



DIP Project Report

License plate detection and recognition

Xu Kangyan & Sun Kaiwen

December 2017

About the project:

Target: detect license plate and recognize the number.

Simple case:

No detection, directly segment the numbers and make recognition.

Simple background, clear license plate, no distortion.

Hard case:

Detect the license plate firstly from a vehicle.

Transformation and distortion.

1. Motivation

Teacher has introduced us some projects on the class. Among those projects, we are more interested in license plate recognition, and we also think it's not too difficult to design. As a result, we choose this theme as our DIP project.

2. Methodology

This part will display the important parts of our codes, after that the methodology we used will be explained.

```
I=imread ('8.jpg');  
figure(1),subplot(4,2,1),imshow(I);title('The original License');  
  
Ig=rgb2gray(I);  
figure(1),subplot(4,2,2),imshow(Ig);title('grey image');  
  
[d,c]=size(Ig);% row:d , column:c  
Ig=imcrop(Ig,[c/40 d/7 153*c/160 5*d/7]); % no change  
figure(1),subplot(4,2,3),imshow(Ig);title('delete frame');
```

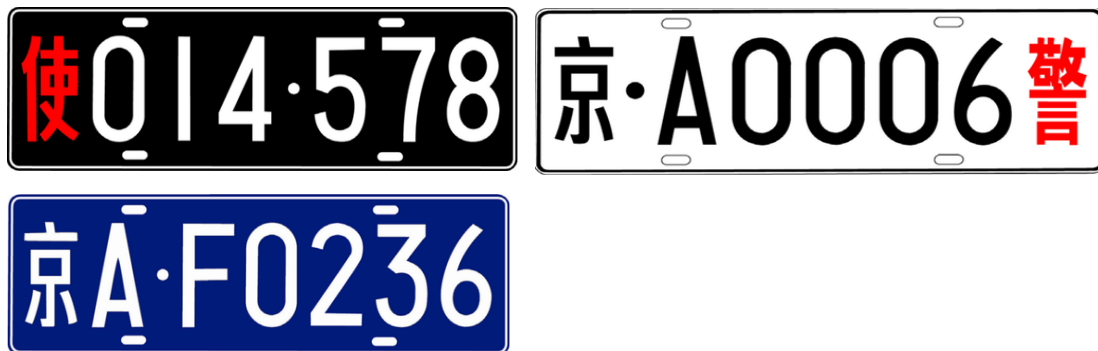
In this part, we first changed the image into gray image, then removed the frame of the plate.

```

%black=0(0),white=1(256)
if Ig(1,1)>125 % avoid different background problem
Ig=im2bw(Ig,0.4);
Ig=~Ig;
else
Ig=im2bw(Ig,0.29); % 0.29 no change
end
figure(1),subplot(4,2,4),imshow(Ig);title('binaryzation');

```

This part is used to distinguish different type of license plate. The parameter is decided by experimenting.



These three plates are examples.

```

SE1=ones(2,1); % (2,1) no change
Ig=imerode(Ig,SE1);
figure(1),subplot(4,2,5),imshow(Ig);title('erosion');

```

This part use erosion method. The size of the square 'SE1' is determined by many tests.

```

SE2=ones(3,2); % (3,2) no change
Ig=imdilate(Ig,SE2);
figure(1),subplot(4,2,7),imshow(Ig);title('dilation');

```

This part use dilation method. The size of the square 'SE2' is also determined by many tests.

**All of the codes above are used to divide the words on the plate.
After several operations, we get images of independent letters.**

```

DD1=imresize(c1,[64 32],'nearest');
t=0;
for m=35:70
    fname = strcat('sample/',num2str(m),'.jpg');
    ADD2 = im2bw(imread(fname));
    ADD2 = imresize(ADD2,[64 32],'nearest');
    for i=1:64
        for j=1:32
            ADD(i,j)= xor(ADD1(i,j),ADD2(i,j));

        end
    end
    P(1,m)=sum(sum(ADD));
    if(P(1,m)>t)
        t=P(1,m);
        k(1,1)=m;
    end
end
end

```

This part is used to recognize the word. We compared the image we divided with those in the library. The specific method is to add the white bottom black word image we have divided with the black bottom white word images in the library. We used the exclusiveOR (xor) logic — that is to say only when the two parts are different, we will get a “1”, otherwise we will get “0”. As a result, we know the largest sum is the word we should choose.

