# Image Restoration – A Use Case

Thai Quang Phat

Supervisor: Prof. Nguyen An Khuong, PhD.Candidate Nguyen Hoang Minh

November 22, 2024

# Outline

- Problem
- Approach
- Experimental Results
- Conclusion
- Limitations & Future Work

# Problem

- Problem Statement
- Assumptions

# Problem Statement

Given a blurred image of a heart, the task is to recover the degraded function. It is known that, the bottom right corner crosshair image before degraded, is 3 pixels wide, 30 pixels long, and had an intensity of 255.

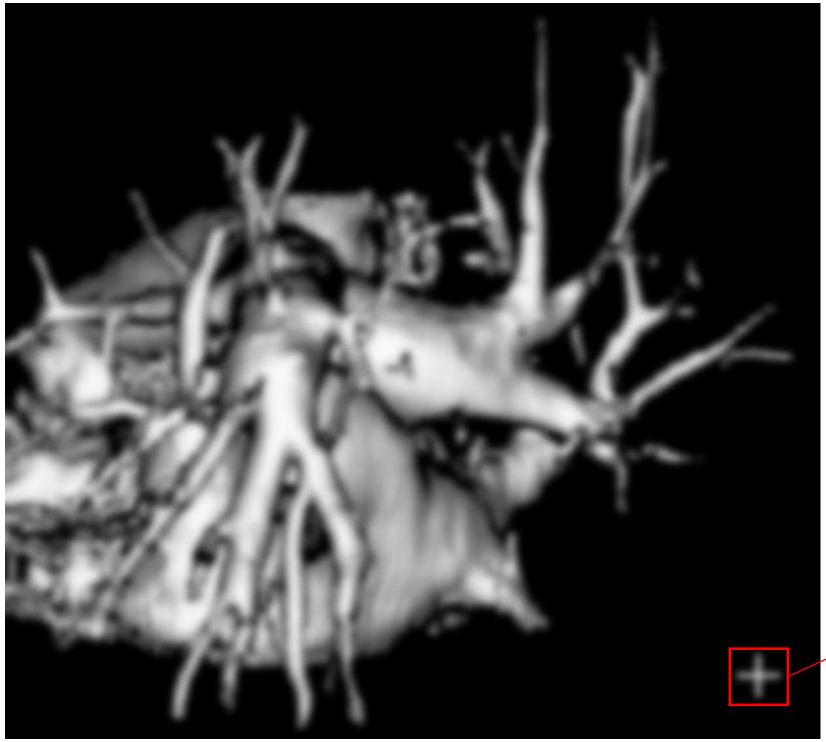


Figure. Blurred heart

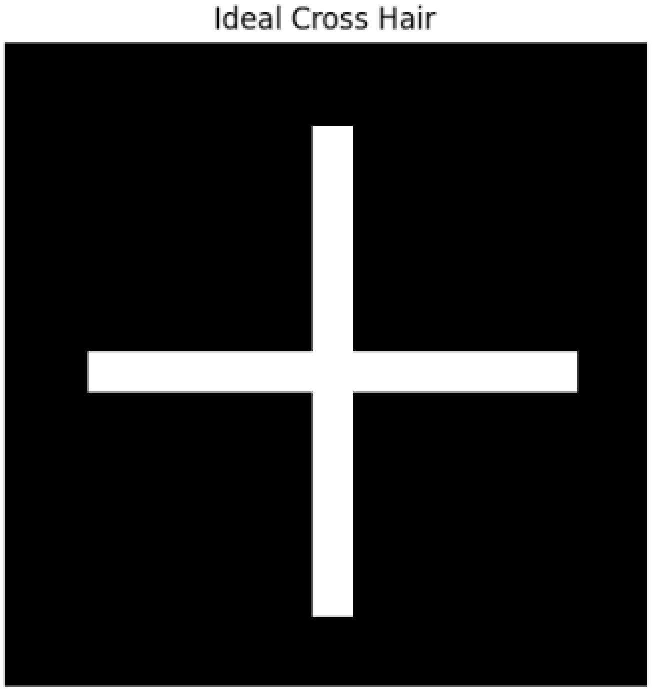



Figure. Crosshair before blur

# Assumptions

Upon solving the problem, we consider the following assumptions

1. **No knowledge of the original heart image:** We do not have the detailed heart image, thus at the restored result, we accept to our understanding.
2. **Gaussian or Butterworth:** We assume the blurred image is filtered using either Gaussian or Butterworth lowpass filters, other methods are not considered.

# Approach

- Phase explanation

# Phase Explanation. Noise Remove

We use the contraharmonic mean filter. This filter specialize in removing any salt-and-pepper type noise. The formula for the filter is given as

$$\boxed{\hat{f}(x,y) = \frac{\sum_{(r,c)\in W} g(r,c)^{Q+1}}{\sum_{(r,c)\in W} g(r,c)^{Q}}}$$

In which

- $\hat{f}(x,y)$ is the output pixel value at position $(x, y)$.
- $g(x,y)$ is the input pixel at position $(x, y)$.
- $W$ is the neighborhood of the pixel $(x, y)$.
- $Q$ is the order of the contrast filter, where
  - $Q > 0$ removes pepper noise (dark pixels).
  - $Q < 0$ removes salt noise (bright pixels).

We choose $Q = 2$, and the neighborhood W of each pixel will be its 8-neighbor pixels.

# Phase Explanation. Noise Remove

The purpose of this phase is to smoothened the image, since as observed, the heart image after blurring is affected by small noises which are originally from the image, this phase helps removes those noises, resulting a better result output.

