Contents

Chapter 1

Part - I

1.1 Edge Detection

Writing and testing a MATLAB script that perform the following:
Edge Detection [1] using various filters and display [2] RGB and gray image.

1.2 MATLAB Code

```
%% Function where our program starts
function part1()
%--------------------------------------------------------------------
% USAGE: part1
% Output: Applying various edge detection operations
%--------------------------------------------------------------------
    % Given the Image filename/path
    pic1_name = 'lena_color.tiff';
    pic2_name = 'lena_gray.pgm';
    % Calls the imagecheck function to check image type
    img1_type = imagecheck(pic2_name);
    % Reads the image specified the image name
    var_gmat = imread(pic2_name,'pgm');
    % Calls the imagecheck function to check image type
    img2_type = imagecheck(pic1_name);
    % Reads the image specified the image name
    var_cmat = imread(pic1_name,'tiff');
    % Performing edge detection for grayscale/RGB image
    grayOutput_1 = processImgEdge(img1_type,var_gmat,'sobel');
    grayOutput_2 = processImgEdge(img1_type,var_gmat,'roberts');
```

```
    grayOutput_4 = processImgEdge(img1_type,var_gmat,'laplacian');
    grayOutput_5 = processImgEdge(img1_type,var_gmat,'prewit');
    rgbOutput_1 = processImgEdge(img2_type,var_cmat,'sobel');
    rgbOutput_2 = processImgEdge(img2_type,var_cmat,'roberts');
    rgbOutput_4 = processImgEdge(img2_type,var_cmat,'laplacian');
    rgbOutput_5 = processImgEdge(img2_type,var_cmat,'prewit');
    figure(1)
    subplot(2,5,1); imshow(var_gmat);
    title('Original Image');
    subplot(2,5,2); imshow(grayOutput_1);
    title('Sobel');
    subplot(2,5,3); imshow(grayOutput_2);
    title('Roberts');
    subplot(2,5,4); imshow(grayOutput_5);
    title('Prewit');
    subplot(2,5,5); imshow(grayOutput_4);
    title('Laplacian');
    subplot(2,5,6); imshow(var_cmat);
    title('Original Image');
    subplot(2,5,7); imshow(rgbOutput_1);
    title('Sobel');
    subplot(2,5,8); imshow(rgbOutput_2);
    title('Roberts');
    subplot(2,5,9); imshow(rgbOutput_5);
    title('Prewit');
    subplot(2,5,10); imshow(rgbOutput_4);
    title('Laplacian');
end

function[img_check] = imagecheck(var_img)
%--------------------------------------------------------------------
% USAGE: x = imagecheck(image_path);
% Inputs: image_path = Image filename/URL
% Output: x = Type of the Image('GrayScale Image'/'Color Image')
%--------------------------------------------------------------------
    % var_info get information about the image file
    var_info = imfinfo(var_img);
    % Checks if the image is grayscale/color and return image type
    if(strcmp(var_info.ColorType,'grayscale'))
        img_check = 'GrayScale Image';
    elseif(strcmp(var_info.ColorType,'truecolor'))
```

```
        img_check = 'Color Image';
    else
        img_check = 'Invalid';
    end
end

%% Function to process the image
function[mOut] = processImgEdge(var1,var2,var3)
%-------------------------------------------------------------
% USAGE: processImgEdge(typeOfImage,MatOfImage,Filtername);
% Inputs: var1 = Type of Image
%         var2 = Matrix of Image
%         var3 = Filter name
% Output: mOut - Image after applying edge filter
%-------------------------------------------------------------
    if(strcmp(var1,'GrayScale Image'))
        if(strcmp(var3,'sobel'))
            mOut = edge(var2, 'sobel');
        end
        if(strcmp(var3,'roberts'))
            mOut = edge(var2, 'roberts');
        end
        if(strcmp(var3,'prewit'))
            mOut = edge(var2, 'prewitt');
        end
        if(strcmp(var3,'laplacian'))
            h = fspecial('log');
            mOut = imfilter(var2, h);
        end
    end
    if(strcmp(var1,'Color Image'))
       v = rgb2gray(var2);
       if(strcmp(var3,'sobel'))
            mOut = edge(v, 'sobel');
        end
        if(strcmp(var3,'roberts'))
            mOut = edge(v, 'roberts');
        end
        if(strcmp(var3,'prewit'))
            mOut = edge(v, 'prewitt');
        end
```

```
        if(strcmp(var3,'laplacian'))
            h = fspecial('log');
            mOut = imfilter(v, h);
        end
    end
end
```

