Final Report - Team Bellissimo

Project ID: 14

Project Title: Weak-Light Image Enhancement Method Based on Adaptive Local Gamma Transform and Color Compensation

Github Link: https://github.com/Digital-Image-Processing-IIITH/dip-project-bellissimo

Team Members

1. Ainesh Sannidhi (2019101067)
2. Anandhini Rajendran (2019101055)
3. Kunwar Maheep Singh (2019101075)
4. Mayank Jain (2019101023)

Weak light image enhancement using adaptive gamma transform:
For a general image with uneven low light, a gamma transform will not enhance the image evenly. As such an adaptive gamma transform approach was followed in this paper.

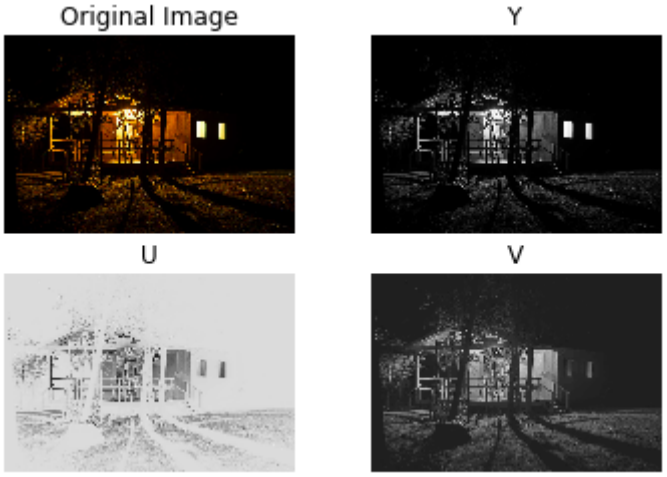
In the illumination-reflection model the brightness of any pixel in an image can be considered a product of its illumination component(I) and the reflective component(R).

$$F(x,\ y)\ =\ I(x,\ y)\ *\ R(x,\ y)$$

The illumination component spectrum is usually concentrated in a low frequency region which reflects the lighting environment during image capture. If this can be extracted from the image and made even then the effect of uneven lighting can be removed from the image.

1. **Conversion from RGB to YUV space**
   Human eyes are more sensitive to luminescence than color. As such the paper aims to enhance luminescence to correct uneven illumination. This information cannot be captured effectively in RGB space and therefore we work in the YUV space where each color corresponds to two chrominance components(U, V) and one brightness component(Y). This luminescence component(Y) will be enhanced to fix uneven lighting while leaving the U and V components unchanged.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$



2. **Extraction of illumination component using fast guided filter**

We need to extract the illumination component from a scene to correct the effect of uneven lighting. Many different algorithms were considered like mean filter, gaussian filter, bilateral filter. Out of these mean and gaussian filters did not do a good job of preserving the original image and bilateral filter while edge preserving was not computationally feasible. Finally, fast guided filter was chosen as it was both fast and edge preserving.

Here for input image p, output image q and guide I, for every pixel s we have:

$$q_j = a_s * I_j + b_s, \forall_j \in \omega_s$$

Where j is the pixel index and a_s and b_s are linear transform factors.

The reconstruction difference between p and q is minimized for the following choice of a and b:

$$a_s = \frac{1/|\omega| \sum I_j p_j - \mu_s \bar{p}_s}{\sigma_s^2 + \xi},$$

$$b_s = \bar{p}_s - a_s \mu_s,$$

where ω_s and μ_s are the variance and the mean value of the guided image I within the window ω_s, respectively; ξ is a parameter that controls the degree of smoothness of the filter. |ω| the pixel number of ω_s; and $\bar{p}_s$ is the mean value of the input image p.

