Purvang Lapsiwala

UIN = 662689378

Homework- 4

Answer (1):

(a)

Display of Image

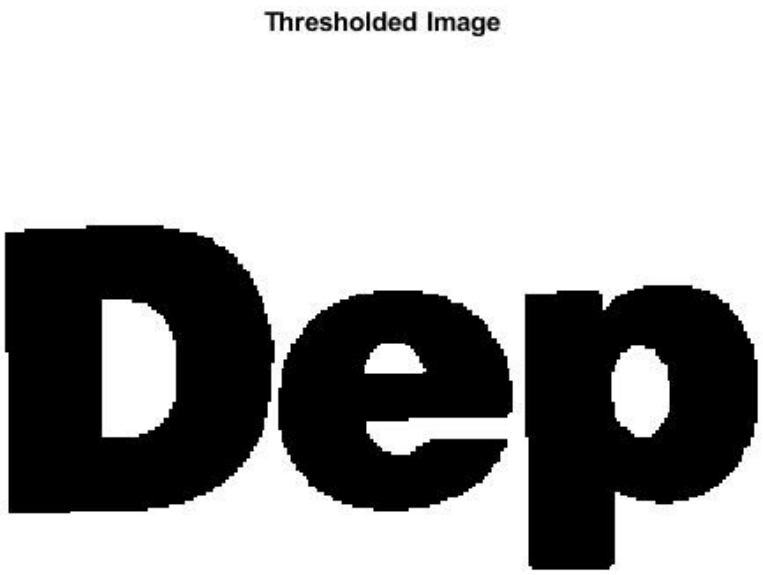
Original Image

(b)

As given image was already black and white image, simple method is followed to do thresholding.

In thresholding, if pixel value is greater than 0, then it is taken as 1 else 0.

Image after thresholding operation



(c)

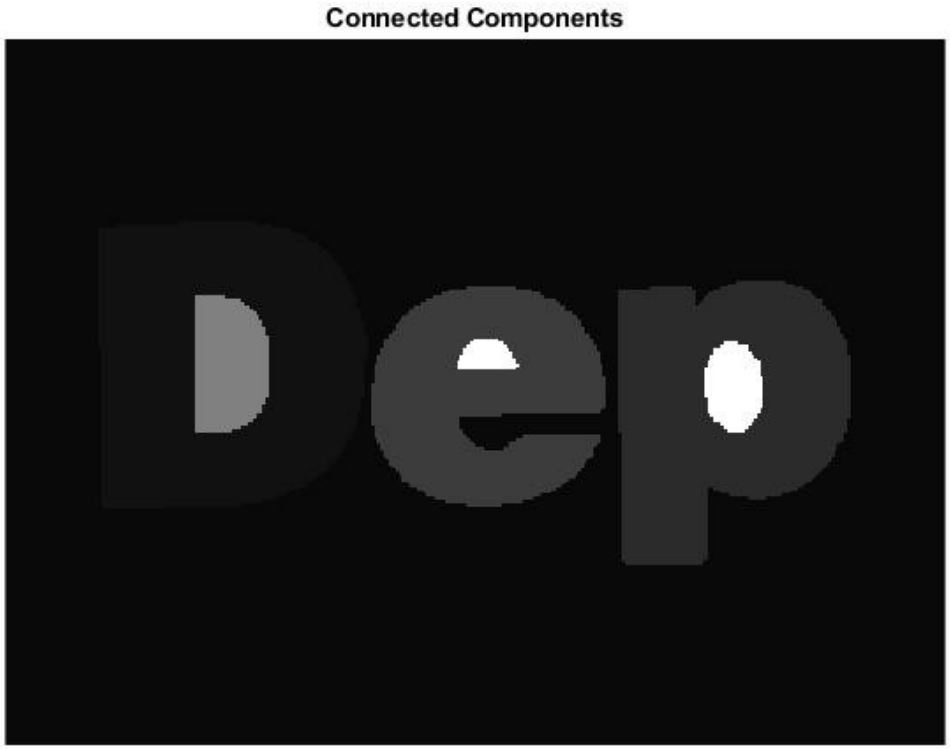
By applying four connectivity algorithm, there are 7 connected components detected in to given image. This also can be verified by visual inspection of given figure. Background of an image is considered as first component, three black letters are considered as 3 separate components and three while holes within each letter are another 3 components, total we have 7 connected components.

Final output of an image after doing second pass and replacing black and white pixel values with connected component value is below.

(d)

Algorithm is performed in two passes.

Algorithm first labels all the components by checking the northside and the westside neighbors. After that it found out all the components that are connected to each other. In the next scan of the image, all the connected components were given the same label. Then, mapping current range to a new range of 0 to 255 is done. Thus, we can see the different shades in the final image. By this way, components or objects can be differentiate by same object ended up with similar label of connection.