1.3 MATLAB Output

Edge detection filters are applied for both RGB and Grayscale images and displayed in the plot as below:

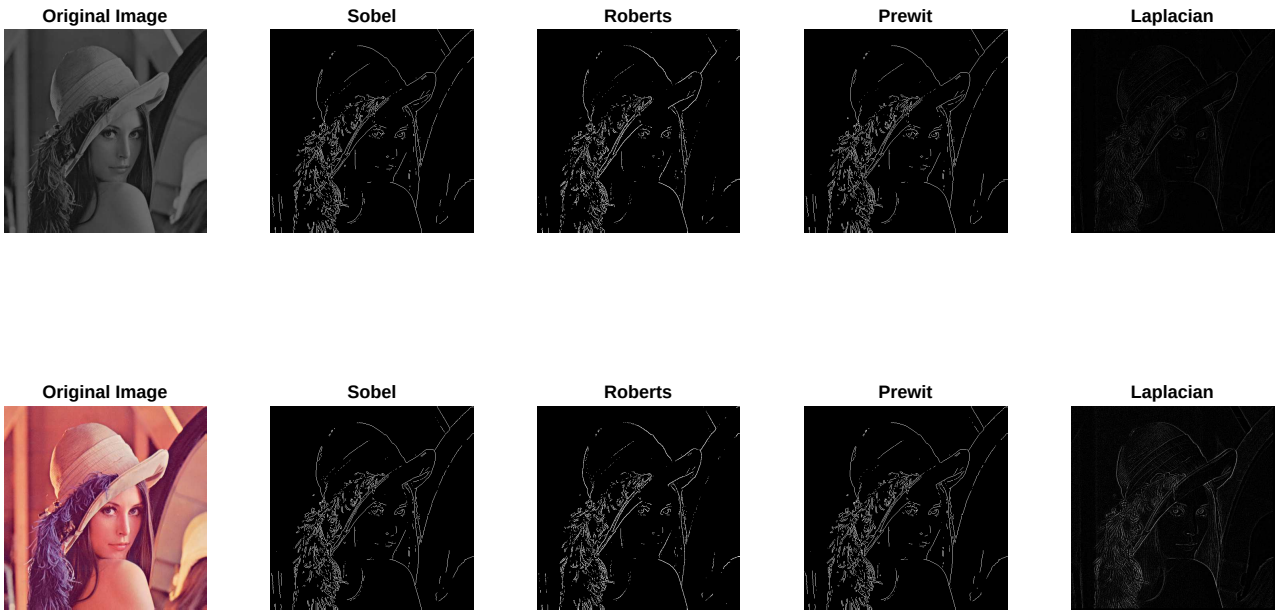

Figure 1.1: Before and After Edge Detection.

Chapter 2

Part - II

2.1 Image Scaling

Writing and testing a MATLAB script that perform the following:
Scaling [3] of the Image using nearest neighbor, bilinear and interpolation methods.