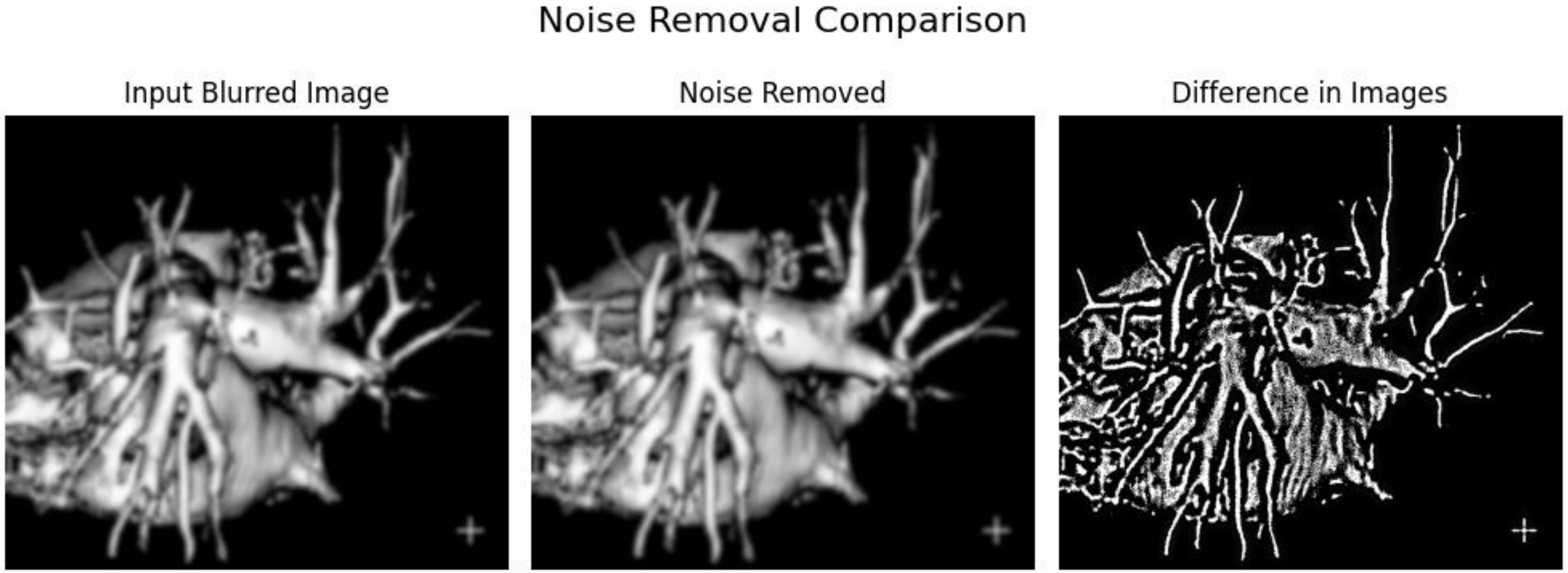
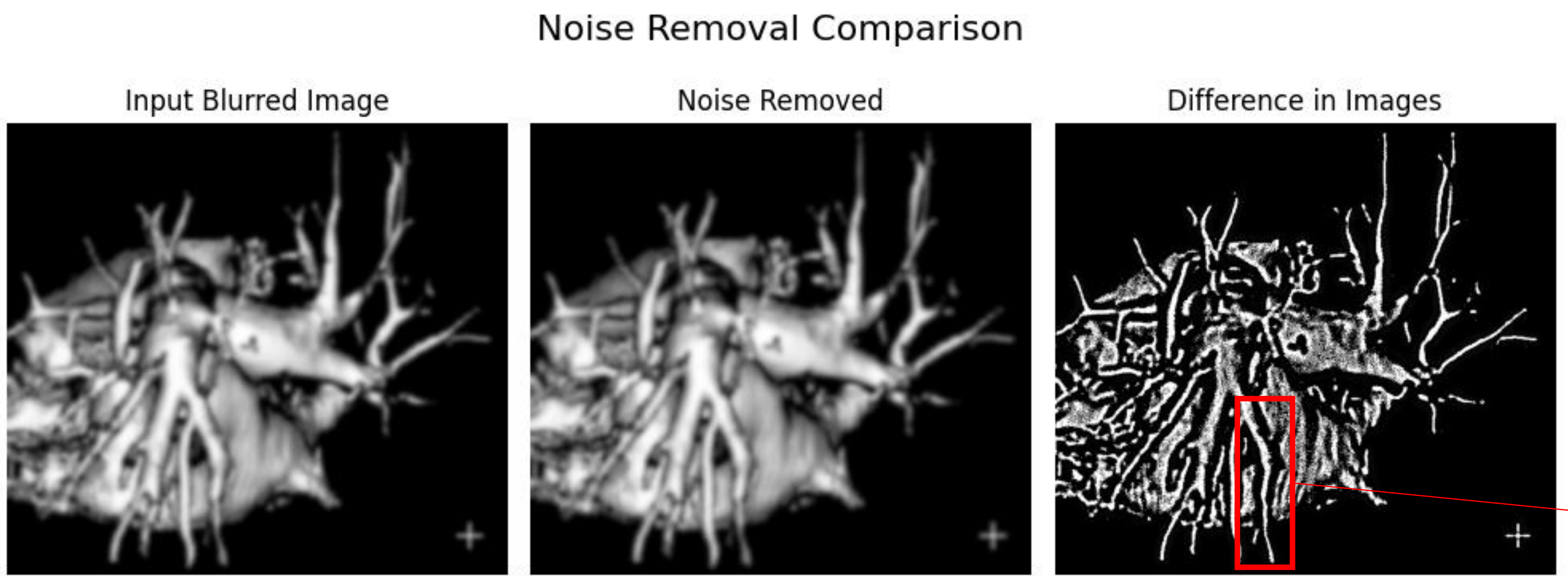



