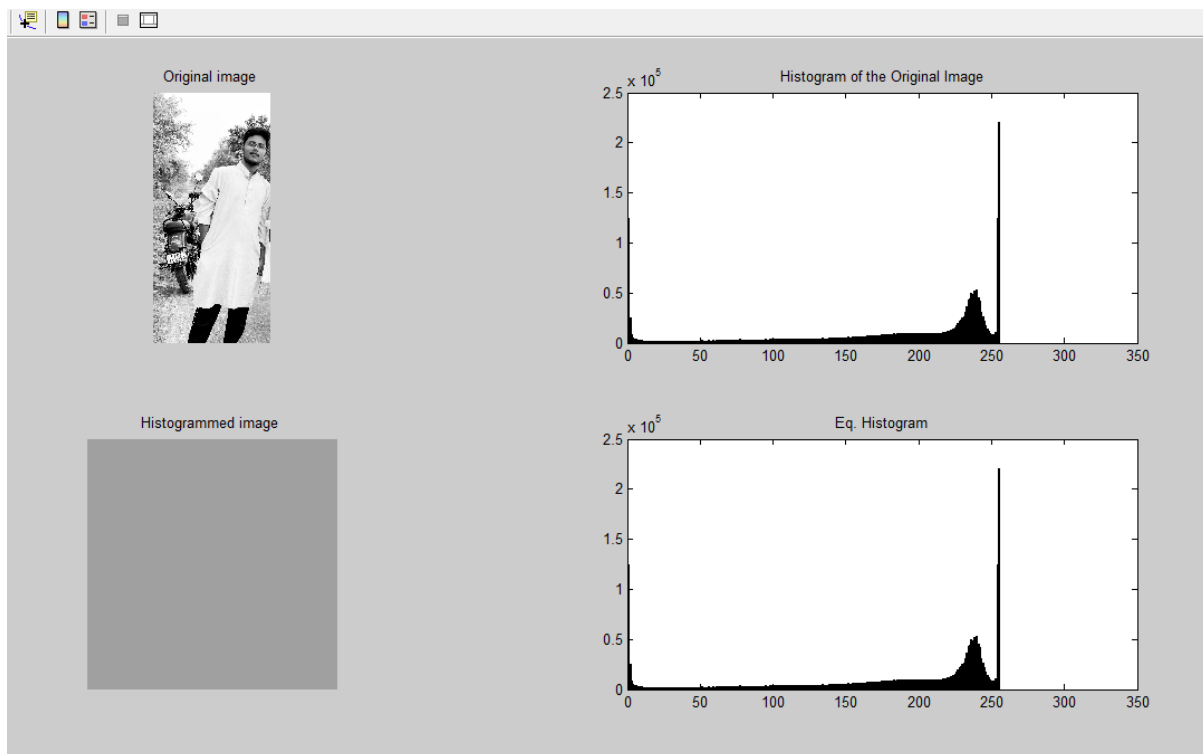


The image shows a MATLAB Editor window titled "Editor - C:\Users\Admin\Documents\MATLAB\p6.m*". The window contains a script for performing Histogram Equalization. The script starts with clearing the workspace and closing all figures. It then reads an image file 'D:\RS\1671458913727.jpg' and converts it to grayscale. The script calculates the histogram of the grayscale image and uses it to calculate the cumulative distribution function (CDF). The CDF is then used to calculate the new pixel values for the equalized image. The script is as follows:

```
1 % Practical 6 - Write a program to perform Histogram Equalization
2 - clc;
3 - clear all;
4 - close all;
5
6 - a = imread('D:\RS\1671458913727.jpg');
7 - a1 = double(a);
8 - a2 = rgb2gray(uint8(a1));
9 - [row col] = size(a2);
10 - c = row*col;
11 - h = zeros(1,300);
12 - z = zeros(1,300);
13 - for m = 1:1:row
14 -     for n = 1:1:col
15 -         t = a2(m,n);
16 -         h(t+1) = h(t+1)+1;
17 -     end
18 - end
19
20 - pdf = h/c;
21 - cdf(1) = pdf(1);
22 - for x=2:1:256
23 -     cdf(x) = pdf(x)+cdf(x-1);
24 - end
25
26 - new = round(cdf*256);
27 - new = new+1;
28 - for p = 1:1:row
29 -     for q = 1:1:col
30 -         temp = a2(p,q);
31 -         b = (temp);
