Yaskawa Group 2 Design Proposal:

Identifying Cap and Orienting Test tubes

Chin Wei Wu
Liu Ya
Jiaoran Wang

AME 547, Fall 2019
Aerospace and Mechanical Engineering Department
University of Southern California
12 November 2019

Table of Content

1. **Summary and Problem Statement**

This document presents the proposed design of Yaskawa Test Tube Sorting and Fetching Group 2. The Yaskawa design is separated into 3 groups, where group 1 is responsible for singulating the test tubes with different cap and tube sizes from bulk. Group 2 is responsible for identifying the caps and then orientating and scanning the test tubes. Group 3 is tasked to place the scanned test tubes into the puck, so the test tubes can be transported to where the test tube has been classified to go. Hence, the design project gives significant challenges to propose a design that has precision and accuracy as well as efficiency (1000 test tubes/hr). This report will include our top two designs and will be evaluated using the concept scoring method by the AHP (Analytic Hierarchy Process) matrix our group generated from the project needs.

2. **Design Needs and Specifications**

In order to evaluate and determine which design option can satisfy the needs more effectively and efficiently, our group will first evaluate the importance of the needs of the project. After talking to our project advisor (Eric), our group determined speed rate, cost, reliability, precision/accuracy, safety (test tubes), workspace and user-friendly are the main factors that need to be considered while evaluating the design.

Table 1. AHP Matrix for Project Needs

	Speed Rate	Cost	Reliability	Precision/ Accuracy	Safety (test tube)	Workspace	User-friendly	Total	Weighting
Speed Rate	1	2	1/2	1/3	1/4	1	2	7.08	0.1
Cost	1/2	1	1/2	1/4	1/5	1/2	1/3	3.28	0.05
Reliability	2	2	1	2	1/3	2	1	10.33	0.15
Precision/ Accuracy	4	4	1/2	1	1/3	2	1	12.83	0.18
Safety (Test tubes)	4	5	3	3	1	4	3	23	0.32
Workspace	1	2	1/2	1/2	1/4	1	1/2	5.75	0.08
User-friendly	1/2	3	1	1	1/3	2	1	8.83	0.12

	Total	71.12	

From the given needs to consider the project design, our team ranked the needs against each other in order to determine the weighting of each factor. According to Table 1, the most important need is the safety of the test tube. This is certainly important because our team wants to assure that our design proposal will not damage the test tubes when going through the process of sorting and scanning. The next most important need is precision and accuracy. This is another very important factor because a precise and accurate will allow the sorting and scanning system to be more efficient and minimizing the chances of failure.

3. Design Proposal

Following the creation of our AHP needs matrix, we began generating design concepts for our system. The role of recording our brainstorming sessions was passed between teammates so all ideas were evenly shared. Design concepts were mostly pitched in the form of sketches or rough verbal descriptions. Resulting from this first part of brainstorming was an overabundance of ideas. The second part of our process focused on combining and refining ideas. Similar ideas were combined into more complex concepts and ideas that lacked solid sketches or descriptions were improved or discarded. In the end, our group came up with two designs for scanning and orienting the test tubes. The designs will be evaluated from concept scoring the two designs in order to determine which design scores better in terms of satisfying the project needs.

3.1 Tube Sample Analysis

3.1.1 Tube Information

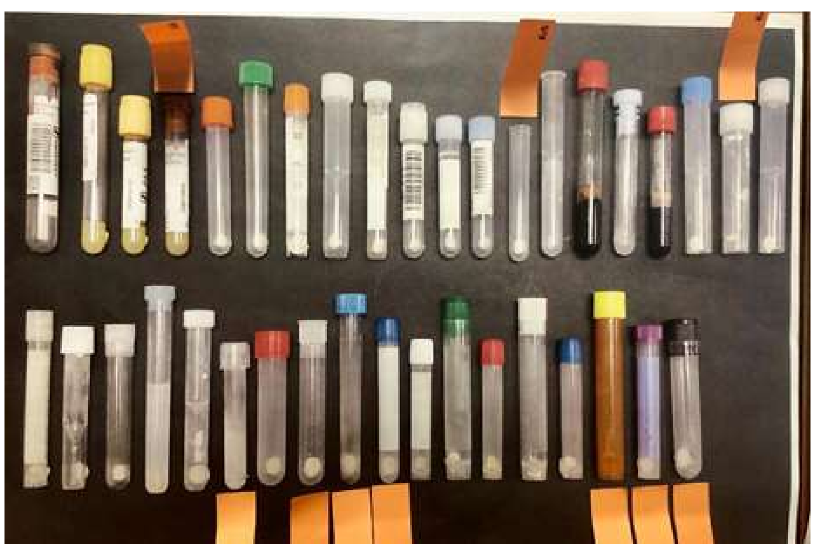


Figure 1. Tube Samples

Table 2. Types of Test Tubes

Test Tube Type	Body	Cap	Bottom	Material /Color
	Transparent-plastic Cylinder Length:82-100mm Diameter: 12mm-18mm	Plat/rounded: (9-19mm) Brim: (2-19mm)	Indent plat curved arc (up to 17mm)	Mixed-color Rubber (red & yellow, black & green) Plastic(transparent/colored)

3.1.2 Special Test tubes sizes for 3D Modeling

Table 3. Selected Test tube for 3D Model

Tube Number	Body	Cap	Bottom	Material /Color
No.1	Cylinder(85mm)	Indent	Cone(14mm)	Mixed-color Rubber (red & yellow)
No.2	Cylinder(60mm)	Outside	Hemispheric(R=5mm)	Plastic

Consider the system need to fit all the tube sizes, the diameter of rollers wrapped with rubber is 20mm along with 5mm interval between them in System A and B. Based on the tube samples, the length of roller in System A is determined to be 20mm and 18mm in System B.

Among practical industrial use, the length of roller system can be adjusted by moving the barriers.

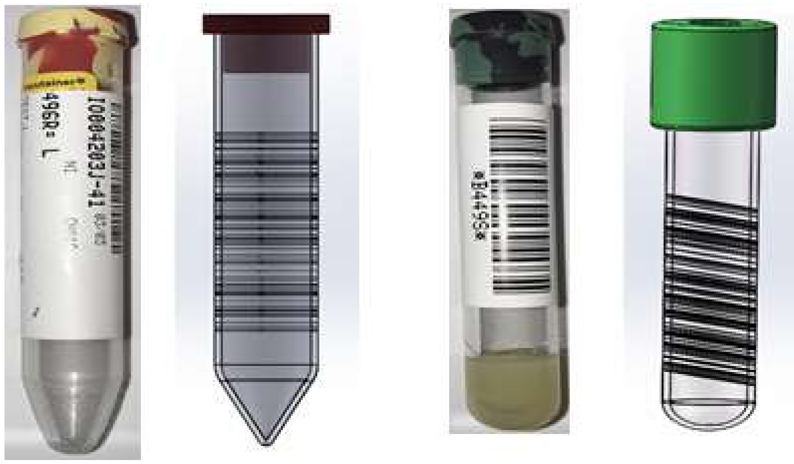


Figure 2. Tube Models (Left: No.1 Right: No.2 Tube)

3.2 Workspace

1. Workspace Size is given by Yaskawa industry, where the whole system would be in a transparent box. The system was designed to work in a people-free area.
2. Lighting Requirement
3. Temperature Requirement

3.3 Design Proposal A: Roller-Convoy Sorting System

Our team's first design proposal uses rollers, conveyors and OpenCV detector to determine the cap and orient the test tubes. First, the test tubes will be fed individually by group 1. Then the test tubes will go on a conveyor that can transport the test tubes into the OpenCV detector. The test tube will be placed perpendicular to the conveyor's direction of motion. The conveyor's speed will be at least 0.05m/s. The test tubes will be transferred with 10 test tubes side by side on the conveyor (800mm) with interval of 80mm. This will assure that the test tubes are carefully transferred and scanned by the OpenCV detector while maintaining at an adequate speed rate.

