# Assignment-1: Image Effect Implementation (Part-1)

Sumaiya Tarannum Noor
Student ID: 2425410650
email: sumaiya.tarannum@nothsouth.edu

February 7, 2025

## I. Introduction

In this modern age where images and photos play a significant role in our daily life, image effects can become very useful for conveying messages through using them. One of the most important aspects of using image effect is to increase visual appeal. By creating a lucrative version of an ordinary image, a user can attract more attention to their work. Sometimes, it is much needed because of the natural and artificial light distribution during a photo shoot. For example, using mirror effect in an image can make certain pieces more appealing. One of the most important reasons of using image effect is to convey certain messages. To give an example, vignetting effect can help focusing on the center part of an image. Posterize effect can show a version of any image using limited colors, hch can be really helpful. Also, nightvision and photocopy effect can show two very different aspects of same image.

These effects can also be useful for correcting and enhancing images, creating realistic or artistic expressions, creating psycological and emotional impact etc. No matter what is the reason for using image effects, they have become an integral part of our daily communicational, education and social life. This report aims toward discussing the experiments of using 5 image effects, which are Posterize, Nightvision, Photocopy, Vignetting and Mirror. Gradually we will discuss the methodology of the experiments. For the experimens, we are using MATLAB, we will also discuss the code for creating these effects on a certain image. In the end we will conclude this report with a proper closing discussion.

Refer papers like this [1].

## II. Method

The method of these experiments are simple. We are taking an image which is named ”MY_IMAGE”

Fig. 1: The image that we are using to perform our experiments.

We will be discussing all five image effets one by one below:

### A. Posterize

Posterize effect is the method of represntig an image withing limited color usage. In this process, the colour used in an image gets replaces by the most near colour that we are allowed to use. For example, we are using 4 colours for posterizing an image, an the colours are red, green, blue and yellow. If the given image has pink, it will be replaced with red.

Pixels are represented using the values from [0, 255]. Therefore, there are 256 colours to represent any images. In our experiment, we are using 4 colours. They can also be called as levels. Usually there are 4-8 levels of colours that are used for poterizing an image.

The equation that I have used to perform the Posterizing effect on "My_Image" is shown below:

$$OutputImage, posterized\_im = \frac{\text{round}(InputImage, im \times (numberofcolours - 1))}{(numberofcolours - 1)} \tag{1}$$

Posterize effect can be used in artistic and stylized imaged, like in the Comic books, pop arts, vintage posters etc. It can also be used in edge detection and feature extraction, compression and data reduction. It is very useful in medical and scintific imaging.

### B. Nightvision

Nightvision effect is used to enhance the image visibility in low-light conditions by creating an infrared version. By using the effect the input image gets a green hue all over it. Human eyes are conditioned to see green better than other colors, so it makes it easier for anyone to see the image in the dark.

The question that I have used to perform Nightvision effect on "MY_Image" is shown below:

$$\text{Output Red}, R_{\text{out}} = \frac{\text{Input Green}, G_{\text{in}}}{2} \tag{2}$$

$$\text{Output Blue}, B_{\text{out}} = 2 \times R_{\text{out}} \tag{3}$$

$$\text{Output Green}, G_{\text{out}} = 2 \times B_{\text{out}} \tag{4}$$

The Nightvision image effect is achieved through adjusting RGB colour channels. The night vision effect is widely used in a variety of fields, including security, military operations, and gaming applications, to enhance visibility in low-light or dark environments.

### C. Photocopy

The photocopy effect is a process that transforms an image to look like a photocopy, typically by converting it to grayscale and applying a thresholding technique. In this method, a threshold value is used to determine how pixel values in the image are processed. Generally the values are distributed between 0 and 255. For example, say the threshold value is 100, the pixel values that are less then 100 will be replaced by 0 and the pixel values that are greater than 100 will be replaced by 255. The equation that I have used to perform Photocopy effect is shown below:

$$\text{Output Image}, O = \text{Convert to Grayscale}(I) \tag{5}$$

$$\text{If the pixel value of the input image } I(x, y) > 100, \quad \text{then the output pixel value } O(x, y) = 255 \tag{6}$$

$$\text{If the pixel value of the input image } I(x, y) \leq 100, \quad \text{then the output pixel value } O(x, y) = \frac{I(x, y) \times (100 - I(x, y))}{100^2} \tag{7}$$

The Photocopy effect is used for image simplification and to create high-contrast images. Tasks like document processing, text enhancement and creating artistic designs can be benefitted using this effect.

