.

Title	Image Effect Implementation, Part:2
Student Name	Jubayer Hossain Arnob
Student ID	1813124642
Email address	jubayer.arnob@northsouth.edu

Question

Implement and explain your approach to implement the following image effects. **(a)** Histogram Equalization **(b)** Old image **(c)** Sketch **(d)** Edge glow.

Instructions

- Both explanations and Matlab codes should be included in **THIS** document. No need to submit separate files for codes.
- Include some input and output of your code. In addition to the sample input/output (from code\SampleIN-OUT), also include more input/output using other photos of your choice.
- You can also include sample output of the intermediate steps of your calculation to explain the reason/effect with/without any particular step.
- Note that, only producing the output is not the only end goal here. I will also mark your understanding of the work from your explanation. Try to illustrate as elegant as you can with figures, tables, algorithms, charts, references, etc. like a research paper.

Introduction

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. In this report I am going to discuss about assignment 2 that includes 4 image processing tasks. Challenges are to give the following effects (Histogram equalization, old image effect, sketch effect and edge glow effect) on an input image using MATLAB tool. My approach on the implementation is explained here.

Method

Histogram Equalization Effect:

To be able to apply this effect first I need to calculate frequency for each pixel in input image. This is the histogram of the input image. Later I will normalize histogram for each pixel for this have divided each pixel frequency buy the size of the image. [f(i) / (m*n)] where m, n=row, column and f(i) is frequency of certain pixel.

After that I have calculated CDF and later found the transformation function that will be used to get expected output. Formula: S=round(255*CDF)

Applying S on input image will return the output of histogram equalization effect.

Old Image Effect:

This task can be divided into 3 steps. First step is to implement Sepia effect on the input image and get the sepia effect output. The next step is to add noise effect on the sepia image. This is image 2 that is created by two effects: Sepia and Noise. The final step, now I have to blend image 2 and input image in such a way that will create the expected output. Here, I have used blend formula: old image = Input image/3 + image2 – 60.

Sketch Effect:

I have used these 2 filters here. Filter A: [-1 -2 -1; 0 0 0; 1 2 1] and Filter B: [-1 0 1; -2 0 2; -1 0 1]. As the input image is RGB. I have applied these filters for each of the 3 planes of the RGB image that means filter applied for R, B and G plane. This will give us edge filtered effect for all three planes of the input image.

Later I have converted all 3 back to uin8. So, it makes again an RGB image with edge effect. The next step is to convert the output RGB into gray scale image by the help of RGB2GRAY library. This will bring us a result output that is with black background with white edges. So, by inversing it I get the expected output. For this I have used 255-pixel value for each iteration.

Edge Glow Effect:

The first half work here is similar to the Sketch effect that I have discussed. From using same filters on each of 3 planes of RGB image and again adding together to make the new RGB image with edge effect is same as before. Now, for this effect I don’t convert the RGB to gray here, instead the work is to give blur effect on edges.

Which means, after getting the RGB output after applying filters, now the task is to give blur effect on it that will work as glow effect on edges. In this assignment I have used gaussian filter with 3x3 filter and sigma 3. Suppose the output is image 2. Now if I blend the input image and this image 2, it will give us the result that we are calling as edge glow effect on image.

Experiment

Histogram Equalization Effect:

Following the Method of Histogram equalization effect that is discussed earlier I came up with few examples here. Figure 1,2,3 right below. This image shows that this effect works really well to get more detailed information in output image from the input one. From 3rd example it is clear that it is a good effect for low light images too.

