgContour, text5, (700, 795),  cv2.FONT_HERSHEY_SIMPLEX,
1.5, (0, 255, 255), 3)
    text2 = "Biscuits QC Approved: " + str(goodcount)
    cv2.putText(imgContour, text2, (750, 845),  cv2.FONT_HERSHEY_SIMPLEX,
1.3, (0, 255, 0), 3)
    text3 = "Biscuits QC Rejected: " + str(badcount)
    cv2.putText(imgContour, text3, (750, 895),  cv2.FONT_HERSHEY_SIMPLEX,
1.3, (0, 0, 255), 3)
    text4 = "Total Biscuits: " + str(totalbiscuits)
    cv2.putText(imgContour, text4, (750, 945),  cv2.FONT_HERSHEY_SIMPLEX,
1.3, (255, 255, 0), 3)




    return a

# Webcam video is obtained.
video = cv2.VideoCapture(1)
video.set(3,1280)
video.set(4,960)
video.set(10, brightness_default)
video.set(11, contrast_default)
video.set(12, saturation_default)
video.set(13, hue_default)
video.set(14, gain_default)
video.set(15, exp_default-9)
video.set(17, whitebalance_default)

# The video is obtained with the lowest possible exposure so that the
video can be as sharp as possible.
# This low exposure is made possible by a faster shutter speed, which in
turn allow for a sharp image
# i.e., the image is not blurry.

def empty(a):
    pass

# Trackbars are made to control the video processing parameters while code
is running
```

```
cv2.namedWindow("TrackBars")
cv2.resizeWindow("TrackBars", 1080, 480)
cv2.createTrackbar("B min", "TrackBars", 129, 255, empty)
cv2.createTrackbar("B max", "TrackBars", 255, 255, empty)
cv2.createTrackbar("G min", "TrackBars", 0, 255, empty)
cv2.createTrackbar("G max", "TrackBars", 255, 255, empty)
cv2.createTrackbar("R min", "TrackBars", 0, 255, empty)
cv2.createTrackbar("R max", "TrackBars", 112, 255, empty)
cv2.createTrackbar("Brightness", "TrackBars", brightness_default, 255,
empty)
cv2.createTrackbar("Contrast", "TrackBars", contrast_default, 255, empty)
cv2.createTrackbar("Saturation", "TrackBars", saturation_default, 255,
empty)
cv2.createTrackbar("Hue", "TrackBars", hue_default, 255, empty)
cv2.createTrackbar("Gain", "TrackBars", gain_default, 255, empty)
cv2.createTrackbar("Shutter Speed", "TrackBars", exp_default, 9, empty)
cv2.createTrackbar("White Balance", "TrackBars", whitebalance_default,
10000, empty)
cv2.createTrackbar("Dilation", "TrackBars", 0, 100, empty)
cv2.createTrackbar("Gaussian Blur", "TrackBars", 15, 100, empty)

h1 = 1
h2 = 1
l=1
k=1
A_zeros = np.zeros([1,3])
d_time = time.time()


while True:
    kernel = np.ones((5, 5), np.uint8)
    Y_min = cv2.getTrackbarPos("B min", "TrackBars")
    Y_max = cv2.getTrackbarPos("B max", "TrackBars")
    Cr_min = cv2.getTrackbarPos("G min", "TrackBars")
    Cr_max = cv2.getTrackbarPos("G max", "TrackBars")
    Cb_min = cv2.getTrackbarPos("R min", "TrackBars")
    Cb_max = cv2.getTrackbarPos("R max", "TrackBars")
    brightness = cv2.getTrackbarPos("Brightness", "TrackBars")
    contrast = cv2.getTrackbarPos("Contrast", "TrackBars")
    saturation = cv2.getTrackbarPos("Saturation", "TrackBars")
    hue = cv2.getTrackbarPos("Hue", "TrackBars")
    gain = cv2.getTrackbarPos("Gain", "TrackBars")
```

