# Feature extraction

## Laplacian of Gaussian for edge detection

<giới thiệu về LOG>

1. Gaussian blur for image smoothing

Firstly, gausian filtering is applied to the region of interest (ROI) to achieve image blurring, which serves as a foundational step in enhancing feature detection by blur the small details

The Gaussian filter is a low-pass filter that attenuates high-frequency noise while preserving the overall structure of the image. By applying this filter to the palm print's ROI, the texture variations caused by external factors, such as minor wrinkles or noise, are smoothed out. This smoothing not only reduces unwanted noise but also highlights the broader patterns of ridges and valleys in the palm print, which are crucial for subsequent feature extraction steps.

<math part optional>

The mathematical basis of the Gaussian filter involves convolving the image with a Gaussian kernel, defined as:

G(x,y)=12πσ2e−x2+y22σ2G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}G(x,y)=2πσ21​e−2σ2x2+y2​

Where:

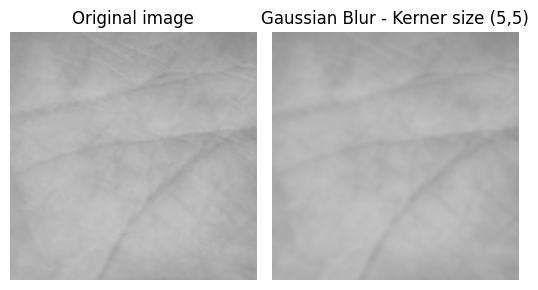
* G(x,y)G(x, y)G(x,y) is the Gaussian function at pixel (x,y)(x, y)(x,y),
* σ\sigmaσ is the standard deviation of the Gaussian distribution, controlling the degree of smoothing. A larger kernel size or higher σ\sigmaσ value would result in greater blurring, which could obscure finer details, whereas smaller values may fail to sufficiently suppress noise.

<math part optional>

In our implementation, the Gaussian filter parameters were carefully chosen to balance noise reduction and feature preservation. A larger kernel size or higher σ\sigmaσ value would result in greater blurring, which could obscure finer details, whereas smaller values may fail to sufficiently suppress noise.

the kernel size for the Gaussian filter was set to xxx, with the standard deviation (σ\sigmaσ) adjusted to achieve a balance between noise suppression and feature preservation. This step ensured that the palm print’s broader ridge and valley patterns were emphasized while finer, irrelevant details were subdued.

Once the Gaussian filter was applied, the ROI exhibited a smoother texture, making the unique patterns of the palm print more distinguishable for subsequent stages of processing, such as feature encoding and matching. The results confirmed that Gaussian filtering effectively enhances the clarity and quality of the ROI without compromising the critical features necessary for authentication.



2. Laplacian

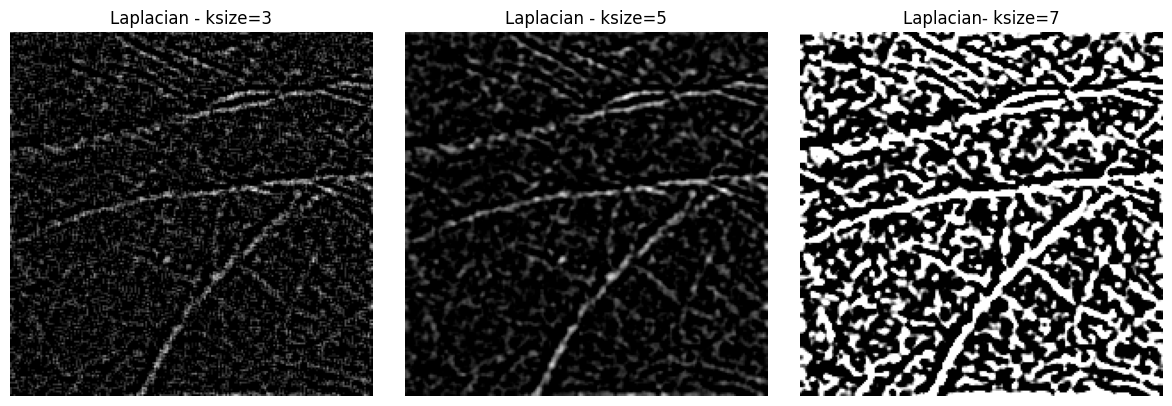
Following Gaussian blurring, the Laplacian of Gaussian (LoG) method was applied to detect edges in the ROI. The LoG technique combines Gaussian smoothing with the Laplacian operator, effectively identifying areas in the image where intensity changes abruptly.

The Laplacian operator calculates the second derivative of the image, and its kernel emphasizes regions of rapid intensity change, which correspond to edges. By first applying Gaussian smoothing, the noise in the image is reduced, minimizing false edge detection. The LoG kernel is defined as:

LoG(x,y)=−1πσ4(1−x2+y22σ2)e−x2+y22σ2\text{LoG}(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}}LoG(x,y)=−πσ41​(1−2σ2x2+y2​)e−2σ2x2+y2​

In our approach, we utilized a Gaussian kernel of size xxx and a Laplacian kernel of size yyy. The combined effect of these kernels allowed for robust edge detection, focusing on the primary patterns and structures in the palm print, such as ridges and minutiae.