Figure. Noise removed and the difference image

# Phase Explanation. Noise Remove




As observed from the difference image, we have the following conclusions

1. The black lines are where there are no difference, which means there is a high probability that there is no noise.
2. The bright areas shows a dotted style, which can be treated as noise, removing this enhances the resulting image.

# Phase Explanation. Blurred Crosshair Cropped

This step is an important step, as it is the only information of the image that we have. We attempt by cropping out the blurred crosshair.

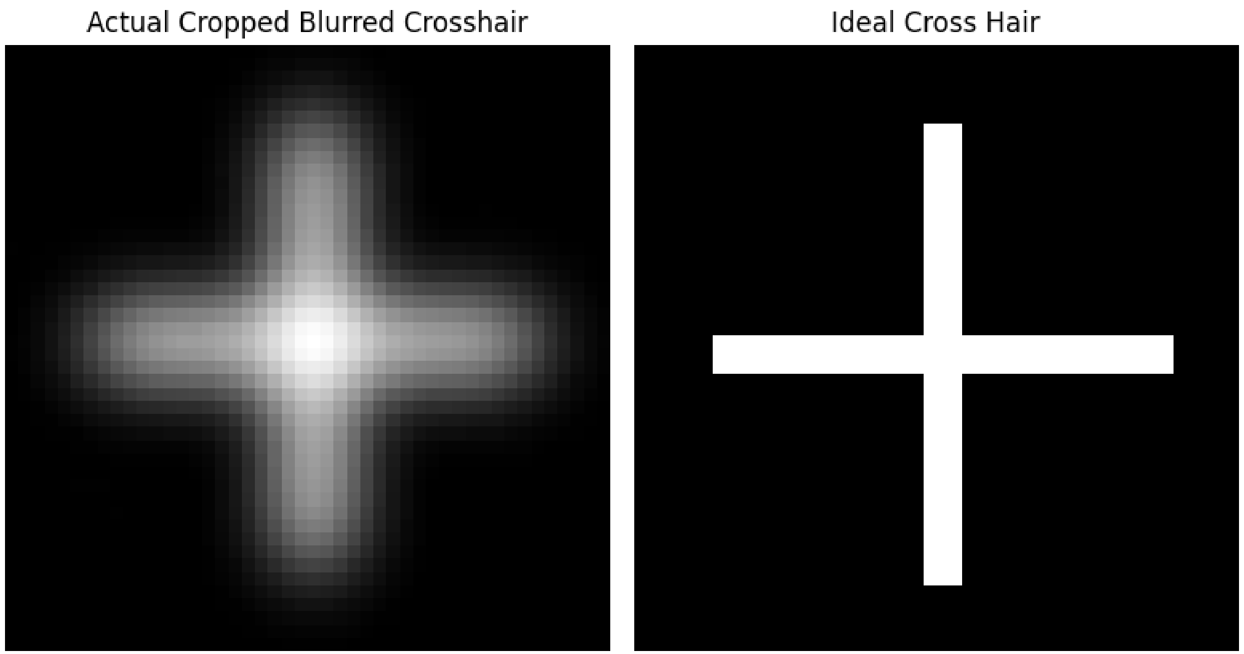



Figure. The cropped image compared to its original

# Phase Explanation. Fourier Transformed

Since we are working on the frequency domain, it is crucial to apply Fourier transform before applying any filters.

Let *f* be any input image size *M* x *N*, we denote *F* as its Fourier transform, which is obtained by

$$\boxed{F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)}}$$

Fourier transform allows us to convert from the spatial domain, to the frequency domain. We do not discuss about why in this particular presentation.

# Phase Explanation. Estimated Gaussian Filter

Let us define $F_{\text{blurred}}$ and $F_{\text{ideal}}$ respectively represents the Fourier Transformed of the blurred cropped crosshair and the ideal crosshair, $H$ be the estimated Gaussian filter.

$$\boxed{F_{\text{blurred}} = F_{\text{ideal}} \cdot H \Leftrightarrow H = \frac{F_{\text{blurred}}}{F_{\text{ideal}}}}$$

We use the following Gaussian filter

$$\boxed{H(u,v) = e^{\frac{-D(u,v)^2}{2D_0{}^2}}}$$

Where $D_0$ is the cutoff frequency, $D(u,v)$ is the distance from each pixel to the center of the image, in this case the center of the Fourier Transformed image.