2.2 MATLAB Code

```
%% Function where our program starts
function part2()
%-----------------------------------------------------------------
% USAGE: part2
% Output: Applying various rescaling operations
%-----------------------------------------------------------------
    % Given the Image filename/path
    pic1_name = 'lena_color.tiff';
    pic2_name = 'lena_gray.pgm';
    % Calls the imagecheck function to check image type
    img1_type = imagecheck(pic2_name);
    % Reads the image specified the image name
    var_gmat = imread(pic2_name,'pgm');
    % Calls the imagecheck function to check image type
    img2_type = imagecheck(pic1_name);
    % Reads the image specified the image name
    var_cmat = imread(pic1_name,'tiff');
    % Performing rescaling for grayscale/RGB image
```

```
    grayOutput_1 = imresize(var_gmat, 0.3, 'nearest', 0);
    grayOutput_2 = imresize(var_gmat, 0.5, 'bilinear', 0);
    grayOutput_3 = imresize(var_gmat, 0.8, 'Bicubic', 0);
    rgbOutput_1 = imresize(var_cmat, 0.4, 'nearest', 0);
    rgbOutput_2 = imresize(var_cmat, 0.7, 'bilinear', 0);
    rgbOutput_3 = imresize(var_cmat, 0.9, 'Bicubic', 0);
    figure(1)
    h1=subplot(2,4,1); imshow(var_gmat);
    axis on
    title('Original Image');
    h2=subplot(2,4,2); imshow(grayOutput_1);
    axis on
    title('Nearest Neighbor');
    h3=subplot(2,4,3); imshow(grayOutput_2);
    axis on
    title('Bilinear');
    h4=subplot(2,4,4); imshow(grayOutput_3);
    axis on
    title('Bicubic');
    h5=subplot(2,4,5); imshow(var_cmat);
    axis on
    title('Original Image');
    h6=subplot(2,4,6); imshow(rgbOutput_1);
    axis on
    title('Nearest Neighbor');
    h7=subplot(2,4,7); imshow(rgbOutput_2);
    axis on
    title('Bilinear');
    h8=subplot(2,4,8); imshow(rgbOutput_3);
    axis on
    title('Bicubic');
    linkaxes([h1,h2,h3,h4,h5,h6,h7,h8])
end

function[img_check] = imagecheck(var_img)
%------------------------------------------------------------------
% USAGE: x = imagecheck(image_path);
% Inputs: image_path = Image filename/URL
% Output: x = Type of the Image('GrayScale Image'/'Color Image')
%------------------------------------------------------------------
    % var_info get information about the image file
```

```
    var_info = imfinfo(var_img);
    % Checks if the image is grayscale/color and return image type
    if(strcmp(var_info.ColorType,'grayscale'))
        img_check = 'GrayScale Image';
    elseif(strcmp(var_info.ColorType,'truecolor'))
        img_check = 'Color Image';
    else
        img_check = 'Invalid';
    end
end
```

2.3 MATLAB Output

Scaling of the images is performed as follows as displayed below:

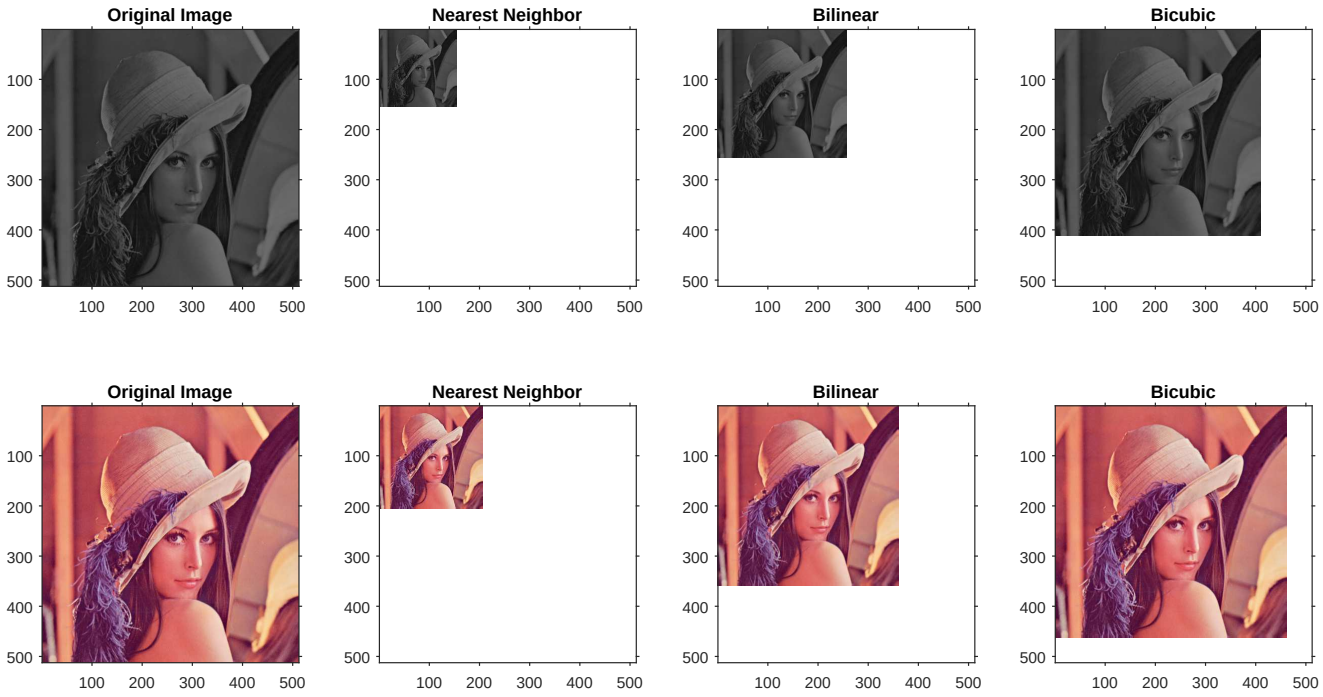

Figure 2.1: Before and After scaling.

Chapter 3

Part - III

3.1 Image Rotation

Writing and testing a MATLAB script that perform the following:
Rotating [4] the image at various angle and displaying the image.

3.2 MATLAB Code

```
%% Function where our program starts
function part3()
%-----------------------------------------------------------------
% USAGE: part3
% Output: Applying various rotation operations
%-----------------------------------------------------------------
    % Given the Image filename/path
    pic1_name = 'lena_color.tiff';
    pic2_name = 'lena_gray.pgm';
    % Calls the imagecheck function to check image type
    img1_type = imagecheck(pic2_name);
    % Reads the image specified the image name
    var_gmat = imread(pic2_name,'pgm');
    % Calls the imagecheck function to check image type
    img2_type = imagecheck(pic1_name);
    % Reads the image specified the image name
    var_cmat = imread(pic1_name,'tiff');
    % Performing rescaling for grayscale/RGB image
    figure(1)
```

```
    h1=subplot(2,3,1); imshow(var_gmat);
    axis on
    title('Original Image');
    h2=subplot(2,3,2); imshow(imrotate(var_gmat, 45, 'crop'));
    axis on
    title('Rotated - Cropped');
    h3=subplot(2,3,3); imshow(imrotate(var_gmat, 45, 'loose'));
    axis on
    title('Rotated - Not Cropped');
    h4=subplot(2,3,4); imshow(var_cmat);
    axis on
    title('Original Image');
    h5=subplot(2,3,5); imshow(imrotate(var_cmat, 45, 'crop'));
    axis on
    title('Rotated - Cropped');
    h6=subplot(2,3,6); imshow(imrotate(var_cmat, 45, 'loose'));
    axis on
    title('Rotated - Not Cropped');
end

function[img_check] = imagecheck(var_img)
%--------------------------------------------------------------------
% USAGE: x = imagecheck(image_path);
% Inputs: image_path = Image filename/URL
% Output: x = Type of the Image('GrayScale Image'/'Color Image')
%--------------------------------------------------------------------
    % var_info get information about the image file
    var_info = imfinfo(var_img);
    % Checks if the image is grayscale/color and return image type
    if(strcmp(var_info.ColorType,'grayscale'))
        img_check = 'GrayScale Image';
    elseif(strcmp(var_info.ColorType,'truecolor'))
        img_check = 'Color Image';
    else
        img_check = 'Invalid';
    end
end
```