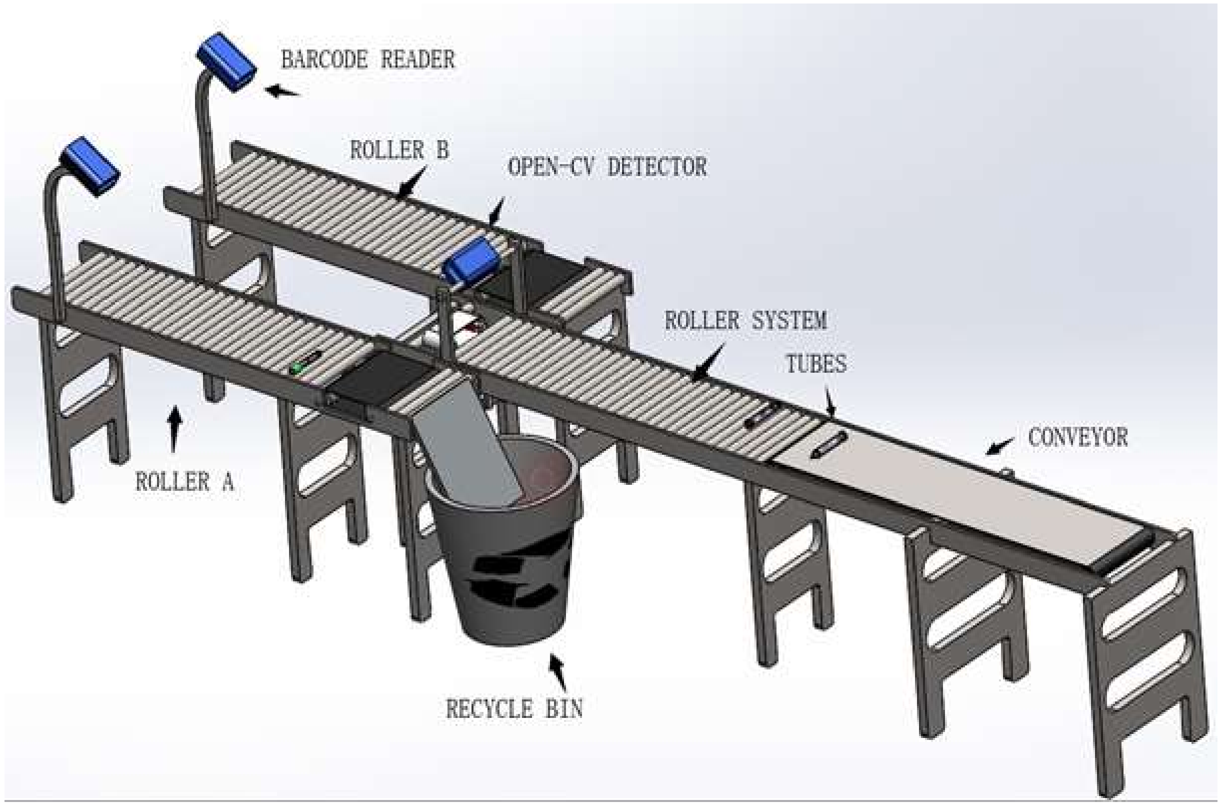


Figure 3. System A: Roller-Convoy Sorting System

The OpenCV detector can analyze the shape and color of the test tubes and detect which end of the test tube is the cap. It will take on average 1 seconds per tube. The OpenCV scanner will then transfer the information regarding the test tubes to the horizontal roller that can divide the test tubes into two different sections. One section consists of test tubes with their caps on the right side and the other section that has test tubes with their caps on the left side. The middle conveyor will move slower in order to accurately and safely move the test tube to the correct section.