### D. Vignetting

Vignetting Effect can be used to draw peoples attention onto the Center of an image. Vignetting effect darkens or lightens the corner of the image. The equation that I have used to create vignetting effect is shown below:

$$C_x = \left\lfloor \frac{\text{cols}}{2} \right\rfloor, \quad C_y = \left\lfloor \frac{\text{rows}}{2} \right\rfloor \tag{8}$$

$$M = \sqrt{C_x^2 + C_y^2} \quad \text{where } M \text{ is the maximum distance from the center to any corner.} \tag{9}$$

$$D = \sqrt{(j - C_x)^2 + (i - C_y)^2} \quad \text{where } D \text{ is the distance from the pixel } (i, j) \text{ to the center.} \tag{10}$$

$$\text{Darkness Weight} = 1 - \frac{D}{M} \quad \text{where } D \text{ is the distance from the pixel } (i, j) \text{ to the center, and } M \text{ is the maximum distance.} \tag{11}$$

$$\text{Output Pixel}(i, j) = \text{Input Pixel}(i, j) \times \left(1 - \frac{D}{M}\right) \tag{12}$$

Vignetting effect is used in photography and filming, medical imaging and graphic design and editing.

### E. Mirror

The mirror effect is a digital image processing technique that creates a symmetrical reflection of an image, either horizontally, vertically, or diagonally. This effect is widely used in various fields for artistic, practical, and analytical purposes.

The equation that I used for Mirror effect is shown below:

$$\text{Output Image}(i, j) = \text{Input Image}(i, \text{Total Columns} - j + 1) \tag{13}$$

The mirror effect is a versatile tool with applications in photography, design, gaming, medical research, security, fashion, and architecture. Whether for artistic creativity, scientific analysis, or functional improvements, it remains as a powerful image-processing technique.

## III. Experiment

I have performed all of my experiments using MATLAB. All of the experiment for all five image effects are done using the simlar process. First the image was read and then the equations were used to get the desired results.

### A. Results

All the resultant images are shown in this section for all five image effects:

*1) Posterization:* Applying equation no. 1 in MATLAB, I was able to create the Posterize effect on "MY_IMAGE." The resultant image is shown below:

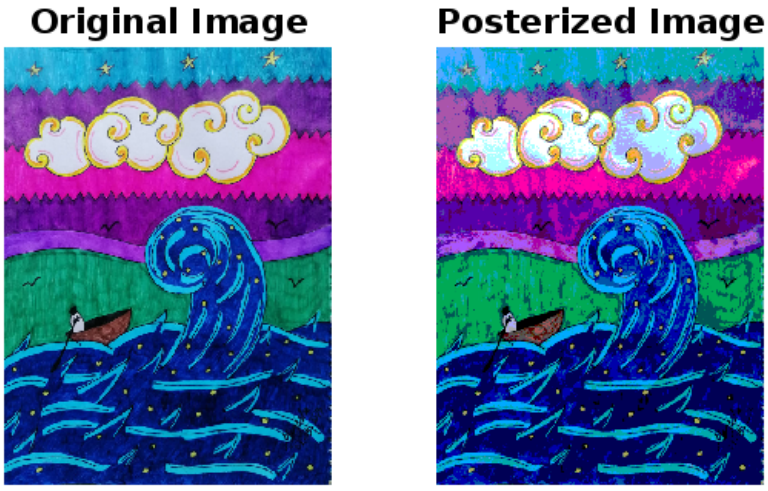


Fig. 2: The Posterize Effect.