Answer (2):

(a)

Display of original image

(b)Gaussian pyramid levels are shown below.

First Level

Second Level

Third Level

Forth Level

Fifth Level

(c)

Images obtained by Laplacian pyramid are shown.

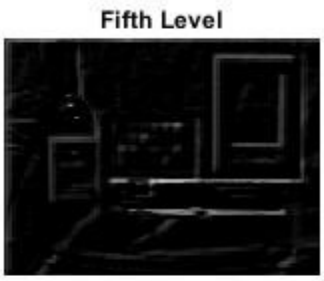
First Level


SIMI
BEOGRAD
COLOR APPEARANCE
MODELS
GOODMAN INTRODUCTION TO FOURIER OPTICS

Second Level

Third Level

Forth Level



(d)

On applying the Gaussian pyramid to given image, final image output is observed to be smooth than original image simultaneously we also reduce resolution to its half in every operation. So, We get the blurred image or lower frequency content image by using a binomial kernel and by reducing the image pixels to its half after every operation performed. In gaussian pyramid, image is subject to repeated smoothing and subsampling

On the other hand, when we apply the Laplacian Kernel, we get only the high frequency content as a result after subtraction. This is done by applying a Binomial Kernel and scaling down the image to half its original value. The Laplacian pyramid can thus be used to represent images as a series of band-pass filtered images, each sampled at successively sparser densities. Each band of the Laplacian pyramid is the difference between two adjacent low-pass images of the Gaussian pyramid.This scaled down image again up sampled to the same ratio and then it is subtracted from the original to get the final output.

Matlab Code:

Answer (1):

```
clc;
close all;
clear all;
original_img = imread('Connected.bmp');
size_of_image = size(original_img);
no_of_rows = size_of_image(1);
no_of_cols = size_of_image(2);
figure(1);
imshow(original_img);
title('Original Image')
label = 1;
num = 1;

for i = 1:no_of_rows
```

```
    for j = 1:no_of_cols
        if(original_img(i,j) > 0)
            threshold_img(i,j) = 1;
        else
            threshold_img(i,j) = original_img(i,j);
        end
    end
end
figure(2);
imshow(threshold_img);
title('Thresholded Image')

for i = 1:no_of_rows
    for j = 1:no_of_cols
        if (i ==1)
            if(j==1)
                inter_img(i,j) = label;
            else
               if((threshold_img(i,j) == threshold_img(i,j-1)))
                   inter_img(i,j) = inter_img(i,j-1);
               else
                   label = label+1;
                   inter_img(i,j) = label;
               end
            end
        else
            if(j==1)
                if(threshold_img(i,j) == threshold_img(i-1,j))
                    inter_img(i,j) = inter_img(i-1,j);
                else
                    label = label+1;
                    inter_img(i,j) = label;
                end
            else
                if(threshold_img(i,j) ~= threshold_img(i,j-1) &&
threshold_img(i,j) ~= threshold_img(i-1,j))
                    label = label+1;
                    inter_img(i,j) = label;
                else
                    if(threshold_img(i,j) == threshold_img(i,j-1) &&
threshold_img(i,j) == threshold_img(i-1,j))
                        inter_img(i,j) = min(inter_img(i,j-1),inter_img(i-
1,j));
                        if(inter_img(i,j-1) ~= inter_img(i-1,j))
                            match1(num) = inter_img(i-1,j);
                            match2(num) = inter_img(i,j-1);
                            num = num+1;
                        end
                    else
                        if(threshold_img(i,j) == threshold_img(i-1,j))
                            inter_img(i,j) = inter_img(i-1,j);
                        else
                            inter_img(i,j) = inter_img(i,j-1);
                        end
                    end
                end
            end
```

```
        end
    end
end

labels = zeros(max(max(inter_img)),1);
pixel = 1;
for i = 1:length(match1)
   if(labels(match1(i)) == 0 && labels(match2(i)) == 0)
       labels(match1(i)) = min((match1(i)),(match2(i)));
       labels(match2(i)) = min((match1(i)),(match2(i)));
   elseif(labels(match1(i)) == 0 || labels(match2(i)) == 0)
       if(labels(match1(i)) ~= 0)
           labels(match2(i)) = labels(match1(i));
       else
           labels(match1(i)) = labels(match2(i));
       end
   elseif(labels(match1(i)) ~= 0 && labels(match2(i)) ~= 0)
       previous = max(labels(match1(i)),labels(match2(i)));
       labels(match1(i))= min(labels(match1(i)),labels(match2(i)));
       labels(match2(i)) = min(labels(match1(i)),labels(match2(i)));
       for k=1:length(labels)
           if(labels(k) == previous)
               labels(k) = min(labels(match1(i)),labels(match2(i)));
           end
       end
   end
end

for i=1:no_of_rows
    for j=1:no_of_cols
        if(labels(inter_img(i,j)) == 0)
            final_img(i,j) = inter_img(i,j);
        else
            final_img(i,j) = labels(inter_img(i,j));
        end
    end
end

sum = 0;
for i=1:length(labels)
    if(labels(i) == 0)
        sum = sum+1;
    end
end

if (sum==0)
    no_of_connected_components = length(unique(labels));
else
    no_of_connected_components = length(unique(labels)) - 1 + sum;
end

max_value=max(max(final_img));
final = round(final_img*(256/max_value+1));
figure(3);
imshow(uint8(final));
title('Connected Components')
```