Figure 1: Histogram Equalization Effect on RGB flower image

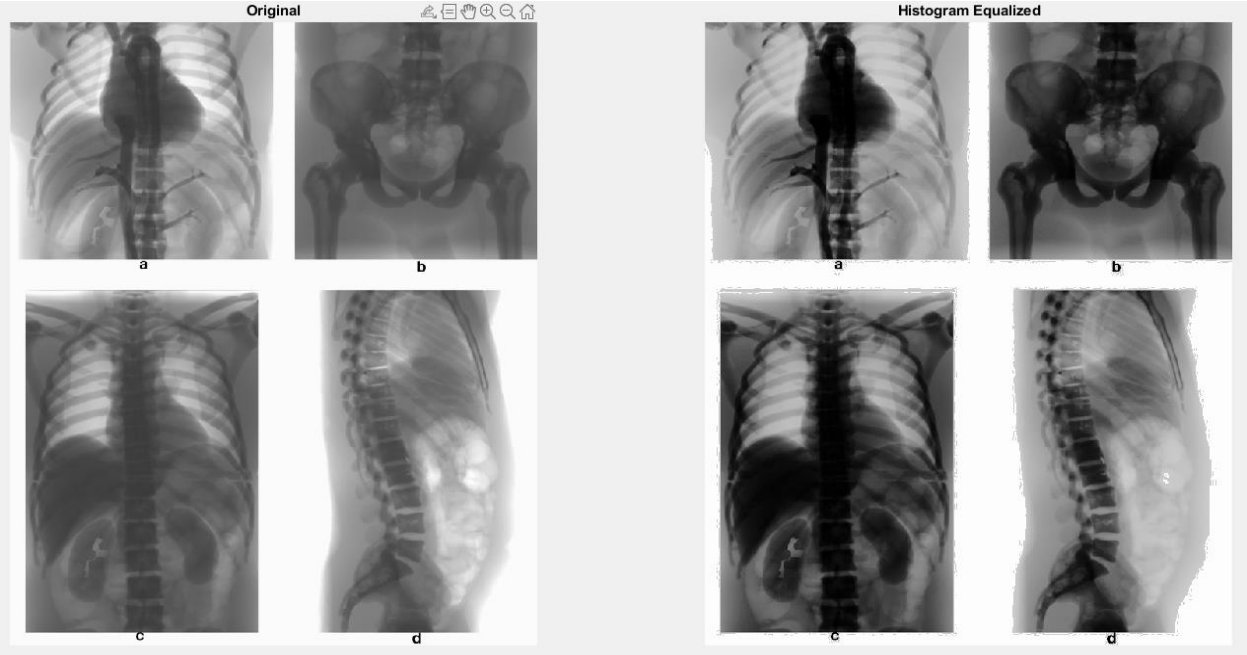

Figure 2: Histogram Equalization Effect on Medical Images

Figure 3: Histogram Equalization Effect on low light Xray image.

Old Image Effect:

Experiment here is done as described in method section. Here are two examples (fig 4,5) of the input image with the old image effect as output.

Figure 4: Old Image Effect on a clean RGB Image.

Figure 5: Old Image Effect on clean HD image.

Using figure 5 here are the three steps shown by which I have achieved old image effect result.
Step 1: Input image RGB to SEPIA conversion. [1]

Figure 5.1: Sepia effect

Step 2: Output Sepia image with Noise effect. (I used Gaussian effect with 0.2 parameter) [2]

Figure 5.2: Noise effect

Step 3: Noise sepia image blend with original image. Here this self-made blend formula used, old image = Input image/3 + image2 – 60.

Figure 5.3: blend effect (final output)

Sketch Effect:

Experiment here is done as described in method section. Here are two examples (fig 6,7) of the input image with the sketch effect as output.

Figure 6: Sketch effect on RGB image of house

Figure 7: Sketch effect on RGB image of birds on water

Using figure 7 here are the three steps shown by which I have achieved sketch effect result.

Step 1: Filter A: [-1 -2 -1; 0 0 0; 1 2 1] and Filter B: [-1 0 1; -2 0 2; -1 0 1] applied on each R, G, B plane of the input image and converted all 3 back to uin8. The output is below:

Figure 7.1: Apply Sobel filter on RGB image

Step 2: Applied RGB2GRAY function to the filtered RGB output and the outcome is given.

Figure 7.2: RGB output to gray image conversion

Step 3: The above result is still a sketch but in negative effect. To make it white page sketch I had to inverse it. For this each pixel value = 255 – pixel value. The output is satisfying.