I also have tried several kerner size and found that this is the most effective size for palm print edge detction



## Gabor filter for edge detection

1. Introduction

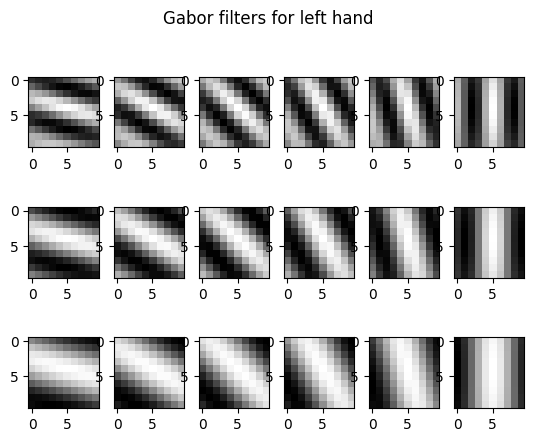
<about gabor wavelets>

2. implementation [ref]

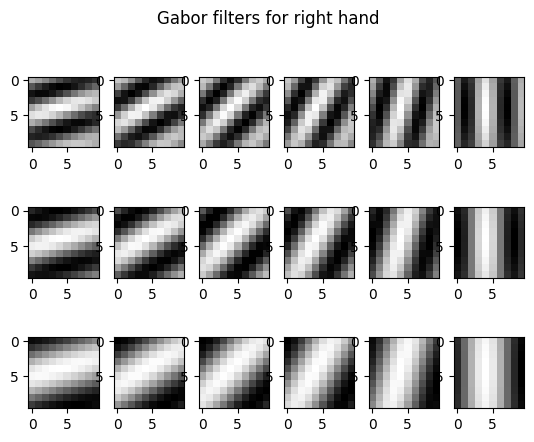
a. Creating gabor filter

The first step is to generate a set of Gabor filters tailored for the palm print images. A custom Gabor filter class (Gabor) was implemented, which takes as input the number of rows and columns (R and C) for the filter size, the number of orientations, and the scale for the wavelets. The filter design uses six different orientations (0°, 30°, 60°, 90°, 120°, and 150°) to capture ridge information from different directions, and two scales to capture both fine and coarse ridge details.

The class calculates a set of Gabor filters by iterating over the specified orientations. For each orientation, the gabor\_wavelet function is called, which generates the Gabor wavelet by calculating its frequency components and applying a Gaussian envelope. The filter is then optionally flipped (useful for distinguishing between left and right palms). The resulting Gabor filters are stored in a set and are ready for use in the next step.



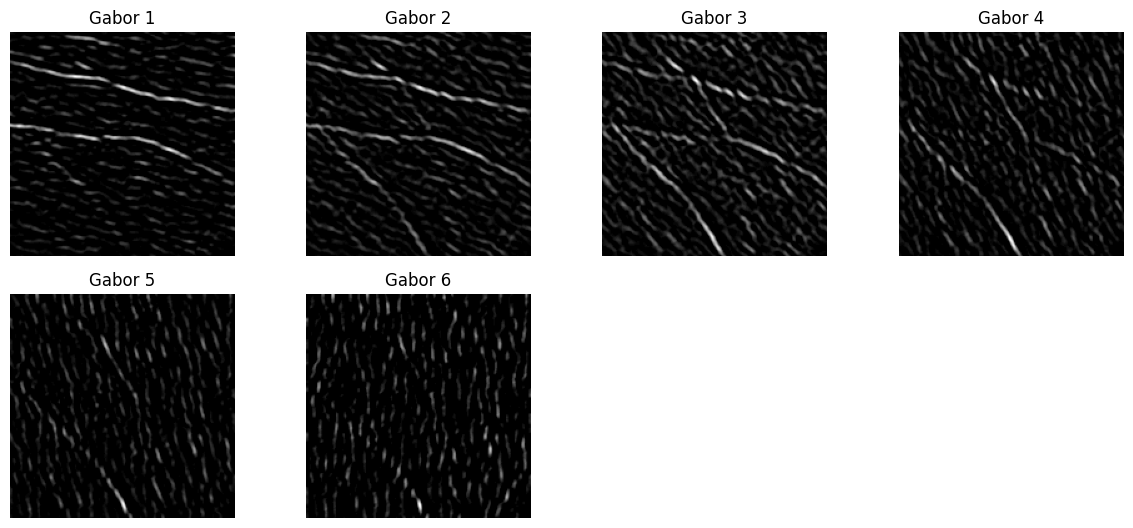
Due to the different in the direction of palm vein between to hand, another gabor set of filter is generated with another direction which is suitable for right hand



b. Applying gabor filters to palm print images

Once the Gabor filters are designed, they are applied to the palm print images. For each hand, a corersponding filters set will be applied

The application of the filters involves convolving the palm print with each of the Gabor filters. This process highlights the ridge patterns at different orientations and scales. Each filter response produces a feature map that represents the strength of the ridges in the palm print at the corresponding orientation and scale.



c. Post processing to remove Small object and refine ridge detection

## Histogram of oriented gradient

# Reference