Answer (2):

```
clc;
clear all;
close all;
original_img = imread('image.bmp');
figure(1);
imshow(original_img);
title('First Level');
gaussian_kernel=[1/16 1/4 6/16 1/4 1/16];
padded_img1 = padarray(original_img,[2 2], 0);

for i=3:+2:948
    for j=3:+2:1268
        final_img1((i),(j))= gaussian_kernel(1,1)*padded_img1(i,j-2) +
gaussian_kernel(1,2)*padded_img1(i,j-1) +
gaussian_kernel(1,3)*padded_img1(i,j) +
gaussian_kernel(1,4)*padded_img1(i,j+1) +
gaussian_kernel(1,5)*padded_img1(i,j+2);
    end
end
temp11=final_img1(3:2:end,:);
temp12=transpose(temp11);
final_img1=temp12(3:2:end,:);
final_img1=transpose(final_img1);
figure(2);
imshow(final_img1);
title('Second Level');

padded_img2 = padarray(final_img1,[2 2],0);
for i=3:+2:473
    for j=3:+2:633
        final_img2((i),(j))= gaussian_kernel(1,1)*padded_img2(i,j-2) +
gaussian_kernel(1,2)*padded_img2(i,j-1) +
gaussian_kernel(1,3)*padded_img2(i,j) +
gaussian_kernel(1,4)*padded_img2(i,j+1) +
gaussian_kernel(1,5)*padded_img2(i,j+2);
    end
end
temp21=final_img2(3:2:end,:);
temp22=transpose(temp21);
final_img2=temp22(3:2:end,:);
final_img2=transpose(final_img2);
figure(3)
imshow(final_img2);
title('Third Level');

padded_img3 = padarray(final_img2,[2 2],0);
```

```
for i=3:+2:237
    for j=3:+2:317
        final_img3((i),(j))= gaussian_kernel(1,1)*padded_img3(i,j-2) +
gaussian_kernel(1,2)*padded_img3(i,j-1) +
gaussian_kernel(1,3)*padded_img3(i,j) +
gaussian_kernel(1,4)*padded_img3(i,j+1) +
gaussian_kernel(1,5)*padded_img3(i,j+2);
    end
end
temp31=final_img3(3:2:end,:);
temp32=transpose(temp31);
final_img3=temp32(3:2:end,:);
final_img3=transpose(final_img3);
figure(4)
imshow(final_img3);
title('Forth Level');

padded_img4 = padarray(final_img3,[2 2],0);
for i=3:+2:118
    for j=3:+2:158
        final_img4((i),(j))= gaussian_kernel(1,1)*padded_img4(i,j-2) +
gaussian_kernel(1,2)*padded_img4(i,j-1) +
gaussian_kernel(1,3)*padded_img4(i,j) +
gaussian_kernel(1,4)*padded_img4(i,j+1) +
gaussian_kernel(1,5)*padded_img4(i,j+2);
    end
end
temp41=final_img4(3:2:end,:);
temp42=transpose(temp41);
final_img4=temp42(3:2:end,:);
final_img4=transpose(final_img4);
figure(5)
imshow(final_img4);
title('Fifth Level');

%%

clc;
clear all;
close all;
original_img= imread('image.bmp');
figure(1);
imshow(original_img);
title('First Level');
Laplacian_Kernel=[1/16 1/4 6/16 1/4 1/16];
padded_img1 = padarray(original_img,[2 2],0);
for i=3:+2:948
    for j=3:+2:1268
        final_img1((i),(j))= Laplacian_Kernel(1,1)*padded_img1(i,j-2) +
Laplacian_Kernel(1,2)*padded_img1(i,j-1) +
Laplacian_Kernel(1,3)*padded_img1(i,j) +
Laplacian_Kernel(1,4)*padded_img1(i,j+1) +
Laplacian_Kernel(1,5)*padded_img1(i,j+2);
    end
end
temp11=final_img1(3:2:end,:);
```

```
temp12=transpose(temp11);
final_img1 = temp12(3:2:end,:);
final_img11 =transpose(final_img1);
figure(2);
imshow(final_img11);


[a1 b1] = size(final_img11);
scale = 2;
M = zeros(scale.*a1,scale.*b1);
for count1 = 1:scale:a1.*scale
    for count2 = 1:scale:b1.*scale
        for count3 = 1:scale
            for count4 = 1:scale
                M1(count1+count3+1,count2+count4+1) =