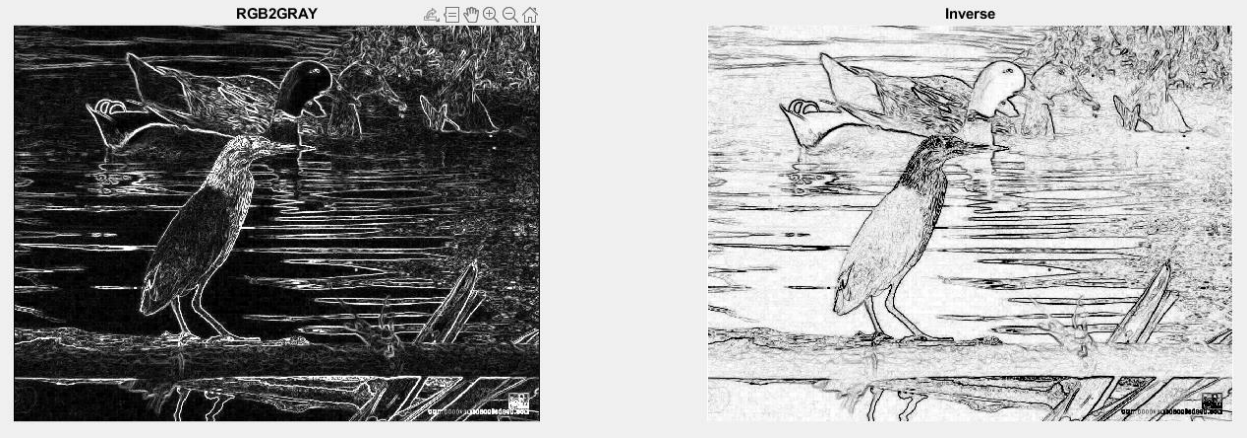

Figure 7.3: Sketch effect on RGB image of birds on water (Final Output)

Edge Glow Effect:

Experiment here is done as described in method section. Here are two examples (fig 8,9) of the input image with the old image effect as output.