# Phase Explanation. Estimated Gaussian Filter

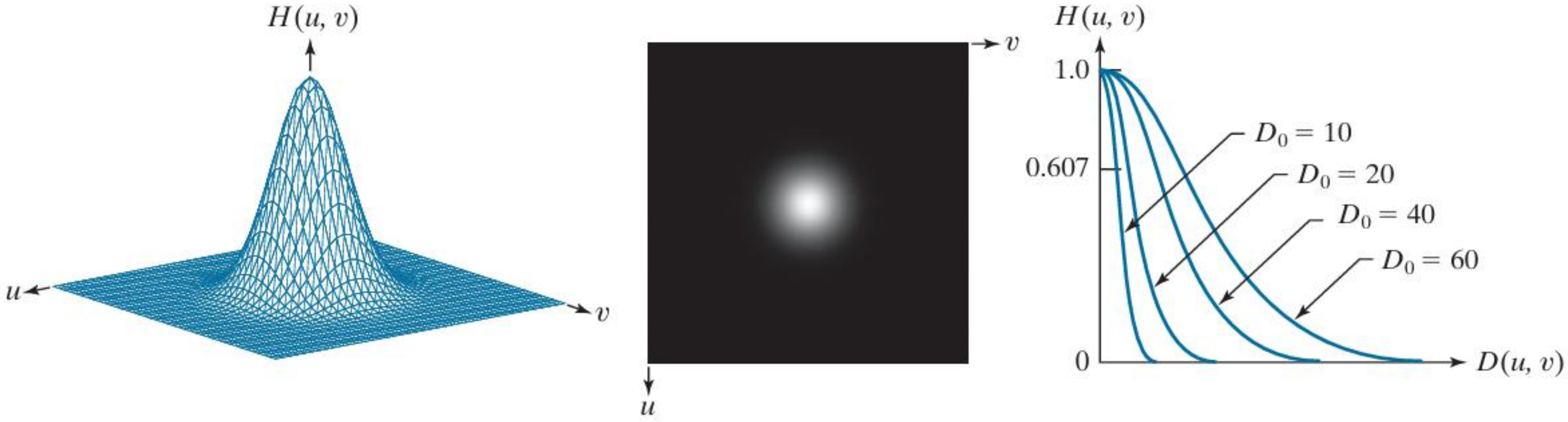



Figure. The Gaussian Lowpass Filter visualization and the cutoff frequency

# Phase Explanation. Estimated Gaussian Filter

Recall that we have the following equivalent

$$\boxed{H(u,v) = e^{\frac{-D(u,v)^2}{2D_0^2}} \Leftrightarrow D_0 \approx \sqrt{\left|\frac{-D(u,v)^2}{2\ln H(u,v)}\right|}}$$

Thus, to obtain the estimated cutoff frequency, the simplest approach is to take the average

$$\boxed{D_0 = \frac{1}{M \cdot N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sqrt{\left|\frac{-D(u,v)^2}{2\ln H(u,v)}\right|}}$$

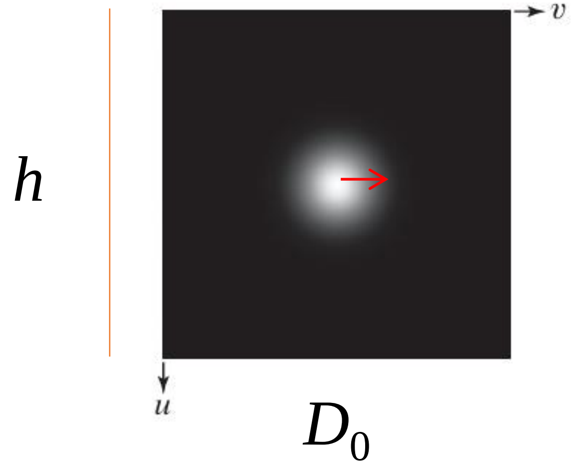
# Phase Explanation. Estimated Gaussian Filter

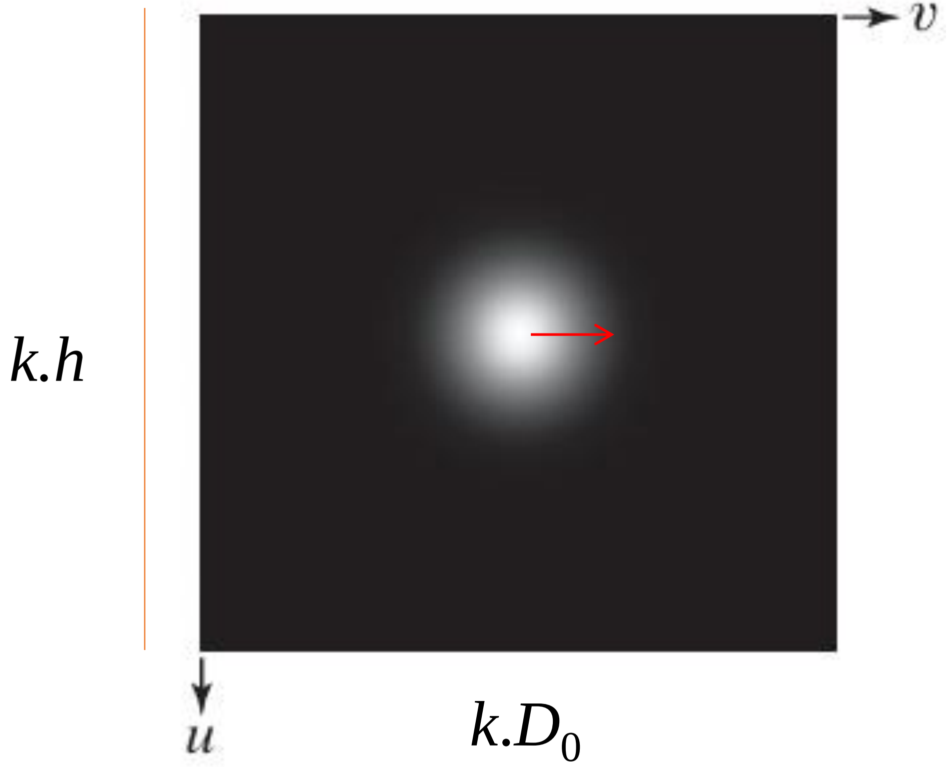
From here on, for simplicity, we let the following term be $D_0$

$$\boxed{D_0 = \frac{1}{M \cdot N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \sqrt{\left| \frac{-D(u,v)^2}{2 \ln H(u,v)} \right|}}$$

# Phase Explanation. Scale for Input Image

We take advantage of the cutoff frequency mentioned earlier, for a larger scale image, we would like to scale the cutoff frequency approximately as the ratio between the large and cropped image.