*2) Nightvision:* Applying equation for 2, 4, and 3 on ”MY_IMAGE,” I was able to create this Nightvision effect:

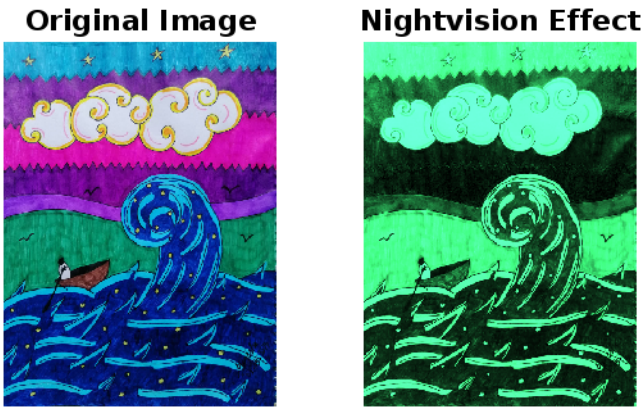


Fig. 3: The Nightvision Effect.

*3) Photocopy:* Applying euation no 7, I was able to create this Photocopy effect on ”MY_IMAGE”:


Fig. 4: The Photocopy Effect.

*4) Vignetting:* Applying equation for 8, 9, 10, 11 and 12 on ”MY_IMAGE,” I was able to create this Vignetting effect:

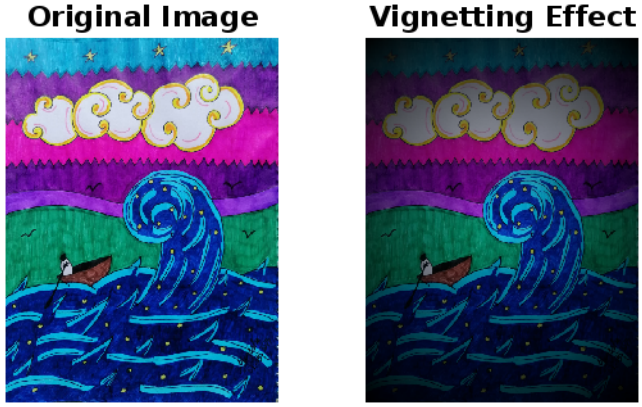


Fig. 5: The Vignetting Effect.

*5) Mirror:* Applying euation no 13, I was able to create this Mirror effect on ”MY_IMAGE”:

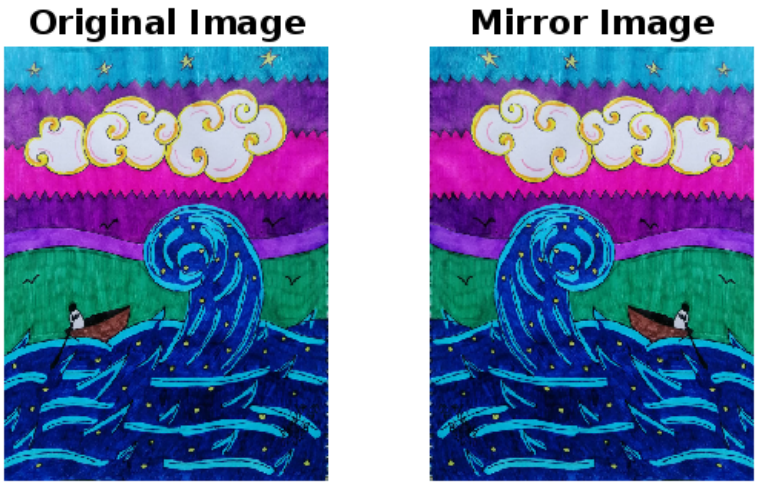


Fig. 6: The Mirror Effect.

## IV. Conclusion

Image processing is a part of our daily life. From taking nice photos to medical imaging to gaming, it is used everywhere. Some of the image processing techniques are called image effects. It is important that we use them in the right way for right purpose.