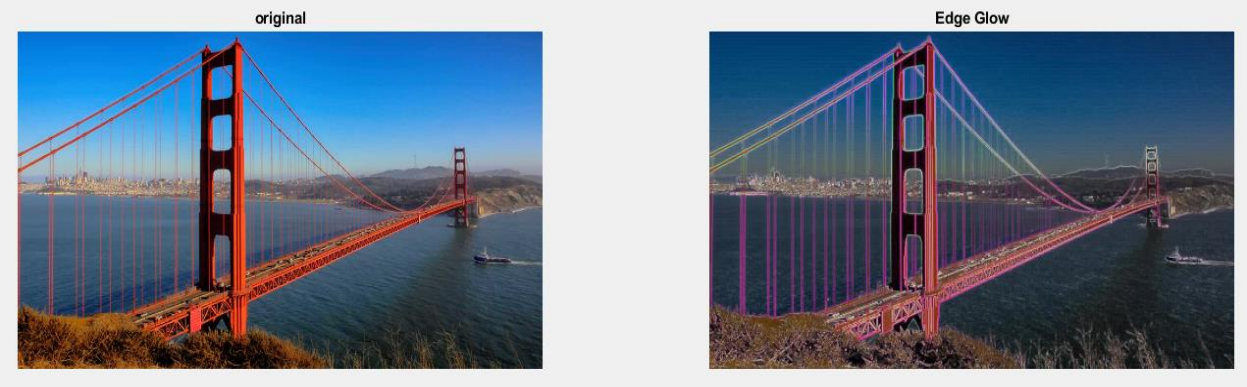

Figure 8: Edge Glow Effect on RGB image of bridge

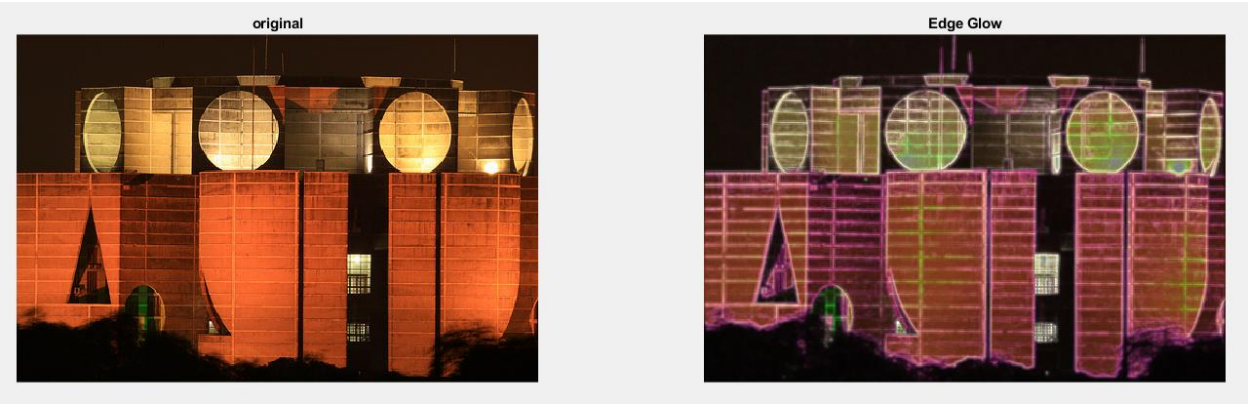

Figure 9: Edge Glow Effect on RGB image of Architecture

Using figure 9, here are the three steps shown by which I have achieved edge glow effect.
Step 1: Is same as step1 in sketch effect method. The output is here:

Figure 9.1: Apply Sobel filter on RGB image

Step 2: To give the glow effect I have used to use blur effect on the edges. As the last output gives us a edge filtered output. On this, I have used gaussian blur filter with size 3x3 and sigma 3. The blur filter effect is here:

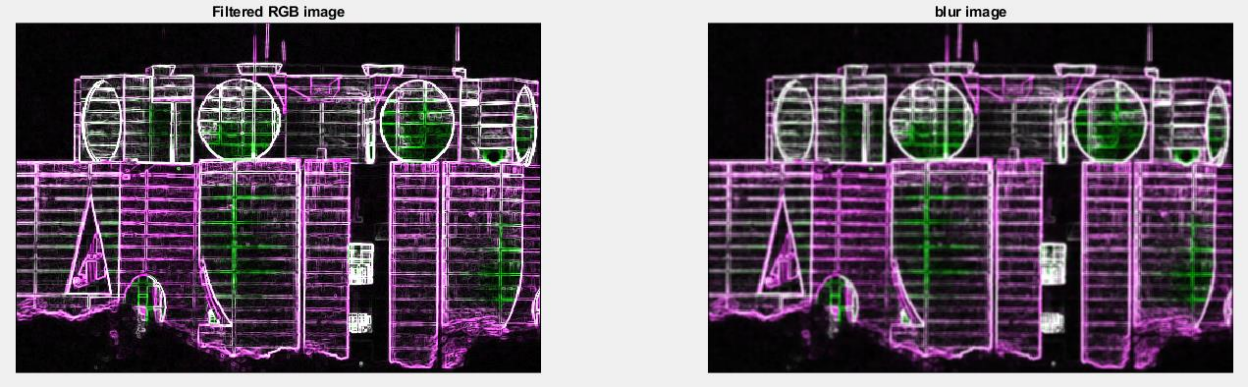

Figure 9.2: Blur effect on the output image.