32 -         t = b;
33 -         z(t+1) = z(t+1)+1;
```

```
34 -     end
35 - end
36
37 - b = b-1;
38 - subplot(2,2,1)
39 - imshow(uint8(a2))
40 - title('Original image');
41
42 - subplot(2,2,2)
43 - bar(h)
44 - title('Histogram of the Original Image');
45
46 - subplot(2,2,3)
47 - imshow(uint8(b))
48 - title('Histogrammed image');
49
50 - subplot(2,2,4)
51 - bar(z)
52 - title('Eq. Histogram');
```



```

Editor - D:\RS\P7\P7.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
% Practical 7 - Write a program to perform Erosion and Dilation
clc;
clear all;
close all;

w = [0 0 0 0 1 1 1 1 1 1 1 0 0 0;
     0 0 0 0 1 1 1 1 1 1 1 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 1 1 1 1 1 0 0 0;
     0 0 0 0 1 1 1 1 1 1 1 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 0 0 0 0 0 0 0 0];

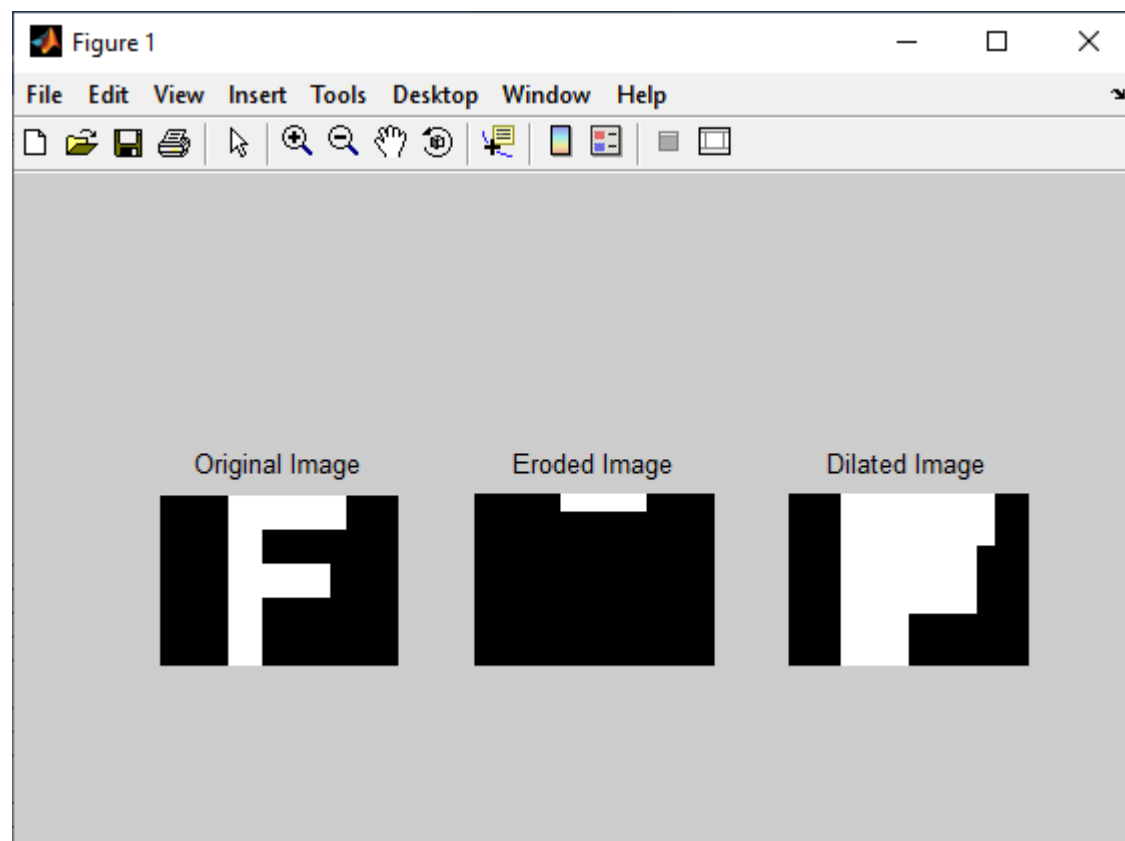
disp(w);
se1 = strel('square',3);
disp(se1);
IM1 = imerode(w, se1);
IM2 = imdilate(w, se1);

subplot(1,3,1);
imshow(w);
title('Original Image')

subplot(1,3,2);
imshow(IM1);
title('Eroded Image')

subplot(1,3,3);
imshow(IM2);
title('Dilated Image')

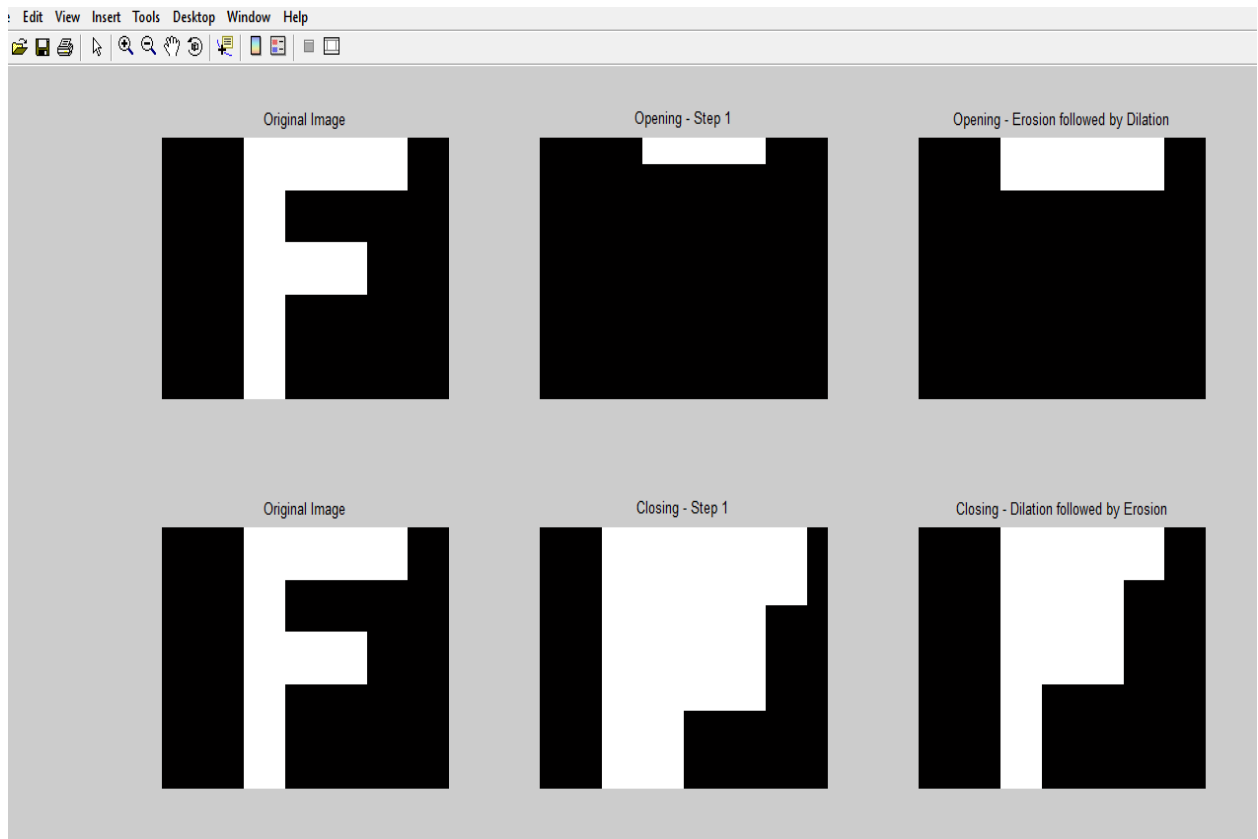
```



```

Editor - D:\RS\P8\P8.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
1 % Practical 8 - Write a program to perform Opening and Closing