This encourage us to just scale the cutoff frequency.

# Phase Explanation. Scale for Input Image

Thus, the estimated *H’* Gaussian filter for the input heart image will have the following formula

$$\boxed{H(u,v) = e^{\frac{-D(u,v)^2}{2D_0^2}}} \xrightarrow{\text{Scaling}} \boxed{H'(u,v) = e^{\frac{-D(u,v)^2}{2(kD_0)^2}}}$$

# Phase Explanation. Restored Image

Having the estimated $H'$, obtained from scaling the cutoff frequency $D_0$ from $H$, we can obtain the restored image, using the Wiener filtering

$$\boxed{F_{\text{restored}} = \frac{F_{\text{input}} \cdot \bar{H}'}{|H'|^2 + K}}$$

$F_{\text{input}}$, $F_{\text{restored}}$ respectively be the Fourier transform of the input and the restored image, $K$ will be a constant being estimated.

# Phase Explanation. Restored Image

This phase is not necessary for the problem itself, but it is a way to verify whether the obtained function is acceptable or not.

# Phase Explanation. Convert to Spatial Domain

Lastly, we use the inverse Fourier transform to bring the image back from the frequency domain to its spatial domain.

$$\boxed{f_{\text{restored}}(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{n=0}^{N-1} F_{\text{restored}}(u, v) e^{i2\pi(ux/M + vy/N)}}$$

# Phase Explanation. Post Processing

We introduce two extra steps in enhancing the image quality: brightening and sharpening.

For brightening, we use thresholding, a technique which selects pixels at a certain level of intensity and add to its intensity.

$$f_{\text{brighten}} = \begin{cases} f_{\text{restored}} + \alpha & \text{if } \; f_{\text{restored}} \geq t \\ f_{\text{restored}} & \text{otherwise} \end{cases}$$

Where *a* is the added intensity, *t* is the threshold level.

# Phase Explanation. Post Processing

For sharpening, we opt to use the Laplacian sharpening kernel, and apply convolution to the restored image

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Apply convolution to obtain the final sharpened image

$$f_{\text{sharpened}} = f_{\text{restored}} \star L$$

# Experimental Results

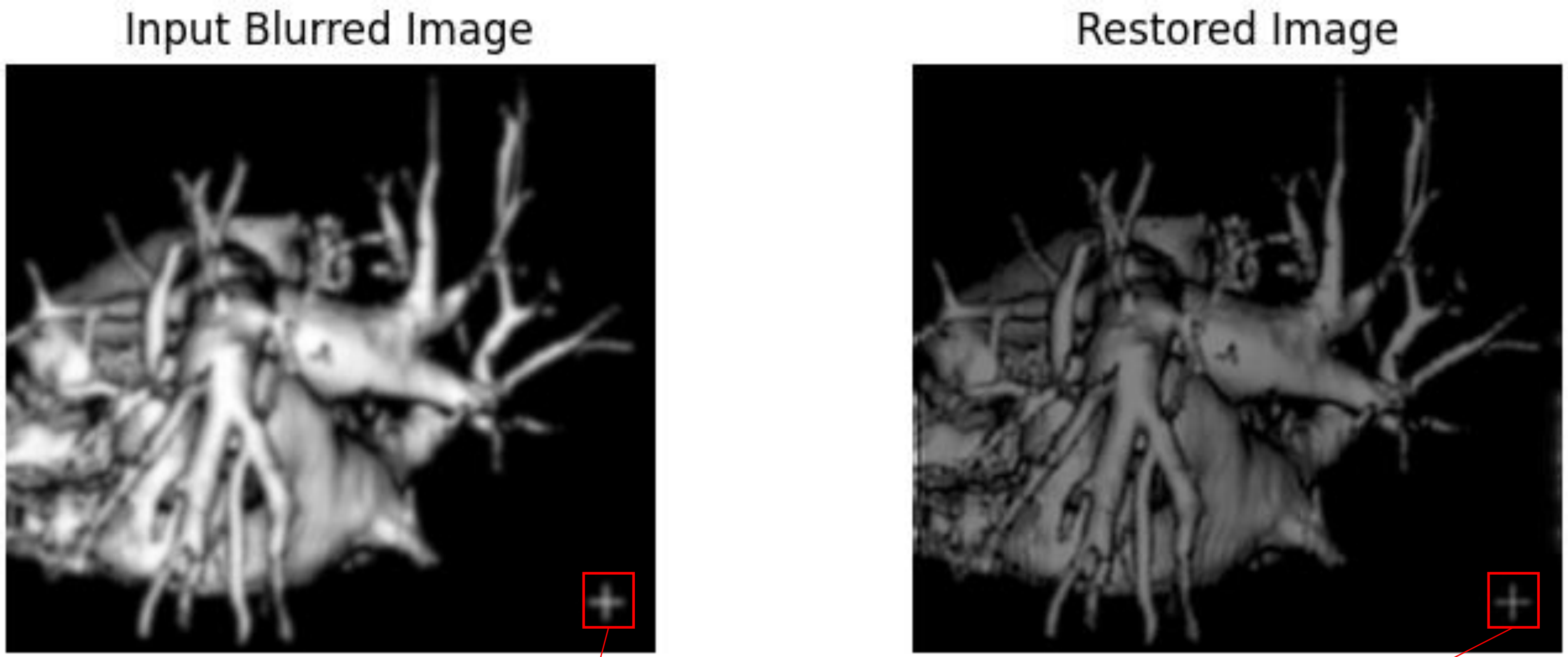
- Parameters setting
- Image results
- Conclusion