```
    exp = cv2.getTrackbarPos("Shutter Speed", "TrackBars")
    exp = exp-9
    whitebalance = cv2.getTrackbarPos("White Balance", "TrackBars")
    dl = cv2.getTrackbarPos("Dilation", "TrackBars")
    gb = cv2.getTrackbarPos("Gaussian Blur", "TrackBars")

    video.set(10, brightness)
    video.set(11, contrast)
    video.set(12, saturation)
    video.set(13, hue)
    video.set(14, gain)
    video.set(15, exp)
    video.set(17, whitebalance)
    # cv2.VideoCapture.set)

    success, img = video.read()
    # out.write(img)

    # Obtained video is converted to YCrCb colour space
    imgYCrCb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)

    # Upper and lower limits of the colour space is defined
    lower = np.array([Y_min, Cr_min, Cb_min])
    upper = np.array([Y_max, Cr_max, Cb_max])
    mask = cv2.inRange(imgYCrCb, lower, upper)

    imgResult = cv2.bitwise_and(img, img, mask=mask)

    # Gaussian blur is applied on the mask, which is further dilated.
    imgblur = cv2.GaussianBlur(mask, (15, 15), 15)
    imgdilate = cv2.dilate(imgblur, kernel, iterations=0)

    # Blurring and dilation allows for the noise in the image to be
reduced and
    # also to allow the Canny function to find edges easily
    ab = np.ones([1,2])*2
    cab = np.ones([1,2])*2
    ab = getContours(imgdilate, img, ab)
    # print ("ab", ab)
    Detection_time = time.time()
    Detection_time = Detection_time - (d_time - 1)
```

```
    # The standard and sub-standard biscuits are sorted
    # using their x-coordinate and stored in a matrix.
    data = ab[np.argsort(ab[:, 0])]
    # print ("this is dataaaa before sorting", dataa)



    imgStack = stackImages(1, ([img, mask, imgblur],[mask, imgblur,
imgdilate]))
    cv2.imshow("Stacked Images", imgStack)
    # cv2.imshow("Output", img)
    A = np.ones([1,2])*2 # A zero array of 1x2 is made because in our case
we will consider only 2 bicuits in a row.
    data = np.delete(data, 0, 0) # Delete the 1 extra entry which was
earlier used for stacking.

    dataa = data[:, 1]

    # To correctly account for biscuit postion in x-axis
    # so that correct decision communicated to corresponding arduino.
    if len(dataa) == 1:
        if 960>data[0,0]>480:
            A[0,1] = dataa[0]
        else:
            A[0, 0] = dataa[0]
    if len(dataa) == 2:
        A[0, 0] = dataa[0]
        A[0, 1] = dataa[1]
    # Store only decision array in A
    #                 # Also, A knows the position of good and bad
biscuits present in a row.
    #                 # The size of A = No. of biscuits + 1

    # A = np.delete(A, 0) # Delete the 1 extra entry which was earlier
used for stacking.
    length_A = len(A)

    A = np.append(A, Detection_time+Time_delay) # Introduce time for each
row of biscuit

    A_zeros = np.vstack([A_zeros, A])
```

```
    execution_time = A_zeros[0,2]


    # Lower 1 means 1st servo motor (from left if POV is conveyor forward
direction)
    # and vice versa
    # while Lower 2 means 2nd servo motor
    if execution_time <= Detection_time:
        if (A_zeros[0, 0] == 1 and (h1 == 1 or h1 == 2)):
            Arduino_Serial.write(b'L')
            print("Lower 1")
            h1 = 0
        if (A_zeros[0, 0] == 0 and (h1 == 0 or h1 == 2)):
            Arduino_Serial.write(b'H')
            print("Higher 1")
            h1 = 1
        if (A_zeros[0, 0] == 2 and (h1 == 0 or h1 == 1)):
            print("Nothing 1")
            h1 = 2

        if (A_zeros[0, 1] == 1 and (h2 == 1 or h2 == 2)):
            Arduino_Serial.write(b'P')
            print("Lower 2")
            h2 = 0
        if (A_zeros[0, 1] == 0 and (h2 == 0 or h2 == 2)):
            Arduino_Serial.write(b'Q')
            print("Higher 2")
            h2 = 1
        if (A_zeros[0, 0] == 2 and (h2 == 0 or h2 == 1)):
            print("Nothing 2")
            h2 = 2

        A_zeros = np.delete(A_zeros, 0, 0) # Deletes preceding row
everytime it runs.