2 |
3 - clc;
4 - clear all;
5 - close all;
6
7 - w = [0 0 0 0 1 1 1 1 1 1 1 0 0;
8       0 0 0 0 1 1 1 1 1 1 1 1 0 0;
9       0 0 0 0 1 1 0 0 0 0 0 0 0 0;
10      0 0 0 0 1 1 0 0 0 0 0 0 0 0;
11      0 0 0 0 1 1 1 1 1 1 0 0 0 0;
12      0 0 0 0 1 1 1 1 1 1 0 0 0 0;
13      0 0 0 0 1 1 0 0 0 0 0 0 0 0;
14      0 0 0 0 1 1 0 0 0 0 0 0 0 0;
15      0 0 0 0 1 1 0 0 0 0 0 0 0 0;
16      0 0 0 0 1 1 0 0 0 0 0 0 0 0];
17
18 - disp(w);
19 - se1 = strel('square',3);
20 - disp(se1);
21 - O1 = imerode(w, se1);
22 - O2 = imdilate(O1, se1);
23
24 - C1 = imdilate(w, se1);
25 - C2 = imerode(C1, se1);
26
27 - subplot(2,3,1);
28 - imshow(w);
29 - title('Original Image')
30
31 - subplot(2,3,2);
32 - imshow(O1);
33 - title('Opening - Step 1')
34
35 - subplot(2,3,3);
36 - imshow(O2);
37 - title('Opening - Erosion followed by Dilation')
38
39 - subplot(2,3,4);
40 - imshow(w);
41 - title('Original Image')
42
43 - subplot(2,3,5);
44 - imshow(C1);
45 - title('Closing - Step 1')
46
47 - subplot(2,3,6);
48 - imshow(C2);
49 - title('Closing - Dilation followed by Erosion')
50

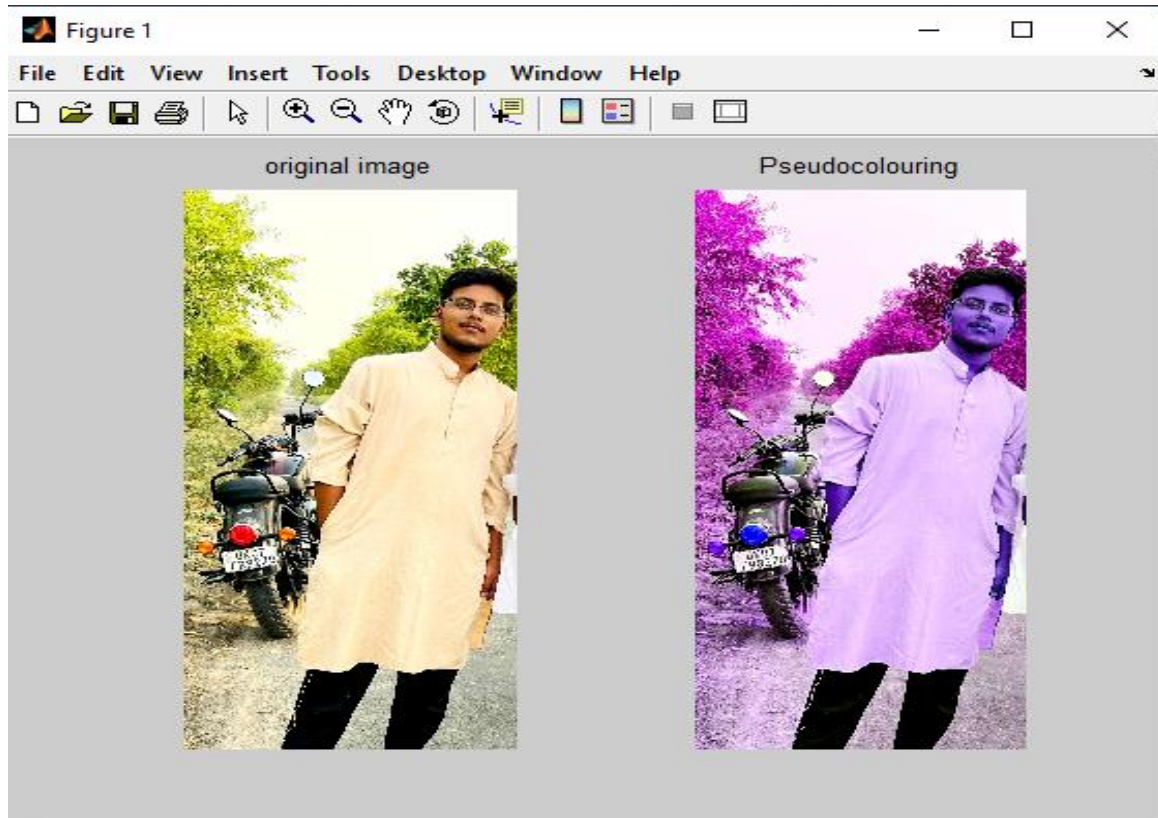
```



```
Editor - D:\RS\P9\P9A.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons]
[Icons]
[Icons]
1 % Practical 9a - RGB planes