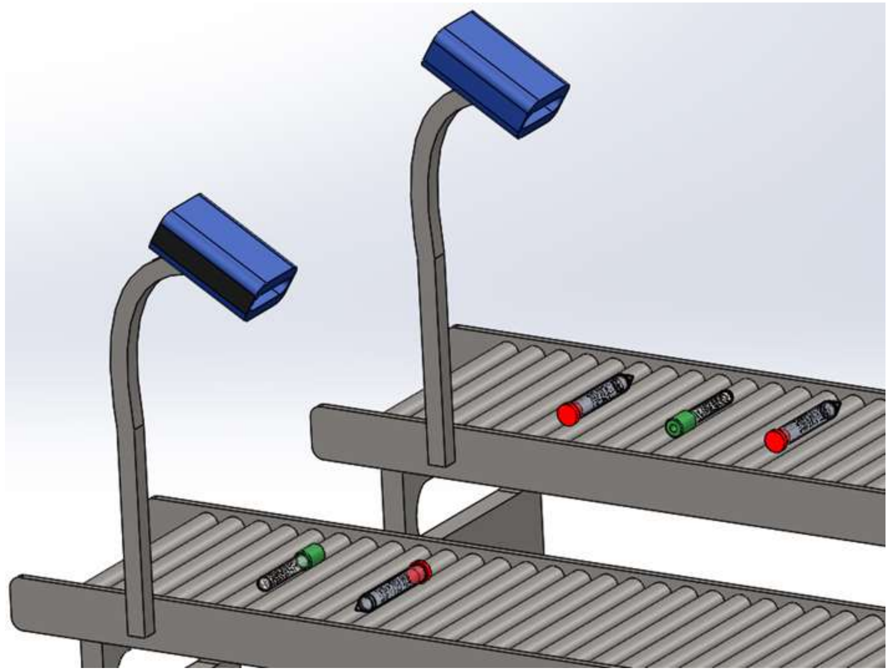


Figure 4. Barcode Reader

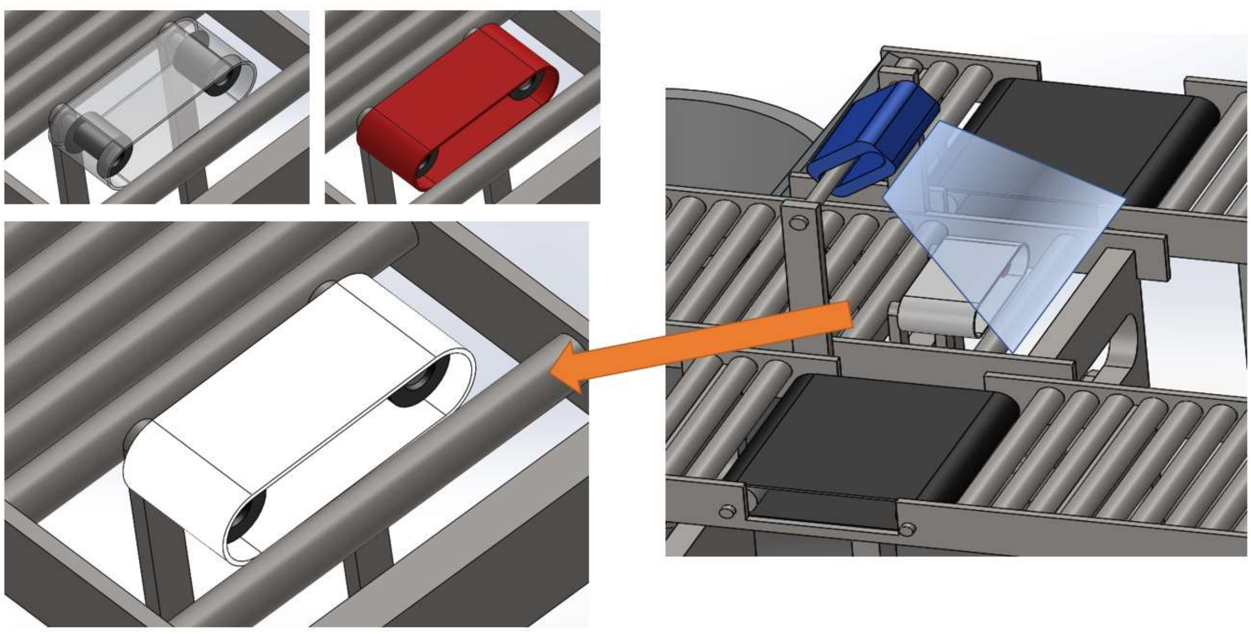


Figure5. OpenCV Detector

Transparent colors and white do not affect color recognition. Other colors, especially primary colors, do not only affect the recognition of opencv colors and shapes. In order to recognize the cap rapidly with certain accuracy. The detecting area (mainly the middle conveyor could just be White No-reflection Material.

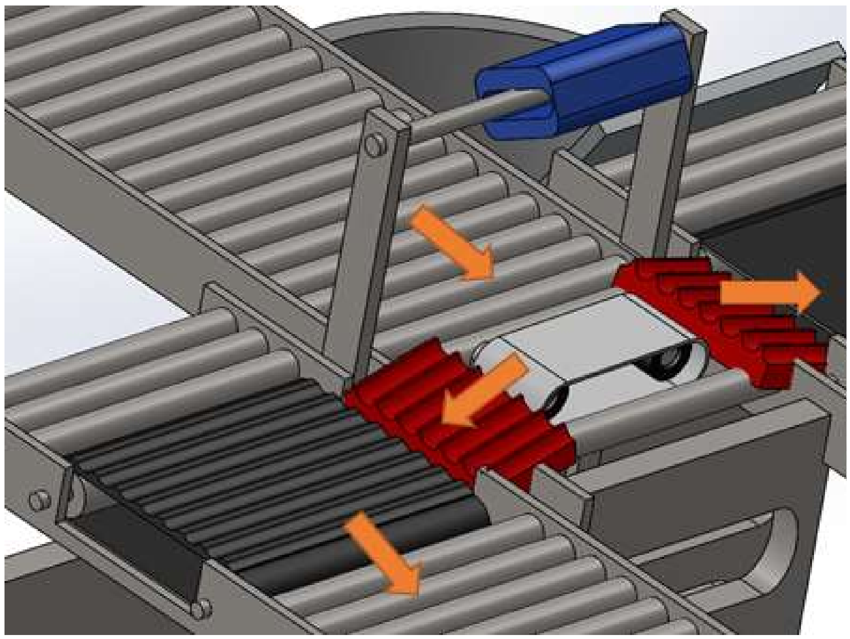


Figure 5. Close up on the middle conveyor for OpenCV Detector

As shown in the figure above, the middle conveyor will move the test tubes to two different sections. The test tube will go through the red part that helps aligning the test tubes so it can precisely slide down to the rigid conveyor. The rigid conveyor assures that the test tubes are perpendicular to the motion of the conveyor, making it easier for the test tubes to travel down the rollers. The test tubes will then travel down the rollers and will be scanned by the barcode scanners at each section.

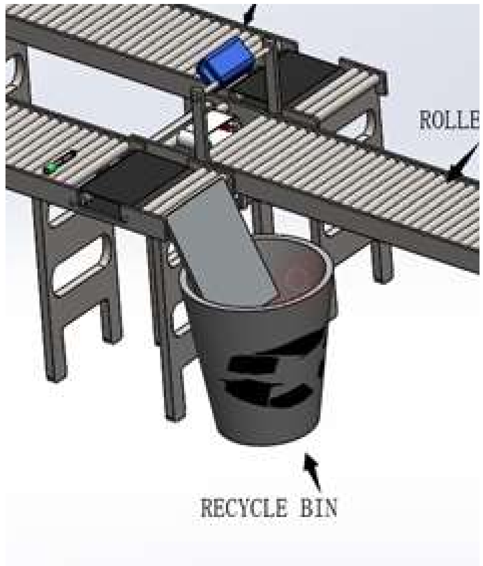


Figure 6. Recycle Bin Provided in the Sorting System

Recycle bin is used in the design proposal for test tubes that fail to be successfully scanned by the OpenCV detector. The bin with the failed test tubes can then go through the process again to make sure that all test tubes are scanned and are brought to its correct place.

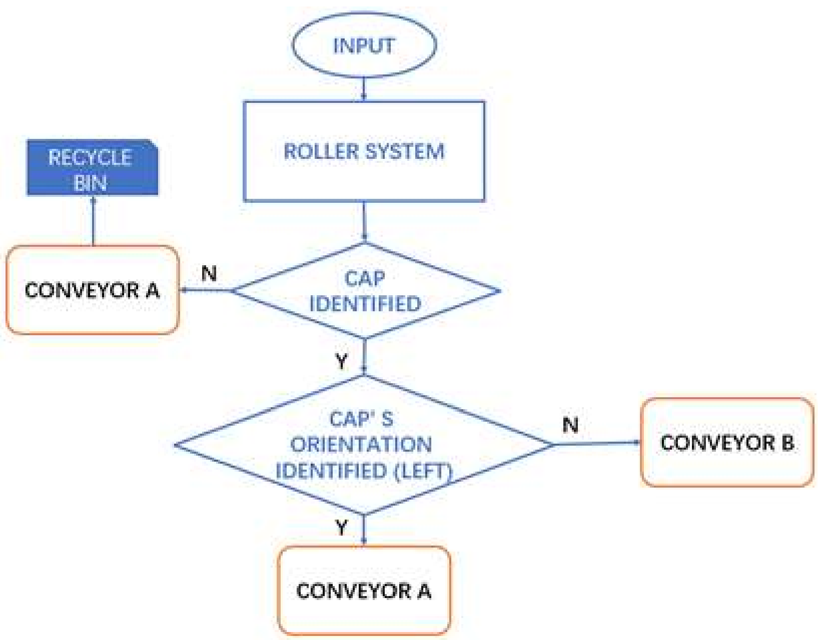


Figure 7. Sorting System Block Diagram

3.4 Design Proposal B: Circular Track Sorting System

Our group's second proposed design is a circular track, where the specially designed trays can circulate in a loop to assure that all the test tubes are differentiated accordingly. After the test tubes are fed one by one vertically by group 1, the test tubes will be moved by a conveyor. During the time when the test tube travels down, the test tubes will be scanned by the OpenCV detector. The detector will then determine the location of the cap and will transmit the information regarding the position of the cap into the tray mechanism.