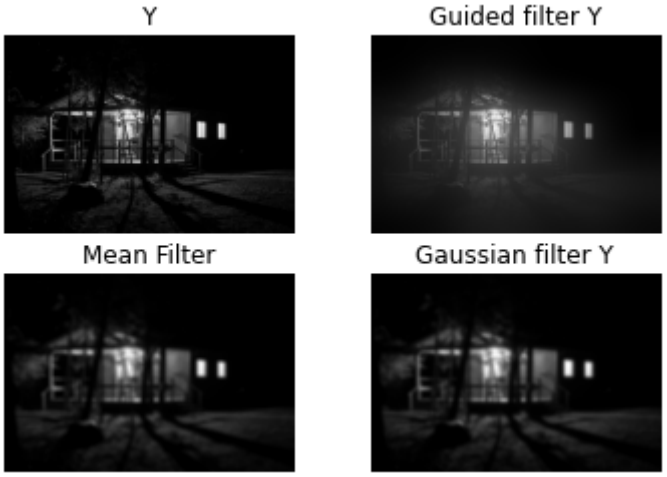
This gives output:

$$q_j = \frac{1}{|\omega|} \sum_{s:j \in \omega_s} (a_s I_j + b_s),$$

$$q_j = \bar{a}_j I_j + \bar{b}_j,$$

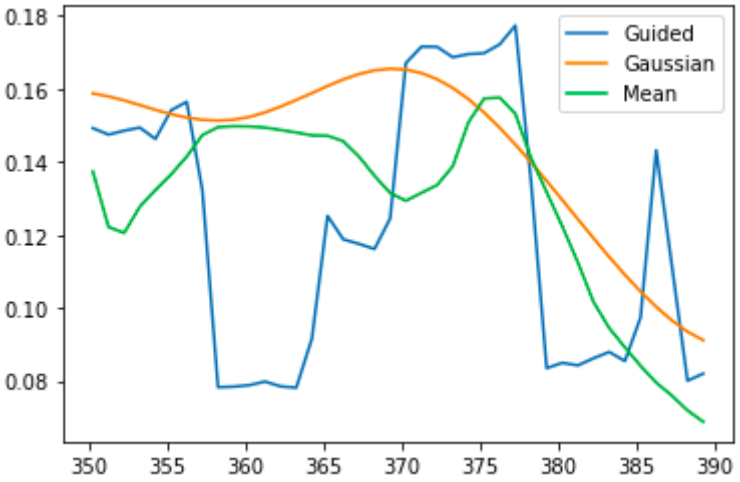
where $\bar{a}_j$ and $\bar{b}_j$ are the mean values of a and b, respectively, within the neighborhood window ω_s centered on pixel j.

In the paper the guidance image was not specified, neither was a standard given to choose the parameter ξ. After reading about self guided filter and using some hints from the flowchart in the paper, we assumed the guide to be the image itself and ξ to be the standard deviation of all the pixels in the image.



Instead of multiscale illumination which is the weighted mean of guided filters with different window sizes, single scale illumination was chosen for computational benefit with window size as $floor(min(h,\ w)\ /\ 4)$ where $h,\ w$ are the dimensions of the image.

We obtain final output $G = Y * G^F$ where G^F is the guided filter.



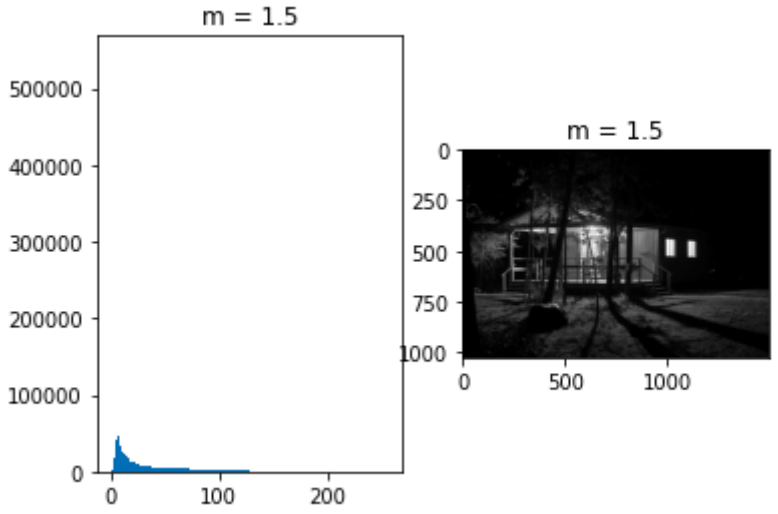
One-dimensional plot of the illumination components extracted with different algorithms.