2 -
3 - clc;
4 - clear all;
5 - close all;
6 -
7 - img = imread('D:\RS\1671458913727.jpg');
8 - subplot(1,4,1);
9 - imshow(img);
10 - title('Original Image')
11 -
12 - redp = img(:,:,1);
13 - greenp = img(:,:,2);
14 - bluep = img(:,:,3);
15 -
16 - subplot(1,4,2);
17 - imshow(redp);
18 - title('Red plane')
19 -
20 - subplot(1,4,3);
21 - imshow(bluep);
22 - title('Blue plane')
23 -
24 - subplot(1,4,4);
25 - imshow(greenp);
26 - title('Green plane')
```



```
Editor - D:\RS\P9\P9B.m
File Edit Text Go Cell Tools Debug Desktop Window Help
[Icons]
1 %Practical 9b - Pseudocoloring
2 - clc;
3 - clear all;
4 - close all;
5
6 - img=imread('D:\RS\1671458913727.jpg');
7 - subplot(1,2,1);
8 - imshow(img);
9 - title('original image')
10
11 - redp=img(:,:,1);
12 - greenp=img(:,:,2);
13 - bluep=img(:,:,3);
14
15 - OP(:,:,1)=greenp;
16 - OP(:,:,2)=bluep;
17 - OP(:,:,3)=redp;
18 - subplot(1,2,2);
19 - imshow(OP);
20 - title('Pseudocolouring');
```



```

Editor - D:\RS\P10\P10.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 % Practical 10 - Write a program to perform smoothing on a given image
2 clc;
3 clear all;
4 close all;
5
6 img = imread('D:\RS\1671458913727.jpg');
7 grayImage = rgb2gray(img);
8 subplot(2,3,1);
9 imshow(grayImage);
10 title('Original Image');