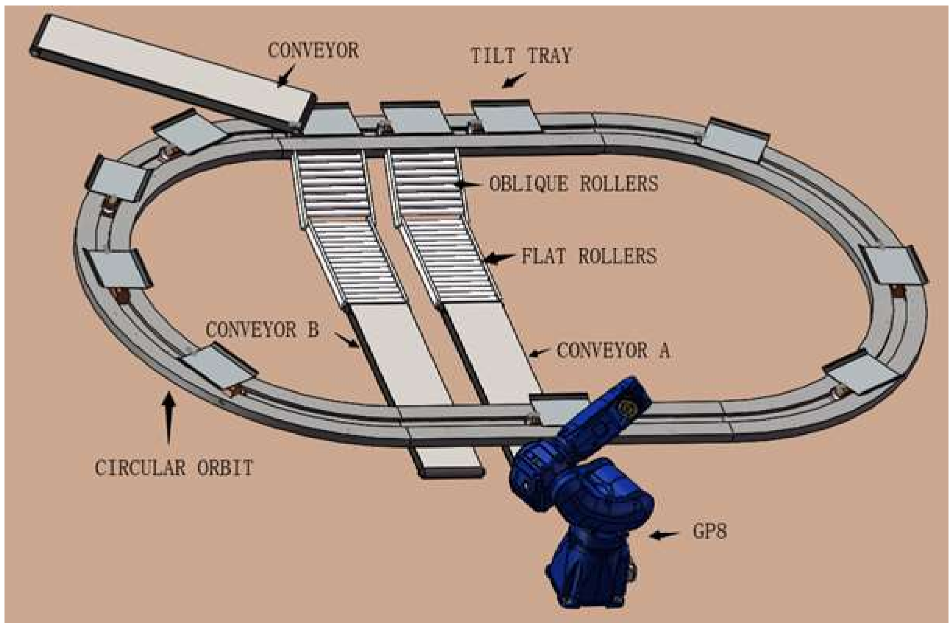


Figure 8. System B: Circulating Loop Sorting System

The conveyor will then place the test tube into the tray and depending on the position of the test tube, the tray will tilt and slide the test tube down. The trays will tilt at different positions depending on whether the cap of the test tube is located at the back or front.

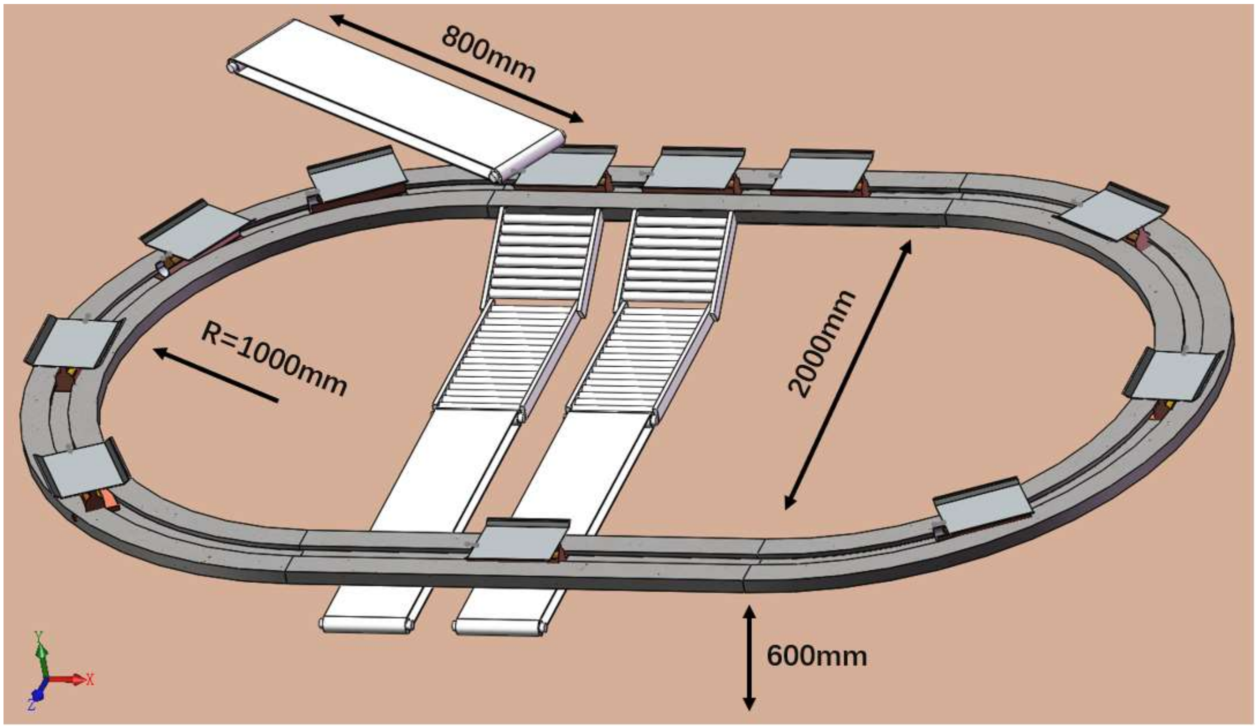


Figure 9. System Size

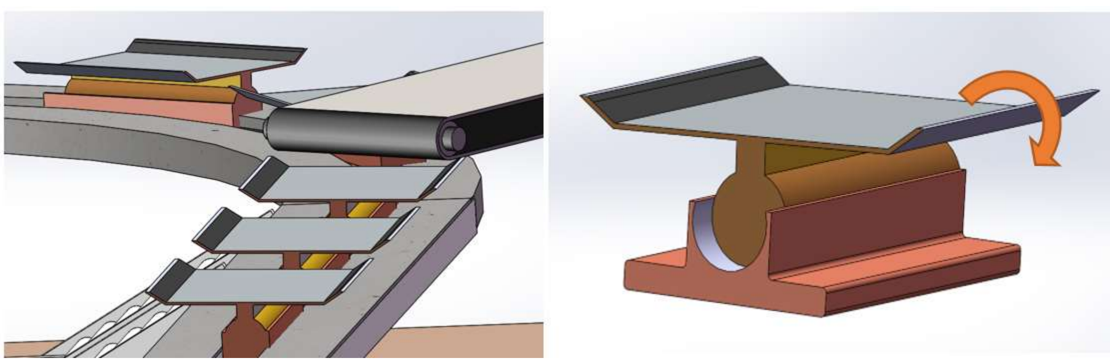


Figure 10. Close up on Tray Mechanism

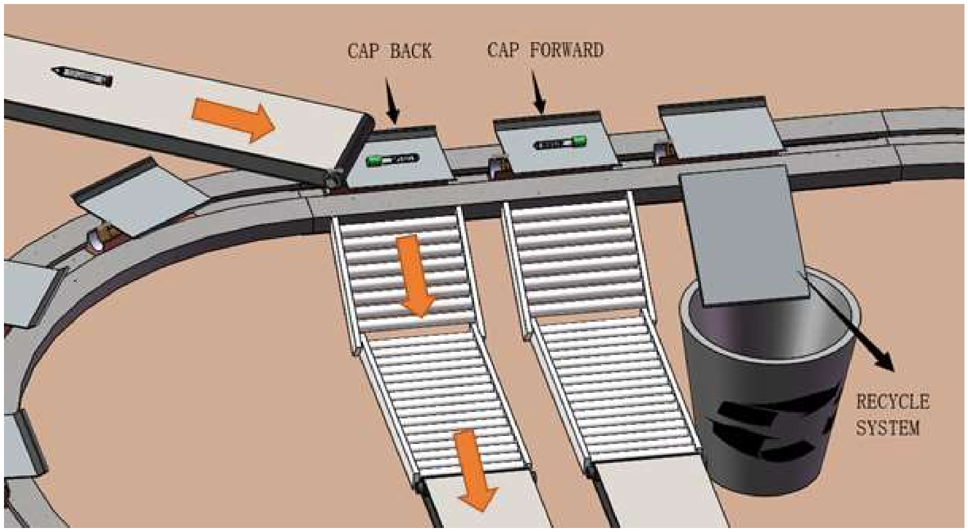


Figure 11. Divided rollers by test tube cap (Front and Back)