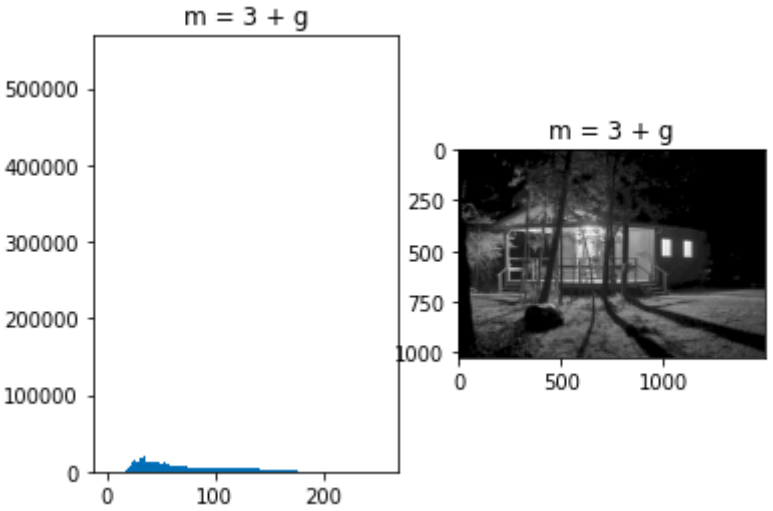
3. **Local gamma transform**

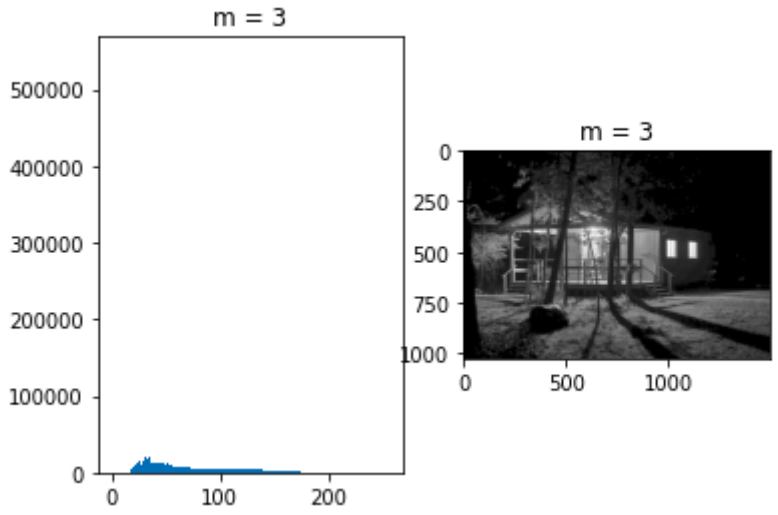
Traditional gamma transform can give reasonable results for uniform underexposed or overexposed images but for non uniform exposure we use local gamma transform. Hence we choose $\gamma = m^{2G(x,y)-1}$ where G(x, y) is the illumination extracted component. For images with excessively high contrast, a γ value greater than 1

