GROUP ASSIGNMENT

CT036-3-3-IPPR

IMAGE PROCESSING, COMPUTER VISION AND PATTERN RECOGNITION

Lecture:	**Dr. V. SIVAKUMAR**
Group Member:	**Gan Qian Chao TP055444**
	Sia De Long TP060810
	Tai Yi Yong TP059588
Intake Code:	**APU3F2309CS(IS)/APD3F2209CS(IS)**
Hand out date:	**28th September 2022**
Hand in date:	**23rd December 2022**

Table of Contents

Contribution Matrix

Name / Task	Gan Qian Chao (TP055444)	Tai Yi Yong (TP059588)	Sia De Long (TP060810)
Project Report			
Acknowledgement	- Acknowledge		
Abstract			- Abstract
Introduction		- Introduction	
- Description and justification of algorithm/techniques - Experimental result	- Latex and cloth glove identifier - Dirty and stain defect detector - Finger not enough defect detector - Hole(palm) defect detector	- tearing on cloth gloves detector - open seam on cloth glove detector - stitching run off on cloth glove detector	- Plastic Glove Detection - Touching Defect Detection - Double Dipping Defect Detection - Thinner Defect Detection
- Description and justification of obtained result - Critical comments, analysis and future work direction	- Latex glove defect detection	- cloth glove defect detection	- Plastic Glove Defect Detection
Conclusion	- Conclusion		
Coding Implementation			
GUI design	-GUI coding		
Latex glove detection	- Latex glove detection coding		
Cloth glove detection		-Cloth glove detection coding	
Plastic glove detection			-Plastic glove detection coding
Signature	GAN	Tai	Sia

1 Acknowledgements

In this IPPR assignment, the researchers would like to show his great appreciation towards their IPPR lecturer, Dr. V. Sivakumar for providing some useful guides on the researches starting from assignment documentation, dataset collection and code development. The researchers would like to express his gratitude towards the group members who willing to take time to provide support from the documentation and code development. With their help, the researchers can cooperate with each other and always be motivated on delivering the assignment on time. Last but not least, the researchers appreciate the helping hand provided by their friends who had provide emotion and technical support to the researchers. They had helped the researchers on reviewing their opinions and ideas towards the proposed system.

2 Abstract

In this report, the researcher will be working in a group to develop a solution to detect the selected type of glove defect by applying image processing and computer vision techniques. Firstly, the tasks for identifying type of glove and defects including gathering the dataset for respective task are distributed evenly to the researchers. Then the researchers will describe the proposed algorithm used to achieve the assignment objectives with given justification, with that, the snippet of source code will be provided to showcase the application of the proposed algorithm on MATLAB platform. Lastly, the researchers will implement the solution to a graphical user interface and evaluate the system on the accuracy for the defect detection while providing critical analysis for cases that have fail or false detection.

3 Introduction

Image processing can be known as a method that convert an image into a digital aspect and apply functions on it in order to get an outcome which is extracting a certain part of the image or enhancing it visually. Image processing is one of the fastest growing technologies in this era, this technology is widely used by several business sectors, and graphical design forms the main core of the research in the engineering field and computer science industries as well (Great Learning Team, 2022).

In this project, the researchers are tasked to create a system that is able to detect gloves, and the system will be able to recognize the defects on it and segment it. The researchers shall conduct research on what techniques shall be used on the project, and provide justification for the choices made during the research, including the libraries and data chosen for the research. In short, the researchers of this project are tasked to create, design, and implement a gloves defect detection system, provide a throughout test on the system and evaluate its accuracy and precision.

4 Description and Justification of Algorithm and Techniques

4.1 Latex glove detection

The latex glove can be identified by color threshold the glove image. Since the latex glove is in blue color, the blue color threshold is done by transform the image into HSV form with hue value of 201 degree to 240 degree. (WorkWithColor.com, 2013) After calculating the percentage of blue color, it will be defined as latex glove if the percentage of blue pixel is high.

4.2 Algorithm for dirty and stain defect

Figure 4.1: Dirty and stain (Cortex Robotics, 2022)

For detecting dirty and stain on the glove shown as Figure 4.1, the glove area of latex glove will be extracted using Otsu method to perform dynamic threshold segmentation towards the glove image in different glove orientation shown as Figure 4.2 after transform to HSV form. (Wang, 2020)

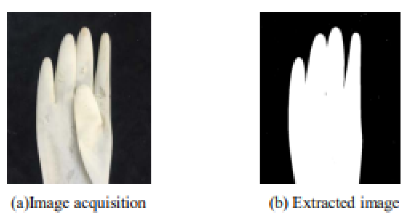


Figure 4.2: Glove area extraction (Wang, 2020)

The median filter is applied to remove noises in image for an accurate extraction. The reduction process will be done to remove unrelated region besides the defect. The closing operation is performed to smoothen the defects extracted shown as Figure 4.3. The bounding box is created on the defect area with a rectangular shape for identify the structural outlines of the current defect shown as Figure 4.4. (Wang, 2020)

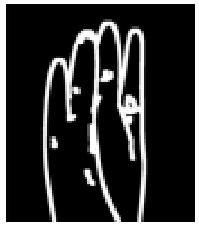


Figure 4.3: Defect area extraction using closing operation (Wang, 2020)

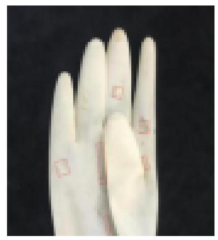


Figure 4.4: Bounding box of dirty and stain defect (Wang, 2020)

The defect area extracted is then extract hue value to identify the significant change of color. Once the area of defect has high percentage of significant different in hue value, it will be identified as dirt and stain defect shown as Figure 4.5.

Figure 4.5: Dirt Defect Identification

4.3 Algorithm for Finger Not Enough Defect

Figure 4.6: Finger not enough

For the detection of finger not enough defect as Figure 4.6, the image is transformed to YCbCr form to enable the skin color of the hand can extracted through Otsu Method. The median filter is done to filter out pixels which is not related to the area of glove for an accurate result. (Perimal, Basah, Safar, & Yazid, 2018)

After that, the center of the palm and wrist region need to be removed since they are not involved in the finger detection. (Perimal, Basah, Safar, & Yazid, 2018) With these concepts, opening operation with large enough kernel so that the fingers can be chopped off successfully. (MathWorks, 2019) After the palm region is extracted, the subtraction operation is done to remove the palm region to obtain fingers area shown as Figure 4.7. If the number of fingers is lesser than 5, it means that there is finger not enough defect in the glove.

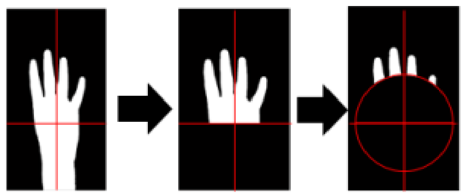


Figure 4.7: Finger counting (Perimal, Basah, Safar, & Yazid, 2018)

4.4 Algorithm for Hole Defect (Latex glove palm)

Figure 4.8: Hole (Palm) (Cortex Robotics, 2022)

The method of finding hole defect as Figure 4.8 on palm which is only can be found in latex glove is similar as dirty and stain defect detection as the steps like the extraction of glove with the Otsu method, reduction to remove non-defect region in glove area is still the same. The difference of recognizing hole is to do the skin color detection in the defect region which the defect region extracted is transform to YCbCr form which can extract out the skin color in the region. If the skin region has higher percentage in the defect region, it will be declared as hole defect.

4.5 Plastic Glove Detection

Since the plastic glove have the color of black, HSV layers should not be very helpful on detecting the glove in the image as glove and background will be mixing together. Hence the convert of image from RGB to grayscale method is chosen for highlighting the glove area pixels. The grayscale image should be studied by plotting the number of shade level for pixel value of 0 to 255 on a histogram (Kaler, 2016), so that the range of value for the black glove can be determined based on the pixel's distribution. With the base binary image produced, further image processing such as filling the hole and connecting all the pixels together which are median filter (Maheswari & Radha, 2010) and dilation (Sebastián, et al., 2018) will be used to make sure that the image should have just only one white area representing the glove area for the usage of other detection.

4.6 Algorithm for Touching Defect Detection

With the glove area acquired by using the plastic glove detection algorithm, the touching defect is next to be detected. The hole of touching defect is detected from converting the image to HSV layer because the pixels of the holes will be more significantly on the saturation layer as the whole glove is in black color. The defect pixels are acquired by using a defined saturation value and gone through noise removal technique including median filter (Maheswari & Radha, 2010) and dilation (Sebastián, et al., 2018) to limiting the number of separate defect area with defined maximum number for better detection performance. However, this process should also capture the wrist area pixels which can be removed if combining defect area and glove area image together. Lastly, all the separated defect area can have a label itself and will only be identified as touching defect if the area of it is within the defined range.

4.7 Algorithm for Double Dipping Defect Detection

With the glove area acquired by using the plastic glove detection algorithm, the center of the glove area should be determined with an exact coordinate. So that it can be used to form a black circle surrounding the coordinate with a given radius size determined by the glove area size. The coordinate of the center must be flexible that able to adapt to the change of direction of glove image so the circle pixel is removed correctly, while the radius size is also flexible to the glove size to ensure that fingers and wrist area is separated completely. Noise removal are applied again after the pixel removal to remove any unrelated pixel and for better accuracy. Lastly, the number of separated white pixel area is the factor that determine double dipping defect where it should have at least 6 area which are 5 fingers and 1 wrist area. This condition is defined because double dipping should cause at least two fingers linked together which making counting fingers can determine double dipping situation.

4.8 Algorithm for Thinner Defect Detection

Thinner defect is detected using the algorithm for touching defect detection as the step of detection is completely same, so both of the algorithm will be implemented together as one. However, the factor that differ between touching and thinner is that the area of the defect area where thinner defect should have a bigger size of area than touching defect.

4.9 Cloth gloves detection

Cloth gloves detection can be done by using entropy filter. Entropy filter measures the number of bits required to encode the image (Julio et al, 2019). The reason that entropy filtering is used is because that cloth gloves has more details than latex and rubber gloves when looked closely. The details include the seam lines, cloth texture and more.

4.10 Algorithm and techniques for tearing detection on cloth gloves

The main cause of tearing on cloth gloves are sharp cuts, worn out and more. it is similar, but different to holes on gloves as holes on gloves are made during manufacturing, and tearing are done when the user is using or by accident. Color image segmentation can be used on the detection of tearing on cloth gloves. Color image segmentation can be defined as the separation of image based on the features and Color of the image pixels assuming that monotone Color in the image relate to the different clusters and objects in the image. The result of segmentation depends on the used Color workspace, hence different projects or section will use different Color workspace to solve the problem (Ding, 2014).

For the technique used for tearing detection, the RGB image is first converted to hue, saturation, and value (HSV), then the image is converted to HSV array, and the arrays are used for thresholding. The outcome image will then undergo a series of noise removing operation, and the final outcome will be displayed as follows.

4.11 Algorithm for open seam detection

For open seam on cloth gloves, it is usually caused by the improper seaming during the manufacturing process, and it can be visually identified as from the outside, it is viewed as a hole on the suture. Hence, the same process on detecting tearing on gloves can be applied on the seaming detection, but different threshold value, noise removing methods and image array is used as the defect is smaller compared to tearing.

4.12 Algorithm for open stitching detection

Open stitches are mainly caused by the improper stitching process during the manufacturing phase. These types of defects can be identified if there are threads coming out at the end of the gloves. For the open stitching detection in this project, the researcher has used Color image segmentation to differentiate the gloves, the thread and the background of the image, the segmented image is then converted to binary image, then the defect is then segmented out using the AND operator.

5 Experimental Results

5.1 Latex glove detection

```
% Load the image and convert it to the HSV color space
I = imread('D:/Degree 3 - Image Processing/dataset/dirty and stain/side(left)/side_left_dirty(15).jpeg');
subplot(1,3,1), imshow(I), title('Original Image')
Ihsv = rgb2hsv(I);
subplot(1,3,2), imshow(Ihsv), title('HSV Image')

% Extract the hue channel
hue = Ihsv(:,:,1);
saturation = Ihsv(:,:,2);
value = Ihsv(:,:,3);
% Threshold the hue channel to identify pixels with a hue value in the range of blue colors
bluemask = hue >= 0.55 & hue <= 0.67 & saturation >=0.2 & value >=0.2;
subplot(1,3,3), imshow(bluemask),title ('Blue region in image')
% Count the number of blue pixels
numblue = nnz(bluemask);

% Calculate the percentage of the image that is blue
percentblue = numblue / numel(bluemask) * 100;

% Display the result
disp(['Percentage of blue: ' num2str(percentblue) '%'])

%Condition
if (percentblue>= 5)
    disp('This is a latex glove')
else
    disp('This is not a latex glove')
end
```

Figure 5.1: Latex glove detection

The latex glove detection is done with coding shown as Figure 5.1.

