ECE415 -- Homework 3

Name: Wang Yunjuan; UIN:661180568

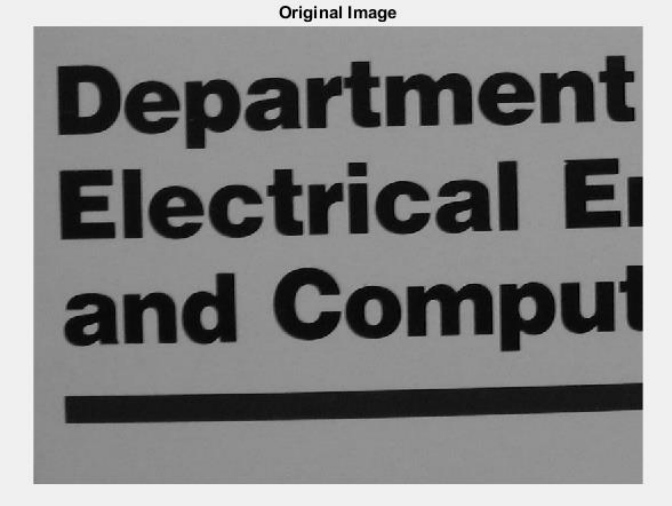
Problem 1

Load the grayscale image Text.bmp into Matlab

1) Display the image

Load the grayscale image Image.bmp into MATLAB and display the image. The picture is shown as follows.

```
im=imread('Text.bmp');
imshow(im);
```



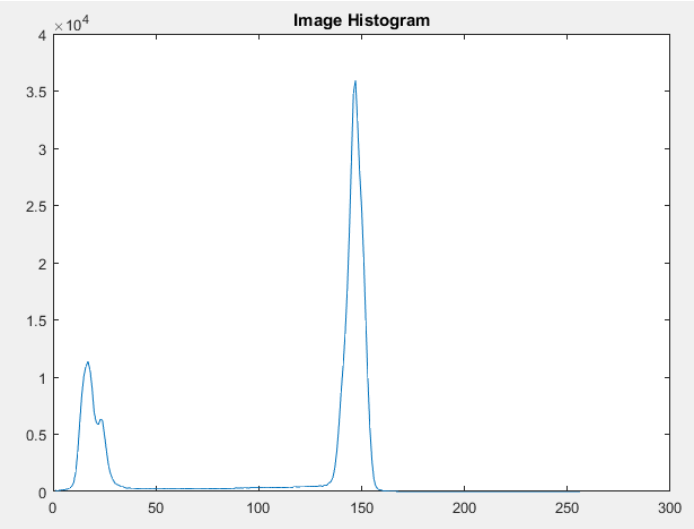
2) Plot a histogram of image pixel intensities.

I first write a function *histogram* to calculate the histogram for the intensity image, which can implement the built-in function of *imhist*. The MATLAB code is as follows.

```
function y=histogram(I)
[row col]=size(I);
h=zeros(1,256);
for i=1:row
    for j=1:col
        h(I(i,j)+1)=h(I(i,j)+1)+1;
    end
end
y=h;
end
```

Then I use function *histogram* I have write above to display the histogram of picture. We can easily find a valley in this curve. We roughly evaluate the pixel value of this valley as 40 and choose 40 as

our threshold. The result is a binary image. We display the original image and the image after thresholding together to make a comparation.



3) Perform thresholding on the image to create a binary image. Describe the method and threshold used. Display the binary image.

We can easily find a valley in this curve in (2). We roughly evaluate the pixel value of this valley as 100 and choose 100 as our threshold.

MATLAB code:

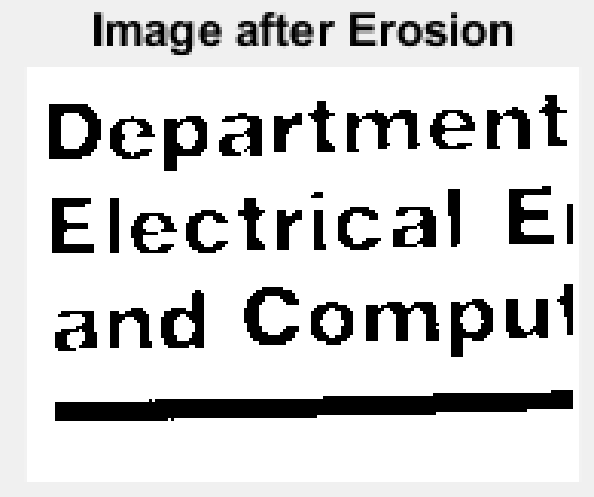
```
clear;clc;
im=imread('Text.bmp');
[row, col]=size(im);
for i=1:row
    for j=1:col
        if im(i,j)<100
            output(i,j)=0;
        else
            output(i,j)=255;
```

```
        end
    end
end
output=uint8(output);
h2=histogram(output);
imshow(output);
title('Image after Thresholding');
```

4) Apply 5 times the erosion morphological operator on the binary image from part (3). Display the resulting image. What size kernel did you use?