Step 3: Now, the last task is to blend the current output image with the original image in such way that the image kept less changed with the edges glow. To do that I used overlay style blend method on both images and the output is this:

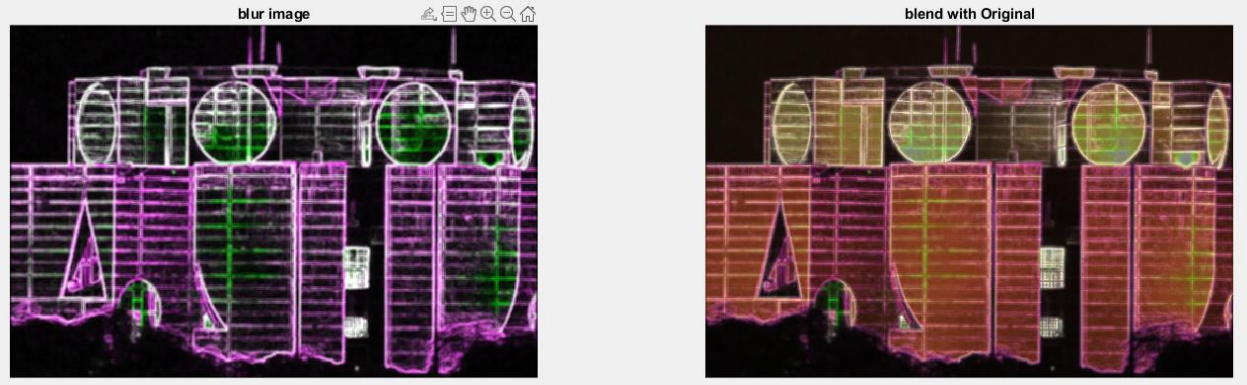

Figure 9.3: Overlay blending on original image with edge blur image. (Final output)

Conclusion

The tasks in this assignment have been really interesting from solving problems to trying new things to experiment and manipulate images. All though there are some issues where one must be careful while doing the tasks. A very important thing is the dimension start here from (1,1) not (0,0). It is important to notice while calculating with matrices. One need to be careful doing the conv method as from where the filter starts and where to stop by row and by column. I faced a problem that I wasn't able to work with gray images as input which is nicely handled by using 'if' method. Again, outputting gray images, one must be aware of using mat2gray function for the output. I faced error so I understood the problem and solved it. I believe after doing this assignment I have enough motivation and interest to learn more and do projects and paper writing in image processing sector.

Reference

1. RGB2SEPIA: How do I create vintage images in MATLAB? - Stack Overflow
2. Add noise to Image: Add noise to image - MATLAB imnoise (mathworks.com)
3. Overlay style blending: Edge Detection and Image Overlay - MATLAB & Simulink (mathworks.com)
4. Ideas from lecture 3 & 4
5. Class practice codes

Appendix

Code 1 (Histogram Equalization Effect)

```
clc
clear all
close all

im = imread('histeq_in.jpg');
if size(im,3) == 3 % if input is rgb
     im = rgb2gray(im);
end
im = double(im);

r = size(im,1);
c = size(im,2);
n = r*c; %total number of pixels
ah = uint8(zeros(r,c));

f = zeros(256,1); %frequency/histogram
pdf = zeros(256,1); %normalized histogram, probability(f)
cdf = zeros(256,1);
s = zeros(256,1); %equalized histogram

%calculating histogram
for i=1:size(im,1)
    for j=1:size(im,2)
        value = im(i,j);
        f(value+1)=f(value+1)+1;
    end
end

%calculating pdf,cdf,transformation function
a = 0;
for i=1:size(f)
     pdf(i) = (f(i)/n);
     a = a+pdf(i);
     cdf(i) = a;
     s(i) = round(cdf(i)*255);
end

%applying transformation function, s on image
for i=1:size(im,1)
    for j=1:size(im,2)
        ah(i,j) = s(im(i,j)+1);
    end
end

subplot(1,2,1);
imshow(mat2gray(im));title("Original")
subplot(1,2,2);
imshow(mat2gray(ah));title("Histogram Equalized");
```

