The University of Windsor

ELEC4490: Sensors and Vision Systems

Summer 2020

Final Project Report

10.2 Labels on Glue

Friday, August 14, 2020

Emmanuel Mati

104418019

Research

The project at hand is intended to develop code to recognize labels on glue bottles and to categorize them. The bottles fall into one of three categories starting with whether they even have a label or not. Next, the code must be able to identify whether the label on the glue bottle is straight; this part is interpreted as also whether or not the label is placed in the center of the bottle. Finally, the code must check to see if the labels are torn or folded. A sample of glue bottles is shown below.

Figure 1 Given test image, Glue1

To begin the project, a cell array is used to import all of the test images that will be used to categorize the glue bottles. Then a looping is used to loop through all of the different functions that are utilized by the project.

```
%Clearing Workspace
close all,clear all,clc;

%Calling all of our sample images into a single cell array
bottles = {imread('Glue1.jpg'), imread('Glue2.jpg'), ...
    imread('Glue3.jpg'), imread('Glue4.jpg'),...
    imread('Glue5.jpg'), imread('Glue6.jpg')};

for i=1:6
    [widths, mask,maskedImage] = InputImage(bottles{i});
    labelChecker(mask, bottles{i}); %%checking labels with mask
    isIntact(widths, maskedImage);
end
```

The first step that I needed to tackle was creating a mask for our glue bottles and to remove the background of the sample. I did this by adjusting each of the RGB elements by converting the sample image using im2bw and adjusting the threshold until the only thing remaining was the outline of the glue bottles. Then the outlines were combined and holes filled to provide a mask of the glue bottle shown below.

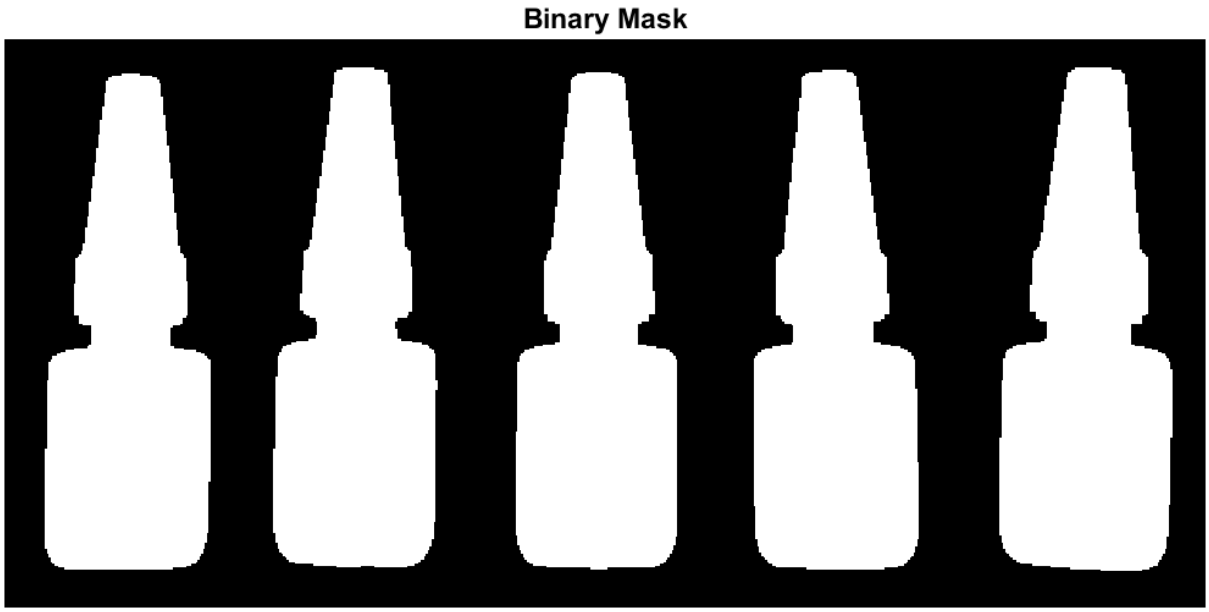

Figure 2 Mask for glue bottles

```
%%Function that creates our mask
function [widths, mask, maskedImage] = InputImage(sampleImage)
figure;imshow(sampleImage);title('Input Image');%displaying input image

%adjusting the rgb compents to build our mask
r = ~im2bw(sampleImage(:,:,1),0.3);g = ~im2bw(sampleImage(:,:,2),0.25);b = ~im2bw(sampleImage(:,:,3),0.19);
mask=(r.*g).*b; %matrix element-wide operations
%filling up holes
mask = ~imfill(~mask, 'holes');
%Morphological operations to remove noise
SE = strel('square',5);
mask = imopen(~mask, SE);
figure;imshow(mask);title('Binary Mask');
```

With the mask created, we could now use it to remove the background of the samples and also calculate how wide each glue bottles' base-width is. Note that to do this, I simply anded all

of the RGB components of the image with the mask. Then, I looped horizontally through each pixel of the binary image to calculate and store each bottle's width in an array for later use.

