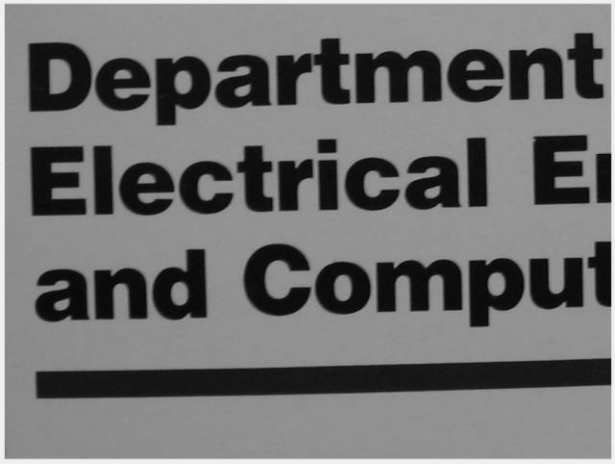
Vishesh Chanana 665319085

COMPUTER VISION

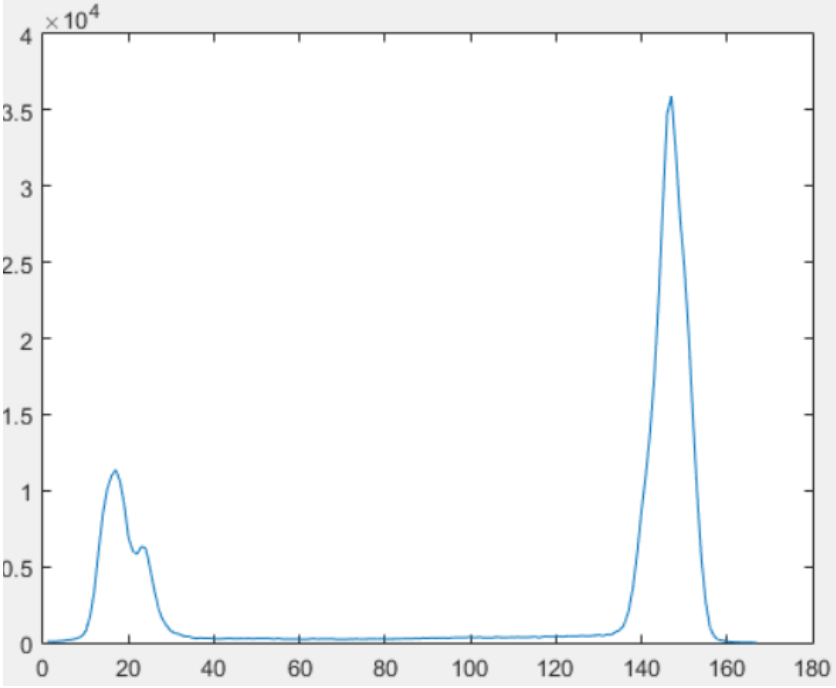
Homework – 3

Answer 1:

Display Image



Histogram



Histogram of an image is basically the graph of all the pixel intensities of the image. It tells us the number of times a particular intensity occurs in the image.