Code 2 (Old Image Effect)

```
clc
```

```
clear all
close all

original = imread('old_in.jpg');% // Read in your image here
r = size(original,1);
c = size(original,2);

% RGB2SEPIA CONVERSION
im = original;
inputRed = im(:,:,1); %// Extract each colour plane
inputGreen = im(:,:,2);
inputBlue = im(:,:,3);
% Create sepia tones for each channel
outputRed = (inputRed * .393) + (inputGreen *.769) + (inputBlue * .189);
outputGreen = (inputRed * .349) + (inputGreen *.686) + (inputBlue * .168);
outputBlue = (inputRed * .272) + (inputGreen *.534) + (inputBlue * .131);
% convert back to uint8
sepia = uint8(cat(3, outputRed, outputGreen, outputBlue));
imshow(sepia); %// Show sepia image

% ADD NOISE
im2 = imnoise(sepia,'gaussian');
imshow(im2);

% BLEND
old = uint8(zeros(r,c));
for i=1:r
    for j=1:c
        for k=1:3
            old(i,j,k)=im(i,j,k)/3+im2(i,j,k)-60; %RANDOM TRY
        end
    end
end

subplot(1,2,1);
imshow(original);title("original")
subplot(1,2,2);
imshow(old); title("old effect")
```

Code 3 (Sketch Effect)

```
clc
clear all
close all

im = imread('sketch_in.jpg');

filterX = [-1 -2 -1; 0 0 0; 1 2 1];
filterY = [-1 0 1; -2 0 2; -1 0 1];

%Applying filter on R plane
im1 = double(im(:,:,1));
out1 = zeros(size(im1));
```

```
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im1(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out1(row,col) = val;
    end
end
%Applying filter on G plane
im2 = double(im(:,:,2));
out2 = zeros(size(im2));
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im2(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out2(row,col) = val;
    end
end

%Applying filter for B plane
im3 = double(im(:,:,3));
out3 = zeros(size(im3));
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im1(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out3(row,col) = val;
    end
end

% convert back to uint8
o = uint8(cat(3, out1, out2, out3));
% imshow(o);

%RGB2GRAY conversion
gray = rgb2gray(o);
% imshow(gray);

%inversting color
f = zeros(size(gray));
for i=1:size(gray,1)
    for j=1:size(gray,2)
        f(i,j)=255-gray(i,j);
    end
```

```
end

%output
subplot(1,2,1);
imshow(im);title("original")
subplot(1,2,2);
imshow(mat2gray(f)); title('Sketch Effect')
```

Code 4 (Edge Glow Effect)

```
clc
clear all
close all

im = imread('Edge_Glow_in.jpg');

filterX = [-1 -2 -1; 0 0 0; 1 2 1];
filterY = [-1 0 1; -2 0 2; -1 0 1];

%Applying filter on R plane
im1 = double(im(:,:,1));
out1 = zeros(size(im1));
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im1(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out1(row,col) = val;
    end
end

%Applying filter on G plane
im2 = double(im(:,:,2));
out2 = zeros(size(im2));
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im2(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out2(row,col) = val;
    end
end

%Applying filter on B plane
im3 = double(im(:,:,3));
out3 = zeros(size(im3));
for row = 2 : size(im,1) - 1
    for col = 2 : size(im,2) - 1
```

```
        temp1 = row - 1;
        temp2 = col - 1;
        sub_block = im1(temp1:temp1 + 2,temp2:temp2 + 2);
        valx = sum(sum(sub_block .* filterX));
        valy = sum(sum(sub_block .* filterY));
        val = sqrt(valx.^2 + valy.^2);
        out3(row,col) = val;
    end
end
% convert back to uint8
out = uint8(cat(3, out1, out2, out3));
% imshow(out);

% BLUR EDGE OUTPUT
sigma = 3;
filter = fspecial('gaussian',[3,3],sigma);
imeglow = imfilter(out,filter);
% imshow(imeglow);

% BLEND
r=size(im,1);
c=size(im,2);
a = 0.5;
eglow = uint8(zeros(r,c));
for i=1:r
    for j=1:c
        for k=1:3
            eglow(i,j,k)=a*im(i,j,k)+(1-a)*imeglow(i,j,k); %RANDOM TRY
        end
    end
end

subplot(1,2,1);
imshow(im);title("original")
subplot(1,2,2);
imshow(eglow); title("blend with Original")
```