# Parameter Settings

By explanations above, combine with some experiment, we show below the parameters for the problem.

| Parameter | Meaning | Value |
|---|---|---|
| $D_0$ | Cutoff frequency for the blurred crosshair | 2.851 |
| $K$ | The estimate constant for Wiener filter | 0.005 |
| $k$ | The scaling coefficient | 11.500 |
| $a$ | The added intensity | 30.000 |
| $t$ | The threshold level | 5.000 |

# Image Results




The restored image is said to be better than the blurred image, here are the reasons:

**Crosshair restored:** Observing the crosshair at the corner, it is seen that the recovered crosshair resembles more of the ideal crosshair (although darker, this is a result of the Wiener filter).

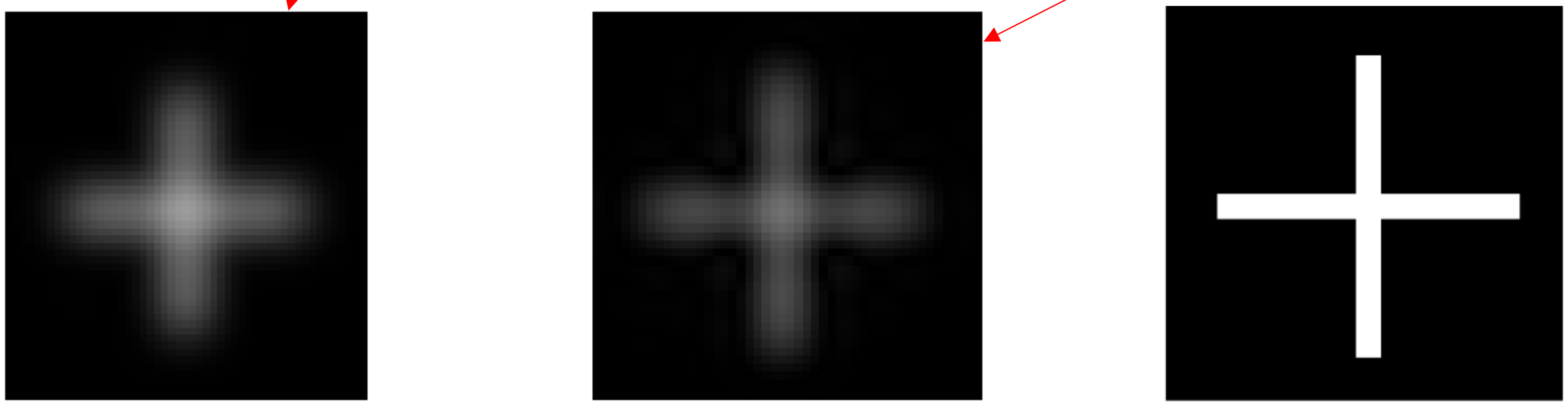
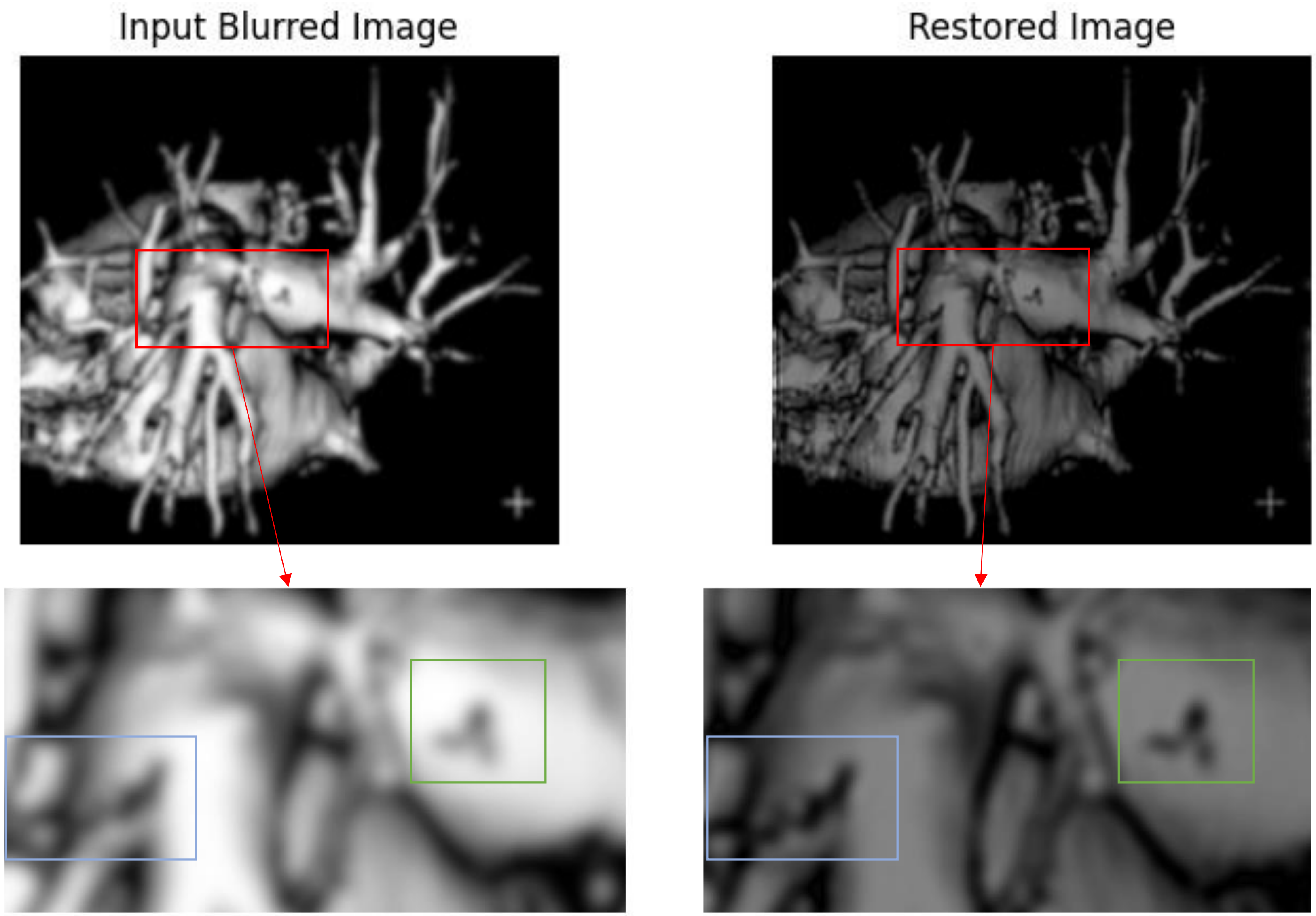