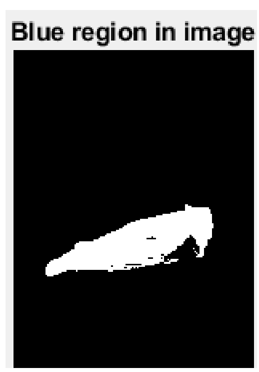


Figure 5.2: Blue region mask

The hue value of the image is threshold to form blue mask shown as Figure 5.2 which blue percentage exceed 5% is declared as latex glove

5.2 Dirt and stain defect detection + Hole (on palm) defect detection

```
% read the image
im = imread('D:/Degree 3 - Image Processing/dataset/dirty and stain/fingertip/bad_2/fingertip_dirty(1).jpeg');
figure,imshow(im),title('Original Image');
% Transform image to HSV
Ihsv = rgb2hsv(im);
%Extract saturation channel
I1= Ihsv(:,:,2);
%Thresh away low saturation pixel
thresholded = I1 > 0.4; %% Threshold to isolate lungs
thresholded = bwareaopen(thresholded,100);  % remove too small pixels
I2=thresholded.*I1; %apply threshold
%Otsu threshold for getting glove region
BW = graythresh(I2);
I3 = imbinarize(I2,BW);
% remove noise of glove
I3 = imfill(I3,'hole');
I3=medfilt2(I3,[5,5]);
% erode the glove region
se = strel("disk",5);
I3 = imerode(I3,se);
% ensure there is only one white region in the binary image
I3 = bwareafilt(I3,1);
%mask out glove region
I4 = I2.*I3;
```

Figure 5.3: Image saturation thresholding

For detection for dirty and stain and hole defects, it can be done with coding as Figure 5.3.

Figure 5.4: Saturation threshold image

The saturation channel is extracted and apply threshold to produce result shown as Figure 5.4.

Figure 5.5: Glove region

The Otsu threshold method is used to apply graythresh for obtain glove region shown as Figure 5.5.

```
%Single threshold segmentation
s = size(I4);
segment = zeros(s(1),s(2));
for i = 1:s(1)
    for j = 1:s(2)
        %side 0.68, fingertip palm 0.5
        if(I4(i,j)>=0.5)
            segment(i,j) = 0;
        else
            segment(i,j)=255;
        end
    end
end
%show defects
I4 = I3-(~segment);
%smoothen defects using closing operation
I4 = imclose(I4,se);
```

Figure 5.6: Image segmentation

Figure 5.7: Possible defects in glove

The segmentation of image is done with the coding shown as Figure 5.6. The possible defects are shown in the Figure 5.7.

```
%Blob up possible defect region
[Ilabel, num] = bwlabel(I4);
disp("Total possible defect="+num);
Iprops = regionprops(Ilabel,'BoundingBox','Area');
Ibox = [Iprops.BoundingBox];
IArea = [Iprops.Area];
Ibox = reshape(Ibox, [4 num]);
newBox = [];
%Remove too big or too small defect (for dirty and stain defect + hole)
for i = 1: length(IArea)
    disp("i="+i);
    disp("Area="+IArea(i));
    if ((IArea(i)>300)&&(IArea(i)<6000))
        newBox(:,end+1) = Ibox(:,i);
    end
end
[rows,columns] = size(newBox);
```

Figure 5.8: Bounding box creating

The defect size filtering is performed in Figure 5.8.

```
defect_name = [];
not_defect = [];
for cnt = 1:columns
    x = newBox(1,cnt);
    y = newBox(2,cnt);
    w = newBox(3,cnt);
    h = newBox(4,cnt);
    size(segment);
    defect_mask = imcrop(segment,[x, y, w, h]); % Img Cutted or extrated from bounding box
    defect_region = imcrop(im,[x, y, w, h]);
%     Resize the mask image to have the same size as the input image
    defect_mask = imresize(defect_mask, [size(defect_region,1) size(defect_region,2)]);
    defect_mask = cast(defect_mask,'uint8');

%     Replicate the mask image across the additional channels of the input image, if necessary
    defect_mask = repmat(defect_mask, [1,1,3]);
    defect_region(~defect_mask)=255;
    figure('Name',sprintf('Defect %d',cnt)), subplot(2,2,1),imshow(defect_region), title('Defect detected');
    subplot(2,2,2),imshow(defect_mask), title('Segmented mask');
```

Figure 5.9: Defect area extraction

The coding shown as Figure 5.9 is to mask out defect in bounding box.

```
defect_hsv = rgb2hsv(defect_region);
% Extract the hue channel
hue = defect_hsv(:,:,1);
saturation = defect_hsv(:,:,2);
value = defect_hsv(:,:,3);
% Threshold the hue channel to identify pixels with a hue value in the range of blue colors
bluemask = hue >= 0.55 & hue <= 0.67 & saturation >=0.2 & value >=0.4;
subplot(2,2,3), imshow(bluemask),title ('Blue region in defect region');
% Count the number of blue pixels
numblue = nnz(bluemask);
% Calculate the percentage of the image that is blue
percentblue = numblue / numel(bluemask) ;
numberwhite = nnz(defect_mask);
percentwhite = numberwhite / numel(defect_mask);
percent = percentblue/percentwhite * 100;

% Display the result
disp(['Percentage of blue: ' num2str(percent) '%'])
```

Figure 5.10: Blue color detection

The blue color detection is done in coding shown as Figure 5.10.

```
%Condition
if (percent< 50)
    % Convert the image to the YCbcr color space
    ycbcrImg = rgb2ycbcr(defect_region);

    % Define the lower and upper bounds of the skin color cluster
    lower = [80, 110, 129];
    upper = [210, 145, 165];

    % Threshold the HSV image to create a binary mask
    mask = ycbcrImg(:,:,1) >= lower(1) & ycbcrImg(:,:,1) <= upper(1) & ...
           ycbcrImg(:,:,2) >= lower(2) & ycbcrImg(:,:,2) <= upper(2) & ...
           ycbcrImg(:,:,3) >= lower(3) & ycbcrImg(:,:,3) <= upper(3);
    subplot(2,2,4), imshow(mask),title ('Skin region in defect region');
    % Use the mask to segment the image into regions
    regions = regionprops(mask, 'Area');
    % Compute the total area of the image
    totalArea = size(defect_region, 1) * size(defect_region, 2);
    % Compute the percentage of skin-colored regions in the image
    skinCoverage = 100 * sum([regions.Area]) / totalArea;

    % Print the percentage of skin-colored regions
    fprintf('Skin coverage: %.2f%%\n', skinCoverage);

    if (skinCoverage > 1 )
        %Hole defect
        defect_name = cat(2, defect_name,{'Hole'});
    else
        %Dirty and stain
        defect_name = cat(2, defect_name,{'Dirty and Stain'});
    end


   else
       %fail defect
       not_defect = cat(2, not_defect,cnt);
       not_defect = sort(not_defect, 'descend');
   end
end
```

```
%Remove fail defect
for i = 1: length(not_defect)
    newBox(:,not_defect(i)) = [];
end
```

Figure 5.11: Hole/dirty+stain defects identification

The skin coverage test is done for the identification process shown as 5.11. The hole defect is identified with more than 1% of skin coverage in defect. Otherwise, it will be identified as dirty and stain.

```
figure, imshow(im), title('Image with defect');
%Draw bounding box
% Set the font size and color for the label
fontSize = 8;
fontColor = 'red';
%Start draw with rectangle
hold on;
for cnt = 1:size(newBox,2)
    rectangle('position', newBox(:,cnt),'EdgeColor','r');
    text(newBox(1,cnt), newBox(2,cnt)-25, defect_name(cnt), ...
     'FontSize', fontSize, 'Color', fontColor);
end
hold off;
```

Figure 5.12: Drawing bounding box for defects

Figure 5.13: Image with defect bounding box

With the experimental coding in Figure 5.12. the image with defect bounding box is produced shown as Figure 5.13.

5.3 Finger not enough defect detection

```
% read the image
img = imread('D:/Degree 3 - Image Processing/dataset/finger not enough/palm/palm_finger_not_enough(18).jpeg');
subplot(2,2,1), imshow(img),title('Original image');
%convert to hsv
img_hsv = rgb2hsv(img);
%Extract saturation channel
img_hsv = img_hsv(:,:,2);

%Glove region extraction
bw_img = im2bw(img_hsv,0.45);
%smoothen boundaries and remove noises
se= strel('disk',5);
bw_img = imclose(bw_img,se);
bw_img = bwareaopen(bw_img,100000);
bw_img = imfill(bw_img,'holes');
subplot(2,2,2), imshow(bw_img),title('Glove region');

%Palm region extraction
open = strel('disk',50);
palm_img = imopen(bw_img, open);
subplot(2,2,3), imshow(palm_img),title('Palm region');

%Reduction of palm area in glove region
finger = bw_img-palm_img;
finger = im2bw(finger);
%Accept size with greater or equal to 5000 pixels (finger size)
sizeThreshold = 5000;
finger = bwpropfilt(finger, 'Area', [sizeThreshold Inf]);
subplot(2,2,4),imshow(finger), title('Fingers extracted');

%Record number of fingers
[label,num]  = bwlabel(finger);
fprintf('Number of fingers: %d',num);

if (num ~= 5)
    disp('-> Finger not enough');
else
    disp('-> Not finger not enough');
end
```

Figure 5.14: Finger counter

The finger counter is implemented with the experimental coding shown as Figure 5.14.

Figure 5.15: Glove region extraction

The glove region is extracted shown as Figure 5.15.

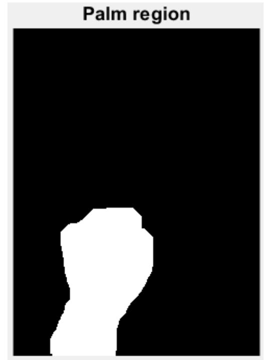


Figure 5.16: Palm region extraction

The palm region is extracted with opening operation shown as Figure 5.16.

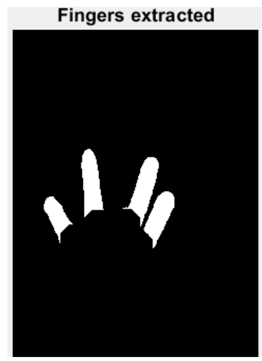


Figure 5.17: Finger extraction result

Result of fingers extraction is shown as Figure 5.17.

```
% Load image
img = imread('D:/Degree 3 - Image Processing/dataset/finger not enough/palm/palm_finger_not_enough(2).jpeg');
figure('Name','Original'),imshow(img),title('Original Image');
% Transform image to HSV
img_hsv = rgb2hsv(img);
%Extract saturation channel
img_hsv = img_hsv(:,:,2);

%Thresh away low saturation pixel
%if palm 0,3, fingertip 0.4, side 0.41
bw_img = im2bw(img_hsv,0.3); %Threshold to isolate glove + finger region
bw_img = bwareaopen(bw_img,300);  % remove too small pixels
orientation = "palm";
%use for side orientation
if(orientation=="side")
    dilate = strel('disk',3);
    bw_img = imdilate(bw_img,dilate);
end
%remove noise
bw_img = medfilt2(bw_img,[5 5]);
%Otsu threshold method for glove region
BW = graythresh(img_hsv);
binarized = imbinarize(img_hsv,BW);
%Remove noise of glove region
binarized = imfill(binarized,'hole');
binarized = bwareaopen(binarized,300);
```

Figure 5.18: Glove and object extraction

The coding for finger not enough defect extraction is shown as Figure 5.18.

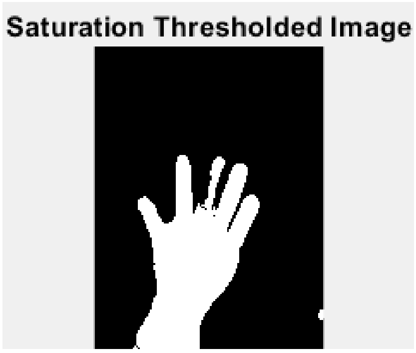


Figure 5.19: Saturation threshold

After the saturation threshold, the hand region is reviewed as Figure 5.19.

