Xi'an Jiaotong-Liverpool University

西交利物浦大学

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE330 Lab Report 5

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Engineering

Student Name : Lingxuan Kong
Student ID : 1405867

Contents

1 Task 1 **2**

2 Task 2 **5**

2.1 Transform Binary Photo 5

2.2 Crop Image 6

2.3 Detection by Erosion 10

2.4 Hit-Miss Detection 12

3 Conclusion **14**

1 Task 1

This first was ask the students to write algorithms, which can count the coins in a photo automatically. The original photo is a color photo have three floors. Thus, we need to first transform this photo into a binary photo, and then do some significant operations to obtain the numbers of coins.

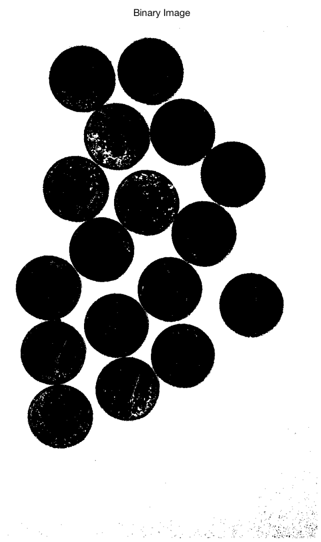

Figure 1: Binary Image

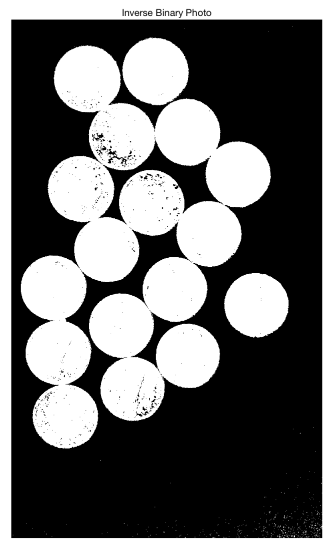

Figure 2: Inverse Binary Image

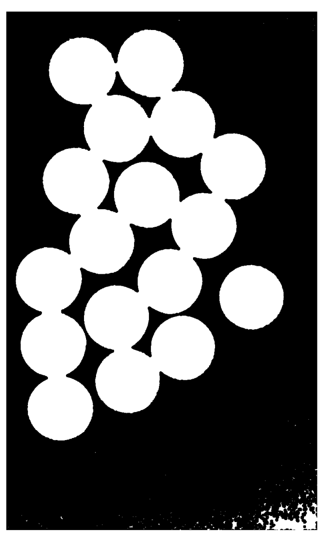

Figure 3: Apply Close

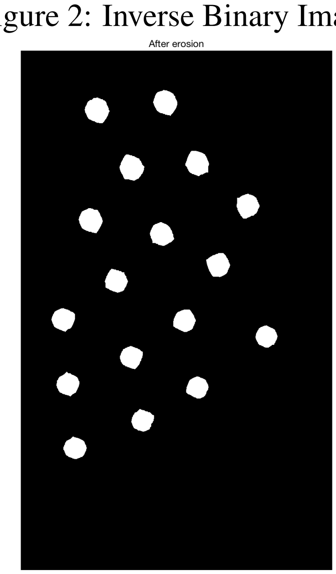

Figure 4: Apply Erosion

Then we observe the figure1 to figure4,

- Figure1 shows the binary image of the original photo. The background is black, represent the coins in this photo, and the foreground is white, as shown in figure1.
- Figure2 shows the inverse binary image of figure1, which means transform the background to foreground, foreground to background. As shown in figure2, the foreground, which represent the coins, shows the white color, and the background shows the black color. Inverse the background and foreground can make a different to the results of the following steps.
- The figure3 shows the results of close, we can find that the holes have been filled with this operations. Additional, the close operation can remove the small holes in the foreground and enlarge the region of foreground. Then we observe the coins in the figure3, we can find that some coins have been connected together, that's the reason why we call this operation **close**.
- The figure4 displays the result of erosion, which can decrease the region of foreground. In another word, erosion can remove the pixels on the object boundaries and make them open. Before counting the number of coins, the white points in the figure4, which represent the coins should be single, not connected, or the results of counting will be unreasonable. Thus, the size and shape of the structuring element will play an important roles in erosion. So, the students should test the SE to make the results is reasonable. After that, we can use the matlab build-in function to count the number of coins, it should be **17**.