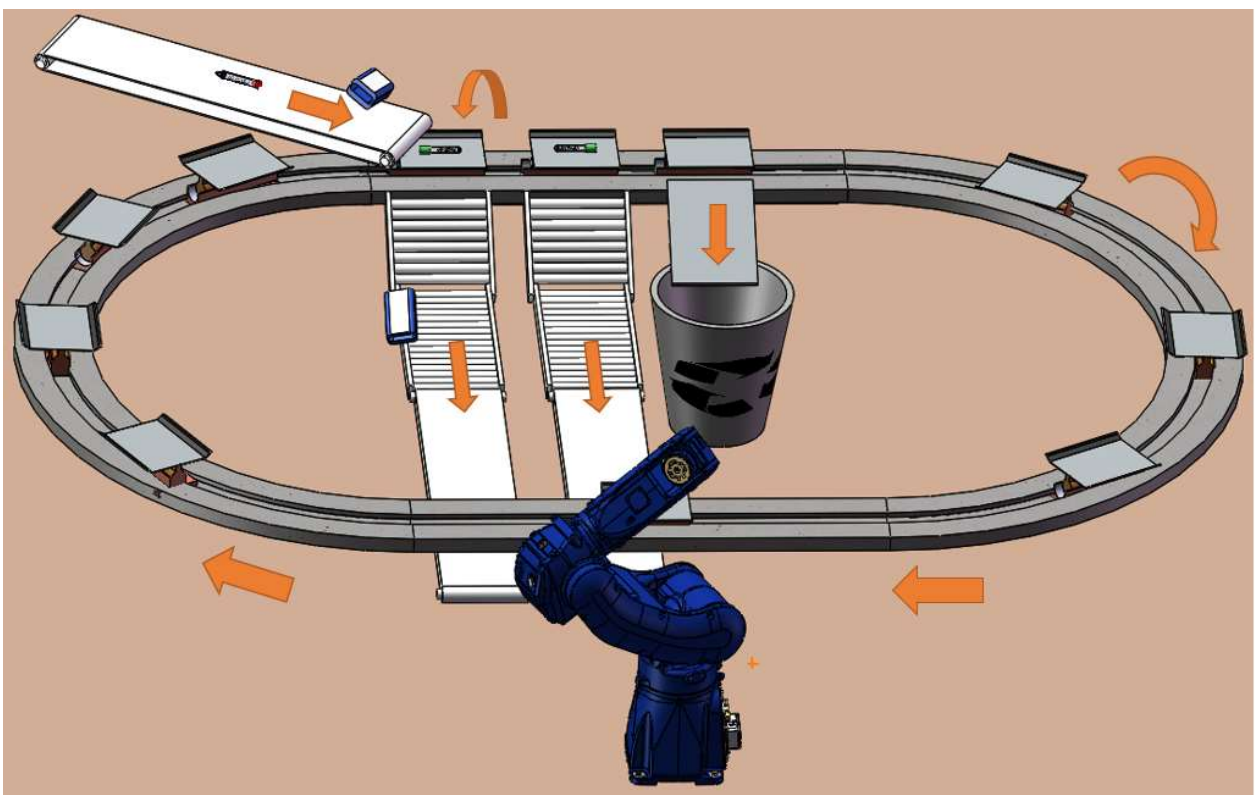


Figure 12. Overall Movement

After the test tubes have been divided accordingly, the test tubes will then roll down from the rollers and travel on the conveyor for barcode scanning.

4. OpenCV Detector

4.1 Color detection from OpenCV Detector

The OpenCV detector holds an important role in our proposed design because it is able to detect the cap of the test tubes by scanning its color and shape. The assurance of scanning orientating the test tubes heavily depends on the accuracy of the OpenCV scanner. Through MatLab, the OpenCV detector can detect the shades of color as well as its shape. The OpenCV scanner determines the object colors by detecting the RGB (red, green, blue) component value and the grayscale value.

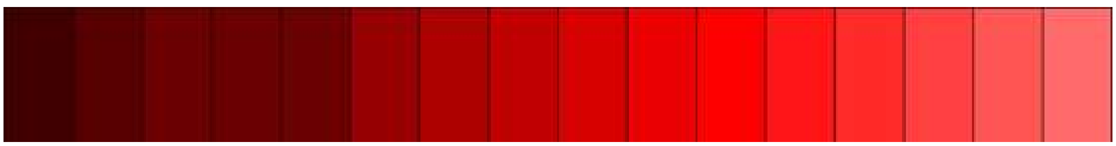


Figure 11. Spectrum of the color red

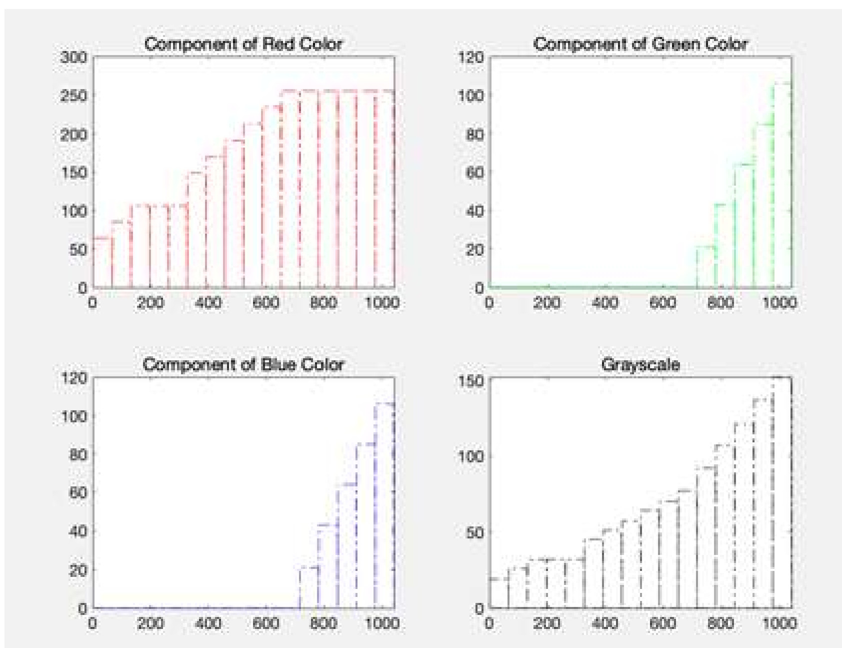


Figure 12. RGB value for the spectrum of colors

When the OpenCV scanner scans through the red spectrum of colors from left to right, the data will show shows charts related to the RGB and grayscale value as shown in Figure 11. The values across from 0-1000 have increased for both the red, green, blue and grayscale value. This proves that the color of red is getting brighter. The same method can be applied when scanning the test tubes. The OpenCV scans the test tube and determines the RGB and grayscale value for the test tube.