Vishesh Chanana 665319085

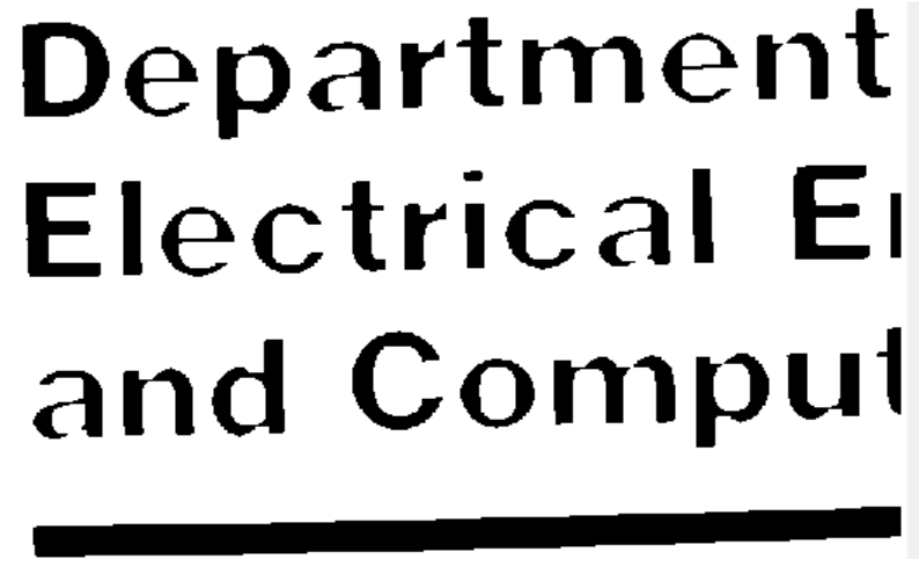
Thresholding

The mean of all the values of the pixels of the image is taken as the threshold of the image. Any pixel that is less than this value is taken as a 0 and any value greater than the threshold is taken as 1. This gives
Threshold value is selected as the mean of all the intensities present in the input image. At the end we get a black and white image.

Erosion

The erosion is a type of morphological operator and it is used to erode away the boundary regions between the foreground and the background.

Kernel Size = 3x3



Dilation

The Dilation operator is used to enlarge the boundaries of the image. In our experiment, we had to dilate an eroded image.

Kernel Size = 3x3

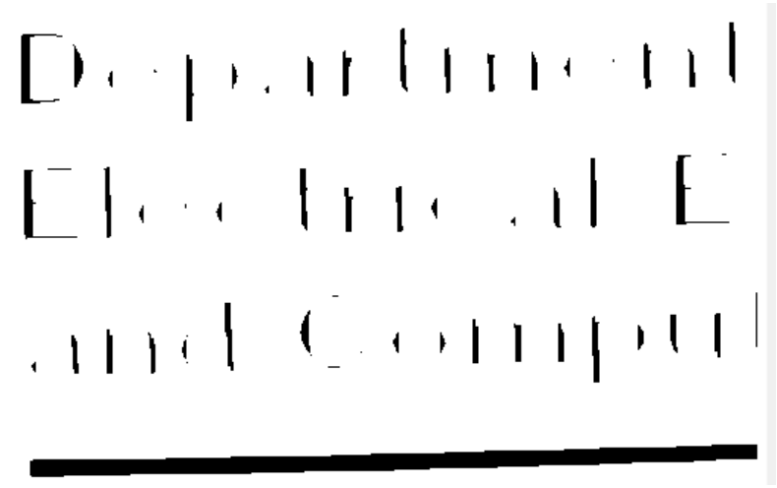
6. The output from the previous step and the original image are almost similar. The only difference between them is the added noise.

7. Now, we have to find the difference between the original image and the dilated image. According to the concept of erosion and dilation, we should get a black image as the output. As, we dilate and erode, some noise is also added inside of our image. The white pixels that are visible in the following image correspond to the noise.

Eroding the image 10 times

As we keep on eroding the image, the lines become thinner and thinner.