Then this report will provide the code in matlab, and analysis how the algorithm works.

- **4-5** Apply the 'hsv' operation to show the rgb photo, and task the second floor to represent the photo in gary level.
- **6-8** Obtain the threshold voltage of the photo, then transform it into binary photo and display it.
- **9-10** Inverse the binary photo, exchange the background and foreground, and display it.
- **11-13** Define the structuring element and apply the close to this photo, then display it.
- **14-16** Set another structuring element, which should be disk, then apply the erosion operation and display the results.

- **17-18** Perform the counting function to obtain the number of coins, the number should be 17.

```
1 a=imread('Coins.jpg');%read the original photo
2 a=im2double(a);%change it to double
3 subplot(2,3,1),imshow(a),title('Original Photo');
4 a=rgb2hsv(a);%use hsv to display its RGB photo
5 a=a(:,:,2);%task the second floor of its RGB
6 x=graythresh(a);%obtain the threshold value
7 b=im2bw(a,x);%transform the photo into binary
8 subplot(2,3,2),imshow(b),title('Binary Image');
9 b=¬b;%inverse it, background to foreground
10 subplot(2,3,3),imshow(b);
11 se=strel('disk',8);%set the SE
12 c=imclose(b,se);%apply close
13 subplot(2,3,4),imshow(c);
14 se1=strel('disk',80);%set SE
15 d=imerode(c,se1);%apply erosion
16 subplot(2,3,5),imshow(d);
17 [L,num]=bwlabel(d);%count the foreground points in ths photo
18 num;%obtain the number of coins
```

2 Task 2

2.1 Transform Binary Photo

In this task, we need first transform the image to the binary photo, the foreground, which should represent the object is white, and the background is black.

Figure 5: The car licence in binary photo

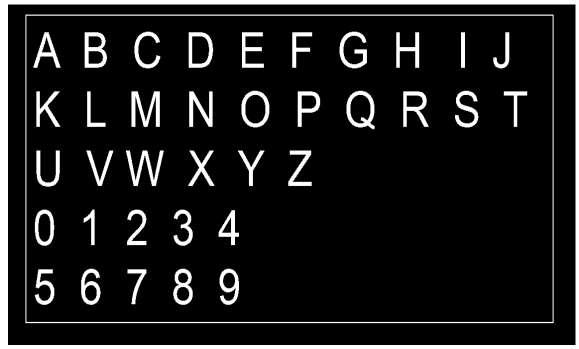

Figure 6: The template in binary photo

The figure5 and figure6 shows the two RGB image in binary image. As we see, the foreground represent the single word using the white color, and the background using the white color.

```
a=imread('car_license_plate.png');%read the original car license
b=imread('alphanumeric_templates.png');%read the template
%obtain the threshold value of two images
x1=graythresh(a);
x2=graythresh(b);
%change the RGB image into binary image
a_bin=im2bw(a,x1);
b_bin=im2bw(b,x2);
%display the image
figure,imshow(a_bin);
figure,imshow(b_bin);
%write the image into computer
imwrite(a_bin,'Car_2bw.png');
imwrite(b_bin,'Al_2bw.png');
```

2.2 Crop Image

However, if we want to achieve the Car License Plate Recognition, we must crop the image to many single word. And compare the single word one by one with the template database to found the correct word. Then this report will show the results after cropping:

- As the figure7 shown, this car license has been cut into 8 different characters, **E K F H 1 Q O C**.
- The figure8 display the results of cropping template, it shows 36 different char, **A-Z, 0-9**.
- We can use the matlab build-in function **imcrop** to crop the image. Because the there is no chinese in template, so the chinese word has been abandoned in this task. In another, this task only fouces on how to recognize the English word and the number from 0-9.
- It is significant to add the boundary to the word **I**.