3.3 MATLAB Output

Displaying the image before and after rotation is shown below:

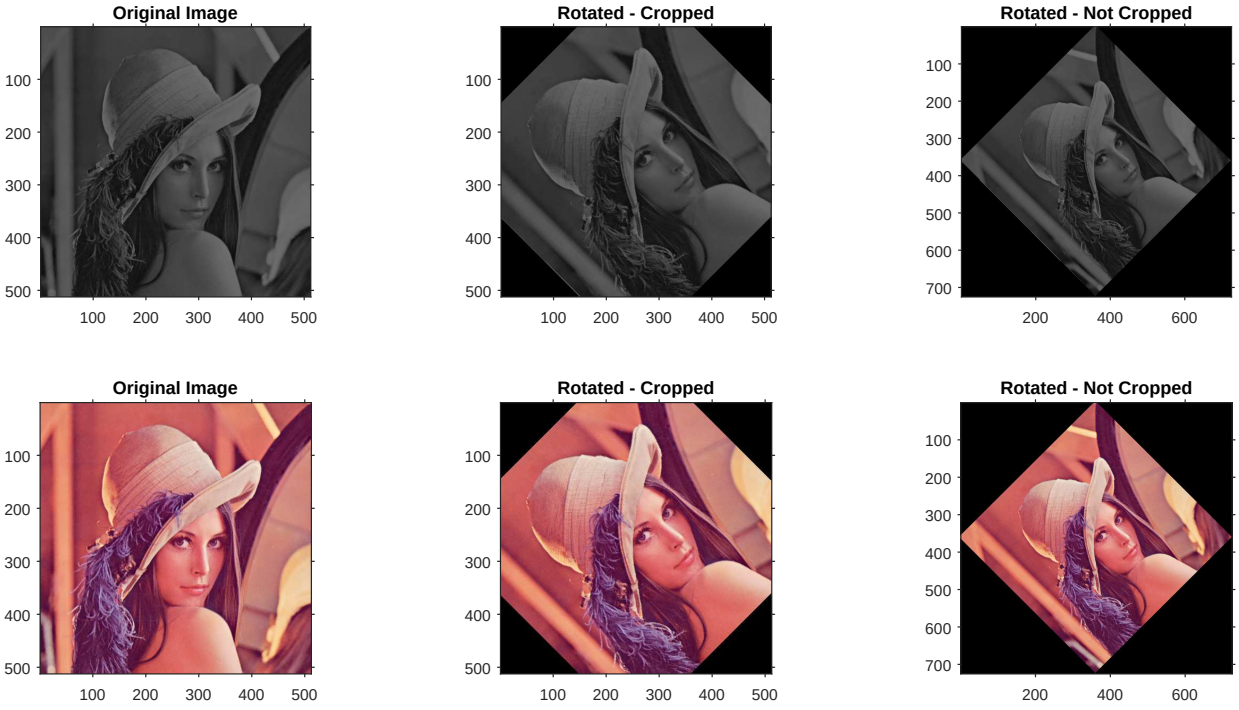

Figure 3.1: Before and After rotation.

References

[1] Documentation. (2016) Matlab - detect the edge. [Online; accessed 16-June-2016]. [Online]. Available: http://www.mathworks.com/help/images/ref/edge.html

[2] Documentation. (2016) Matlab - display image. [Online; accessed 16-June-2016]. [Online]. Available: http://www.mathworks.com/help/matlab/ref/imshow.html

[3] Documentation. (2016) Matlab - resize the images. [Online; accessed 16-June-2016]. [Online]. Available: http://www.mathworks.com/help/images/ref/imresize.html

[4] Documentation. (2016) Matlab - rotate the image. [Online; accessed 16-June-2016]. [Online]. Available: http://www.mathworks.com/help/matlab/ref/imrotate.html