    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

APPENDIX III: BILL OF MATERIALS

Table 11: Bill of Materials

Sr.	Type	Part	Item	Description	Number	Unit Cost	Total
1	**Prototype 1**	Enclosure	Webcam	Logitech	1	5190	5190
2			DC Lights	12V	1	120	120
3			Chloroform	Acrylic	1	100	100
4			Potentiome	Fan Dimmer	1	150	150
5			Steel Rod	3mm dia	1	30	30
6		Flap	Acrylic	1×2 Feet,	1	480	480
7		Encoder	Wheel	Slotted	1	50	50
8			Bearings	3mm dia	2	50	100
9	**Prototype 2**	Enclosure	Switch	Toggle	1	50	50
10			DC Lights	12V	1	150	150
11			Steel Rod	3mm dia	1	30	30
12			Rivets	3.2x5mm	20	1.5	30
13			DC Jack	Generic	1	50	50
14			Sheet	23 Gauge,	1	260	550
15			DC Adapter	12V; 3A	1	350	350
16			Nuts,	M3	35	5.5	192
17			Micro-B	-	1	130	130
18			USB Type B	-	1	135	135
19			Screws	M3x8mm	30	5	150
20			Velcro	-	3	99	99
21		Flap Assembly	Flaps	3D Printing	40	10/gram	800
22			Flap Holder	3D Printing	150	12/gram	1800
23			Nuts,	M3	35	5.48571	192

24			Screws	M3x8mm	30	5	150
25			Servo	SG-90	3	350	1050
26	**Miscellaneous**		AC lights	220V	8	25	200
27			AC Strip		1	40	40
28			Conveyor	3 pin Light	2	100	200
29			Convyor	200mA,	5	20	100
31			Wire	Generic	4ft	20	80
32			L-Key	3mm	1	60	60
33			L-Key	2.5mm	1	60	60
34			L-key Bolts	M3x8mm	30	8	240
35			L-Key Bolts	M4x15mm	15	6.7	100
36			Voltage	7850	3	30	90
37			Screws,	M2x12mm	8	12	96
38			FICS Banner	2x5 ft	1	1000	1000
39			Conveyor	3 pin Light	1	120	120
40			Scotch		1	50	50
41			Paper Tape	Normal	1	100	100
	Total						**PKR 14664 / $73.32**