## V. Appendix

**Code for Posterize:**

```
% Read the image
im = imread('MY_IMAGE.jpg');

% Convert the image to double
im_double = im2double(im);

% Define the number of Colours for posterization
colours = 4;

% Quantize the image
posterized_im = round(im_double * (colours - 1)) / (colours - 1);

% Convert back to uint8 for display
posterized_im_uint8 = im2uint8(posterized_im);

% Display the original and posterized images
figure;
subplot(1, 2, 1);
imshow(im);
title('Original Image');

subplot(1, 2, 2);
imshow(posterized_im_uint8);
title(['Posterized Image']);
```

**Code for Nightvision:**

```
% Read the image
im = imread('MY_IMAGE.jpg');

% Split the image into R, G, B channels
R = im(:,:,1);
G = im(:,:,2);
B = im(:,:,3);

% Apply the transformations
% Output R = Input G / 2
R_out = G / 2;

% Output B = 2 x Output R
B_out = 2 * R_out;

% Output G = 2 x Output B
G_out = 2 * B_out;

% Combine the channels into a new image
nightvision_im = cat(3, R_out, G_out, B_out);

% Convert the image back to uint8
nightvision_im_uint8 = im2uint8(nightvision_im);

% Display the original and nightvision images
figure;
subplot(1, 2, 1);
imshow(im);
title('Original Image');

subplot(1, 2, 2);
imshow(nightvision_im_uint8);
title('Nightvision Effect');
```

**Code for Photocopy :**

```
% Read the image
im = imread('MY_IMAGE.jpg');

% Convert the image to grayscale
gray_im = rgb2gray(im);

% Define the threshold
threshold = 100;

% Initialize the output image
output_im = zeros(size(gray_im));

% Apply the threshold transformation
for i = 1:size(gray_im, 1)
    for j = 1:size(gray_im, 2)
        if gray_im(i, j) > threshold
            output_im(i, j) = 255;
        else
            output_im(i, j) = gray_im(i, j) * (threshold - gray_im(i, j)) / (
                threshold^2);
        end
    end
end

% Convert the output image to uint8 for display
output_im_uint8 = uint8(output_im);

% Display the original, grayscale, and transformed images
figure;
subplot(1, 3, 1);
imshow(im);
title('Original Image');

subplot(1, 3, 2);
imshow(gray_im);
title('Grayscale Image');

subplot(1, 3, 3);
imshow(output_im_uint8);
title('Photocopy Effect');
```

**Code for Vignetting :**

```
% Read the image
im = imread('MY_IMAGE.jpg');

% Get the dimensions of the image
[rows, cols, ~] = size(im);

% Calculate the center of the image
center_x = floor(cols / 2);
center_y = floor(rows / 2);

% Calculate the maximum distance (M) from the center to any corner
M = sqrt(center_x^2 + center_y^2);

% Initialize the output image
output_im = zeros(size(im), 'like', im);

% Apply the vignette effect
for i = 1:rows
    for j = 1:cols
        % Calculate the distance (D) from the current pixel to the center
        D = sqrt((j - center_x)^2 + (i - center_y)^2);

        % Calculate the darkness weight
        weight = 1 - D / M;

        % Apply the weight to each channel of the pixel
        output_im(i, j, :) = weight * double(im(i, j, :));
    end
end

% Convert the output image back to uint8 for display
output_im_uint8 = uint8(output_im);

% Display the original and vignetted images
figure;
subplot(1, 2, 1);
imshow(im);
title('Original Image');

subplot(1, 2, 2);
imshow(output_im_uint8);
title('Vignetting Effect');
```

**Code for Mirror :**

```
% Read the image
im = imread('MY_IMAGE.jpg');

% Reverse the order of the pixels in each row to create a mirror image
mirror_im = im(:, end:-1:1, :);

% Display the original and mirrored images
figure;
subplot(1, 2, 1);
imshow(im);
title('Original Image');

subplot(1, 2, 2);
imshow(mirror_im);
title('Mirror Image');
```

## REFERENCES

[1] M. Chen, Z. Kira, G. Alregib, J. Yoo, R. Chen, and J. Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, pages 6320–6329. IEEE, 2019.