I use 3×3 kernel.

```
[0 1 0
 1 1 1
 0 1 0]
```



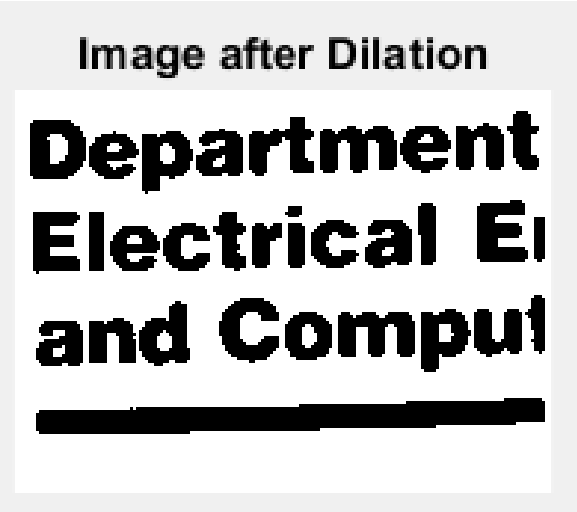
MATLAB code:

```
clear;clc;
im=imread('Text.bmp');
[row, col]=size(im);
for i=1:row
    for j=1:col
        if im(i,j)<100
            output(i,j)=0;
        else
            output(i,j)=255;
        end
    end
end
output=uint8(output);
eout=[255*ones(1,col);output;255*ones(1,col)];
eout=[255*ones(row+2,1) eout 255*ones(row+2,1)];
kernel=[0 1 0
        1 1 1
        0 1 0];
```

```
[krow,kcol]=size(kernel);
for m=1:5
    for i=1:row-2
        for j=1:col-2
            sum=0;
            for k=1:krow
                for l=1:kcol
                    if eout(i-1+k,j-1+l)==0
                        sum=sum+1*kernel(4-k,4-l);
                    end
                end
            end
            if sum>=5
                eout(i,j)=0;
            else eout(i,j)=255;
            end
        end
    end
end
eout=eout(1:row,1:col);
eout=uint8(eout);
imshow(eout);
title('Image after Erosion');
```

5) Apply 5 times dilation morphological operator on image obtained in (4). Display the resulting image. What size kernel did you use?

I use 3×3 kernel $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, same kernel as (4) to apply the dilation.



MATLAB code (part):

```
dout=[255*ones(1,col);eout;255*ones(1,col)];
```

```
dout=[255*ones(row+2,1) dout 255*ones(row+2,1)];
dkernel=[0 1 0
         1 1 1
         0 1 0];
[drow,dcol]=size(dkernel);
for m=1:5
    for i=1:row
        for j=1:col
            sum=0;
            for k=1:drow
                for l=1:dcol
                    if dout(i-1+k,j-1+l)==0
                        sum=sum+1*dkernel(4-k,4-l);
                    end
                end
            end
            if sum>=1
                dout(i,j)=0;
            else dout(i,j)=255;
            end
        end
    end
end
dout=dout(1:row,1:col);
dout=uint8(dout);
imshow(dout);
title('Image after Dilation');
```

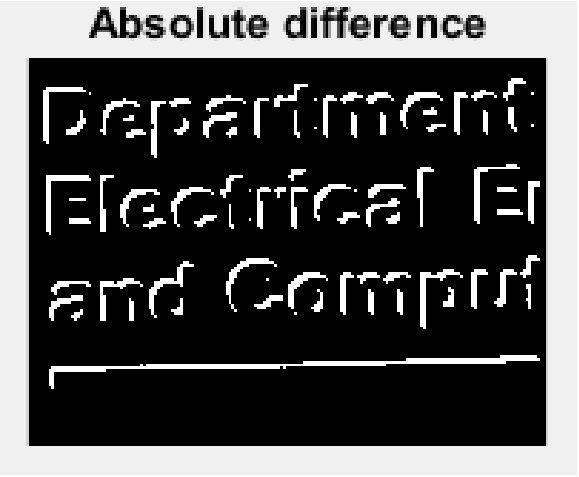
6) Are the image obtained in (5) and the image in (3) identical? Comment!

Image obtained in (5) is similar but not identical to the image in (3). The border of image (5) has more rough edges than image (3).

7) Find the absolute difference of the image created in (3) and image created in (5). Map this difference to full dynamic range of 8 bits and display. Comment!