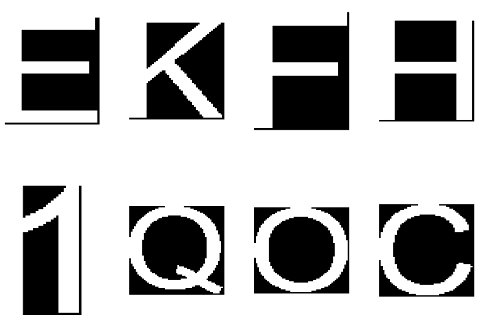

Figure 7: Car license

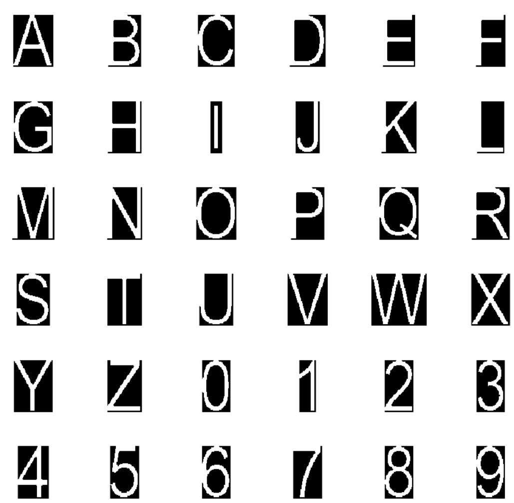

Figure 8: Template Library

```
target=imread('Car_2bw.png');
%calculate the information of foreground
state=regionprops(target,'BoundingBox');
%obtain the previouse position information
```

```
position=cat(1,state.BoundingBox);

%cropping the image into single characters
target_1=imcrop(target,position(5,:));
target_2=imcrop(target,position(6,:));
target_3=imcrop(target,position(7,:));
target_4=imcrop(target,position(8,:));
target_5=imcrop(target,position(9,:));
target_6=imcrop(target,position(10,:));
target_7=imcrop(target,position(11,:));
target_8=imcrop(target,position(12,:));

%save all the characters into one cell
Target=cell(8,1);
%save the cell
Target{1,1}=target_1;
Target{2,1}=target_2;
Target{3,1}=target_3;
Target{4,1}=target_4;
Target{5,1}=target_5;
Target{6,1}=target_6;
Target{7,1}=target_7;
Target{8,1}=target_8;
%display the results after cropping
for i=1:8
    subplot(2,4,i),imshow(Target{i,1});
end
```

```
a=imread('Al_2bw.png');
state=regionprops(a,'BoundingBox');
position=cat(1,state.BoundingBox);

%order the template
Tem_A=imcrop(a,position(2,:));
Tem_0=imcrop(a,position(3,:));
Tem_5=imcrop(a,position(4,:));
Tem_K=imcrop(a,position(5,:));
Tem_U=imcrop(a,position(6,:));
Tem_6=imcrop(a,position(7,:));
Tem_1=imcrop(a,position(8,:));
Tem_B=imcrop(a,position(9,:));
Tem_V=imcrop(a,position(10,:));
```

```
Tem_L=imcrop(a,position(11,:));
Tem_W=imcrop(a,position(12,:));
Tem_2=imcrop(a,position(13,:));
Tem_7=imcrop(a,position(14,:));
Tem_M=imcrop(a,position(15,:));
Tem_C=imcrop(a,position(16,:));
Tem_3=imcrop(a,position(17,:));
Tem_8=imcrop(a,position(18,:));
Tem_X=imcrop(a,position(19,:));
Tem_D=imcrop(a,position(20,:));
Tem_N=imcrop(a,position(21,:));
Tem_4=imcrop(a,position(22,:));
Tem_9=imcrop(a,position(23,:));
Tem_Y=imcrop(a,position(24,:));
Tem_O=imcrop(a,position(25,:));
Tem_E=imcrop(a,position(26,:));
Tem_Z=imcrop(a,position(27,:));
Tem_F=imcrop(a,position(28,:));
Tem_P=imcrop(a,position(29,:));
Tem_G=imcrop(a,position(30,:));
Tem_Q=imcrop(a,position(31,:));
Tem_H=imcrop(a,position(32,:));
Tem_R=imcrop(a,position(33,:));
Tem_S=imcrop(a,position(34,:));
Tem_I=imcrop(a,position(35,:));
Tem_J=imcrop(a,position(36,:));
Tem_T=imcrop(a,position(37,:));

%save into one cell
Tem=cell(36,1);

Tem{1,1}=Tem_A;
Tem{2,1}=Tem_B;
Tem{3,1}=Tem_C;
Tem{4,1}=Tem_D;
Tem{5,1}=Tem_E;
Tem{6,1}=Tem_F;
Tem{7,1}=Tem_G;
Tem{8,1}=Tem_H;
Tem{9,1}=Tem_I;
Tem{10,1}=Tem_J;
Tem{11,1}=Tem_K;
Tem{12,1}=Tem_L;
```

```
Tem{13,1}=Tem_M;
Tem{14,1}=Tem_N;
Tem{15,1}=Tem_O;
Tem{16,1}=Tem_P;
Tem{17,1}=Tem_Q;
Tem{18,1}=Tem_R;
Tem{19,1}=Tem_S;
Tem{20,1}=Tem_T;
Tem{21,1}=Tem_U;
Tem{22,1}=Tem_V;
Tem{23,1}=Tem_W;
Tem{24,1}=Tem_X;
Tem{25,1}=Tem_Y;
Tem{26,1}=Tem_Z;

Tem{27,1}=Tem_0;
Tem{28,1}=Tem_1;
Tem{29,1}=Tem_2;
Tem{30,1}=Tem_3;
Tem{31,1}=Tem_4;
Tem{32,1}=Tem_5;
Tem{33,1}=Tem_6;
Tem{34,1}=Tem_7;
Tem{35,1}=Tem_8;
Tem{36,1}=Tem_9;
%display them
for i=1:36
    subplot(6,6,i),imshow(Tem{i,1});
end
```