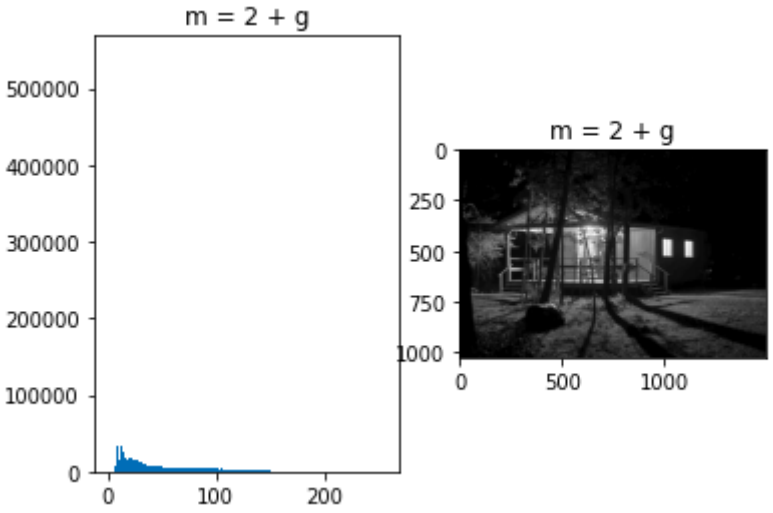
should be adopted, i.e., the value of m should be low, to suppress the illumination intensity. As such an adaptive brightness adjustment function based on local gamma transformation was proposed -

$$O(x,\ y)\ =\ Y(x,\ y)^{\ (2\ +\ G(x,\ y))^{\ [2*G(x,\ y)\ -\ 1]}}$$









4. **Grayscale linear stretching**
   To resolve the problem of gray value concentration, we use simple linear stretching to include the entire grayscale range.

$$Y'(x,y) = \frac{(1 - L_{\min})}{L_{\max} - L_{\min}} O(x,y) + \frac{(L_{\max} - 1)L_{\min}}{L_{\max} - L_{\min}}.$$

   where Y' is the Y component after local gamma transform.

.

After converting to RGB

5. **Color compensation**
   The YUV space is converted back to RGB using the following matrix:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.001 \end{bmatrix} \times \begin{bmatrix} Y' \\ U \\ V \end{bmatrix},$$

   After conversion, decrease in color saturation is fixed using the following transforms:

$$\begin{cases} R' = \varepsilon \times \left[\left(\frac{Y'}{Y} \right) \times (R + Y) + R - Y \right], \\ G' = \varepsilon \times \left[\left(\frac{Y'}{Y} \right) \times (G + Y) + G - Y \right], \\ B' = \varepsilon \times \left[\left(\frac{Y'}{Y} \right) \times (B + Y) + B - Y \right], \end{cases}$$

   with $\varepsilon = 0.5$

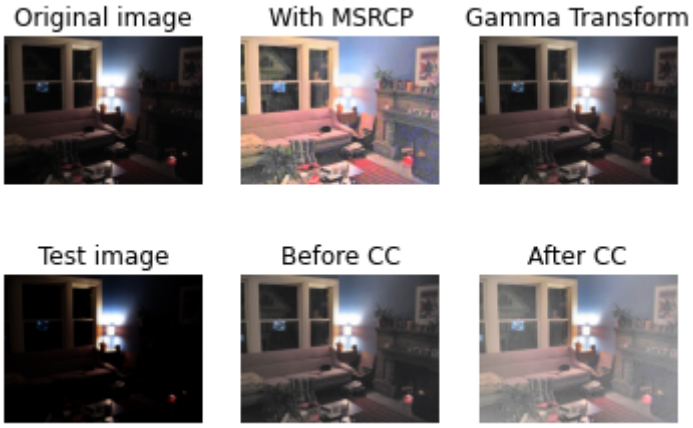
In our tests this led to over-saturation of the images. The results are as follows:

Comparison with other algorithms:
The test image was obtained by a gamma transform with a random value between 1.5 and 2.5.