Figure 3 Background Removed

```
%manipulating eag RGB component to get the mask
rMask = double(sampleImage(:,:,1));gMask = double(sampleImage(:,:,2));bMask = double(sampleImage(:,:,3));
%Removing the background from our input image
maskedImage=cat(3, uint8(rMask.*mask),uint8(gMask.*mask) ,uint8(bMask.*mask));

%finding the width of each bottle's base region
[x,y] = size(mask);
width = zeros(5, 2);
rowPos = 1;
for i=2:1:y
    if mask(floor(x/1.6),i) == 1 && mask(floor(x/1.6),i - 1)== 0
        width(rowPos,1) = i;
    elseif mask(floor(x/1.6),i) == 0 && mask(floor(x/1.6),i - 1) == 1
        width(rowPos,2) = i-1 ;
        rowPos = rowPos + 1;
    end
end
widths = width;
end
```

Now I needed to check and see if the bottle had a label on it or not. I created a function to remove the labels from the glue bottles using im2bw thresholding again. Then, I turned it into a binary image and divided it with our mask. If our image had less than 75% of the pixels that the mask had, then we could say that our image had a label. Otherwise, it did not have a label.

Figure 4 Title says which images had labels

```
%%Question 1)Check to see if label is on the bottle
function labelChecker(binaryMask, glueBottles)
inputImage = glueBottles;
bottle = [0 0 0 0 0];  %used to keep track of the bottles
ourTitle = "Bottles in the following positions have labels: ";

%erode the bottles so that sticker details are emphasized
SE = strel('diamond', 3);
glueBottles = imerode(glueBottles, SE);

[x,y,z]=size(glueBottles);
%separating the 5 glue bottles
bottle(1) = sum(im2bw(glueBottles(x/2:x,1:floor(y/5),:),0.5),'all')/sum(binaryMask(x/2:x,1:floor(y/5)),'all');
bottle(2) = sum(im2bw(glueBottles(x/2:x,floor(y/5):floor(2*y/5),:),0.5),'all')/sum(binaryMask(x/2:x,floor(y/5):floor(2*y/5)),'all');
bottle(3) = sum(im2bw(glueBottles(x/2:x,floor(2*y/5):floor(3*y/5),:),0.5),'all')/sum(binaryMask(x/2:x,floor(2*y/5):floor(3*y/5)),'all');
bottle(4) = sum(im2bw(glueBottles(x/2:x,floor(3*y/5):floor(4*y/5),:),0.5),'all')/sum(binaryMask(x/2:x,floor(3*y/5):floor(4*y/5)),'all');
bottle(5) = sum(im2bw(glueBottles(x/2:x,floor(4*y/5):y,:),0.5),'all')/sum(binaryMask(x/2:x,floor(4*y/5):y),'all');

for i=1:1:5
    if bottle(i) < 0.75
        newText = num2str(i);
        ourTitle = strcat(ourTitle, newText, ', ');
    end
end

figure;imshow(inputImage);title(ourTitle);
end
```

For parts 2 and 3, I accidentally solved both of them using the code needed to solve part 2. This code also solved part 1 too. I used the hough as recommended in the lecture to check the angles of the label. If the sum of all of the hough distances was greater than a certain threshold value, then our image was either folded/ripped, straight, or missing. These threshold values were determined through trial and error. This function took a sample of the glue bottle's, background

and subtracted it from the image to help isolate the lines for the label. Isolating the lines for the labels was a huge challenge. I ended up writing at least 500 lines of code trying to do this but in the end, I could not fully separate the labels from the bottles. Thus the hough transform did not produce very reliable data because the sample edges were not perfectly shaped to represent the labels. I did however tune it enough to produce data that worked for all of the sample images. This was by far the most challenging part of the project and it took me two days before I was getting reasonable outlines for the labels. White labels on a white background are very hard to separate, especially when there are different colours embedded inside of the label.

Figure 5 Answers Question 1, 2 and 3 of the project

```
%%Question 2 & 3, Check to see if the label is not straight or folded/ripped
function isIntact(baseWidths, inputImage)
ourTitle = "";
[x,y, z] = size(inputImage);
Hs = zeros(1,5);
%Using Hough Transform to find if the images edges are level or not
for i=1:5
    gray = rgb2gray(inputImage(floor(x/1.85:x*0.95),baseWidths(i,1):baseWidths(i,2),:));
    sampleGray = rgb2gray(inputImage(floor(x*.89:x*0.9),baseWidths(i,1)+10:baseWidths(i,2)-10,:));
    removeGray = mean(sampleGray(:));
    mask1 = removeGray - 30 < gray;
    mask2 = gray < removeGray+30;
    gray(mask1 & mask2) = 0;
    BW = edge(gray,'canny');
    [H,T,R] = hough(BW,'RhoResolution',0.5,'Theta',-90:0.5:89.5);
    Hs(i)=sum(H,'all');
end

% Displaying List of images and using different thresholds for different
% types of coloured images
for i=1:5
    if Hs(i) < 300000   %blank condition
        ourTitle = strcat(ourTitle,{' '}, num2str(i), ': missing label,');
    elseif Hs(i) < 377000 ||386000 < Hs(i) && Hs(i) < 390000 || 420000 < Hs(i) && Hs(i) < 480000|| 520000 < Hs(i)
        ourTitle = strcat(ourTitle,{' '}, num2str(i), ': not Straight,');
   elseif 480000 < Hs(i) && Hs(i) < 520000|| 540000 < Hs(i) && Hs(i) < 565000|| 570000 < Hs(i) && Hs(i) < 600000
        ourTitle = strcat(ourTitle,{' '}, num2str(i), ': torn/folded,');
    else
        ourTitle = strcat(ourTitle,{' '}, num2str(i), ': passes,');
    end
end
figure;imshow(inputImage);title(ourTitle);
end
```

Overall, this project was a lot more challenging than expected and I spent a lot of time trying to figure out how to interpret the hough transform output data. The hough data varied between different RGB elements and label conditions. Also, image segmentation on a noisy image with multiple colour parameters is very hard to accomplish.