2.3 Detection by Erosion

In this task, we should use erosion to detect the car license using the previous images, which have been cropped.

- This function should have ability to erode the target image with template images one by one, until the results of erosion is foreground. In another word, if the results of erosion is white, which means the character of target is the same of the character of template.
- However, we should do some significant operations to the images of template, because the shape and size if template is not suitable, or there will be some mismatch in this function.
- Thus, we did a erosion to the template images before detection, but the structuring elements is not very big, just removing some pixels from the boundaries from the template images.

The results of this erosion will be shown here:
EIEHIJJC
Which is not match the original car license **EKFH1QOC**. Then this report will provide some reasons for this phenomenon:

(1) We observe the results, it can be found that there are some mismatches after the erosion. Because the erosion can only remove some pixels from the boundaries from the objects.

(2) If the characters have the similar shape. For example, **O** and **Q** has the similar shape, and the erosion may make a mistake between **Q** and **O**.

(3) The erosion can only find somewhere coincidence. In another word, if the two letter have somewhere coincidence after the erosion. Then, this function could make a wrong recolonization here. That's the reason why the erosion can not perform a 100% recognition.

```
%the two input should be the original photo
function [str] = detect_car_license_plate_v1(im_car,im_template)
%call the crop function
target=car_order(im_car);
template=template_word_order(im_template);
%define a new matrix
output=zeros(8,1);
%define a new chart
```

```
word=char(['A':'Z' '0':'9']);
for i=1:8
    for j=1:36
       [L,N]=bwlabel(imerode(target{i,1},template{j,1}));
        if(N)
            output(i)=word(j);
            break;
        end
    end
end
%output the results
fprintf('%s',output);
str=output;
```

2.4 Hit-Miss Detection

In this task, we should use the hit-miss filter to replace the previous erosion. The hit-miss transform can look for particular patterns of foreground and background pixels, so we should first inverse the background and foreground of the template, the results will be shown here:

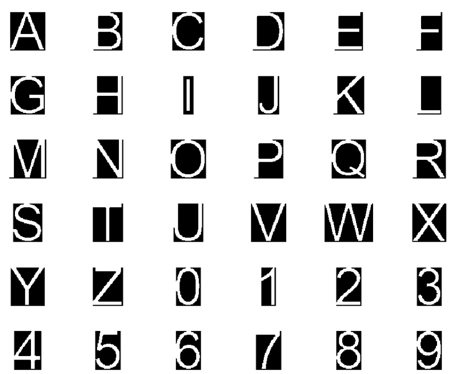

Figure 9: Original Template

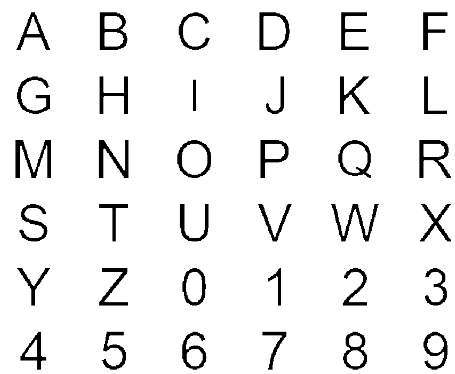

Figure 10: Inverse Template

The figure10 shows the results of inverse template, it can be found that the background and foreground has been exchanged. Additional, we should operate another **Erosion** to the figure10. The procedure of how to apply the hit-miss filter for detection:

- After we obtain the target image and template image, which should be cropped into single letters.
- Apply the inverse change to the template image. This operation will change the foreground to background, the background to foreground of the template image.
- Perform the erosion to both template and inverse-template image. However, the value of SE in inverse should be larger than the original one.
- Apply the hit-miss to the target, using the previous two structuring elements, and then obtain the results.
- The results of hit-miss detection:
  KFH1QOC

Then we observe the results of erosion and hit-miss, we can find that the performance of hit-miss was better than erosion. Then, this report will analysis the reasons of this phenomenon:

(1) The hit-miss not only perform on foreground, but also perform on background.

(2) performs the hit-miss operation defined by the structuring elements SE1 and SE2. The SE1 is the original template image, and the SE2 represents the inverse template image. Then he hit-miss operation preserves pixels in binary image BW whose neighborhoods match the shape of SE1 and don't match the shape of SE2. In another word, it apply **OR** operations to combine the results.

(3) The structuring elements, which should be SE1 and SE2 will representing four corners. In another word, it apply **OR** operations to combine the results.

```
%The two input should be the original images
function str = detect_car_license_plate_v2(im_car,im_template)
%obtain the cropped car images
target=car_order(im_car);
%obtain the cropped template images
template=template_word_order(im_template);
%define a new matrix
output=zeros(8,1);
word=char(['A':'Z' '0':'9']);
%apply the inverse to the foreground
for k=1:36
    template_in{k,1}=¬template{k,1};
end
%apply erosion to the background
for i=1:36
    template_in_back{i,1}=imerode(template_in{i,1},strel('disk',3));
end
%define a new chart
for i=1:8
    for j=1:36
       [¬,N]=bwlabel(bwhitmiss(target{i,1},template{j,1},template_in_back{j,1}));
        if(N)
            output(i)=word(j);
            break;
        end
    end
end
%output the results
str=output;
fprintf('%s',str);
```

3 Conclusion

To conclude, this task focus on the morphological operations, including erosion, dilation, open, and close. The structuring element plays a significant role on those operations. The shape and value of structuring element will make a difference to the morphological operations.

- In the task1, we focus on the close and erosion to a RGB image, in order to count the number of coins. According to this task, we knew that the close can remove the small holes in the foreground, and the erosion can remove the pixels from the boundaries from the object. Then we can use the algorithm to count the number of the coins.

- The task2 concentrate on the pattern recognition by using erosion and hit-miss. According to the results of this task, we can find that the accuracy of recognition on erosion is not high, only 50%. Because the erosion only use one structuring element to do one **AND** operation. If the two letter are similar, it will make a mistake.

- The hit-miss using two different SE, foreground and background, which will increase the accuracy of recognition. Because the hit-miss will do **OR** operation to combine the results. Thus, the hit-miss can perform a better performance for this recognition.