Result 1:

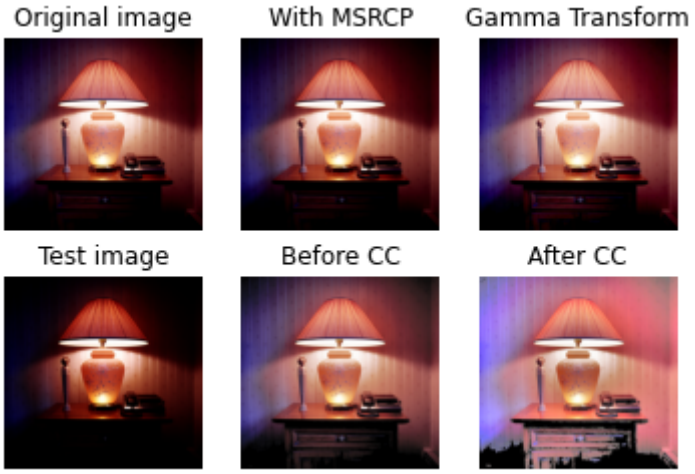
	Gamma Transform	**MSRCP**	**Before color compensation**	**After color compensation**
MSE	2218.37	11063.56	3050.42	13366.66
PSNR	14.67	7.69	13.28	6.87
SSIM	0.42	0.32	0.32	0.29

Result 2:



	Gamma Transform	MSRCP	Before color compensation	After color compensation
MSE	6680.73	23485.68	7769.88	22794.14
PSNR	9.88	4.42	9.22	4.55
SSIM	0.0057	0.0008	0.0027	0.0011

Result 3:



	Gamma Transform	MSRCP	Before color compensation	After color compensation
MSE	11225.37	8152.25	10624.57	22223.53
PSNR	7.62	9.01	7.86	4.66
SSIM	0.15	0.23	0.07	0.04

APPLICATION:

Fingerprint Image Enhancement And It's Feature Extraction For Recognition

Fingerprint recognition is the widely used biometric solution for authentication on computerized systems. The probability of two fingerprints matching is very small ~ 1 in $1.9 * 10^{15}$. Image enhancement is a step in fingerprint extraction. So we are replacing the enhancement used in the paper Fingerprint Image Enhancement And It's Feature Extraction For Recognition with Weak-Light Image Enhancement Method Based on Adaptive Local Gamma Transform and Color Compensation and comparing the results.

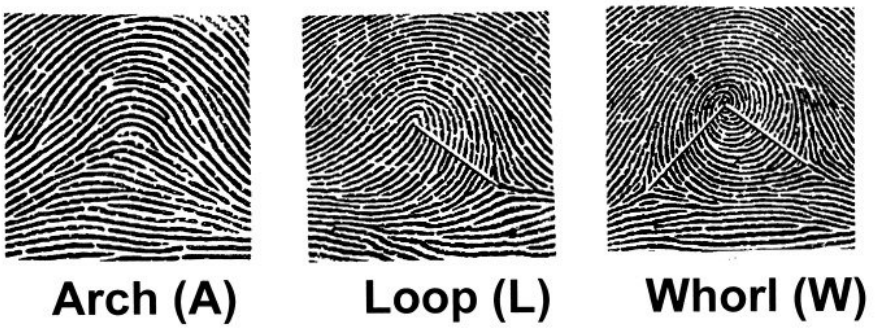
Keywords:

- **Ridge**
  It is a curved line in a finger image. Some ridges are continuous curves and others terminate.

- **Minutiae**
  Minutiae refer to specific points in a fingerprint. These are the small details in a fingerprint that are most important for fingerprint recognition. The **ridge ending** is, as indicated by the name, the spot where a ridge ends. A **bifurcation** is a spot where a ridge splits into two ridges. **Spots** are those fingerprint ridges that are significantly shorter than other ridges.

Patterns:

1. **Arch:** a pattern where the ridge enters one side of the finger, then rises in the center forming an arch and exits on the other side of the finger.
2. **Loop:** In a loop, the ridge enters one side of the finger, then forms a curve, and exits on the same side of the finger from which it entered.
3. **Whorl:** The pattern you have when ridges form circularly around a central point.