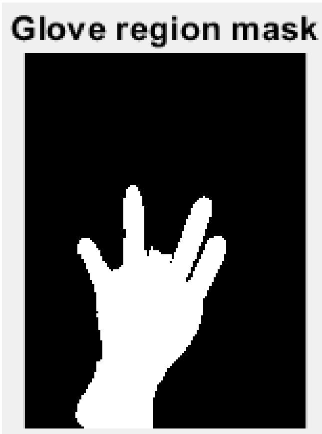


Figure 5.20: Glove region mask

The glove region mask is shown as Figure 5.20.

```
%Obtaining defect regions
%defect_in is defect within glove region
%defect_out is defect out of glove region
defect_in = bw_img-binarized;
defect_out = binarized-bw_img;
defect = defect_in|defect_out;
defect = imfill(defect,'hole');
%filter away noises and small white regions
defect = medfilt2(defect,[7 7]);
defect = bwareaopen(defect,1000);
%smoothen defect region
se = strel('disk',5);
defect = imclose(defect,se);
```

Figure 5.21: Reduction process for defect extraction

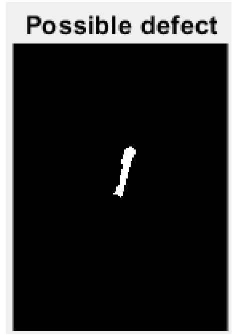


Figure 5.22: Possible defect detected

The missing fingers defect is extracted from the reduction shown in Figure 5.22 with coding in Figure 5.21

```
%Label defect region
[Ilabel, num] = bwlabel(defect);
disp("Number of possible defect=" +num);
Iprops = regionprops(Ilabel,'BoundingBox','Area');
Ibox = [Iprops.BoundingBox];
IArea = [Iprops.Area];
Ibox = reshape(Ibox, [4 num]);
newBox = [];
%remove too small or too big region (for finger not enough)
for i = 1: length(IArea)
    disp("i="+i);
    disp("Area="+IArea(i));
    if ((IArea(i)>5000)&&(IArea(i)<40000))
        newBox(:,end+1) = Ibox(:,i);
    end
end
[rows,columns] = size(newBox);
figure('Name','Pre-processing'),
subplot(2,2,1),imshow(img_hsv),title('HSV Saturation Image');
subplot(2,2,2),imshow(bw_img),title('Saturation Thresholded Image');
subplot(2,2,3),imshow(binarized), title('Glove region mask');
subplot(2,2,4),imshow(defect), title('Possible defect');
```

Figure 5.23: Defect size filtering

The coding for defect size filtering is done shown as Figure 5.23.

```
%Check the skin coverage of all defects detected
not_defect=[];
for cnt = 1:columns
    x = newBox(1,cnt);
    y = newBox(2,cnt);
    w = newBox(3,cnt);
    h = newBox(4,cnt);
    size(img);
    defect_region = imcrop(img,[x, y, w, h]); % Img Cutted or extrated from bounding box
    defect_mask = imcrop(defect,[x, y, w, h]);
    figure('Name',sprintf('Defect %d',cnt)), subplot(1,3,1),imshow(defect_mask), title('Defect mask');
    %     Resize the mask image to have the same size as the input image
    defect_mask = imresize(defect_mask, [size(defect_region,1) size(defect_region,2)]);
    defect_mask = cast(defect_mask,'uint8');

%     Replicate the mask image across the additional channels of the input image, if necessary
    defect_mask = repmat(defect_mask, [1,1,3]);
%     defect = bsxfun(@times, defect_region, cast(defect_mask, 'like', defect_region));
    defect_region(~defect_mask)=255;

    subplot(1,3,2),imshow(defect_region), title('Defect region');
```

Figure 5.24: Defect region extraction

The defect area is obtained from coding shown as Figure 5.24.

```
      % Convert the image to the HSV color space
      ycbcrImg = rgb2ycbcr(defect_region);

      % Define the lower and upper bounds of the skin color cluster
      lower = [0, 110, 129];
      upper = [142, 131, 165];

      % Threshold the HSV image to create a binary mask
      mask = ycbcrImg(:,:,1) >= lower(1) & ycbcrImg(:,:,1) <= upper(1) & ...
             ycbcrImg(:,:,2) >= lower(2) & ycbcrImg(:,:,2) <= upper(2) & ...
             ycbcrImg(:,:,3) >= lower(3) & ycbcrImg(:,:,3) <= upper(3);
      subplot(1,3,3), imshow(mask),title ('Skin region in defect region');
      % Use the mask to segment the image into regions
      regions = regionprops(mask, 'Area');
      % Compute the total area of the image
      totalArea = size(defect_region, 1) * size(defect_region, 2);
      % Compute the percentage of skin-colored regions in the image
      skinCoverage = 100 * sum([regions.Area]) / totalArea;

      % Print the percentage of skin-colored regions
      fprintf('Skin coverage: %.2f%%\n', skinCoverage);

     %Condition
     if (skinCoverage <20)
         not_defect = cat(2, not_defect,cnt);
         not_defect = sort(not_defect, 'descend');
     end

 end

%Remove fail defect
for i = 1: length(not_defect)
    newBox(:,not_defect(i)) = [];
end
```

Figure 5.25: Skin coverage test

The skin coverage test is done with coding in Figure 5.25. If the skin coverage is more than 20%, the defect will be approved as a passed defect.

```
figure, imshow(img), title('Image with defect');
%Draw bounding box
% Set the font size and color for the label
fontSize = 8;
fontColor = 'red';
hold on;
for cnt = 1:size(newBox,2)
    rectangle('position', newBox(:,cnt),'EdgeColor','r');
    text(newBox(1,cnt), newBox(2,cnt)-25, 'Finger not enough', ...
     'FontSize', fontSize, 'Color', fontColor);
end
hold off;
```

Figure 5.26: Bounding box drawing

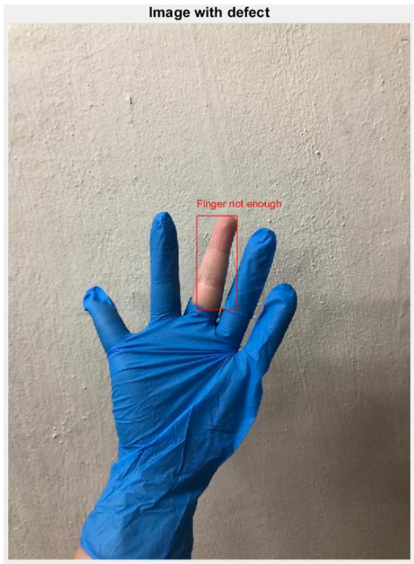


Figure 5.27: Image with defect

For drawing the bounding boxes, the coding in Figure 5.26 is used to produce final result with defect shown as Figure 5.27.

5.4 Plastic Glove Detection

```
% Read original image
img = imread("Palm_Touching(2).jpg");
subplot(1,3,1), imshow(img), title("Original Image")

% Convert image to gray scale
grayimg = rgb2gray(img);
subplot(1,3,2), imshow(grayimg), title("Gray Scale Image")


% Thresholding glove area
glovearea = grayimg < 75;

% remove noise from reflection and only take one white area of the image
glovearea = imfill(glovearea,"holes");
glovearea = bwareafilt(glovearea,1);
subplot(1,3,3), imshow(glovearea), title("Glove Area")

% Calculate black pixel percentage and determine whether it is a plastic
% glove
blackpercentage = nnz(glovearea) / numel(glovearea) * 100;
if (blackpercentage>=10)
    disp("This is a plastic glove")
else
    disp("This is not a plastic glove")
end
```

Figure 5.28: Plastic Glove Detection

The plastic glove detection is done using the code shown as above.

Figure 5.29: Plastic Glove Detection

The gray scale and the binary image for the glove is shown as above and it will be identified as plastic glove if it exceeds 10% of given range of shade level.

5.5 Touching and Thinner Defect Detection

```
% Read original image
img = imread("Palm_Touching(6).jpg");

%% Find Glove Area
% Convert image to gray scale
grayimg = rgb2gray(img);
% Thresholding glove area
glovearea = grayimg < 75;
% remove noise from reflection and only take one white area of the image
glovearea = imfill(glovearea,"holes");
glovearea = bwareafilt(glovearea,1);
subplot(1,3,1), imshow(glovearea), title("Glove Area")
```

Figure 5.30: Clean Plastic Glove Area Detection

The clean plastic glove area detection is done using the code shown as above.

Figure 5.31: Clean Plastic Glove Area Result

The clean plastic glove result is shown as above.

```
%% Find Defect
% Convert image to HSV
hsvimg = rgb2hsv(img);
defect = hsvimg(:,:,2) < 0.3;
for c = 1:10
    defect = medfilt2(defect,[5,5]);
end
defect = bwareaopen(defect,100);
subplot(1,3,2), imshow(defect), title("Defect Area")
```

Figure 5.32: Defect Area Detection

The defect area detection is done using the code above.

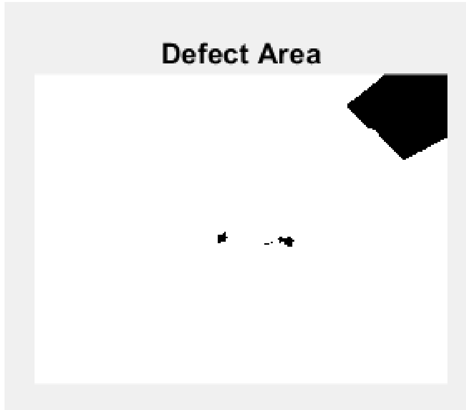


Figure 5.33: Defect Area Result

The defect area is found by taking pixels that within the given saturation value range, then median filter and small pixel removal are performed to remove most of the unrelated pixels.

```
%% Combine Glove Area and Defect Area
% combine two area to remove hand area
binarydefect = glovearea.*defect;
binarydefect = imcomplement(binarydefect);
subplot(1,3,3), imshow(binarydefect), title("Hand and Defect Area")
```

Figure 5.34: Combine Glove and Defect Area

The combination of glove area and defect area is done using the code above.

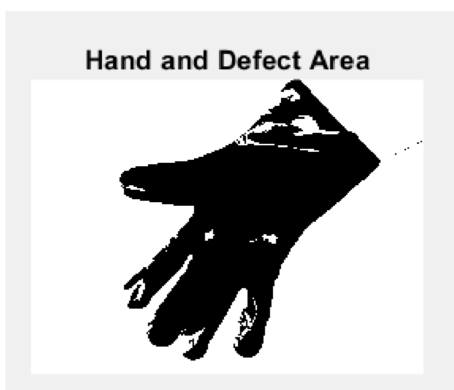


Figure 5.35: Combine Glove and Defect Area Result

The defect area pixels value will be reversed and combine with the glove area which will remove the hand area and show a clean glove area with defect on it for better insight.

```
%% Create box for defect
% find area for all the possible defects
[label, numdefect] = bwlabel(binarydefect);
props = regionprops(label, "BoundingBox", "Area");
box = [props.BoundingBox];
area = [props.Area];
box = reshape(box,[4 numdefect]);
defectBox = [];
defectName = [];

% Remove area that is too small or too big to be a defect
% Identify the type of defect after removal
for c = 1:length(area)
    if((area(c)>6500)&&(area(c)<20000))
        defectName = [defectName "Touching"]
        defectBox(:,end+1) = box(:,c);
    else
        if((area(c)>30000)&&(area(c)<100000))
            defectName = [defectName "Thinner"];
            defectBox(:,end+1) = box(:,c);
        end
    end
end
```

Figure 5.36: Bounding Boxes Creation

The bounding boxes creation and label are done using the code above. The defect will be named as “Touching” or “Thinner” based on the area of it.

```
%% Combine Box and Original Image
namec = 1;

figure, imshow(img), title("Original Image with Bounded Defects")
for c = 1:size(defectBox,2)
    rectangle("Position", defectBox(:,c), "EdgeColor", "r");
    text(defectBox(1,c)-100, defectBox(2,c)-50, defectName(namec), "FontSize", 12, "Color", "r");
    namec = namec + 1;
end
```

Figure 5.37: Bounding Boxes Drawing

The drawing and labeling of bounding boxes on the original image are done using the code above.

Figure 5.38: Touching and Thinner Detection Result

The result of touching and thinner defects detection is shown as above.

5.6 Double Dipping Detection

```
% Read original image
img = imread("Palm_DoubleDipping(15).jpg");

%% Find Glove Area
% Convert image to gray scale
grayimg = rgb2gray(img);
% Thresholding glove area
glovearea = grayimg < 75;
% remove noise from reflection and only take one white area of the image
for c = 1:10
    glovearea = medfilt2(glovearea,[5,5]);
end
glovearea = imfill(glovearea,"holes");
glovearea = bwareafilt(glovearea,1);
subplot(1,3,1), imshow(glovearea), title("Glove Area")
```

Figure 5.39: Clean Plastic Glove Area Detection

The clean plastic glove area detection is done using the code shown as above.

Figure 5.40: Clean Plastic Glove Area Result

The clean plastic glove result is shown as above.

```
%% Find the center and remove the right part of the glove area
glovearea2 = glovearea;
seg = regionprops(glovearea2,'Centroid');
centroid = cat(1, seg.Centroid);
subplot(1,3,2), imshow(glovearea2), title('Centroid of The Glove');
hold(imgca,'on')
plot(imgca,centroid(1), centroid(2), 'r*')
hold(imgca,'off')

x = centroid(1);
y = centroid(2);
```

Figure 5.41: Find the Center of Glove

The finding of center of glove is shown as above.

Figure 5.42: Center of Glove Result

The result of finding the center of glove is shown as above, the coordinate is found by identifying the only white area on the image and plot the center using the x and y.

