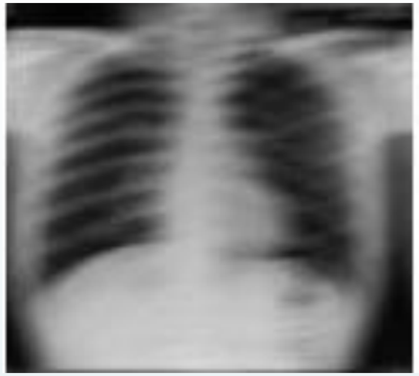
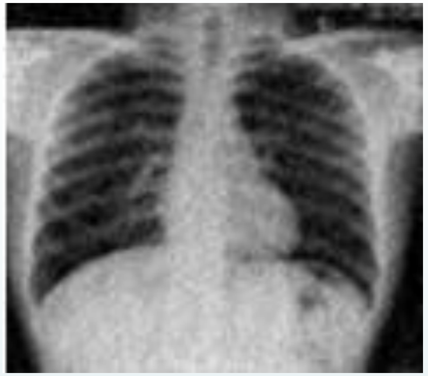


Before



After

# Medical Image Restoration

Dom Carrillo, Josh Dublin

# Goal

- Our goal is to compare the results of the Inverse filter, Wiener filter, and Richardson-Lucy Algorithm and determine which algorithm(s) best restores images.
- These filters were chosen because these are traditional approaches in restoring blurred images
- We hope to provide future researchers of the field with a concise understanding of these algorithms

# Motivation

- Images may be degraded or distorted.
- Crucial information is lost in low quality images
- Image restoration is significant because it allows images to be made clear enough to make accurate observations or descriptions
- An improvement of image restoration techniques will benefit medical fields, forensic analysis, and more.

# Literature Review

| Historical Applications | Modern Day Applications |
|---|---|
| Inverse Filter | All in one restoration imaging via task-adaptive routing |
| Wiener Filter | Energy efficient high-fidelity image reconstruction with memristror arrays |
| Richardson-Lucy Algorithm | Image reconstruction through deep learning |

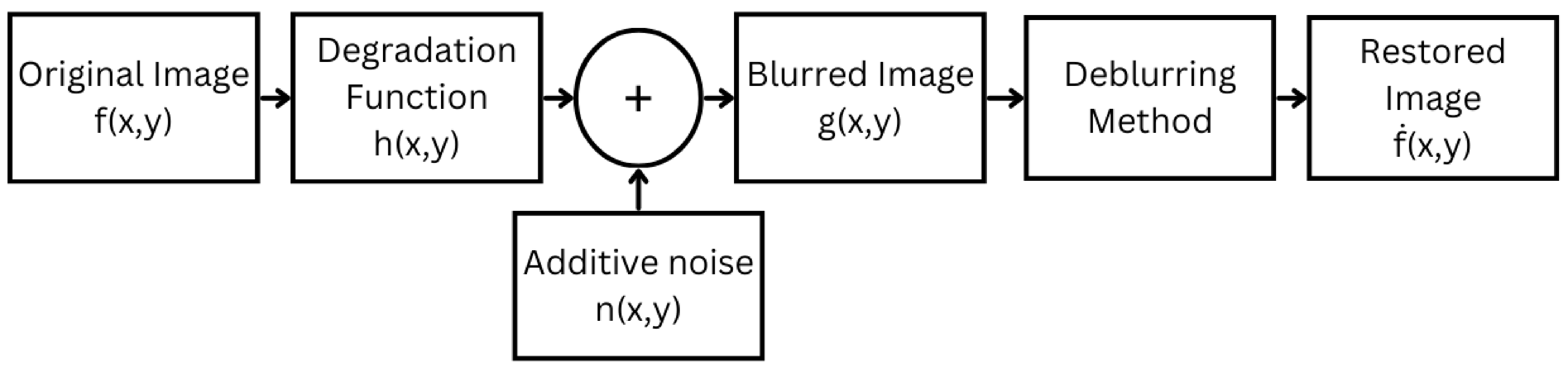
# Literature Review: Established Methods

- Inverse Filter
- Wiener Filter
- Richardson-Lucy Algorithm

# Literature Review: Modern Day Methods

- All in one restoration imaging via task-adaptive routing
- Energy efficient high-fidelity image reconstruction with memristror arrays
- Image reconstruction through deep learning

# Degradation Model

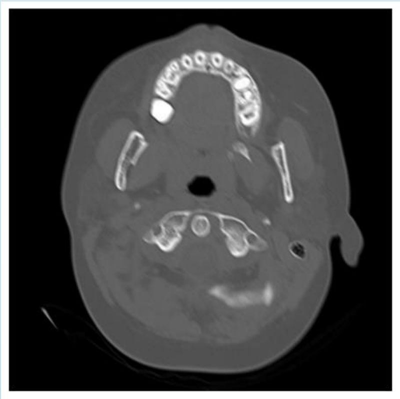


# Blur Methods Used

- Motion Blur
  - Adds an effect to the image as if objects are moving, adding a blur in a specific direction. The format for the motion blurring function is shown in the equation below
    - Θ represents angle of blur
    - L represents the distance of the blur
    - x and y represent spatial coordinates in an image

$$h(x,y) = \begin{cases} \frac{1}{L} & \text{if} \sqrt{x^2+y^2} \leq \frac{1}{L} \text{and} \frac{x}{y} = -\tan(\theta) \\ 0 & \text{Otherwise} \end{cases}$$