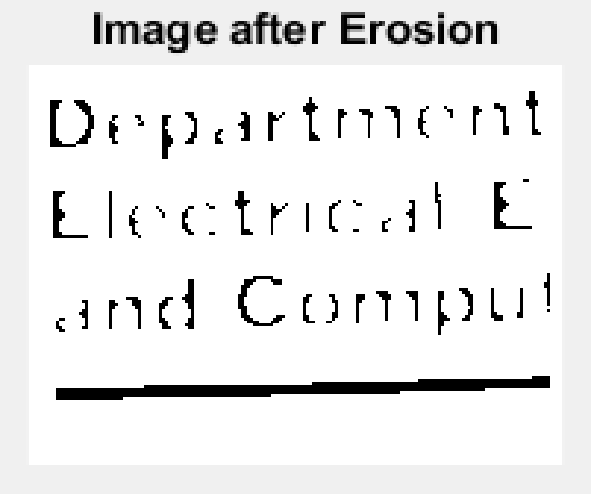
MATLAB code:

```
diff=zeros(row,col);
diff=abs(dout-output);
diff=uint8(diff);
imshow(diff);
title('Absolute difference');
```

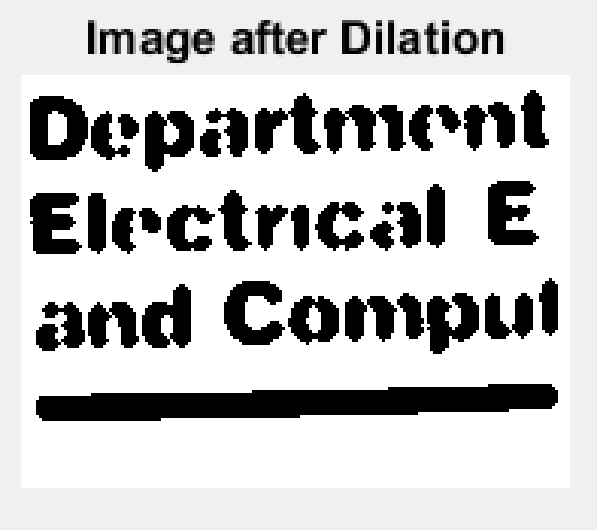


We can find that after erosion and dilation, the text is a little bit shifted down to the bottom right.

8) **Using the same kernel size, apply 10 times the erosion morphological operator on the binary image from part (3). Display the resulting image.**



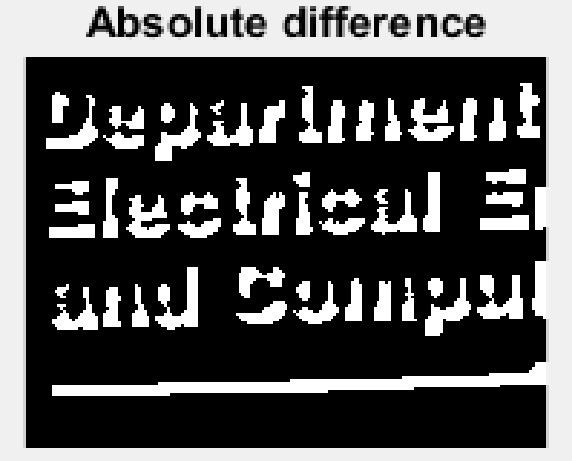
9) **Using the same kernel size, apply 10 times dilation morphological operator on image obtained in (8). Display the resulting image.**



10) **Are the image obtained in (9) and the image in (3) identical? Are the images obtained in (5) and (9) identical? Comment!**

No, they are similar but not identical because of the kernel we choose. The text after erosion and dilation has more white space on the horizontal lines and vertical lines.

11) Find the absolute difference of the image created in (3) and image created in (9). Map this difference to full dynamic range of 8 bits and display. Comment!



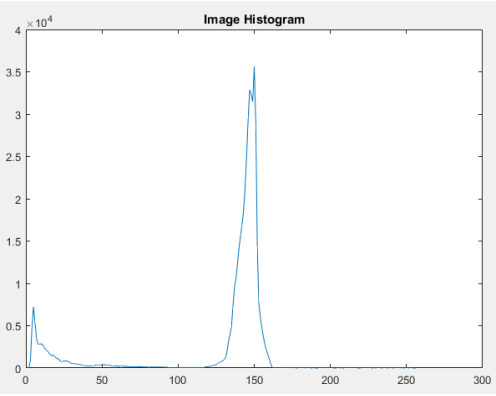
We can find that after erosion and dilation, the text is a little bit shifted down to the bottom right.

Problem 2

1) Display the image.

2) Plot a histogram of image pixel intensities.

I use the histogram function I write, which has mentioned in Problem 1.



3) Perform thresholding to create a binary image. Describe the method and threshold used. Display the image.

We can easily find a valley in this curve in (2). We roughly evaluate the pixel value of this valley as 100 and choose 100 as our threshold.