APPENDIX V: ENGINEERING DRAWING OF FLAP ASSEMBLY V2

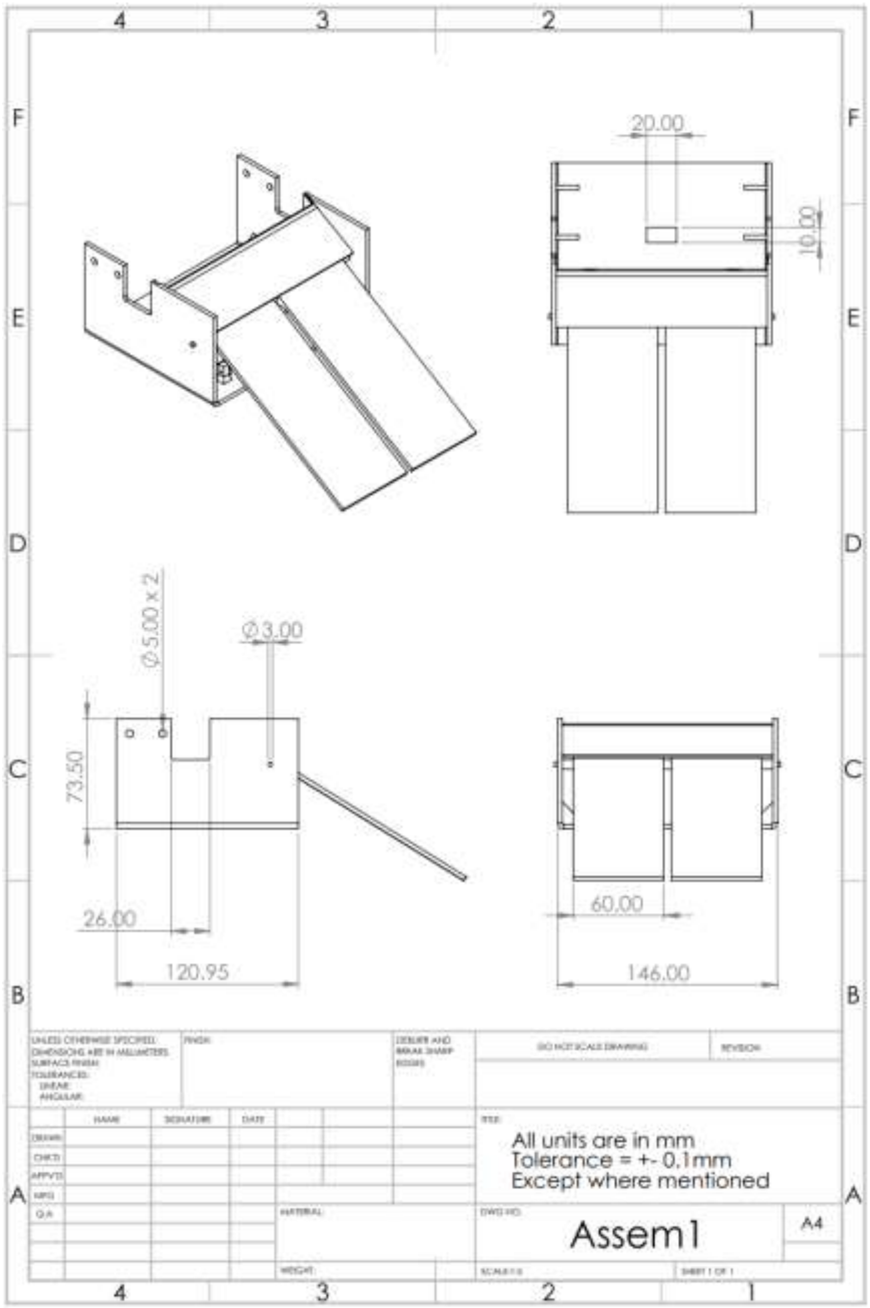

Figure 39: Engineering Drawing for Flap Assembly V2

APPENDIX VI: ENGINEERING DRAWING FOR LIGHT ENCLOSURE