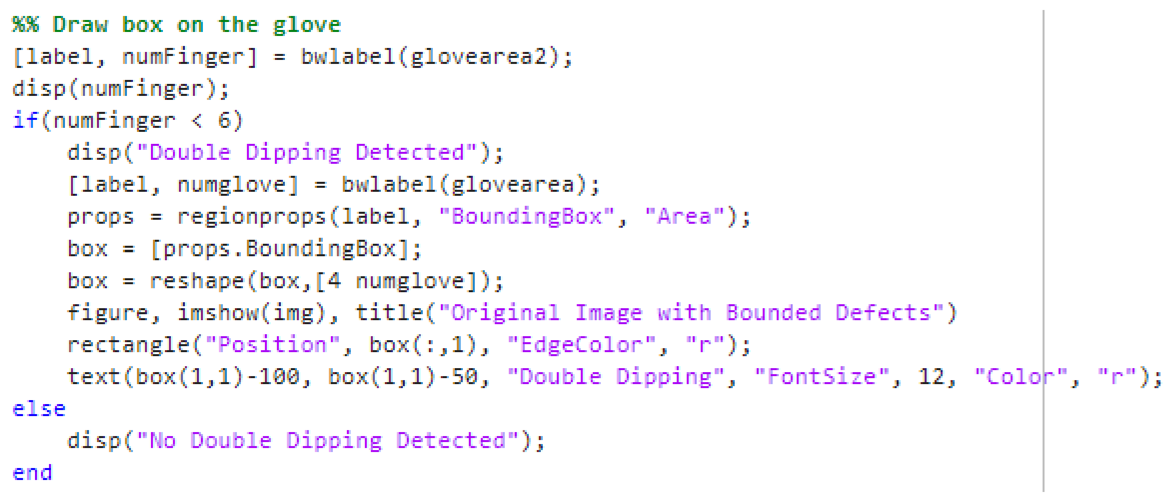
```
%% Remove the glove area with a circle
[col,row] = size(glovearea2);

glovesize = sum(glovearea2(:));
disp(glovesize);
if (glovesize > 3700000)
    r = 1100;
else
    if (glovesize > 3000000)
        r = 700;
    else
        r = 600;
    end
end
for i=1:col
    for j=1:row
        if ((i-y)^2)+((j-x)^2)<(r^2)
            glovearea2(i,j) = 0;
        end
    end
end
for c = 1:10
    glovearea2 = medfilt2(glovearea2,[5,5]);
end

subplot(1,3,3), imshow(glovearea2), title('Circle of Glove Area Removed');
```

Figure 5.43: Circle Area Removal

The circle area removal on glove is shown as above, the radius of the circle will be determined based on the glove area size, then the circle area will be removed surrounding the center point found earlier.

Figure 5.44: Circle Area Removal Result

The result of circle area removal is shown as above.



```
%% Draw box on the glove
[label, numFinger] = bwlabel(glovearea2);
disp(numFinger);
if(numFinger < 6)
    disp("Double Dipping Detected");
    [label, numglove] = bwlabel(glovearea);
    props = regionprops(label, "BoundingBox", "Area");
    box = [props.BoundingBox];
    box = reshape(box,[4 numglove]);
    figure, imshow(img), title("Original Image with Bounded Defects")
    rectangle("Position", box(:,1), "EdgeColor", "r");
    text(box(1,1)-100, box(1,1)-50, "Double Dipping", "FontSize", 12, "Color", "r");
else
    disp("No Double Dipping Detected");
end
```

Figure 5.45: Bounding Boxes Creation and Drawing

The drawing and labeling of bounding boxes on the original image are done using the code above. The glove will be count as having double dipping defect if the number of white areas is less than 6 because a normal glove will have five fingers and one wrist area.

Figure 5.46: Double Dipping Detection Result

The result of double dipping defect detection is shown as above.

5.7 Cloth glove detection

The cloth glove defection codes are as follows:

```
im=imread('TR2.jpg');
E = entropyfilt(im);

Eim = rescale(E);

binary = im2bw(Eim,0.5);

BWao = bwareaopen(binary,2000);

nhood = ones(9);
closeBWao = imclose(BWao,nhood);
figure, imshow(closeBWao)
mask = imfill(closeBWao,'holes');
figure, imshow(mask)
numcloth = nnz(~mask);
clothcover = numcloth / numel(mask) * 100;

disp(['Percentage of cloth in glove image: ' num2str(clothcover) '%'])
```

Figure 5.47: Code for cloth glove detection

The outcome of the code:

```
Percentage of cloth in glove image: 43.2317%
```

Figure 5.48: outcome for cloth glove detection

5.8 Tearing detection

The following code is the segmentation and thresholding for tearing detection:

```
% color image segmentation
TRxh = rgb2hsv(im);
TRR= TRxh(:,:,1);
TRG = TRxh(:,:,2)*2.5;
TRB = TRxh(:,:,3);

TRSh = ((TRG>1)+(TRG<0.58))>0;
TRbw2 = imcomplement(TRSh);
figure, imshow(TRbw2)
```

Figure 5.49: code for color image segmentation for tearing detection

And the outcome of the code:

Figure 5.50: outcome for image segmentation and thresholding

outcome of the noise removal:

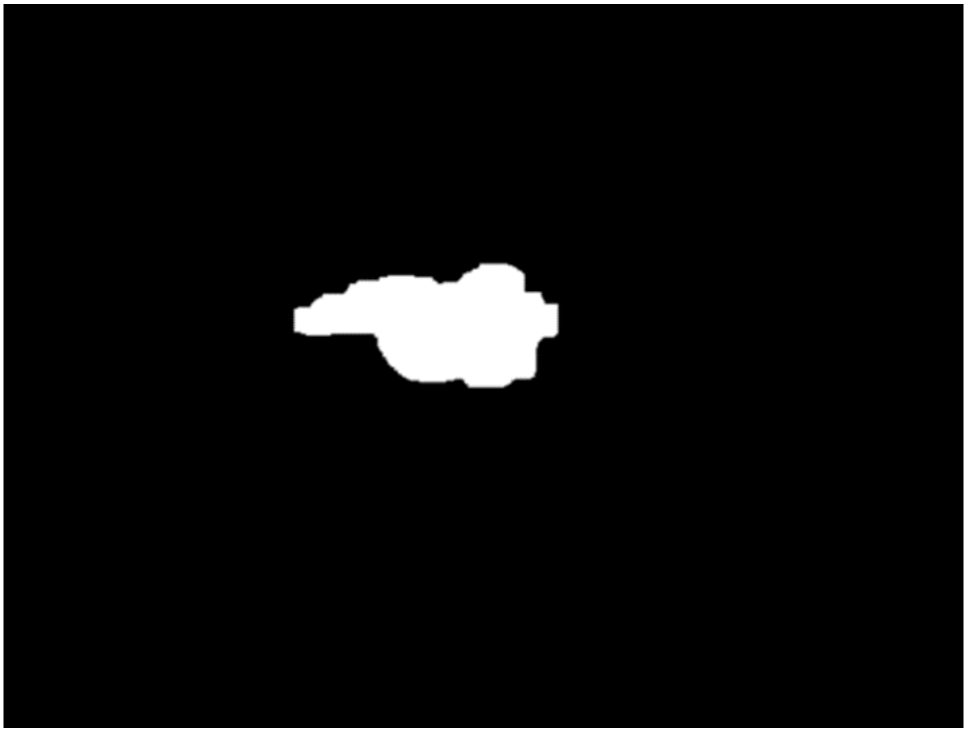


Figure 5.51: outcome of noise removal

Code for drawing bounding box:

```
[BBox, numRegions] = bwlabel(TRBWnobord);
BWfinal = regionprops(BBox, 'BoundingBox');
figure, imshow(im)
hold on;
% Plot the bounding box around each region.
for k = 1 : numRegions
    thisBBox = BWfinal(k).BoundingBox;
    rectangle('Position', thisBBox, 'EdgeColor', 'r','LineWidth',1);
end
```

Figure 5.52: code for bounding box

Result:

Figure 5.53: outcome for image segmentation and thresholding

5.9 Open seam detection

The code below is the code for the color image segmentation of open seam detection:

```
% color image segmentation
OSxh = rgb2hsv(im);
OSR= OSxh(:,:,1);
OSG = OSxh(:,:,2)*2.5;
OSB = OSxh(:,:,3);

[OSH,OSS,OSI] = rgb2hsv(OSR,OSG,OSB);

OSSh = ((OSH>1)+(OSH<0.5))>0;
figure, imshow(OSSh)
```

Figure 5.54: Code for color image segmentation for open seam detection

And the outcome is as follows:

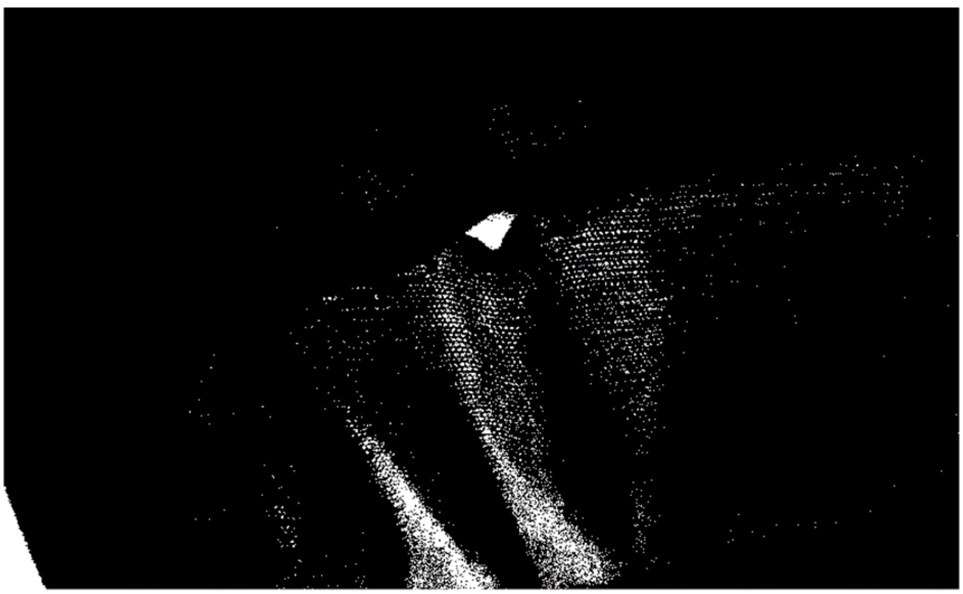


Figure 5.55: Outcome for color image segmentation for open seam detection

The figure is then put into a series of noise removing function:

```
OSShnobord = imclearborder(OSSh,4);
OSSHnoiserm = bwareaopen(OSShnobord, 1000)
OSEh = double(bwmorph(OSSHnoiserm,'erode',3));
OSDh = double(bwmorph(OSEh,'dilate',5));
OSBWnobord = imclearborder(OSDh,4);
figure, imshow(OSBWnobord)
```

Figure 5.56: Code of noise removal for open seam detection

And the final outcome is as follows:

Figure 5.57: Outcome for noise removal for open seam detection

Finally, bounding box is drawn around the detected defect with the following code:

```
[OSBBox, numRegions] = bwlabel(OSBWnobord);
BWfinal = regionprops(OSBBox, 'BoundingBox');

figure, imshow(im)
hold on;
% Plot the bounding box around each region.
for k = 1 : numRegions
    OSthisBBox = BWfinal(k).BoundingBox;
    rectangle('Position', OSthisBBox, 'EdgeColor', 'r','LineWidth',1);
end
```

Figure 5.58: Drawing bounding box

Outcome:

Figure 5.59: Final outcome

5.10 Stitching run off detection

The detection starts with the color image segmentation, and different part of the segmented image is used to identify the area of the gloves and the area of the background. Then, the two area is then combined into the finalized outcome. The code is as follows:

```
clc;
clear;
close all;
im=imread('ST10.jpg');

STxh = rgb2hsv(im);
STR= STxh(:,:,1);
STG = STxh(:,:,2)*5;
STB = STxh(:,:,3);


[STH,STS,STI] = rgb2hsv(STR,STG,STB);

STSbw = im2bw(STS,graythresh(STS));
STSbw2 = imcomplement(STSbw);
STEh = double(bwmorph(STSbw2,'erode',2));
STDh = double(bwmorph(STEh,'dilate',7));
STSBWdfill = imfill(STDh,'holes');
STSbwfinal = imcomplement(STSBWdfill);

STHBW = im2bw(STH);

subplot(121), imshow (STSbwfinal)
subplot(122), imshow (STHBW)

STfinalimage = STSbwfinal & STHBW ;
```

Figure 5.60: Color image segmentation of stitching run off detection

And the outcome is as follows:

Figure 5.61: Glove region(STbwfinal, left) and background region(STHBW, right)

Figure 5.62: Combined image

Then, the outcome is placed into a series of noise removal:

```
STEh = double(bwmorph(STfinalimage,'erode',3));

STnoiserm = bwareaopen(STEh, 2000)

STDh = double(bwmorph(STnoiserm,'dilate',7));
STBWnobord = imclearborder(STDh,4);
figure, imshow(STBWnobord)
```

Figure 5.63: Noise removal for stitching run off

And the outcome of noise removal:

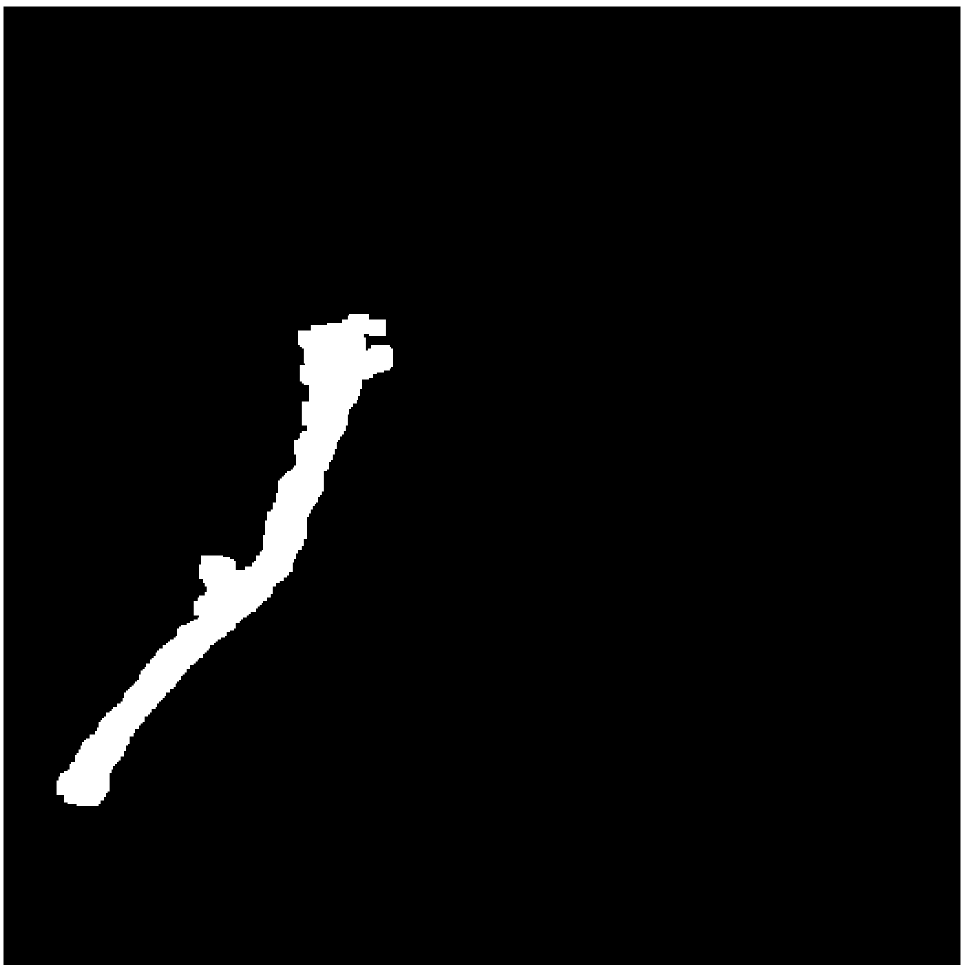


Figure 5.64: Result of noise removal for stitching run off

In the end of the function, bounding box is drawn around the defect:

```
[BBox, numRegions] = bwlabel(STBWnobord);
BWfinal = regionprops(BBox, 'BoundingBox');

figure, imshow(im)
hold on;
% Plot the bounding box around each region.
for k = 1 : numRegions
    thisBBox = BWfinal(k).BoundingBox;
    rectangle('Position', thisBBox, 'EdgeColor', 'r','LineWidth',1);

end
```

Figure 5.65: Drawing bounding box

Outcome:

Figure 5.66: Final result for stitching run off detection

6 Description and Discussion of Obtained Result

6.1 Latex Glove Defect Detection (For the dirty and stain + hole + finger not enough defects)

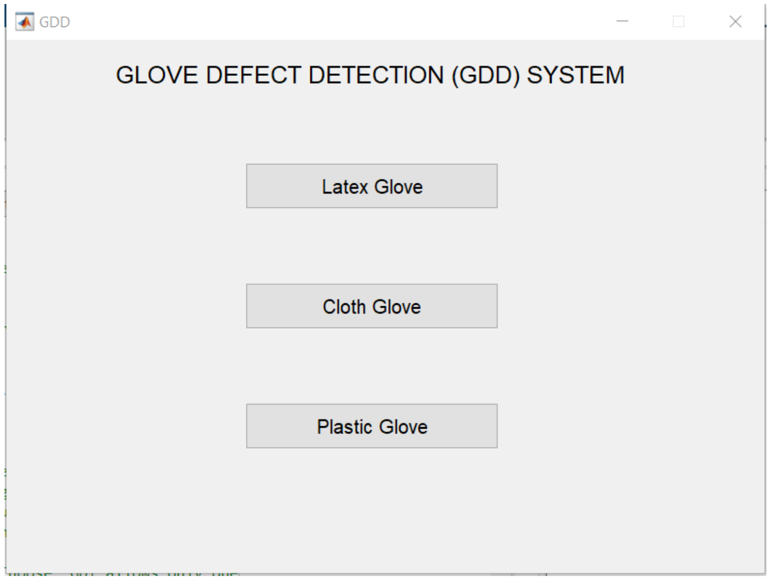


Figure 6.1: Main menu of GDD system

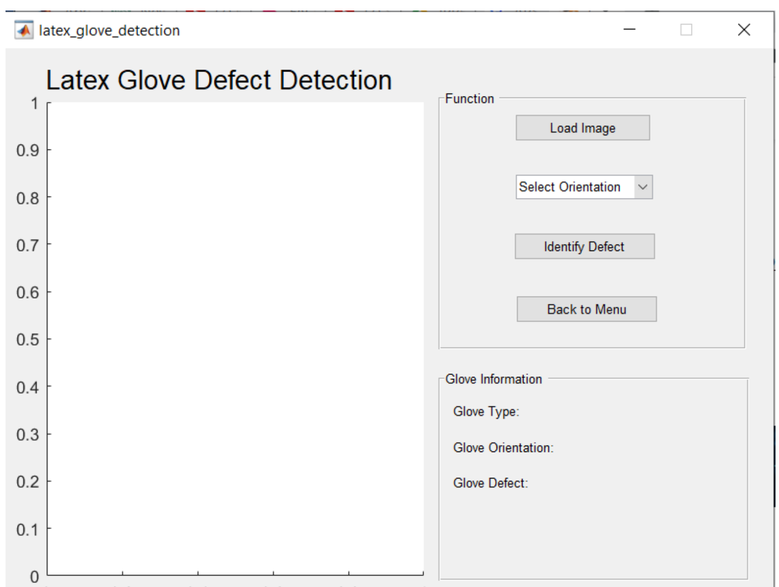


Figure 6.2: Latex glove defect detection page

GUIDE GUI is used in this assignment started with a main menu of type of glove for detecting the glove defect according to glove type shown as Figure 6.1. For detecting the latex glove, the user is required to click on the latex glove button to redirect to the latex glove defect detection page shown in Figure 6.2.

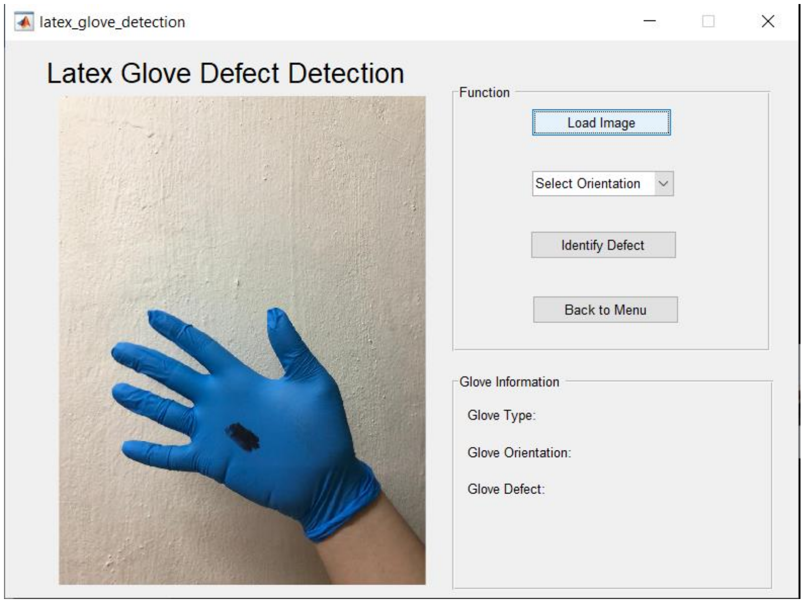


Figure 6.3: Image loaded in GUI

To load the image of glove for defect detection, the load image button is clicked to choose the image from the dataset and shown as Figure 6.3.

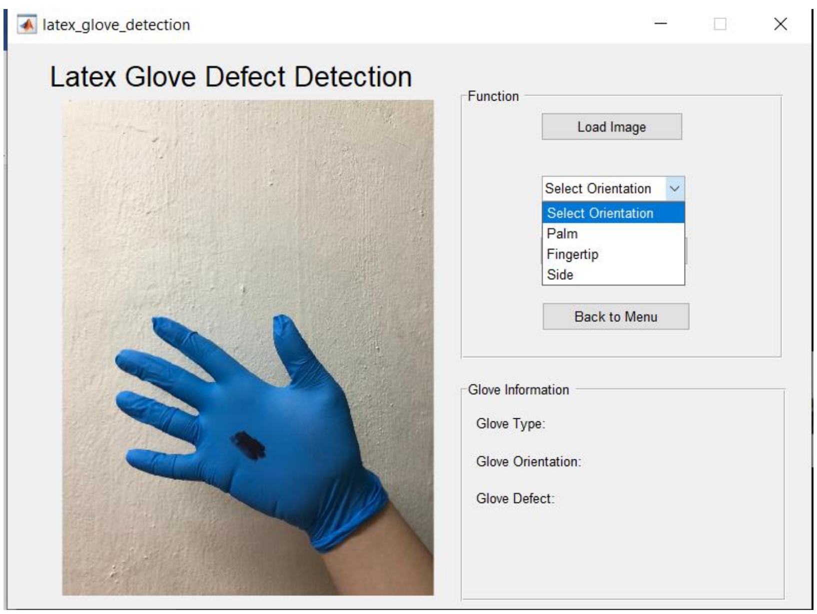


Figure 6.4: Orientation selection

In the selection of the glove orientation, there are 3 type of orientation can be chosen which are the palm, fingertip and side orientation shown in Figure 6.4.

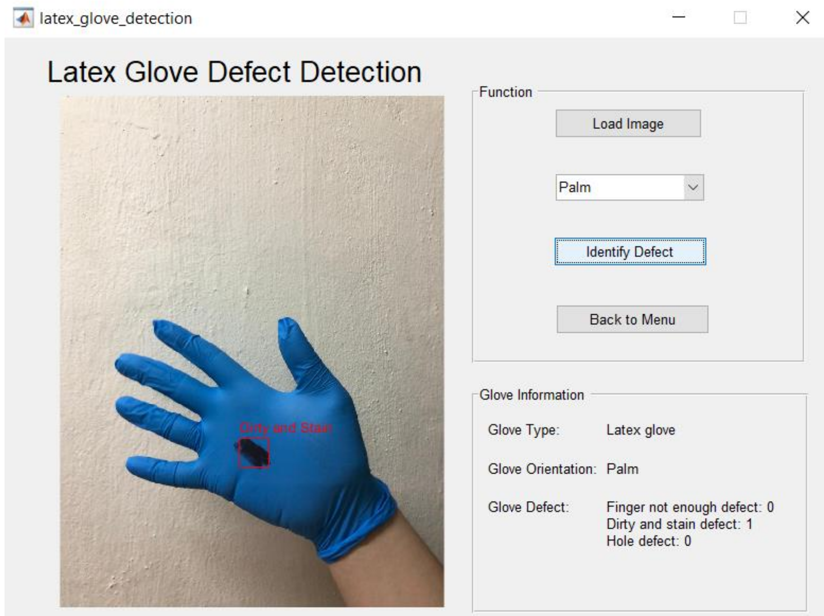


Figure 6.5: Final result of defect detection

After all the previous steps are done, the user can click on the identify defect button to perform defect detection shown as Figure 6.5. The back to menu button can redirect the user back to menu.

6.2 Plastic Glove Defect Detection (For the Touching + Thinner + Double Dipping Defects)

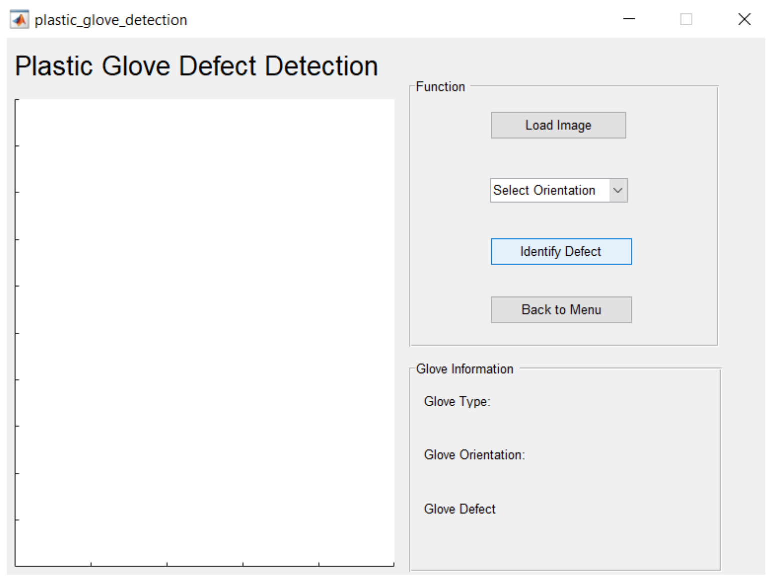


Figure 6.6: Plastic Glove Defect Detection Page

The user can redirect to the plastic glove defect detection page by clicking the button of Plastic Glove shown in Figure 6.2.