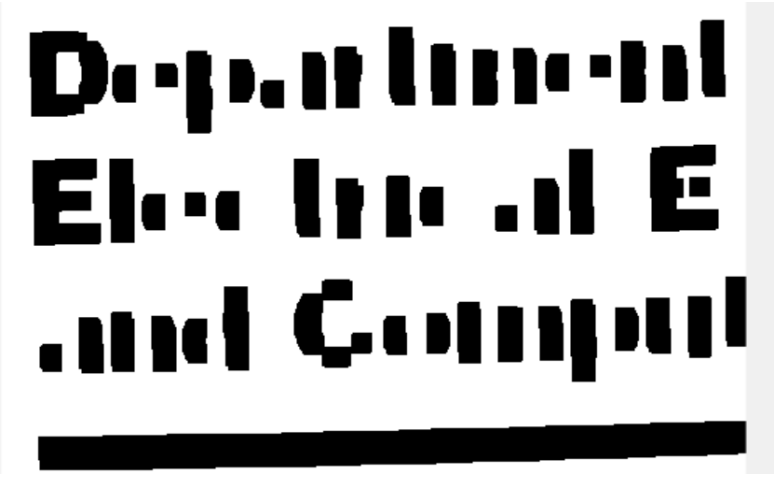
Kernel Size = 3x3



Dilating the image 10 times

Kernel Size = 3x3

We see that we could not retrieve the original image after eroding and dilating 10 times because the added noise is to large and it hampers the process of getting back towards the original image.

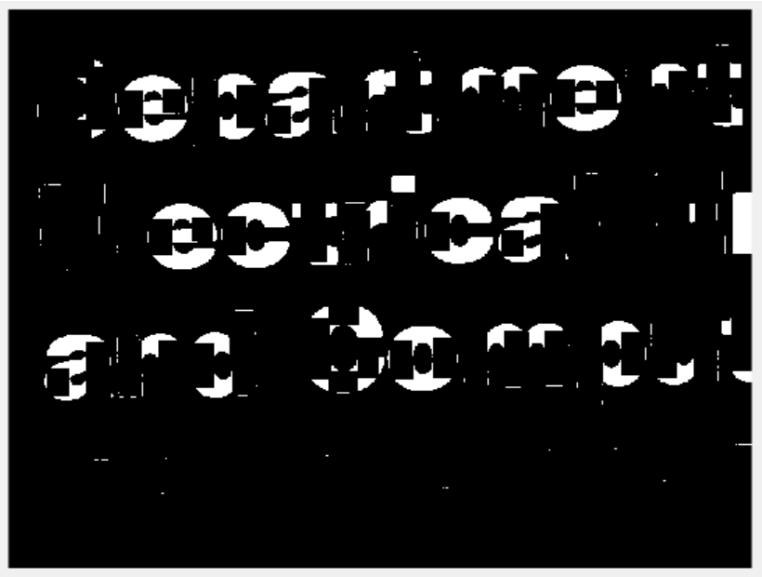


Both the images from 9 and 3 are not same

Both the images from 9 and 5 are not same

Due to the addition of large noise, the images are not same.

Mapping the absolute difference between the output of (9) and the original image to a full dynamic range of 8 bits.



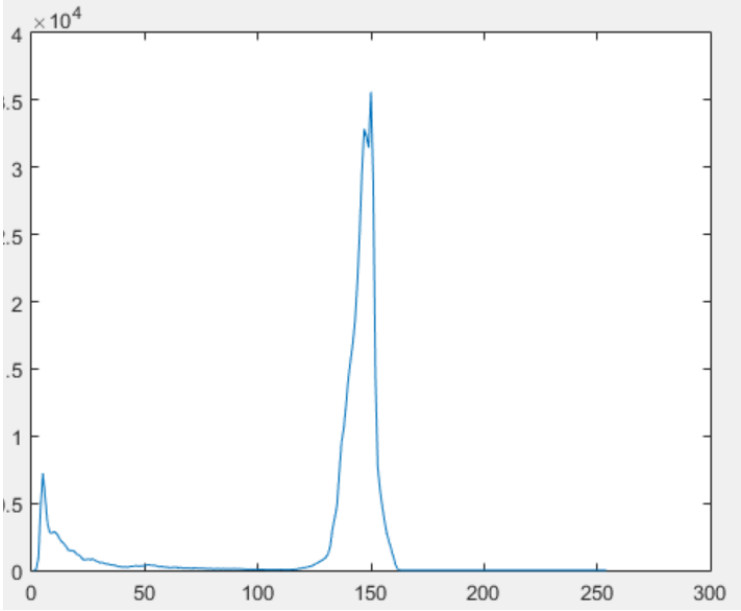
We see some white spots in the absolute difference of the images instead of a full black image that should result from the dilation. As already said, this is due to the added noise while performing the morphological operations on the image.