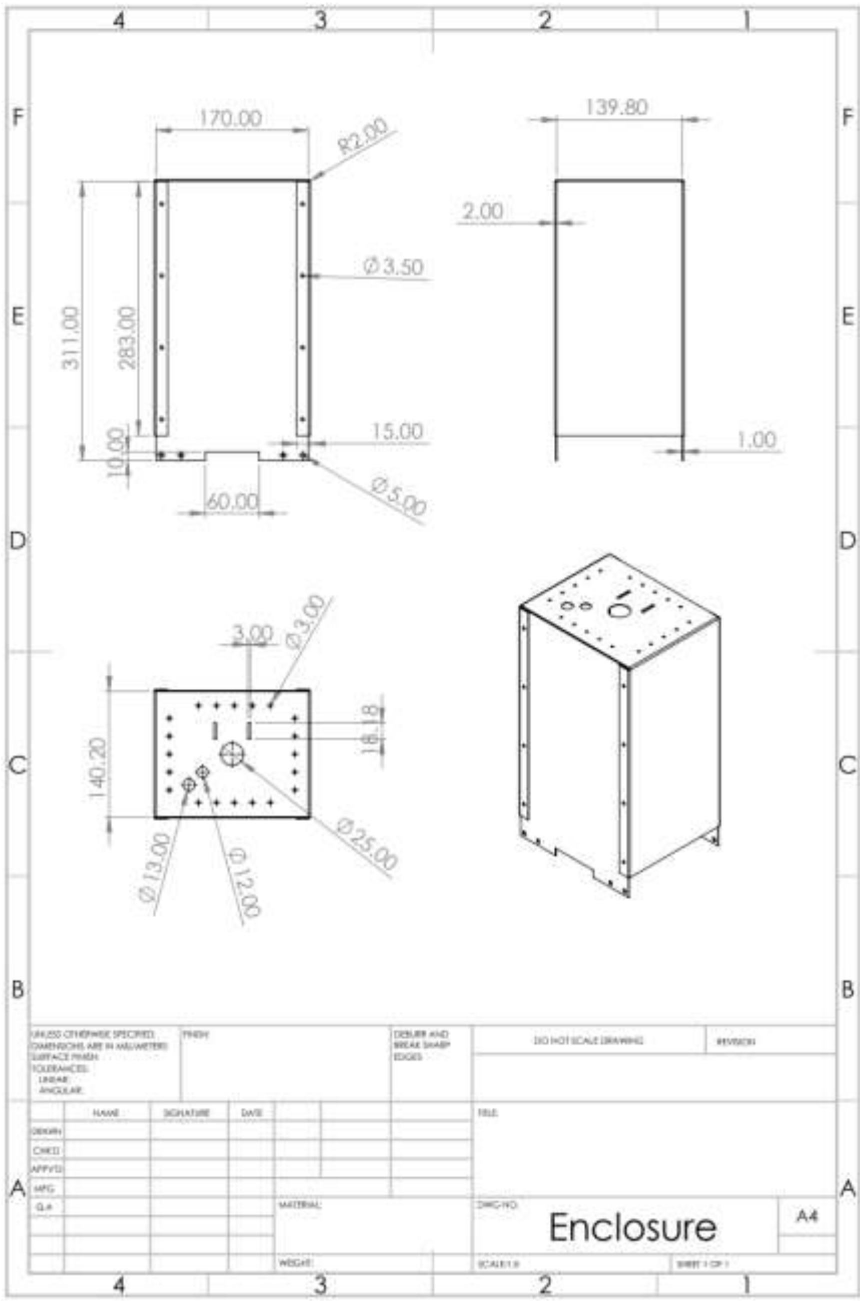

Figure 40: Engineering Drawing for Light Enclosure

APPENDIX VII: ENGINEERING DRAWING FOR ENCODER

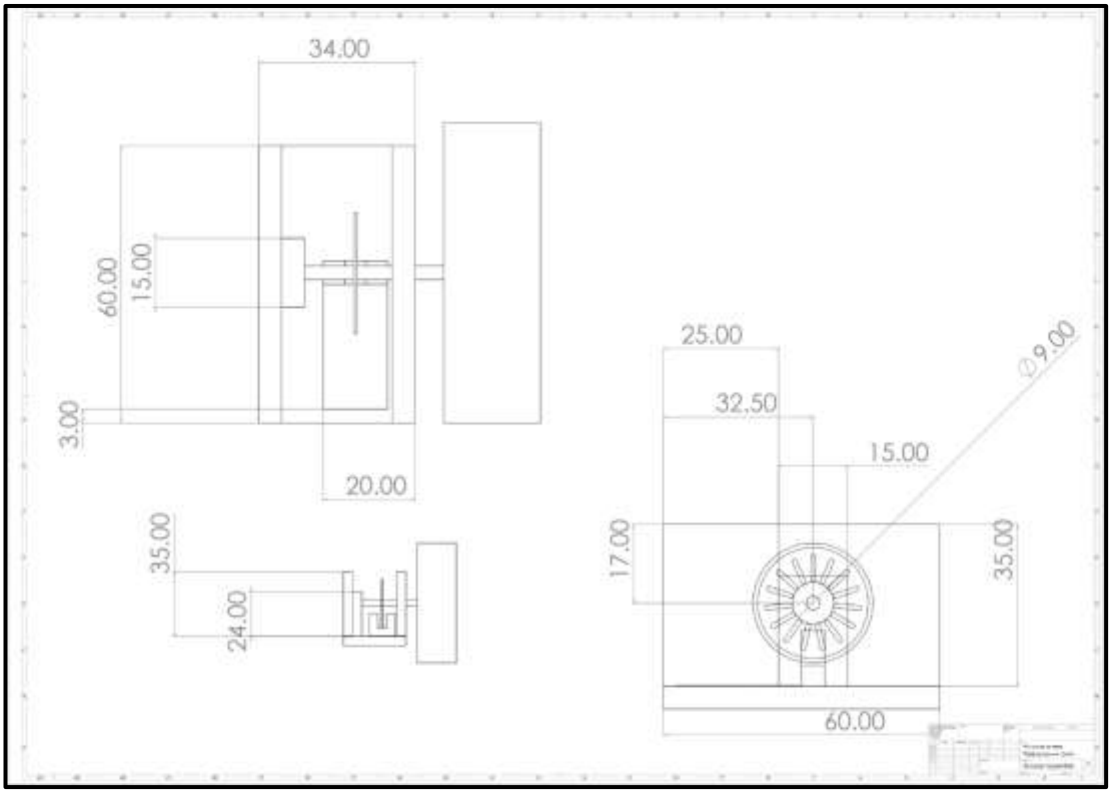