Figure 13. Detecting the test tube’s color with OpenCV scanner

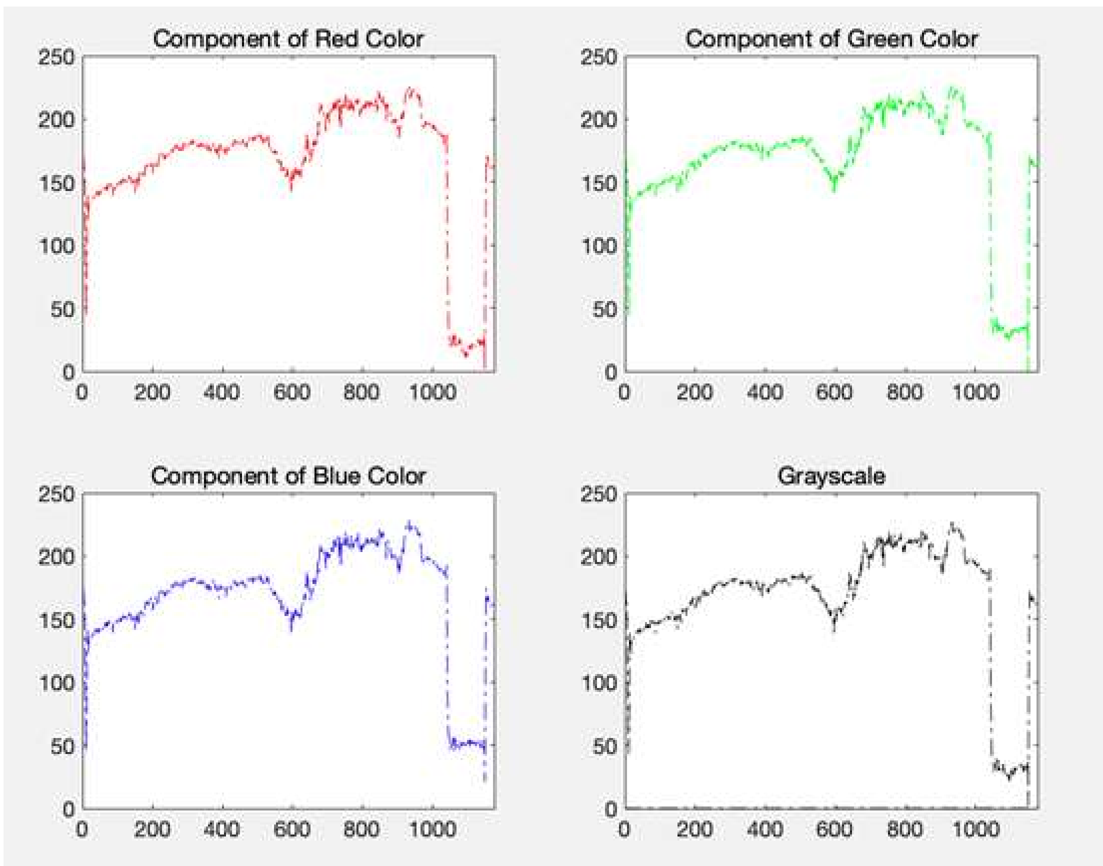


Figure 14. RGB and grayscale value for the test tube in Figure 13

When the test tube is scanned through the OpenCV, the RGB and grayscale value chart will be determined. From the charts, the data is able to determine the position of the cap by the significant change in the value. The change in the value can signify the position of the cap. In this case, the cap is located on the right side of the test tube. In our team’s design process, we plan to use the grayscale value to determine the position of the cap, because of its accuracy and precision in determining the change in color of the cap and the test tube.

4.2 Shape detection from OpenCV Detector

Apart from detecting the test tube's color, OpenCV is also able to detect shapes. The scanner is able to scan the 3-dimensional object and convert the shape into a binary image. The scanner will then remove the background of the test tubes and focus on the shape of the test tube.