4) Process the image to create solid foreground object without any "holes". Describe the exact approach used. Display the image

I first dilate the image 10 times, then I erode the image 10 times. Both of these kernels I use are the same as *Problem 1* 3×3 kernel.

```
[0 1 0
 1 1 1
 0 1 0]
```

MATLAB code:

```
clear;clc;
im=imread('bottle.bmp');
[row, col]=size(im);
%% Threshold
for i=1:row
    for j=1:col
```

```
        if im(i,j)<100
            output(i,j)=0;
        else
            output(i,j)=255;
        end
    end
end
output=uint8(output);
%% Dilation
dout=[255*ones(1,col);output;255*ones(1,col)];
dout=[255*ones(row+2,1) dout 255*ones(row+2,1)];
dkernel=[0 1 0
        1 1 1
        0 1 0];
[drow,dcol]=size(dkernel);
for m=1:10
    for i=1:row
        for j=1:col
            sum=0;
            for k=1:drow
                for l=1:dcol
                    if dout(i-1+k,j-1+l)==0
                        sum=sum+1*dkernel(4-k,4-l);
                    end
                end
            end
            if sum>=1
                dout(i,j)=0;
            else dout(i,j)=255;
            end
        end
    end
end
dout=dout(1:row,1:col);
dout=uint8(dout);
%% Erosion
eout=[255*ones(1,col);dout;255*ones(1,col)];
eout=[255*ones(row+2,1) eout 255*ones(row+2,1)];
kernel=[0 1 0
        1 1 1
        0 1 0];
[krow,kcol]=size(kernel);
for m=1:10
    for i=1:row
```

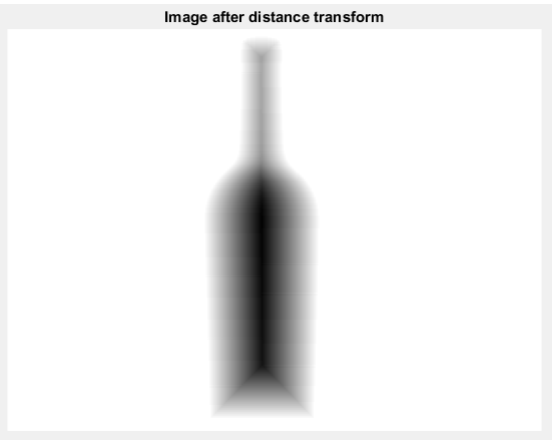
```
        for j=1:col
            sum=0;
            for k=1:krow
                for l=1:kcol
                    if eout(i-1+k,j-1+l)==0
                        sum=sum+1*kernel(4-k,4-l);
                    end
                end
            end
            if sum>=5
                eout(i,j)=0;
            else eout(i,j)=255;
            end
        end
    end
end
eout=eout(1:row,1:col);
eout=uint8(eout);
imshow(eout);
title('Image after Erosion');
```

5) Find the distance transform. What is the maximum value of the distance transform? How many pixels have this value?

The maximum value of the distance transform is 86. There are 47 pixels having this value.

6) Map the values of distance transform to the full dynamic range of 8 bits and display the distance transform. Comment!

We can see that the pixel value is related to the distance between the pixel and the nearest boarder. The more the pixel far from the edge of bottle, the darker the pixel color is.



MATLAB code:

```
%% Two pass algorithm
max=0; % maximum value of the distance transform
count=0;%number of maximum value
disim=[255*ones(1,col);eout;255*ones(1,col)];
disim=[255*ones(row+2,1) disim 255*ones(row+2,1)];
```

```
for i=1:row+2
    for j=1:col+2
        if disim(i,j)==255
            disim(i,j)=0;
        else disim(i,j)=1;
        end
    end
end
for i=2:row+1
    for j=2:col+1
        if disim(i,j)==0
            continue
        else
            disim(i,j)=min(1+disim(i,j-1),1+disim(i-1,j));
        end
    end
end
for i=row+1:-1:2
    for j=col+1:-1:2
        if disim(i,j)==0
            continue
        else
            disim(i,j)=min(min(1+disim(i,j+1),1+disim(i+1,j)),disim(i,j));
            if disim(i,j)>max
                max=disim(i,j);
            end
        end
    end
end
disim=disim(2:row+1,2:col+1);
for i=1:row
    for j=1:col
        if disim(i,j)==max
            count=count+1;
        end
    end
end
disout=disim;
for i=1:row
    for j=1:col
        disout(i,j)=round(255-double(disim(i,j))*255/87);
    end
end
disout=uint8(disout);
```

```
imshow(disout)
```

7) **What are the area, perimeter and centroid of the object in part (4)? How did you compute those?**

The area of the object in part (4) is the number of pixels which is not zero in part (5). There are 72444 pixels in total.

The centroid of the object in part (4) is the centroid of the area of the maximum value of the distance transform. Assume the coordinates of pixel on the top left is (0,0), then the coordinates of centroid is about (289,381).