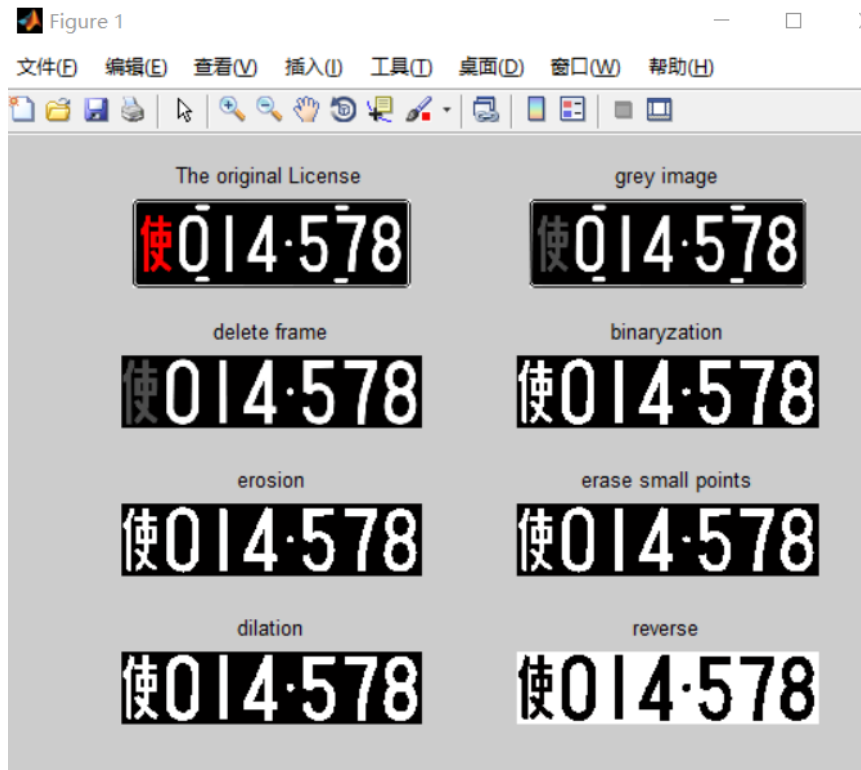


The library we used

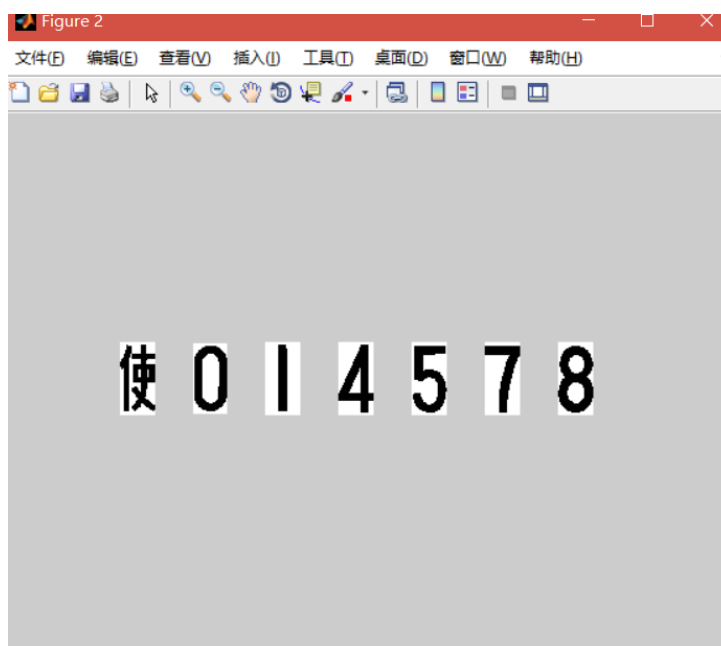
3. Results and analysis

Example 1:

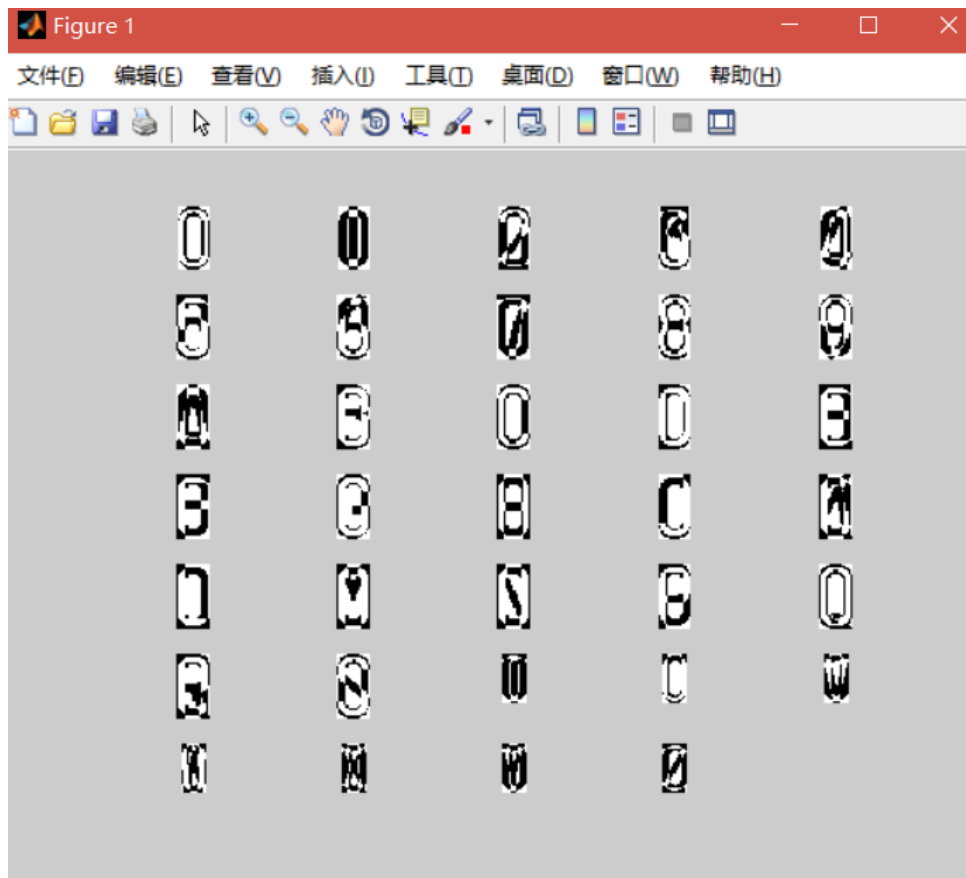
STEP1: The process before dividing, each process is titled.



STEP2: The letters are divided.



STEP3: The process of recognizing. This is an example of recognizing the second letter —— the letter '0'. (Next example is the second letter 'Z'). The sum image is the 'whitest' when it is added with the '0' image in the library.



STEP4: The result of recognizing.

```
命令窗口
>> project3

RESULT =

fx 使014578
```

Example 2:

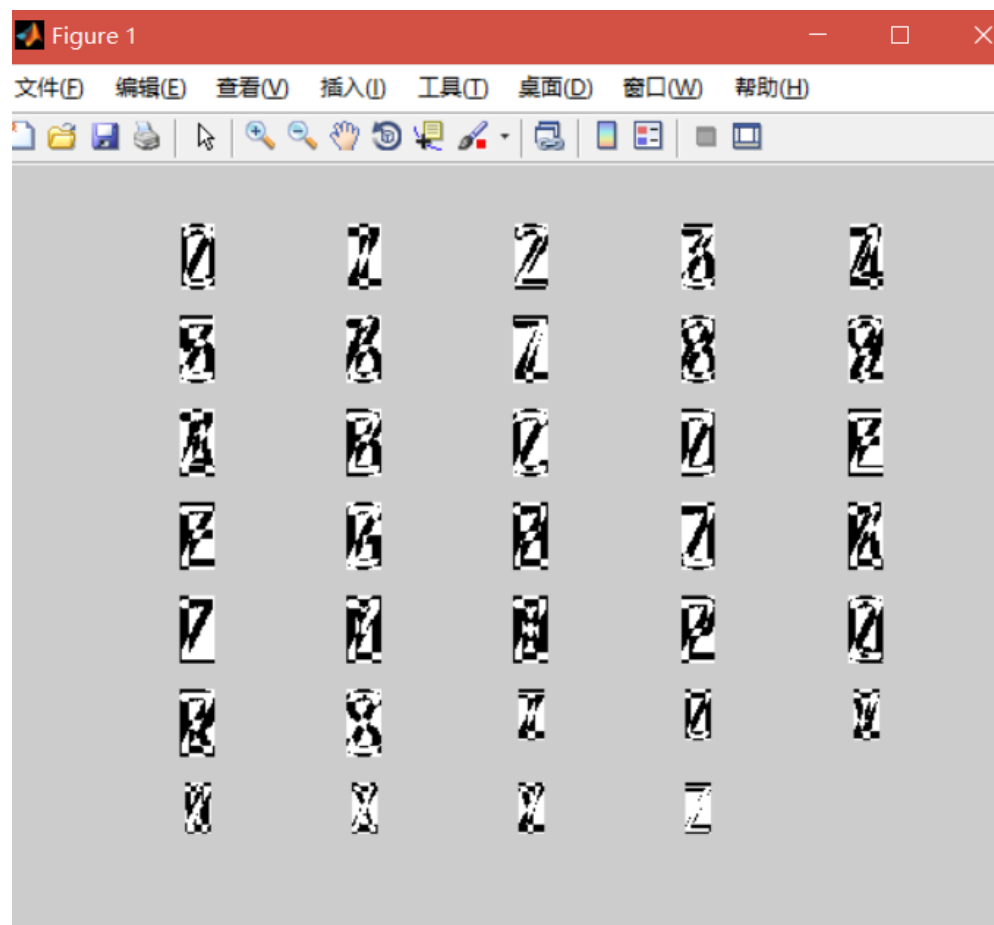
STEP1:



STEP2:



STEP3:



STEP4:

```
>> project3
```

RESULT =

粤ZF023澳

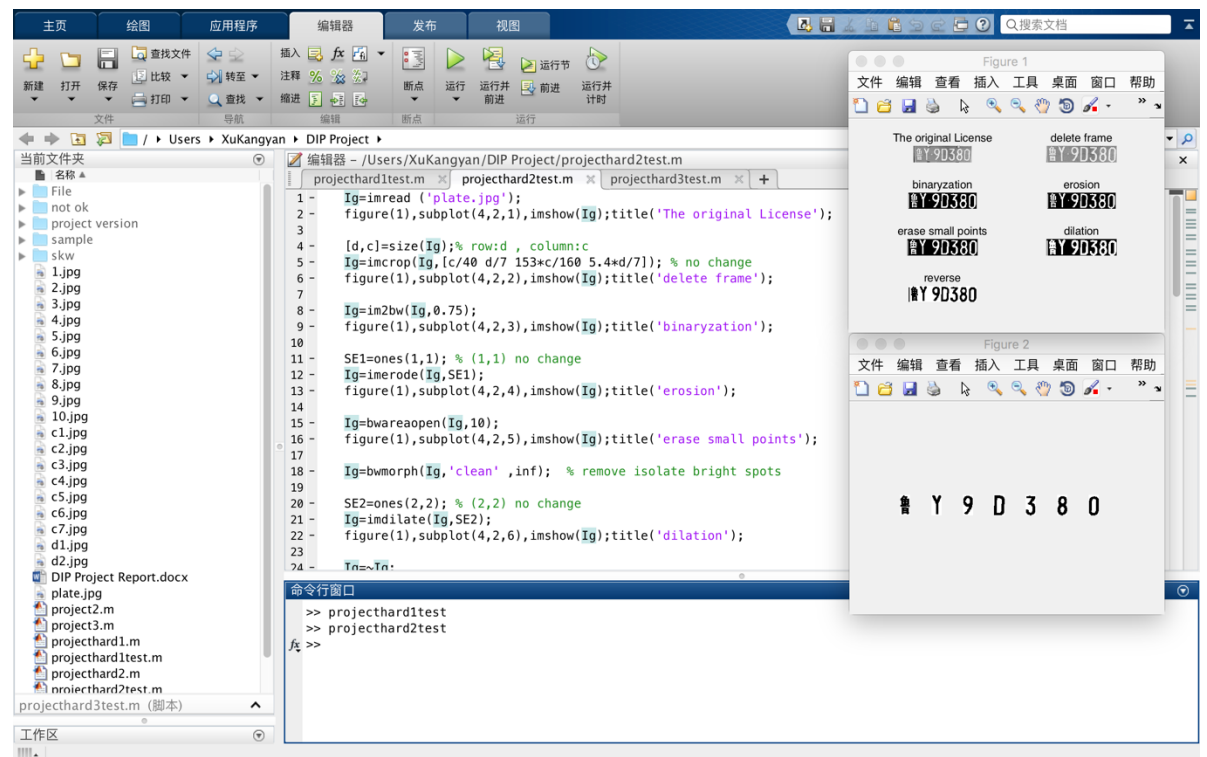
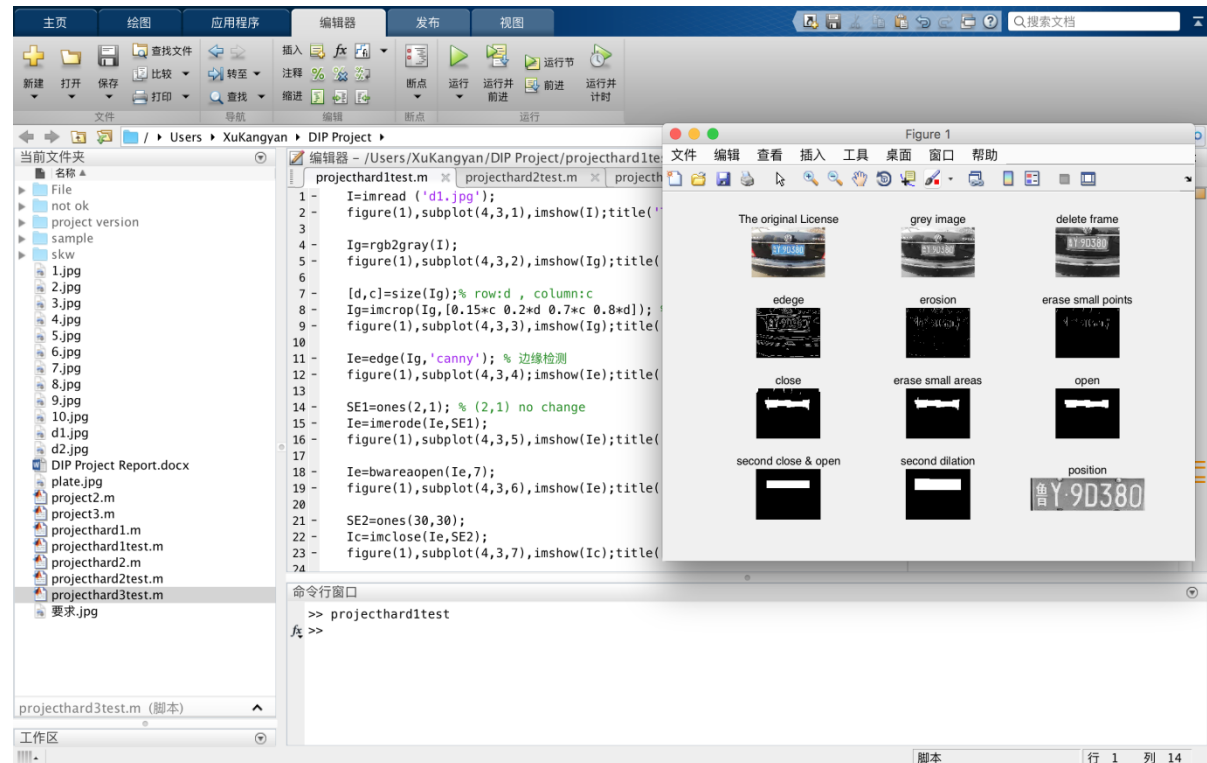
4. Difficulties We Meet