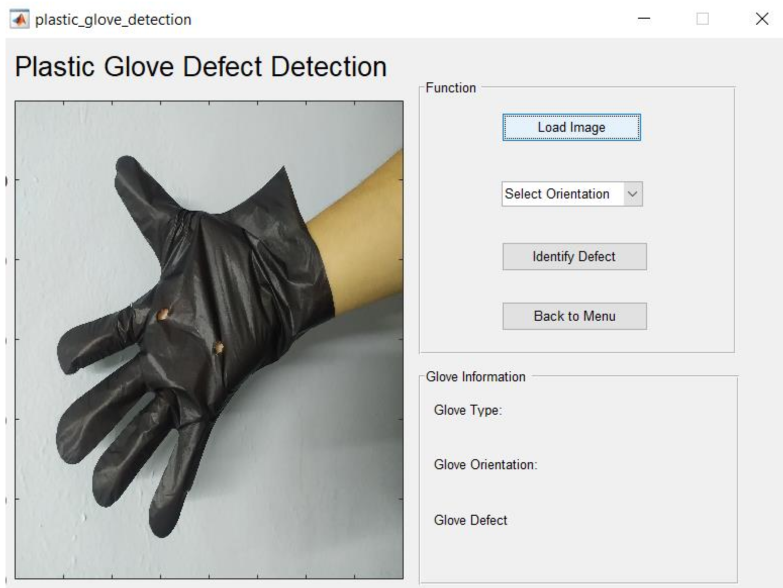


Figure 6.7: Load Image

The user can choose and load an image by clicking the "Load Image" button shown above.

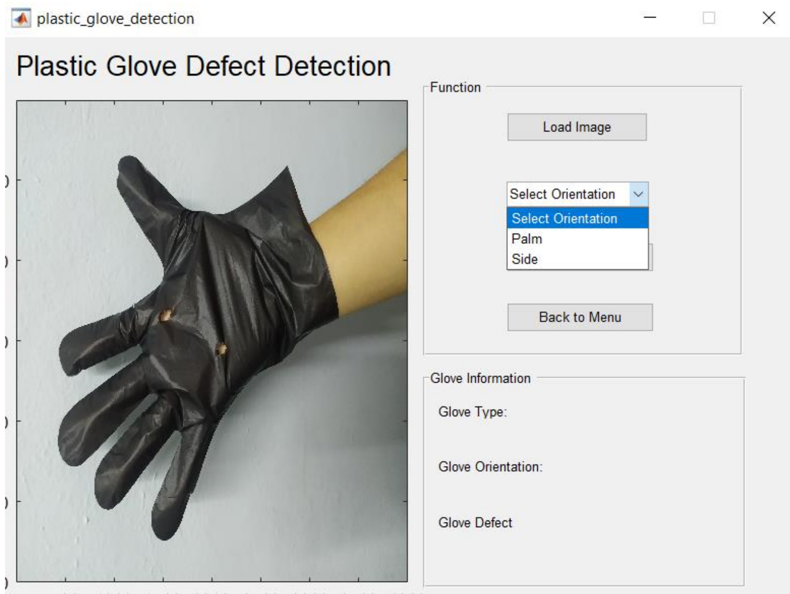


Figure 6.8: Image Orientation

The user can choose the orientation of the image using the pop menu shown above. There are "Palm" and "Side" selection for plastic glove detection.

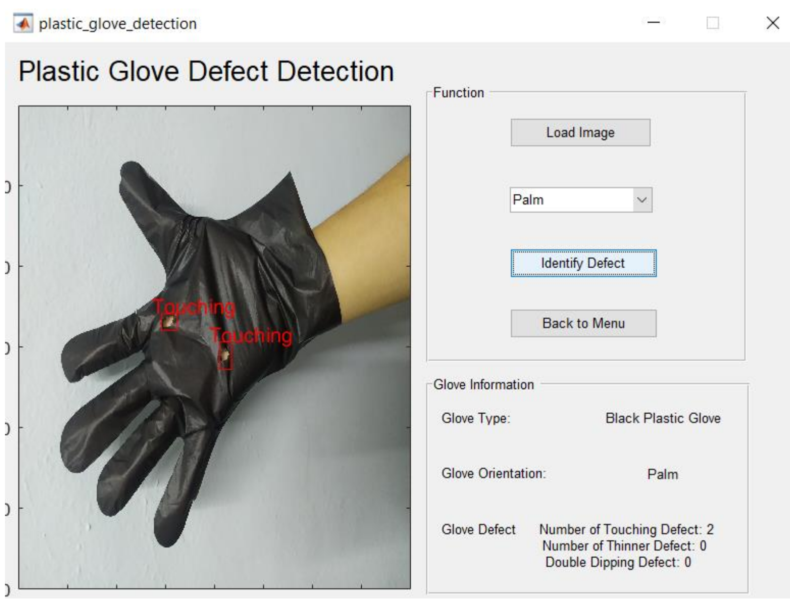


Figure 6.9: Result of Defect Detection

The user can identify defect on the selected image by clicking "Identify Defect" button, the detected defects will then be bound by red boxes shown as above.

6.3 Cloth glove defect detection (For open seam, tearing and stitching run off)

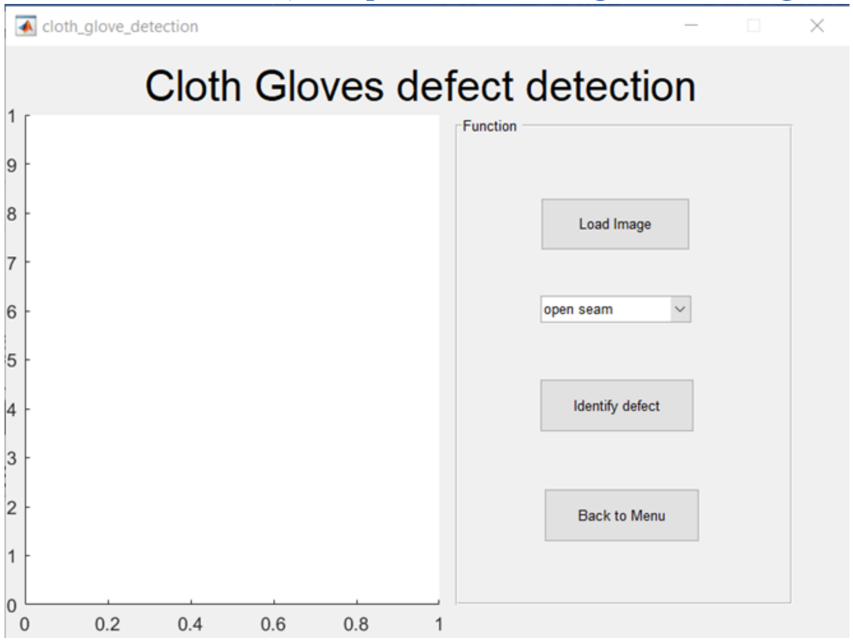


Figure 6.10: Cloth Glove Defect Detection Page

The user can redirect to cloth glove defect detection page by clicking button cloth gloves shown in figure 6.2

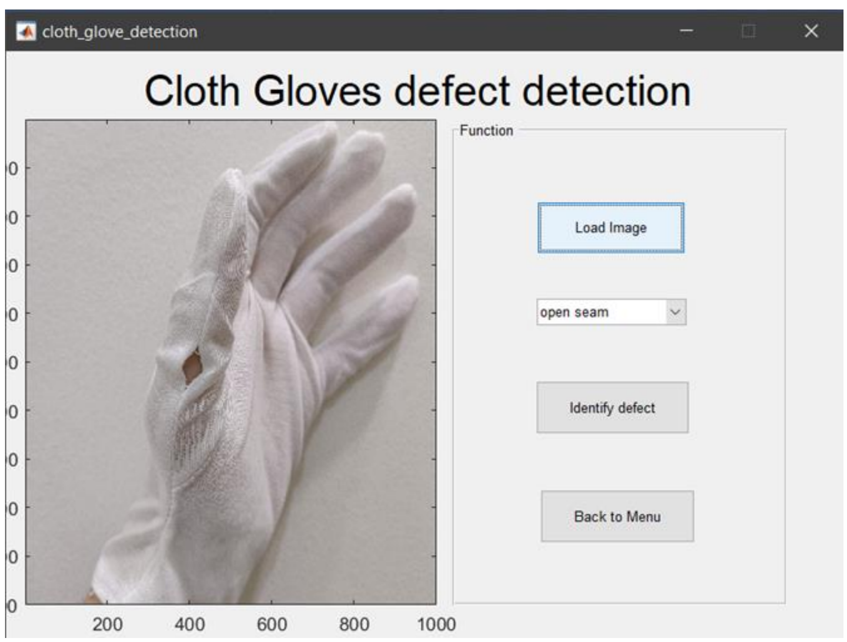


Figure 6.11: loading image

Image can be loaded with the load image button on the top.

Figure 6.12: defect to be detected selection

Defects to be detected can be chosen with the pop up menu as shown in picture.

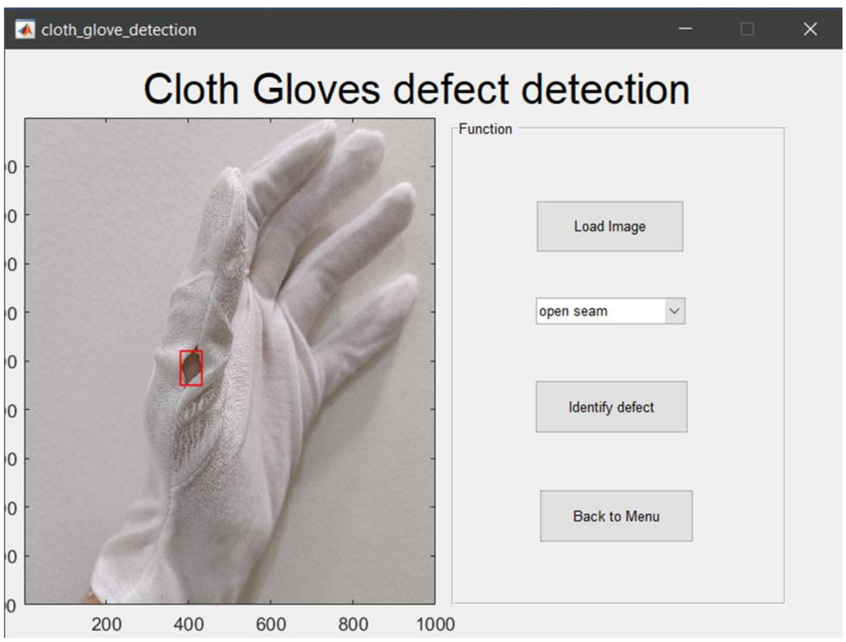


Figure 6.13: Result of Defect Detection

By clicking identify defect, the possible defect will be highlighted on the image as shown above.

7 Critical Comments, Analysis and Future Work Direction

7.1 Latex Glove Defect Detection

Since the glove defect detection (GDD) system is a kind of classification system, the performance analysis is produced to identify the accuracy of each defect detection. The analyses are shown as below

Dirty and stain

Performance analysis for GDD system (Latex glove detection)					
Defect type:	Dirty and stain				
Image defect folder	Image orientation type	Image name	True positive defect (Defect with correct position and type)	False positive defect (Defect with correct position but wrong type/Defect with wrong position)	False negative defect (Defect not detected)
dirty and stain	palm	palm_dirty(1)	1	0	0
		palm_dirty(2)	1	0	0
		palm_dirty(3)	1	0	0
		palm_dirty(4)	1	0	0
		palm_dirty(5)	1	0	0
		palm_dirty(6)	1	0	0
		palm_dirty(7)	1	0	0
		palm_dirty(8)	1	0	0
		palm_dirty(9)	1	0	0
		palm_dirty(10)	1	0	0
		palm_dirty(11)	1	0	0
		palm_dirty(12)	1	0	0
		palm_dirty(13)	1	0	0
		palm_dirty(14)	1	0	0
		palm_dirty(15)	1	0	0
		palm_dirty(16)	1	0	0
		palm_dirty(17)	1	0	0
		palm_dirty(18)	1	0	0
		palm_dirty(19)	1	0	0
		palm_dirty(20)	1	0	0
	fingertip	fingertip_dirty(1)	1	0	0
		fingertip_dirty(2)	1	0	0
		fingertip_dirty(3)	1	0	0
		fingertip_dirty(4)	1	0	0
		fingertip_dirty(5)	1	0	0
		fingertip_dirty(6)	1	0	0
		fingertip_dirty(7)	1	0	0
		fingertip_dirty(8)	1	0	0
		fingertip_dirty(9)	1	0	0
		fingertip_dirty(10)	1	0	0
		fingertip_dirty(11)	1	0	0
		fingertip_dirty(12)	1	1	0
		fingertip_dirty(13)	1	0	0
		fingertip_dirty(14)	1	0	0
		fingertip_dirty(15)	1	0	0
		fingertip_dirty(16)	1	0	0
		fingertip_dirty(17)	1	0	0
		fingertip_dirty(18)	1	0	0
		fingertip_dirty(19)	1	0	0
		fingertip_dirty(20)	1	0	0
		fingertip_dirty(21)	2	0	0
		fingertip_dirty(22)	2	0	0
		fingertip_dirty(23)	2	1	0
		fingertip_dirty(24)	2	0	0
		fingertip_dirty(25)	2	1	0