Method

Image Acquisition

In this step, fingerprint images are obtained from different sensors. These are of poor quality hence we need the enhancement step. The FVC2002 fingerprint dataset is a Fingerprint Verification Competition dataset that was organized back in the year 2000 and then again in the year 2002. This dataset consists of four different sensor fingerprints namely Low-cost Optical Sensor, Low-cost Capacitive Sensor, Optical Sensor, and Synthetic Generator, each sensor having varying image sizes. The dataset has 3200 images in set A, 800 images per sensor. We are taking 3 images from this dataset for further steps.

Image enhancement

Ridges and valleys alternate and flow in a locally consistent direction in an ideal fingerprint picture. An enhancement algorithm's purpose is to increase the clarity of the ridge structures in recoverable parts while marking unrecoverable sections as too noisy to process further. Here we use two enhancement methods to compare the results.

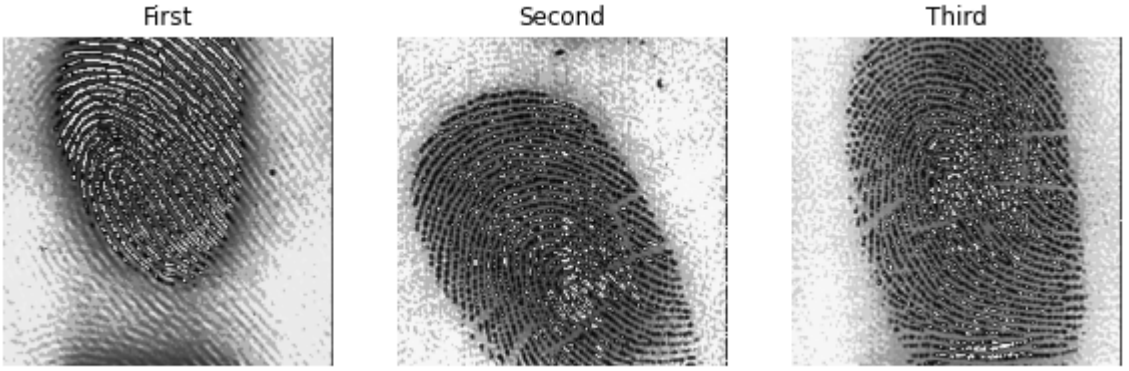
Method 1: Using Fourier domain filtering and histogram equalization.

Fourier Domain Filtering

First, we split images into frequency domain F (u, v) and multiply this frequency domain with a constant such as |F(u, v)| ^k. After doing this, we take inverse Fourier to this altered frequency domain to get the enhanced image.