final_img11(ceil(count1./scale),ceil(count2./scale));
            end
        end
    end
end
inter_img1 = (uint8(M1));
final_img111= original_img-inter_img1;
figure(3);
imshow(final_img111);
title('Second Level');

%

padded_img2 = padarray(final_img11,[2 2],0);
for i=3:+2:473
    for j=3:+2:633
        final_img2((i),(j))= Laplacian_Kernel(1,1)*padded_img2(i,j-2) +
Laplacian_Kernel(1,2)*padded_img2(i,j-1) +
Laplacian_Kernel(1,3)*padded_img2(i,j) +
Laplacian_Kernel(1,4)*padded_img2(i,j+1) +
Laplacian_Kernel(1,5)*padded_img2(i,j+2);
    end
end
temp21=final_img2(3:2:end,:);
temp22=transpose(temp21);
final_img2=temp22(3:2:end,:);
final_img22=transpose(final_img2);
figure(4)
imshow(final_img22);

[a2 b2] = size(final_img22);
scale = 2;
M = zeros(scale.*a2,scale.*b2);
for count1 = 1:scale:a2.*scale
    for count2 = 1:scale:b2.*scale
        for count3 = 1:scale
            for count4 = 1:scale
                M2(count1+count3,count2+count4) =
final_img22(ceil(count1./scale),ceil(count2./scale));
            end
        end
```

```
    end
end
inter_img2 = (uint8(M2));
final_img222= final_img11-inter_img2;
figure(5);
imshow(final_img222);
title('Third Level');

%

padded_img3 = padarray(final_img22,[2 2],0);
for i=3:+2:237
    for j=3:+2:317
        final_img3((i),(j))= Laplacian_Kernel(1,1)*padded_img3(i,j-2) +
Laplacian_Kernel(1,2)*padded_img3(i,j-1) +
Laplacian_Kernel(1,3)*padded_img3(i,j) +
Laplacian_Kernel(1,4)*padded_img3(i,j+1) +
Laplacian_Kernel(1,5)*padded_img3(i,j+2);
    end
end
temp31=final_img3(3:2:end,:);
temp32=transpose(temp31);
final_img3=temp32(3:2:end,:);
final_img33=transpose(final_img3);
figure(6)
imshow(final_img33);

[a3 b3] = size(final_img33);
scale = 2;
M = zeros(scale.*a3,scale.*b3);
for count1 = 1:scale:a3.*scale
    for count2 = 1:scale:b3.*scale
        for count3 = 1:scale
            for count4 = 1:scale
                M3(count1+count3-1,count2+count4-1) =
final_img33(ceil(count1./scale),ceil(count2./scale));
            end
        end
    end
end
inter_img3 = (uint8(M3));
final_img333= final_img22-inter_img3;
figure(7);
imshow(final_img333);
title('Forth Level');

%

padded_img4 = padarray(final_img33,[2 2],0);
for i=3:+2:118
    for j=3:+2:158
        final_img4((i),(j))= Laplacian_Kernel(1,1)*padded_img4(i,j-2) +
Laplacian_Kernel(1,2)*padded_img4(i,j-1) +
Laplacian_Kernel(1,3)*padded_img4(i,j) +
Laplacian_Kernel(1,4)*padded_img4(i,j+1) +
Laplacian_Kernel(1,5)*padded_img4(i,j+2);
```

```
    end
end
temp41=final_img4(3:2:end,:);
temp42=transpose(temp41);
final_img4=temp42(3:2:end,:);
final_img44=transpose(final_img4);
figure(8);
imshow(final_img44);

[a4 b4] = size(final_img44);
scale = 2;
M = zeros(scale.*a4,scale.*b4);
for count1 = 1:scale:a4.*scale
    for count2 = 1:scale:b4.*scale
        for count3 = 1:scale
            for count4 = 1:scale
                M4(count1+count3+1,count2+count4+1) =
final_img44(ceil(count1./scale),ceil(count2./scale));
            end
        end
    end
end
inter_img4 = (uint8(M4));
final_img444= final_img33-inter_img4;
figure(9);
imshow(final_img444);
title('Fifth Level');
```