Figure 41: Engineering Drawing for Encoder

APPENDIX VIII: ENGINEERING DRAWING FOR FLAP ASSEMBLY V1

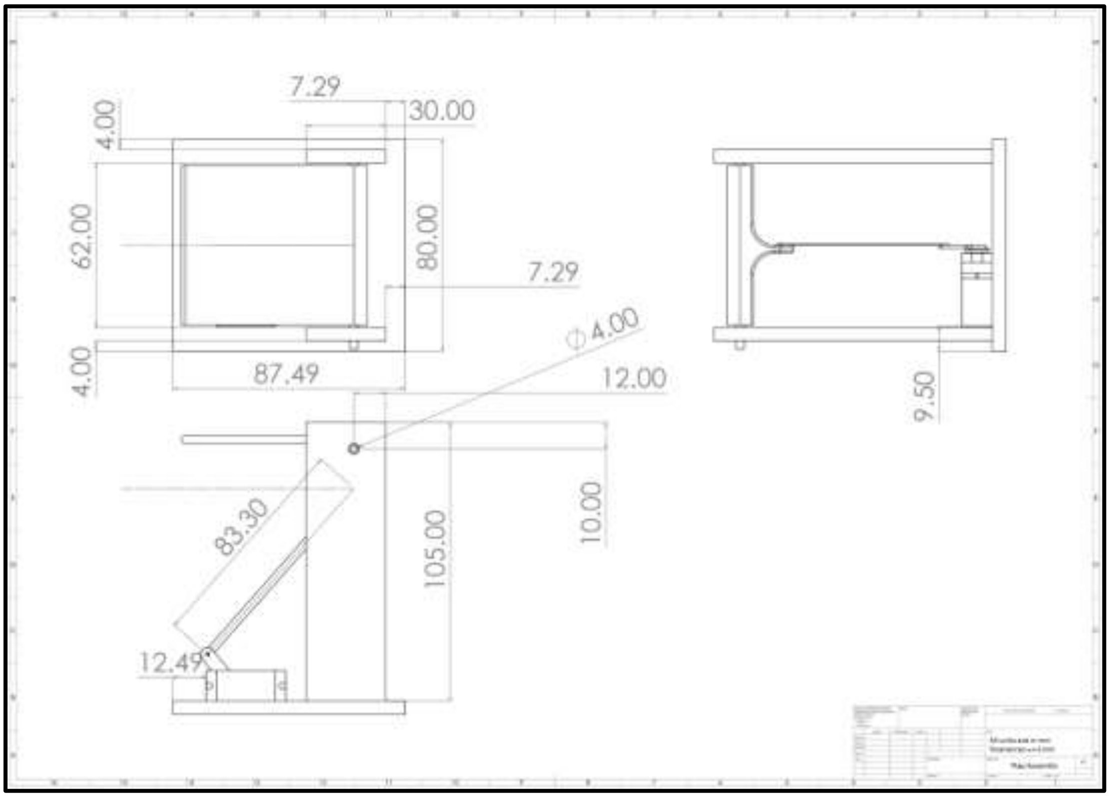

Figure 42: Engineering Drawing for Flap Assembly V1