Answer 2:

Display image

Histogram

Histogram of an image is basically the graph of all the pixel intensities of the image. It tells us the number of times a particular intensity occurs in the image.



Thresholding

The mean of all the values of the pixels of the image is taken as the threshold of the image. Any pixel that is less than this value is taken as a 0 and any value greater than the threshold is taken as 1. This gives

Threshold value is selected as the mean of all the intensities present in the input image. At the end we get a black and white image.

4. Removing the holes from the black and white image

To remove the holes from the black and white image, we use a technique called morphological closing. The morphological closing is used to remove the holes from objects without changing their shape and size. To implement this technique, we need to dilate the image first and then erode it. The dilation first, thickens the boundaries and the erosion, shrinks the boundaries. So, when we apply dilation and erosion once, we see that after the dilation that the holes inside have shrunk a little and when we dilate it the holes get to their original size. So, by hit and trial, when we apply dilation thrice and then erosion thrice, we get the following image. This is because on applying dilation thrice, the holes get closed. When we erode it, we get the boundaries of the image as they were in the original but couldn't revive the holes as the there were no 1s in the image to help the process.

Distance transform

Max value of this distance transform is 87

5 pixels have this maximum value

Mapping of the distance transform to the full dynamic range of 8 bits we get the following image. The distance transform gives us a graylevel output image. In this image the pixel intensities are such that they give the corresponding distance from the background of the image.

Vishesh Chanana 665319085

Area = 74812 pixel^2

Perimeter = 1714 pixels

Centroid = (361.96,408.68)

To find out the area, counted the total number of pixels whose value was greater than or equal to 1 in the distance transformed image.

I found the perimeter by counting the total number of 1s in the image I get after doing the forward and backward transform. I did this because, when we calculate the distance transforms only the outer layer of the image has al 1s in it. The rest either have zeros or a higher value.

I found out the centroid by adding all the indexes of the pixels having a pixel intensity of more than one and then dividing it by 600 for x co-ordinate and 800 for y co-ordinate.

Code:

Answer 1

```
clc;
clear all;
original_image = imread('text.bmp');
%imshow(original_image);

max_intensity = max(max(original_image));
histo = zeros(1,max_intensity+1);

%Histogram creation
for i = 1:600
    for j = 1:800
        temp = original_image(i,j);
        histo(1,temp+1) = histo(1,temp+1) + 1;
    end
end
%plot(histo);

%Thresholding

meanIntensity = mean(mean(original_image));
for i = 1:600
    for j = 1:800
        if original_image(i,j) >= meanIntensity
            tempImage(i,j) = 1;
        else
            tempImage(i,j) = 0;
        end
     end
end
imshow(tempImage);

o = original_image;
x = tempImage;
original_image = tempImage;
structuring_element = [0 0 0;0 0 0;0 0 0];
temp = original_image;
surrounding_size = 3;
```

```
for z =1:10
temp = padarray(temp,[1,1],1);

    for i = 2:601
    for j = 2:801
        m = 1;
        for k = i-1:i+1
            n= 1;
            for l = j-1:j+1
                surrounding(m,n) = temp(k,l);
                n = n+1;
            end
            m = m+1;
        end
        if(surrounding == structuring_element)
            x(i-1,j-1) = 0;
        else
            x(i-1,j-1) = 1;
        end
    end
    end
    temp = x;
end
eroded_image = temp;
imshow(eroded_image);

structuring_element = [1 1 1;1 1 1;1 1 1];
for z =1:10
temp = padarray(temp,[1,1],1);
    for i = 2:601
    for j = 2:801
        m = 1;
        for k = i-1:i+1
            n= 1;
            for l = j-1:j+1
                surrounding(m,n) = temp(k,l);
                n = n+1;
            end
            m = m+1;
        end
        if(surrounding == structuring_element)
            x(i-1,j-1) = 1;
        else
            x(i-1,j-1) = 0;
        end
    end
    end
    temp = x;
end
dilated_image = temp;

imshow(dilated_image);
difference = (dilated_image - original_image);
max_value = max(max(difference));
final_image = round(difference*(256/max_value+1));
```