Figure. The blurred, restored and ideal crosshair image

# Image Results




**Information gained:** take a look at the areas which appears to be a crack in the heart. It is seen in the blurred image, the shape of the triangle crack (in the green rectangle) is not clear, whether in the restored image, we can clearly see the shape of the crack, which resembles a triangle.

Another similar area lies in the blue rectangles. In the blurred image, we barely see the shape of the crack. But in the restored image, we can see details of it.

# Image Results

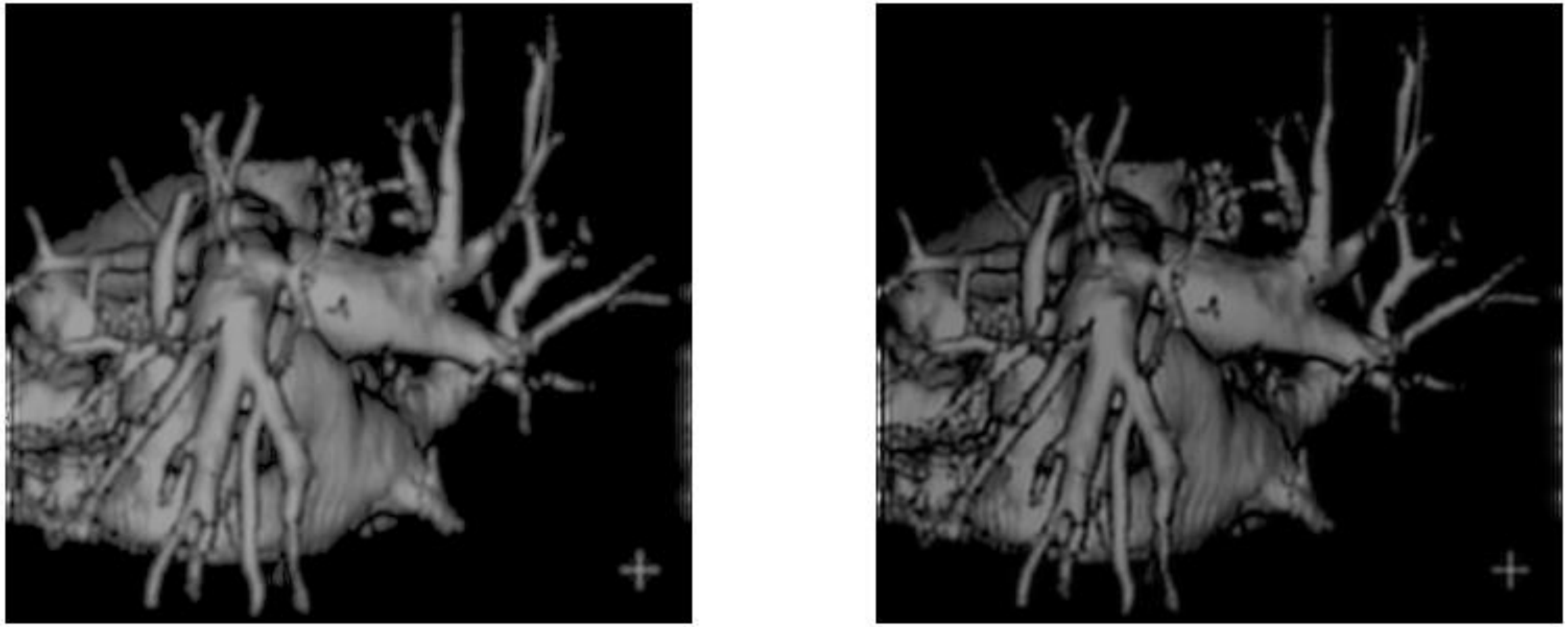


Figure. Brightened and sharpened image

Since these two images are optional steps to enhance the image quality, we only care about the difference between them and the restored image, other conclusions are similar to ones made for the restored image and the blur image.