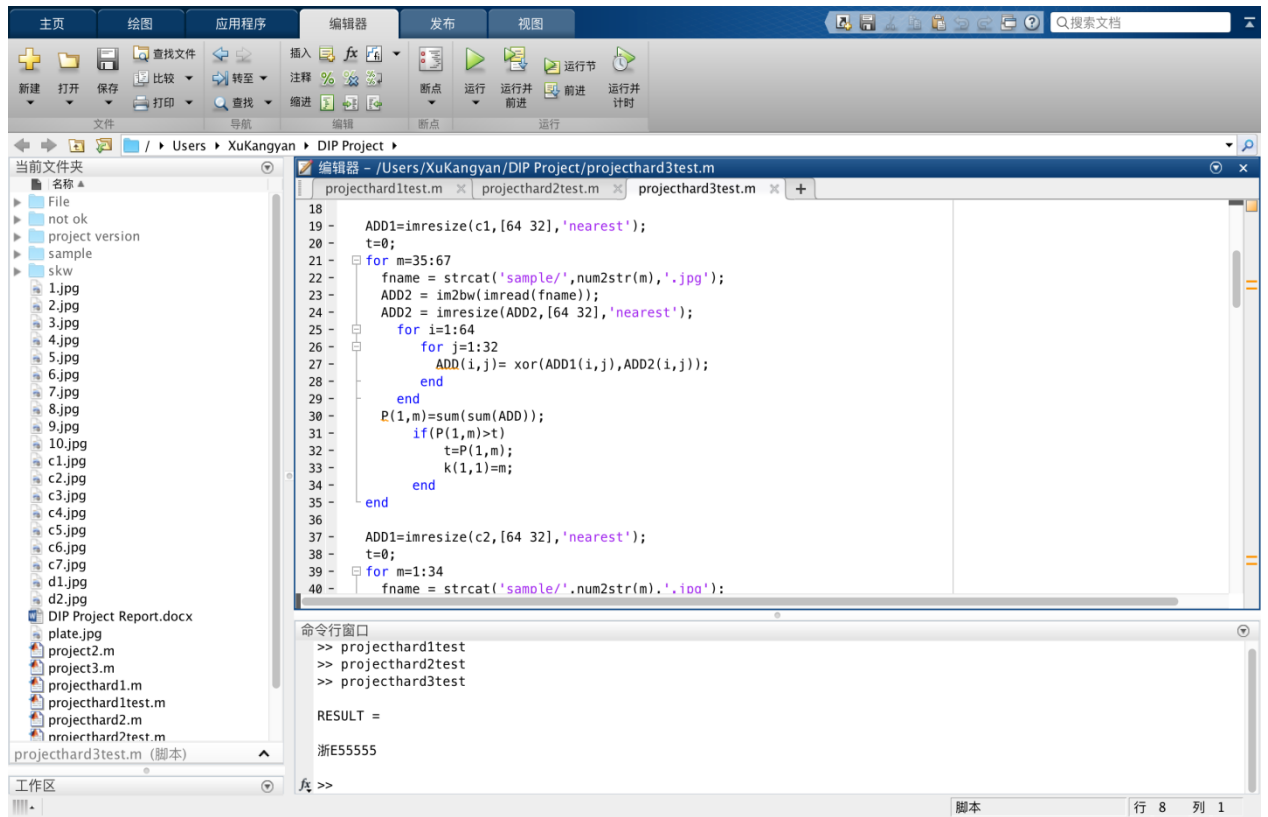
The most serious difficulty we meet is how to tell the difference between '0', 'D' and 'U'. At first the accuracy is not very good. What made us more confused was that we can't find the errors in our codes. Finally, after we checked the parameters of the gray matrix and displayed the image of the sum, we found that the error was in our library — we don't use standard images. So, we modify some images and then solved the problem.

Another difficulty we meet is that some words are too thin — they are more easily to be deleted than the frame. Finally, we deleted the frame first and then solved the problem.

5.Hard Mode

In the hard mode, we want to recognize the plate in more common conditions, such as recognizing a plate from a picture of a car. Unfortunately, though some of our samples are successful, many of them failed. Here is a successful sample.





The core methods are similar to the methods we used in the easy mode, but the conditions required are more severe. If the picture is “too” leap, our code will fail to recognize. (Even though we don’t think the plate is leap, our code thinks it is.)

After all, the code of the hard mode based on the assumption that the plate is the most complicated part of the picture, so we can locate it and then process.