Answer 2

```
clc;
clear all;
original_image = imread('bottle.bmp');
%imshow(original_image);

max_intensity = max(max(original_image));
histo = zeros(1,max_intensity+1);

%Histogram creation
for i = 1:600
    for j = 1:800
        temp = original_image(i,j);
        histo(1,temp+1) = histo(1,temp+1) + 1;
    end
end
%plot(histo);

%Thresholding

meanIntensity = mean(mean(original_image));
for i = 1:600
    for j = 1:800
        if original_image(i,j) >= meanIntensity
            tempImage(i,j) = 1;
        else
            tempImage(i,j) = 0;
        end
     end
end

%imshow(tempImage);
o = original_image;
x = tempImage;
original_image = tempImage;
temp = original_image;
surrounding_size = 3;


structuring_element = [1 1 1;1 1 1;1 1 1];
for z =1:3
temp = padarray(temp,[1,1],1);
    for i = 2:601
    for j = 2:801
        m = 1;
        for k = i-1:i+1
            n= 1;
            for l = j-1:j+1
                surrounding(m,n) = temp(k,l);
                n = n+1;
            end
            m = m+1;
        end
        if(surrounding == structuring_element)
            x(i-1,j-1) = 1;
        else
            x(i-1,j-1) = 0;
        end
    end
    end
    temp = x;
```

```
end

structuring_element = [0 0 0;0 0 0;0 0 0];

for z =1:3
temp = padarray(temp,[1,1],1);

    for i = 2:601
    for j = 2:801
        m = 1;
        for k = i-1:i+1
            n= 1;
            for l = j-1:j+1
                surrounding(m,n) = temp(k,l);
                n = n+1;
            end
            m = m+1;
        end
        if(surrounding == structuring_element)
            x(i-1,j-1) = 0;
        else
            x(i-1,j-1) = 1;
        end
    end
    end
    temp = x;
end

%imshow(temp);
for i =1:600
    for j = 1:800
        if(temp(i,j) ==1)
            temp(i,j) = 0;
        else
            temp(i,j) = 1;
        end
    end
end

%imshow(temp);

tempImage = temp;
tempImage = padarray(tempImage,[1,1],0);

forward pass
forward_temp = tempImage;
backward_temp = tempImage;
for i = 2:601
    for j= 2:801
        if(forward_temp(i,j) >= 1)
            forward_temp(i,j) = min((forward_temp(i,j-1)+1),forward_temp(i-1,j)+1);
        end
    end
end

for i = 601:-1:2
    for j = 801:-1:2
        if(backward_temp(i,j) >= 1)
            temp_min = min((backward_temp(i,j+1)+1),(backward_temp(i+1,j)+1));
            backward_temp(i,j) = min(temp_min,forward_temp(i,j));
```

```
        end
    end
end


final = uint8(backward_temp);
imshow(final)
final_image = final(2:601,2:801);

max_value=max(max(final_image));
final = round(final_image*(256/max_value+1));
final=uint8(final);
area = 0;
for i = 1:600
    for j = 1:800
        if(final(i,j)>0)
            area = area +1;
        end
    end
end


count_peri = 0;
for i = 2:601
    for j = 2:802
        if(backward_temp(i,j) == 1)
            count_peri  = count_peri+1;
        end
    end
end

x_sum = 0;
y_sum = 0;
tot_count = 0
for i = 1:600
    for j = 1:800
        if(final(i,j)>=1)
            x_sum = x_sum + i;
            y_sum = y_sum + j;
            tot_count = tot_count +1;
        end
    end
end

cent_x = x_sum/tot_count;
cent_y = y_sum/tot_count;
```