11
12 grayImage1 = imnoise(grayImage, 'salt & pepper');
13 subplot(2,3,2);
14 imshow(grayImage1);
15 title('Noise Image');
16
17 blurredImage = imfilter(grayImage1, ones(3)/9, 'symmetric');
18 subplot(2,3,3);
19 imshow(blurredImage);
20 title('Blurred Image (3x3)');
21
22 blurredImage = imfilter(grayImage1, ones(5)/25, 'symmetric');
23 subplot(2,3,4);
24 imshow(blurredImage);
25 title('Blurred Image (5x5)');
26
27 blurredImage = imfilter(grayImage1, ones(7)/49, 'symmetric');
28 subplot(2,3,5);
29 imshow(blurredImage);
30 title('Blurred Image (7x7)');
31
32 blurredImage = imfilter(grayImage1, ones(9)/81, 'symmetric');
33 subplot(2,3,6);
34 imshow(blurredImage);
35 title('Blurred Image (9x9)');

```




```
1 % Practical 11 - Perform sharpening on a given image
2
3 -   clc;
4 -   clear all;
5 -   close all;
6
7 -   img = imread('D:\RS\1671458913727.jpg');
8 -   a = rgb2gray(img);
9 -   lap = [1 1 1; 1 -8 1; 1 1 1];
10 -  resp = uint8(filter2(lap, a, 'same'));
11 -  sharpened = imsubtract(a, resp);
12
13 -  subplot(1,3,1);
14 -  imshow(a);
15 -  title('Original Image');
16
17 -  subplot(1,3,2);
18 -  imshow(resp);
19 -  title('Laplacian Filtered Image');
20
21 -  subplot(1,3,3);
22 -  imshow(sharpened);
23 -  title('Sharpened Image');
```