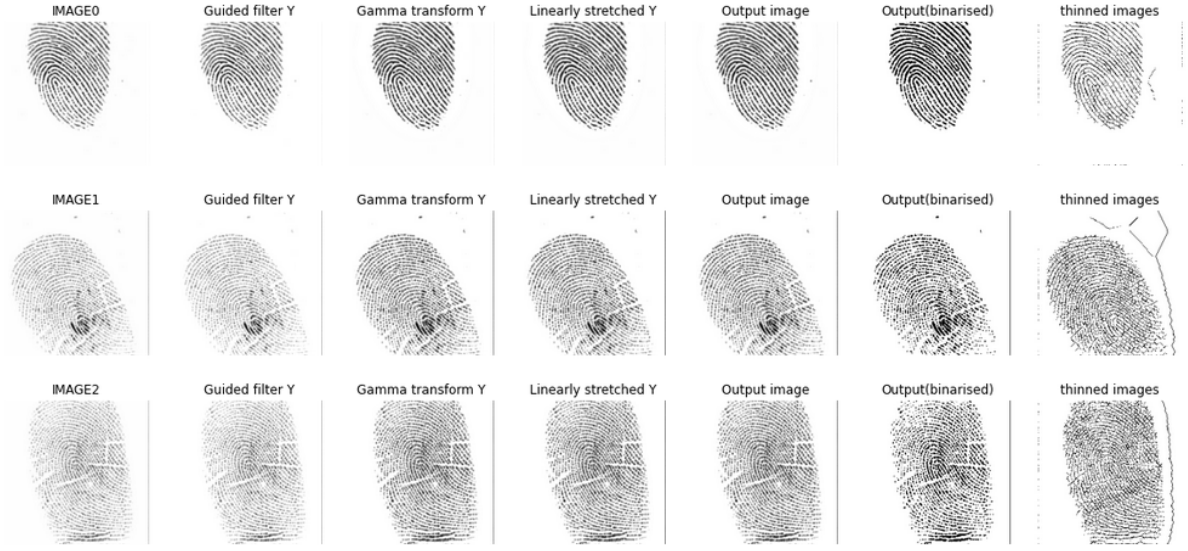
HISTOGRAM EQUALISATION

Next, we apply histogram equalization to get the enhanced image.



Method 2: Using Weak-Light Image Enhancement Method Based on Adaptive Local Gamma Transform and Color Compensation

In this method, we use the YUV space for enhancing images. First, we convert the image from RGB to YUV space, apply guided filtering and local gamma transform on the Y component. Then we apply linear contrast stretching on the Y component. Then we convert back to YUV space.



Binarization

In this process, we convert gray images to binary images. Most minutiae extraction algorithms operate on binary images where the black pixels represent ridges and the white pixels represent valleys. We are using thresholding to convert to a binary image.

Thinning

The images obtained from method 1 need morphological operations like erosion or dilation to obtain a thinned image. We use the process of skeletonization to obtain an image with 1-pixel wide lines.

Steps:

1. Create an empty skeleton
2. Calculate the opening of the image: open
3. Subtract the open from the original image: temp
4. Erode the original image and refine the skeleton using skeleton ∪ temp
5. Repeat steps 2-4 until the original image is completely eroded.

Feature Extraction

The total of the neighbor pixels at every point on a rigid line must equal 6. The summation will be altered if the rigid line is ended, and it will be 7. In a point of bifurcation, the summation will be 5. We must consider a 3x3 window that will scan the entire image with the center pixel dark or zero.

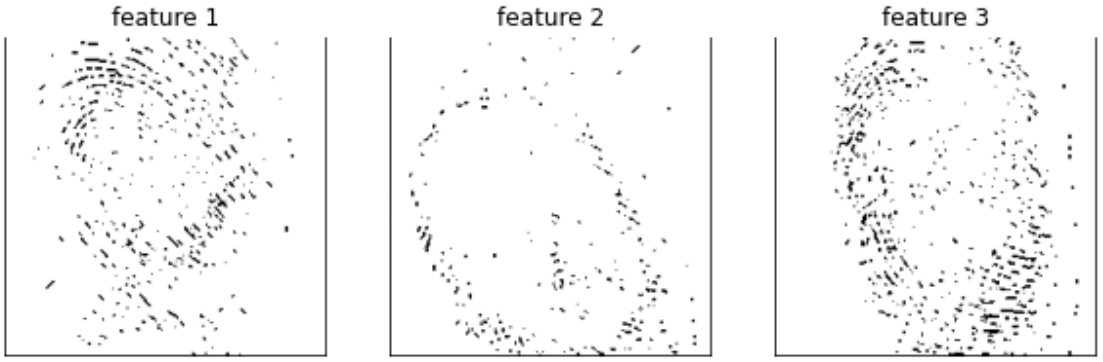
$$P_c = \frac{1}{255}\sum_{i=1}^{8} P_i$$

The sum of 8 neighboring pixels is used to find the minutiae points.

- sum = 6: rigid line
- sum = 5: bifurcation
- sum = 6: termination

The final feature space will contain all bifurcations and terminations. These are features required for fingerprint recognition.

Features Method 1: Using Fourier domain filtering and histogram equalization.



Features method 2: Using Weak-Light Image Enhancement Method Based on Adaptive Local Gamma Transform and Color Compensation

Result:
For the data tested replacing Fourier domain filtering and histogram equalization with weak light image enhancement based on adaptive local gamma transform extracted better features and a more accurate skeleton.