	side(left)	side_left_dirty(1)	1	1	0
		side_left_dirty(2)	1	1	0
		side_left_dirty(3)	1	1	0
		side_left_dirty(4)	1	1	0
		side_left_dirty(5)	1	3	0
		side_left_dirty(6)	1	0	0
		side_left_dirty(7)	1	3	0
		side_left_dirty(8)	1	3	0
		side_left_dirty(9)	1	2	0
		side_left_dirty(10)	1	2	0
		side_left_dirty(11)	1	2	0
		side_left_dirty(12)	1	3	0
		side_left_dirty(13)	1	0	0
		side_left_dirty(14)	1	1	0
		side_left_dirty(15)	1	1	0
		side_left_dirty(16)	1	0	0
		side_left_dirty(17)	1	0	0
		side_left_dirty(18)	1	0	0
		side_left_dirty(19)	1	0	0
		side_left_dirty(20)	1	3	0
	side(right)	side_right_dirty(1)	1	0	0
		side_right_dirty(2)	1	1	0
		side_right_dirty(3)	1	0	0
		side_right_dirty(4)	1	0	0
		side_right_dirty(5)	1	0	0
		side_right_dirty(6)	1	1	0
		side_right_dirty(7)	0	2	0
		side_right_dirty(8)	1	1	0
		side_right_dirty(9)	0	1	0
		side_right_dirty(10)	1	0	0
		side_right_dirty(11)	0	1	0
		side_right_dirty(12)	0	2	0
		side_right_dirty(13)	1	0	0
		side_right_dirty(14)	1	1	0
		side_right_dirty(15)	1	1	0
		side_right_dirty(16)	1	1	0
		side_right_dirty(17)	1	0	0
		side_right_dirty(18)	1	0	0
		side_right_dirty(19)	1	0	0
		side_right_dirty(20)	1	0	0
		Overall of Dirty and Stain	86	42	0
			Precision	Recall/ Accuracy	F1 score
			0.671875	1	0.803738318

Figure 7.1: Performance analysis for Dirty and Stain

Finger not enough

Performance analysis for GDD system (Latex glove detection)					
Defect type:	Finger not enough				
Image defect folder	Image orientation type	Image name	True positive defect (Defect with correct position and type)	False positive defect (Defect with correct position but wrong type/Defect with wrong position)	False negative defect (Defect not detected)
finger not enough	palm	palm_finger_not_enough(1)	1	0	0
		palm_finger_not_enough(2)	1	0	0
		palm_finger_not_enough(3)	1	0	0
		palm_finger_not_enough(4)	1	0	0
		palm_finger_not_enough(5)	1	0	0
		palm_finger_not_enough(6)	1	0	0
		palm_finger_not_enough(7)	1	0	0
		palm_finger_not_enough(8)	1	0	0
		palm_finger_not_enough(9)	1	0	0
		palm_finger_not_enough(10)	1	0	0
		palm_finger_not_enough(11)	1	0	0
		palm_finger_not_enough(12)	1	0	0
		palm_finger_not_enough(13)	1	0	0
		palm_finger_not_enough(14)	0	0	1
		palm_finger_not_enough(15)	0	0	1
		palm_finger_not_enough(16)	0	0	1
		palm_finger_not_enough(17)	1	0	0
		palm_finger_not_enough(18)	1	0	0
		palm_finger_not_enough(19)	1	0	0
		palm_finger_not_enough(20)	1	0	0
		palm_finger_not_enough(21)	2	1	0
		palm_finger_not_enough(22)	2	0	0
		palm_finger_not_enough(23)	2	1	0
		palm_finger_not_enough(24)	2	1	0
		palm_finger_not_enough(25)	2	0	0
	fingertip	fingertip_finger_not_enough(1)	1	0	0
		fingertip_finger_not_enough(2)	1	1	0
		fingertip_finger_not_enough(3)	1	0	0
		fingertip_finger_not_enough(4)	1	0	0
		fingertip_finger_not_enough(5)	1	0	0
		fingertip_finger_not_enough(6)	1	0	0
		fingertip_finger_not_enough(7)	1	1	0
		fingertip_finger_not_enough(8)	1	1	0
		fingertip_finger_not_enough(9)	1	1	0
		fingertip_finger_not_enough(10)	1	1	0
		fingertip_finger_not_enough(11)	1	1	0
		fingertip_finger_not_enough(12)	1	1	0
		fingertip_finger_not_enough(13)	1	2	0
		fingertip_finger_not_enough(14)	1	3	0
		fingertip_finger_not_enough(15)	1	1	0
		fingertip_finger_not_enough(16)	1	1	0
		fingertip_finger_not_enough(17)	1	1	0
		fingertip_finger_not_enough(18)	1	0	0
		fingertip_finger_not_enough(19)	1	1	0
		fingertip_finger_not_enough(20)	1	1	0
	side(left)	side_left_finger_not_enough(1)	0	1	0
		side_left_finger_not_enough(2)	1	1	0
		side_left_finger_not_enough(3)	1	1	0
		side_left_finger_not_enough(4)	0	0	1
		side_left_finger_not_enough(5)	1	1	0
		side_left_finger_not_enough(6)	1	1	0
		side_left_finger_not_enough(7)	1	1	0
		side_left_finger_not_enough(8)	1	1	0
		side_left_finger_not_enough(9)	1	1	0
		side_left_finger_not_enough(10)	1	0	0
		side_left_finger_not_enough(11)	1	0	0
		side_left_finger_not_enough(12)	1	0	0
		side_left_finger_not_enough(13)	0	0	1
		side_left_finger_not_enough(14)	0	0	1
		side_left_finger_not_enough(15)	1	1	0
		side_left_finger_not_enough(16)	1	0	0
		side_left_finger_not_enough(17)	1	1	0
		side_left_finger_not_enough(18)	1	1	0
		side_left_finger_not_enough(19)	1	1	0
		side_left_finger_not_enough(20)	1	1	0

	side(right)	side_right_finger_not_enough(1)	0	0	1
		side_right_finger_not_enough(2)	0	0	1
		side_right_finger_not_enough(3)	0	0	1
		side_right_finger_not_enough(4)	0	0	1
		side_right_finger_not_enough(5)	1	0	0
		side_right_finger_not_enough(6)	1	1	0
		side_right_finger_not_enough(7)	1	1	0
		side_right_finger_not_enough(8)	0	0	1
		side_right_finger_not_enough(9)	1	1	0
		side_right_finger_not_enough(10)	1	1	0
		side_right_finger_not_enough(11)	0	1	1
		side_right_finger_not_enough(12)	0	1	1
		side_right_finger_not_enough(13)	1	1	0
		side_right_finger_not_enough(14)	0	1	1
		side_right_finger_not_enough(15)	1	1	0
		side_right_finger_not_enough(16)	0	0	1
		side_right_finger_not_enough(17)	1	1	0
		side_right_finger_not_enough(18)	0	0	1
		side_right_finger_not_enough(19)	1	0	0
		side_right_finger_not_enough(20)	1	0	0
		Overall of Finger not Enough	73	43	16
			Precision	Recall/ Accuracy	F1 score
			0.629310345	0.820224719	0.712195122

Figure 7.2: Performance analysis for finger not enough

Hole (palm)

Performance analysis for GDD system (Latex glove detection)					
Defect type:	Hole				
Image defect folder	Image orientation type	Image name	True positive defect (Defect with correct position and type)	False positive defect (Defect with correct position but wrong type/Defect with wrong position)	False negative defect (Defect not detected)
hole	palm	palm_hole(1)	1	0	0
		palm_hole(2)	1	0	0
		palm_hole(3)	1	0	0
		palm_hole(4)	1	0	0
		palm_hole(5)	1	0	0
		palm_hole(6)	1	0	0
		palm_hole(7)	1	0	0
		palm_hole(8)	1	0	0
		palm_hole(9)	1	0	0
		palm_hole(10)	1	0	0
		palm_hole(11)	0	0	1
		palm_hole(12)	0	0	1
		palm_hole(13)	0	0	1
		palm_hole(14)	0	0	1
		palm_hole(15)	0	0	1
		palm_hole(16)	0	0	1
		palm_hole(17)	0	0	1
		palm_hole(18)	0	0	1
		palm_hole(19)	0	0	1
		palm_hole(20)	0	0	1
		palm_hole(21)	0	0	2
		palm_hole(22)	2	0	0
		palm_hole(23)	0	2	0
		palm_hole(24)	0	1	1
		palm_hole(25)	0	1	1
		palm_hole(26)	2	0	0
		Overall of Hole	14	4	14
			Precision	Recall/ Accuracy	F1 score
			0.777777778	0.5	0.608695652

Figure 7.3: Performance analysis for hole

Multiple defect (Mixed Defect: Dirty and Stain=4, Hole=2, Finger not enough=1)

Performance analysis for GDD system (Latex glove detection)					
Defect type:	Multiple defect				
Image defect folder	Image orientation type	Image name	True positive defect (Defect with correct position and type)	False positive defect (Defect with correct position but wrong type/Defect with wrong position)	False negative defect (Defect not detected)
multiple	palm	palm_multiple(1)	3	0	4
		palm_multiple(2)	4	2	1
		palm_multiple(3)	4	1	2
		palm_multiple(4)	4	3	0
		palm_multiple(5)	5	1	2
		palm_multiple(6)	5	0	2
		Overall for multiple	25	7	11
			Precision	Recall/ Accuracy	F1 score
			0.78125	0.694444444	0.735294118

Figure 7.4: Performance analysis for multiple defect

Through the analysis shown from Figure 7.1 to 7.4, we can find that the latex glove defect detector is more sensitive to dirty and stain defect and the hole defect has the least sensitive from the recall of the dataset. It is due to the light intensity of hole defect is always not fixed and sometimes the defect cannot be detected as it covered by shadow. From all the analysis, there are still a good accuracy among all of the defect which all the defects include multiple defects data have return more than 0.5 score for the precision, recall and F1 score.

Defect type	TP	FP	FN	Precision	Recall	F1 score
Dirty and stain	86	42	0	0.671875	1	0.803738
Finger not enough	73	43	16	0.62931	0.820225	0.712195
Hole	14	4	14	0.777778	0.5	0.608696
Multiple	25	7	11	0.78125	0.694444	0.735294
Overall	198	96	41	0.673469	0.828452	0.742964

Figure 7.5: Overall performance

From the overall performance, it is considered as a good defect detection as the recall and F1 score is high and around 0.75 as it means that more correct defect is detected in the latex glove defect detection system.

7.2 Future Work for Latex Glove

Since the latex glove detection accuracy is easily affect by the light intensity of the image, an algorithm which is more adaptive than Otsu method to the light coverage in the image should be found and implemented to the system. Besides that, the system is not so good on detect defect which is out of the glove region as it will always mixed with wrist area in image. Therefore, the technique like skin segmentation can be used in future work to remove the wrist area that is not related more effectively from the defect list. Lastly, the hole defect are sometimes defined as dirty and stain in the system, a better recognition method besides the skin coverage detection need to be implemented in future work.

7.3 Cloth Glove Defect Detection

The analysis of the detection is shown as below:

Tearing

defect	image name	true positive	True Negative	false positive	false negative
tearing	TR1	1	0	0	0
	TR2	0	0	0	1
	TR3	1	0	1	0
	TR4	1	0	0	0
	TR5	1	0	1	0
	TR6	1	0	0	0
	TR7	1	0	0	0
	TR8	1	0	0	0
	TR9	1	0	0	0
	TR10	1	0	0	0
	TR11	1	0	1	0
	TR12	1	0	0	0
	TR13	1	0	0	0
	TR14	1	0	0	0
	TR15	1	0	0	0
	TR16	1	0	0	0
	TR17	1	0	0	0
	TR18	1	0	0	0
	TR19	1	0	0	0
	TR20	1	0	0	0
		19	0	3	1
		precision	accuracy	f1 score	
		0.863636364	0.95	0.904761905	

Figure 7.6: Tearing model performance analysis

Open Seam

defect	image name	true positive	True Negative	false positive	false negative
open seam	OS1	1	0	0	0
	OS2	1	0	0	0
	OS3	1	0	0	0
	OS4	1	0	1	0
	OS5	1	0	0	0
	OS6	1	0	0	0
	OS7	1	0	1	0
	OS8	1	0	1	0
	OS9	1	0	1	0
	OS10	0	0	0	1
	OS11	1	0	1	0
	OS12	1	0	0	0
	OS13	1	0	0	0
	OS14	1	0	1	0
	OS15	1	0	0	0
	OS16	1	0	0	0
	OS17	1	0	0	0
	OS18	1	0	1	0
	OS19	1	0	0	0
	OS20	1	0	0	0
		19	0	7	1
		precision	accuracy	recall	
		0.730769231	0.95	0.826086957	

Figure 7.7: Open seam model performance analysis

Stitching run off

defect	image name	true positive	True Negative	false positive	false negative
STITCHING	ST1	1	0	0	0
	ST2	1	0	0	0
	ST3	1	0	0	0
	ST4	1	0	0	0
	ST5	1	0	0	0
	ST6	1	0	0	0
	ST7	1	0	0	0
	ST8	1	0	0	0
	ST9	1	0	0	0
	ST10	1	0	0	0
	ST11	1	0	0	0
	ST12	1	0	0	0
	ST13	1	0	0	0
	ST14	1	0	0	0
	ST15	1	0	0	0
	ST16	1	0	0	0
	ST17	1	0	0	0
	ST18	1	0	0	0
	ST19	0	0	0	1
	ST20	0	0	0	1
		18	0	0	2
		precision	accuracy	recall	
		1	0.9	0.947368421	

Figure 7.8: stitching run off model performance analysis

From the results above, it is clear that all models have high precision as different method is used to counteract their own specific problems. The open seam model has lower precision than both models, and the reason may be that it is used for detecting defects with small area, hence certain noises, such as shadows and lighting are not removed. The stitching run off model has a precision of 1, the reason of that is that the dataset only consists of pictures with clear defect on it. If there are further sample images that consist of smaller stitching defect, the precision of the model will decrease.