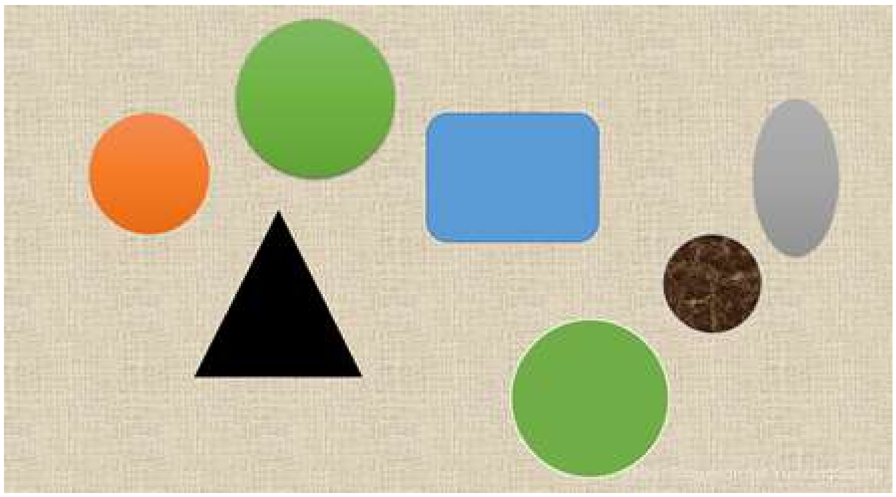


Figure 15. 3D object converted to binary image

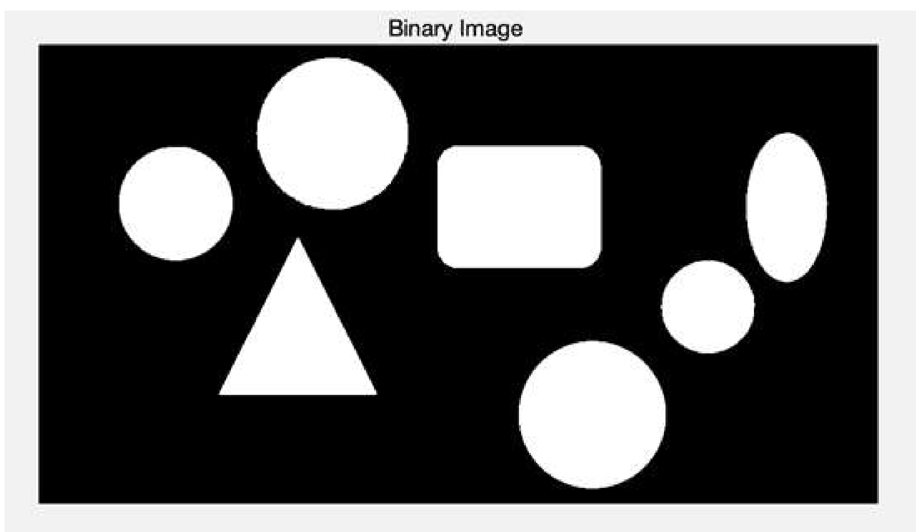


Figure 16. Removed background from the binary image

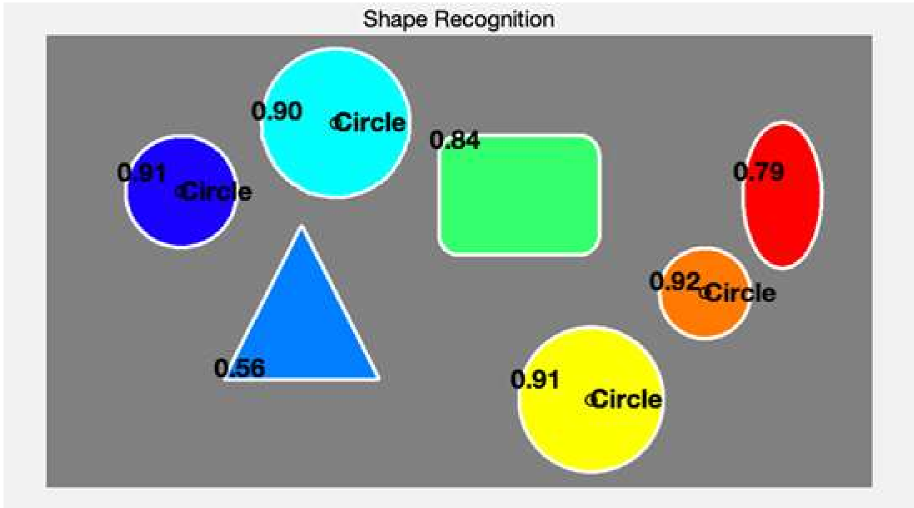


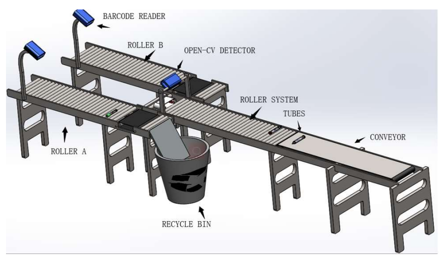
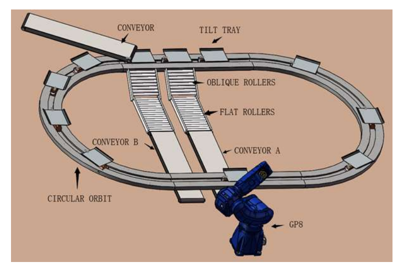
Figure 17. Shape recognition from OpenCV Detector

The OpenCV detector is able to determine the shape and sizes of the object. The OpenCV scanner will be placed at the two sides of the test tube; hence, it is able to determine the position of the cap by comparing the sizes at each end. The size with the bigger surface area will prove that the following side is the cap. Though this method of using OpenCV at each side as compared to the method of using one OpenCV detector is a lot less practical. Hence, our group prefers the method of using color differentiation as the better method for determining the position of the cap.

5. Concept Scoring and Selection

After carefully considering our customer needs and generating two design proposals, the team began the task of determining which concepts would be included in our final design. The screening matrices is used to compare design proposal A and B from each category based on how well the concepts satisfied our project needs (scoring from 1 to 5). We used a basic scoring method to determine the viability of each concept.

Table. 4 Concept Scoring the Two Design Proposals

		Design A		Design B	
					
Selection Criteria	AHP Weight	Score	Weighted Score	Score	Weighted Score
Speed Rate	0.10	3	0.3	4	0.4
Cost	0.05	5	0.25	2	0.1
Reliability	0.15	4	0.6	3	0.45
Position / Accuracy	0.18	3	0.54	4	0.72
Safety (Test tubes)	0.32	3	0.96	4	1.28
Workspace	0.08	4	0.32	1	0.08
User-friendly	0.12	5	0.6	3	0.36
		Sum	3.57	Sum	3.39

As shown in Table 4, our team determined that proposal A is the most viable and satisfies the needs more adequately. The two proposals consist of its own pros and cons. Proposal A is a more compact design that is able to operate at very limited space. The parts needed for the system does not require high skills to operate. Another huge benefit for this type of sorting system is the motion of test tube is kept at very minimal and the system is able to respond to situations where OpenCV detector fails to scan the test tube. The test tubes will be placed into the recycle bin, so the test tubes can be scanned again.

Although sorting system A has many benefits, it consists of a few major flaws. Firstly, the system responds to test tubes that fails to be scanned by dropping the test tubes into the recycling bin. The process of this is inadequate because there might be potential damage to the test tubes during this process. Furthermore, this method of responding failed test tubes requires workers to physically pick up the bin and put it back into the system in order to get it scanned again. This brings inconvenience to the users of this system.

Our team's design proposal B scored lower than A; however, it consists of some benefits that makes it an adequate replacement for proposal A. The circulating sorting system is efficient in that it is able to run at a higher speed than system A; hence, the system works faster in sorting and orientating the test tubes. The system also works better in responding to failed test tubes, as the trays can directly tilt the test tube into the recycling bin, which requires much less time than that of design proposal A.

The disadvantages of proposal B which makes it less desiring is the use of vast amount of space. The design also requires more money to function the system, since the parts are more complicated to build. The maintenance fee will also stand as an issue, because of the complicated design.

6. Conclusion

After using the AHP matrix method of analysing the project needs, our team determined two design proposals that can efficiently identify the cap and orientate the test tubes. Both systems rely on OpenCV detector that is able to use color and shape to determine the location of the test tube's cap. The first design proposal was the Roller-Convoy Sorting System. This system is very simple and effective. The system relies on the middle conveyor to separate the test tubes into two sections. From this, the test tubes will be grouped together with caps on the same side. Our second design proposal is Circulating-Loop Sorting System. This method relies on trays that goes in a circular path and can tilt it's surface to roll the test tubes down the rollers. The rollers will then go into the conveyor and through the barcode scanner.

The two design proposals uses different mechanisms to distribute the test tubes. Through our AHP matrix on design needs, our team is able to determine which design satisfies better from each category. After concept scoring and selection with the AHP matrix, our team determined Roller-Conveyor Sorting System was the better option out of the two. The system stood on top because of its compactness, simplicity and its desirable cost.

7. Appendix

<u>Matlab code of cap identification by color analysis</u>

```
clc;
clear;
Path='/Users/liuya/Desktop/tube.png';%Image Path
Source_img=imread(Path);%Get image information
[r,lie]=size(Source_img(:,:,2));
read_l=20;%Select a line in the image
x=1:lie;
R_value=Source_img(read_l,:,1);%Read red color component
G_value=Source_img(read_l,:,2);%Read green color component
B_value=Source_img(read_l,:,3);%Read blue color component
Gray_img=zeros(1,lie);%Greyscale analysis
   for j=1:lie

Gray_img(read_l,j)=round(Source_img(read_l,j,1)*0.3+Source_img(read_l,j,2)*0.6+Source_img
(read_l,j,3)*0.1);

   end
figure(1);
subplot(2,2,1);
plot(x,R_value,"r-.");
title("Component of Red Color");
subplot(2,2,2);
plot(x,G_value,"g-.");
title("Component of Green Color");
subplot(2,2,3);
plot(x,B_value,"b-.");
```

```
title("Component of Blue Color");
subplot(2,2,4);
plot(x,Gray_img,"k-.");
title("Grayscale");
```