# Image Results

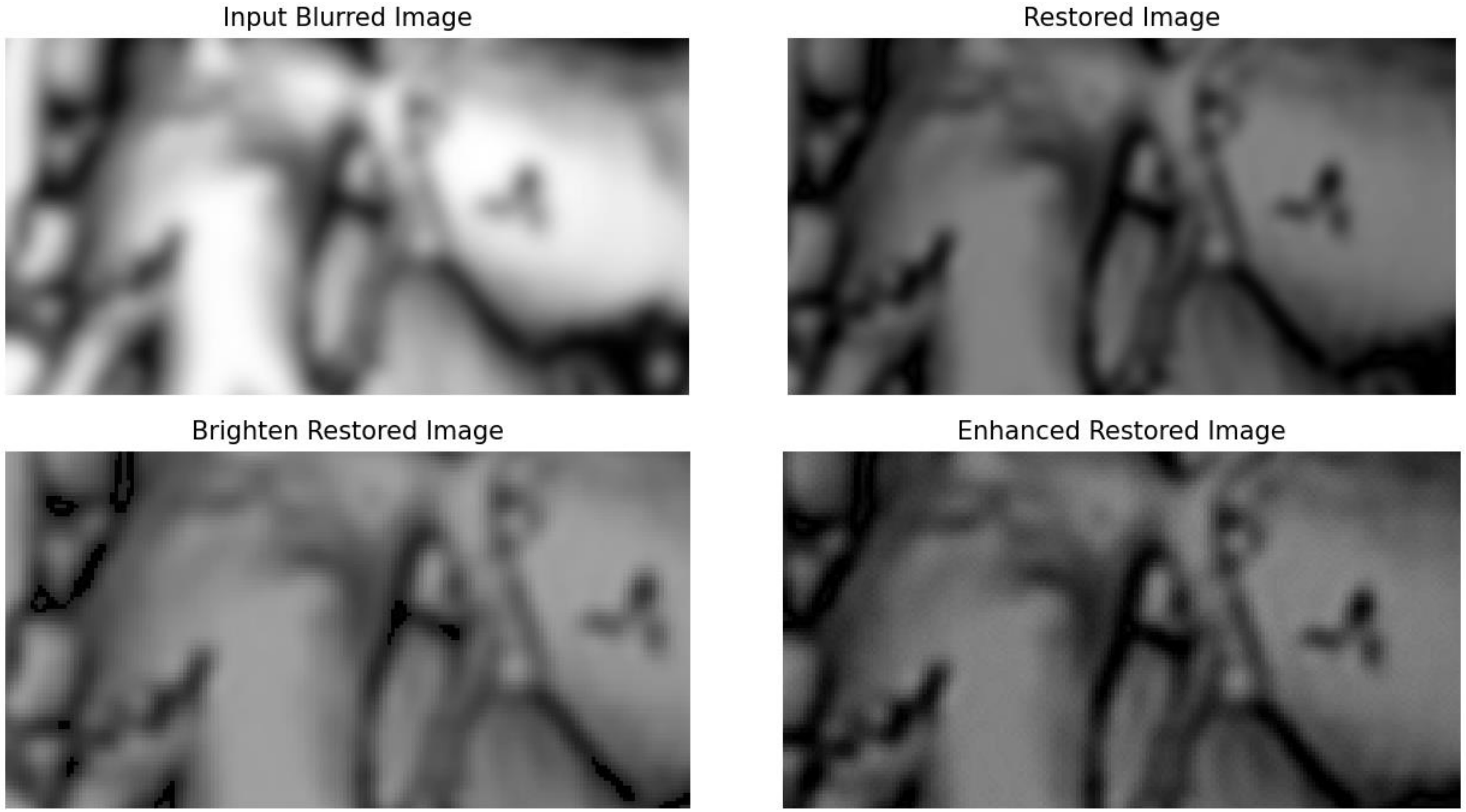



Figure. Crack areas in four images

With the brightened image, the cracks are shown more easier. We notice in the sharpened image, the details shows a more concise shape of how it suppose to be.

# Conclusion

Throughout solving the problem, we could see that

- The estimated Gaussian function is good enough to predict the degraded function of the input heart image.
- Different cutoff radius is applied for different image size.
- Enhancement of an image does not give more information as it already shows. Such work is done for better visualization.

# Limitations and Future Work

- Limitations
- Future Work

# Limitations

1. **Exact image size:** The image we obtain is taken from the internet, which may not correctly have the exact size as the image itself, giving a hard time for estimating.
2. **Limited filters:** Gaussian and Butterworth filters are good filters to estimate the original image in our scenario, there might be other filters which may give better results.
3. **Estimation method:** the complex form of the Gaussian filter makes it hard to apply a machine learning algorithm for estimating the cutoff radius.

# Future Work

1. **Apply machine learning algorithm:** research more on how to apply a combination of Fourier transform with machine learning algorithms, which mostly work on real number domain.
2. **More filters:** explore other filters rather than Gaussian or Butterworth.

# Reference


1. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, Fourth Edition*. Pearson, 2018.