Defect type	TP	FP	FN	Precision	Recall	F1
tearing	19	3	1	0.863636	0.95	0.904761905
open seam	19	7	1	0.730769	0.95	0.826086957
STITCHING	18	0	2	1	0.9	0.947368421
overall	56	10	4	0.848485	0.933333	0.888888889

Figure 7.9: overall performance analysis

The overall performance shows that the model has overall precision of 0.85, recall score of 0.93 and f1 score of 0.89.

7.4 Future Work for Cloth Gloves Detection

For the cloth gloves detection, the texture of cloth is one of the main obstacles for cloth gloves detection due to it is one of the main sources of noise during image processing. The task of noise removal became more challenging when the texture of cloth combined with lighting issues when taking dataset. another future work for the cloth gloves defect detection system is that the researcher will integrate 3 of the defect detection mode into 1 mode, as in the current state, combining 3 of the defect detection mode in the current state will slow down the performance marginally.

7.5 Plastic Glove Defect Detection

To provide a critical analysis from a clearer outcome, the summary of the result for plastic glove defect detection is shown below.

Touching Defect

Defect Type	Image Orientation	Experimental Image	Correct Defect Detection	Incorrect Defect Detection	Fail Detection
Touching	Palm	Palm_Touching(1)	2	0	0
		Palm_Touching(2)	2	1	0
		Palm_Touching(3)	2	0	0
		Palm_Touching(4)	2	1	0
		Palm_Touching(5)	2	0	0
		Palm_Touching(6)	2	1	0
		Palm_Touching(7)	2	0	0
		Palm_Touching(8)	2	1	0
		Palm_Touching(9)	2	1	0
		Palm_Touching(10)	2	0	0
		Palm_Touching(11)	2	1	0
		Palm_Touching(12)	2	0	0
		Palm_Touching(13)	2	1	0
		Palm_Touching(14)	2	1	0
		Palm_Touching(15)	2	0	0
		Palm_Touching(16)	1	0	0
		Palm_Touching(17)	2	1	0
		Palm_Touching(18)	2	1	0
		Palm_Touching(19)	2	0	0
		Palm_Touching(20)	2	1	0
	Side	Side_Touching(1)	1	0	0
		Side_Touching(2)	1	0	0
		Side_Touching(3)	1	0	0
		Side_Touching(4)	1	0	0
		Side_Touching(5)	1	0	0
		Side_Touching(6)	1	0	0
		Side_Touching(7)	1	0	0
		Side_Touching(8)	1	0	0
		Side_Touching(9)	1	0	0
		Side_Touching(10)	1	0	0
		Side_Touching(11)	1	0	0
		Side_Touching(12)	1	0	0
		Side_Touching(13)	1	0	0
		Side_Touching(14)	1	0	0
		Side_Touching(15)	1	0	0
		Side_Touching(16)	1	0	0
		Side_Touching(17)	1	0	0
		Side_Touching(18)	1	0	0
		Side_Touching(19)	1	0	0
		Side_Touching(20)	1	0	0
		Total	**59**	**11**	**0**
		Precision	**0.842857143**		
		Accuracy	**1**		
		F1 Score	**0.914728682**		

Figure 7.10: Summary for Touching Defect Result

For Touching Defect, the result of accuracy showing that there is no any touching defect miss out from the detection. However, there is a few numbers of incorrect detection where all of them is double dipping detected on the image which proved that the touching detection alone is not having issue but double dipping.

Double Dipping Defect

Defect Type	Image Orientation	Experimental Image	Correct Defect Detection	Incorrect Defect Detection	Fail Detection
Double Dipping	Palm	Palm_Double Dipping(1)	1	0	0
		Palm_Double Dipping(2)	1	0	0
		Palm_Double Dipping(3)	1	0	0
		Palm_Double Dipping(4)	1	0	0
		Palm_Double Dipping(5)	1	0	0
		Palm_Double Dipping(6)	1	0	0
		Palm_Double Dipping(7)	1	0	0
		Palm_Double Dipping(8)	1	0	0
		Palm_Double Dipping(9)	1	0	0
		Palm_Double Dipping(10)	1	0	0
		Palm_Double Dipping(11)	1	0	0
		Palm_Double Dipping(12)	1	0	0
		Palm_Double Dipping(13)	1	0	0
		Palm_Double Dipping(14)	1	0	0
		Palm_Double Dipping(15)	1	0	0
		Palm_Double Dipping(16)	1	0	0
		Palm_Double Dipping(17)	1	0	0
		Palm_Double Dipping(18)	1	0	0
		Palm_Double Dipping(19)	1	0	0
		Palm_Double Dipping(20)	1	0	0
		Total	**20**	**0**	**0**
		Precision	**1**		
		Accuracy	**1**		
		F1 Score	**1**		

Figure 7.11: Summary for Double Dipping Defect Result

For Double Dipping Defect, the result of accuracy showing that there is no any double dipping defect miss out from the detection. Besides that, there is no any incorrect defect detection meaning that there is no issue integrating all of the defect together.

Thinner Defect

Defect Type	Image Orientation	Experimental Image	Correct Defect Detection	Incorrect Defect Detection	Fail Detection
Thinner	Palm	Palm_Thinner(1)	1	0	0
		Palm_Thinner(2)	1	0	0
		Palm_Thinner(3)	1	0	0
		Palm_Thinner(4)	1	0	0
		Palm_Thinner(5)	1	1	0
		Palm_Thinner(6)	1	0	0
		Palm_Thinner(7)	1	0	0
		Palm_Thinner(8)	1	0	0
		Palm_Thinner(9)	1	0	0
		Palm_Thinner(10)	1	0	0
		Palm_Thinner(11)	1	0	0
		Palm_Thinner(12)	1	0	0
		Palm_Thinner(13)	1	0	0
		Palm_Thinner(14)	1	0	0
		Palm_Thinner(15)	1	0	0
		Palm_Thinner(16)	1	1	0
		Palm_Thinner(17)	1	0	0
		Palm_Thinner(18)	1	1	0
		Palm_Thinner(19)	1	1	0
		Palm_Thinner(20)	1	0	0
		Total	**20**	**4**	**0**
		Precision	**0.833333333**		
		Accuracy	**1**		
		F1 Score	**0.909090909**		

Figure 7.12: Summary for Thinner Defect Result

For Thinner Defect, the result of accuracy showing that there is no any thinner defect miss out from the detection. However, there is a few numbers of incorrect detection where all of them is double dipping detected on the image which proved that the thinner detection alone is not having issue but double dipping.

Defect Type	Correct Defect Detection	Incorrect Defect Detection	Fail Detection	Precision	Accuracy	F1 Score
Touching	59	11	0	0.842857143	1	0.914728682
Double Dipping	20	0	0	1	1	1
Thinner	20	4	0	0.833333333	1	0.909090909
Overall	99	15	0	0.892063492	1	0.941273197

Figure 7.13: Overall Performance

Firstly, there is no any fail detection from the experimental data showing that all defects are detected with the accuracy of 100% but there are a few numbers of incorrect defect detection where all of them are false positive double dipping detected with the overall precision of 89.21%. However, the defect detection for plastic glove is still acceptable and meeting the requirement since the F1 score have 94.12% which is a relatively high score. To find out the reason of the false positive double dipping detected, one of the analysis examples is shown as below.

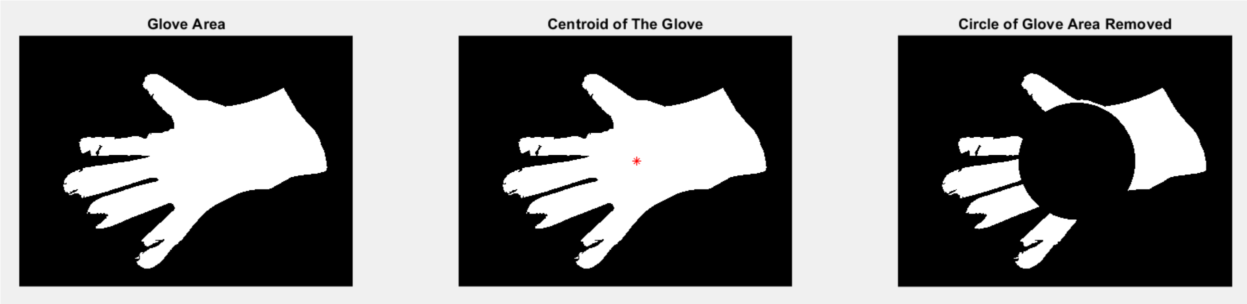


Figure 7.13: Glove Area and Removal for Palm_Touching(2).jpg

The result of touching data gone through the double dipping algorithm is shown above.

Figure 7.13: Area Removal for Palm_Touching(2).jpg

Figure above shows that the area of fingers and wrist are not separated nicely and still connected which is the reason of the false positive happen.

7.6 Future Work for Plastic Glove

Although the performance for touching and thinner detection is high, it might be only occurred for this specific experimental data, hence the testing for these defects can be conducted using more data differ from the used data to further improve the accuracy and robust of detection. The double dipping detection should have a better improvement to decrease the false positive result such that the radius of the circle removal should be more precise with more defect area condition, so that the fingers and wrist area of the glove can be separated correctly for defect detection.

8 Conclusion

In conclusion, the glove defect detection had acquired the researchers with the deep research starting from image acquisition, image pre-processing, defect area extraction and defect recognition for different type of glove defect. It had made the researchers to understand the concept of computer vision from this assignment. With this knowledge, the techniques and knowledge which learned in the module can be implemented in more other defect detection system in future work.

References

Bangare, S. L., Dubal, A., & Bangare, P. S. (2015). Reviewing Otsu's Method For Image Thresholding. *International Journal of Applied Engineering Research*, 21777-21783.

Cortex Robotics. (2022). *Glove AOI Inspection System*. Retrieved from Cortex Robotics: https://cortexrobotics.my/product/glove-aoi-inspection-system/

Great Learning Team. (2022, July 26). Introduction to image processing: What is image processing? Great Learning Blog: Free Resources what Matters to shape your Career! Retrieved December 23, 2022, from https://www.mygreatlearning.com/blog/introduction-to-image-processing-what-is-image-processing/

Kaler, P. (2016). Study of Grayscale imagein Image processing. *International Journal on Recent and Innovation Trends in Computing and Communication*, 309-311.

Khattab, D., Ebied, H. M., Hussein, A. S., & Tolba, M. F. (2014, August 31). Color image segmentation based on different color space models using automatic GrabCut. The Scientific World Journal. Retrieved December 23, 2022, from https://www.hindawi.com/journals/tswj/2014/126025/

Maheswari , D., & Radha, D. (2010). NOISE REMOVAL IN COMPOUND IMAGE USING MEDIAN FILTER . *International Journal on Computer Science and Engineering*, 1359-1362.

MathWorks. (2019, March 10). *Removal of Palm region from binary image of hand*. Retrieved from MathWorks: https://www.mathworks.com/matlabcentral/answers/449274-removal-of-palm-region-from-binary-image-of-hand

Mello Román, J. C., Vázquez Noguera, J. L., Legal-Ayala, H., Pinto-Roa, D. P., Gomez-Guerrero, S., & García Torres, M. (2019, March 4). Entropy and contrast enhancement of infrared thermal images using the multiscale top-hat transform. MDPI. Retrieved December 23, 2022, from https://www.mdpi.com/1099-4300/21/3/244

Perimal, M., Basah, S. N., Safar, M. J., & Yazid, H. (2018). Hand-gesture recognition-algorithm based on finger counting. *ournal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-13)*, 19-24.

Sebastián, S.-C., Ramos-Arreguín, J.-M., Ortiz Echeverri, C., Cabal-Yepez, E., Pedraza-Ortega, J.-C., & Rodriguez-Resendiz., J. (2018). Image dehazing using morphological opening, dilation and Gaussian filtering. *Signal, Image and Video Processing*, 1329–1335.

Wang, X. &. (2020). On-line Defect Detection and Classification of Latex Gloves. *Journal of Physics: Conference Series*, 1-7.

WorkWithColor.com. (2013). *Color Theory*. Retrieved from WorkWithColor.com: http://www.workwithcolor.com/blue-color-hue-range-01.htm