Matlab code of cap identification by shape recognition

```
RGB = imread('/Users/liuya/Desktop/cap.png');%Image Path
figure;imshow(RGB);title('Original Image');
% Convert to grayscale image
I = rgb2gray(RGB);
% Set threshold
threshold = graythresh(I);
% Convert to binary image
bw = im2bw(I,threshold);
bw = im2bw(I,0.69);
% Denoising by domain judgment
[m,n] = size(bw);
for i = 2:m-1
  for j = 2:n-1
      %Same upper and lower element judgment
      if(bw(i,j)~=bw(i+1,j) && bw(i,j)~=bw(i-1,j))
         bw(i,j) = 1;
      %Same left and right element judgment
      elseif(bw(i,j)~=bw(i,j+1) && bw(i,j)~=bw(i,j-1))
         bw(i,j) = 1;
      %Same oblique element judgment
      elseif(bw(i,j)~=bw(i+1,j+1) && bw(i,j)~=bw(i-1,j-1))
         bw(i,j) = 1;
      %Same oblique element judgment
      elseif(bw(i,j)~=bw(i-1,j+1) && bw(i,j)~=bw(i+1,j-1))
         bw(i,j) = 1;
      end
  end
end

for i = 1:m
   for j = 1:n
```

```
        bw(i,j) = ~bw(i,j);
    end
end
% Remove small targets
bw = bwareaopen(bw,30);
% Graphic structural element construction, circular
se = strel('disk',8);
% Off operation
bw = imclose(bw,se);
% Filling hole
bw = imfill(bw,'holes');
% Binary image display
figure(1);imshow(bw);title('Binary Image ');
[B,L] = bwboundaries(bw,'noholes');
figure(2);imshow(label2rgb(L,@jet,[.5 .5 .5]));
hold on;
for k = 1:length(B)
boundary = B{k};
% Display white boundary
plot(boundary(:,2),boundary(:,1),'w','LineWidth',2)
end
hold on;
% Determine circular targets
stats = regionprops(L,'Area','Centroid');
% Set the threshold value and calculate the area
threshold = 0.85;
for k = 1:length(B)
    boundary = B{k};
    delta_sq = diff(boundary).^2;
    % Seek perimeter
    perimeter = sum(sqrt(sum(delta_sq,2)));
    % Calculated area
    area = stats(k).Area;
    metric = 4*pi*area/perimeter^2;
    metric_string = sprintf('%2.2f',metric);
    % Determine the shape according to the threshold
    if metric > threshold
        centroid = stats(k).Centroid;
        plot(centroid(1),centroid(2),'ko');
```

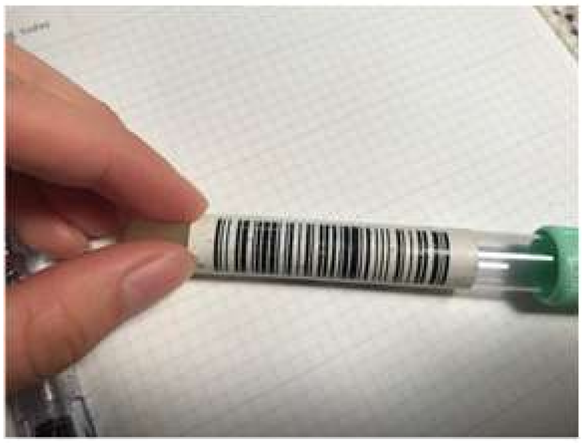
```
    text(centroid(1)-2,centroid(2)-2, 'Circle','Color',...
     'k','FontSize',14,'FontWeight','bold');
  end
    text(boundary(1,2)-10,boundary(1,1)-12, metric_string,'Color',...
     'k','FontSize',14,'FontWeight','bold');
end
title('Shape Recognition')
```

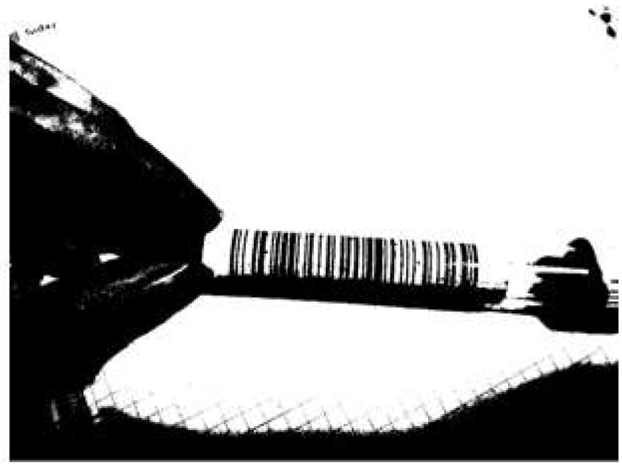
MATLAB code for Barcode Reading

Problems:

The code can not read the bar code in our picture well, as the light and other effects(such as ‘Image pixel’ and ‘sharpness’

The message “Scanning code, please aim at the bar code...” was shown on the screen as the white and black strips are not identified by the code reader





```
clc;

clear all;

close all;

% Start timer to test software performance

[filename,filepath]=uigetfile('*.jpg','Open the files');   %Open the file in your folder

filep=strcat(filepath,filename);

Image=imread(filep);

tic;

   %Read the image and binarize it

level = graythresh(Image);   %Gets the threshold value of image binarization

bw = im2bw(Image,level);   % binarization processing

imshow(bw);
```

```
t = 1;

p = 1;

[m n] = size(bw);  %Preliminary count black and white bar bibliography

q = round(m/2);

    for i=q

        for j=1:n-1

            if bw(i,j)==0&&bw(i,j+1)==1  %The color changes from black to white

                x(t) = j;

                 t = t+1;

            end


        end

    end



for i=q

    for j=1:n-1

        if bw(i,j)==1&&bw(i,j+1)==0  %The color changes from white to black, (white stripes)

            y(p) = j;

             p = p+1;

        end
```

```
    end
end
pin = 0;
 while length(x)~=30||length(y)~=30
      %%Waiting for prompt
        if pin ==0
        display('Scanning code, please aim at the bar code...');
        end
        pin = pin+1;
            % Detect bar code damage, scan line by line
       for pp=q:round(5*m/6)
           t=1;
           p=1;
            if length(x)==30&&length(y)==30
% By judging the number of black and white bars to test the current line whether there is an algorithm error
                break;
            end
                for i=pp
                    for j=1:n-1
                        if bw(i,j)==0&&bw(i,j+1)==1
                            x(t) = j;
                            t = t+1;
```

```
            end
        end
    end
    for i=pp
        for j=1:n-1
            if bw(i,j)==1&&bw(i,j+1)==0
                y(p) = j;
                p = p+1;
            end

        end
    end
  end
end
if length(x)~=30||length(y)~=30
    display('WRONG!');
    return;
end
if i~=round(m/2)
    display(' The bar code has been damaged, but it can still be scanned normally');
end
% Calculate the width of each bar - space, and subtract the values recorded in the xy array
```

```
 for ii=1:30
   if ii==1
       d(ii)=x(ii)-y(ii);      %Calculate the width of the first bar
       d(ii+1)=y(ii+1)-x(ii);  % Calculate the width of the first blank
   end
    if ii>1
        if ii>1&&ii<30
       d(2*ii) = y(ii+1)-x(ii); % Calculate the width of the 2nd ~ 29th empty space respectively
       d(2*ii-1)=x(ii)-y(ii);   % The width of article 2~29 is calculated respectively
    elseif ii==30
        d(ii*2-1)=x(ii)-y(ii);   % The width of the thirtieth article shall be calculated separately
        end
    end
end
  display([' decode',num2str(gen)]);
 % End of test timing
toc;
```

8. Reference

[1] Hongjian Qi. Introduction to Image: MATLAB Image Recognition[EB/OL].https://blog.csdn.net/qhj_miracle/article/details/80958673,2018-07-08.

[2] https://blog.csdn.net/cancumt/article/details/46759789

[3] https://www.motoman.com/search?searchtext=motomini&searchmode=anyword

[4] https://www.falconautoonline.com/sortation-solutions/

[5] https://youtu.be/CVOnt-GJPL4

[6] https://www.youtube.com/watch?v=oOIqtEIpWPA

[7] https://www.youtube.com/watch?v=F-_WSaTn7FA

[8] Xianshang Youling. OpenCV color recognition[EB/OL].https://blog.csdn.net/chen134225/article/details/80812591

[9]https://blog.csdn.net/qq_37269094/article/details/82848589

[10]https://blog.51cto.com/gloomyfish/2104134?lb

[11]Ruikun Liang. OpenCV Shape recognition[EB/OL].https://www.jianshu.com/p/2731f42882f4