Original Image



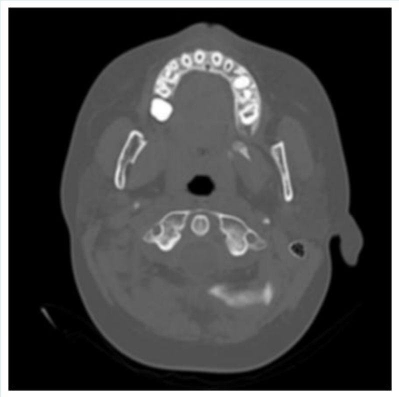
Motion Blurred Image

# Blur Methods Used

- Gaussian Blur
  - Uses a Gaussian function to determine the transformation to each pixel in a two-dimensional image.
    - σ represents the standard deviation of the Gaussian Distribution.
    - X represents the distance from the source on the horizontal axis,
    - y is the distance from the source on the vertical axis.
    - The equation below produces a surface with concentric circles with a Gaussian Distribution from the center point

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Original Image



Gaussian Blurred Image with σ = 0.01

# Additive Noise Methods Used

- Gaussian Noise
  - Additive noise that uses a Gaussian Distribution property and a probability distribution function, shown below for two - dimensional images
    - σ represents the standard deviation of the noise signal
    - X and y represent spatial coordinates of an image

$$\psi(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Original Image

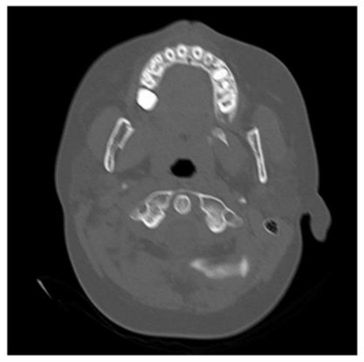
Motion Blurred Image with Gaussian Noise

# Additive Noise Methods Used

- Poisson Noise
  - A signal dependent noise seen in images, also known as quantum noise
  - The equation below shows the degraded image with added Poisson Noise
    - f(i,j) represents the original image
    - λ represents the intensity
    - g(i,j) represents the degraded image
  - The Poisson function shown in the equation takes the mean of the Poisson distribution as input and a Poisson random generation function that produces Poisson random numbers

$$g(i,j) = \frac{1}{\lambda}\text{Poisson}(\lambda f(i,j))$$

Original Image



Motion Blurred Image with Poisson Noise

# Inverse Filter

- The simplest approach to restoration
- The Fourier Transform of a restored image can be taken by dividing it by the degradation function, as shown in equation 1
- The following functions are shown in the equations below with their corresponding Fourier Transform representation
  - $\dot{F}(u,v)$ represents the estimated image
  - F(u,v) represents the Fourier transform of the original image.
  - N(u,v) represents the Fourier transform of the noise added to the image during acquisition (additive noise).
  - H(u,v) represents the degradation function, which represents the effect of distortion on the image, such as blurring.
  - G(u,v) represents the Fourier transform of the degraded image in the frequency domain at coordinates (u,v).
- Equation 1 shows the technique of direct inverse filtering
  - The estimated image is taken by simply dividing the degraded image by the degradation function

$$\dot{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Equation 1

$$G(u,v) = F(u,v) * H(u,v) + N(u,v)$$

Equation 2

# Inverse Filter

- From the previous slide, substituting G(u,v) into the equation of Ḟ(u,v) results into the equation 1.
- Equation 2 represents a simpler way of expressing the function Ḟ(u,v). Divide both the numerator and denominator by H(u,v) and then simplifying each term in the equation.

$$\dot{\mathrm{F}}(u,v) = \frac{F(u,v) * H(u,v) + N(u,v)}{H(u,v)}$$

Equation 1

$$\dot{\mathrm{F}}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Equation 2

# Inverse Filter Problems

- N(u,v) is unknown
  - The function can be estimated, but the exact value is never known
- A small value of H(u,v) allows $\frac{N(u,v)}{H(u,v)}$ to dominate over F(u,v). This results in a more distorted image.
  - If $\mathrm{H(u,v)}$is very small, $\frac{N(u,v)}{H(u,v)}$ will be much larger than F(u,v)

$$\dot{\mathrm{F}}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

# Wiener Filter

- Incorporates the degradation function and statistical characteristics of noise into the restoration process
- The goal is to minimize the mean-squared error between the original and estimated image, as shown in the equation below where:
  - $F(u,v)$ represents the Fourier transform of the original image at frequency coordinates (u,v)
  - $\dot{F}(u,v)$ represents the Fourier transform of the estimated (restored) image at frequency coordinates (u,v)
- Removes additive noise and inverts the blurring
- Wiener filter is also known as the minimum mean square error or least square error filter

$$e^2 = E\left[\left(F(u,v) - \dot{F}(u,v)\right)^2\right]$$

# Wiener Filter

- Equation 1 shows the function for the estimated image with the Wiener Filter function where:
  - $\dot{F}(u,v)$ is the estimated image
  - $W(u,v)$ is the Wiener Filter Function
  - $G(u,v)$ is the degradation function
- Equation 2 shows the Wiener Filter function where:
  - $P_s(u,v)$ is the power spectrum of the original image obtained by the Fourier transform
  - $P_n(u,v)$ is the power spectrum of the additive noise obtained by the Fourier transform
  - $H(u,v)$ is the degraded image
  - $H^*(u,v)$ is the complex conjugate of H(u,v)

$$\dot{F}\,(u,v) = W(u,v)G(u,v)$$

Equation 1

$$W(u,v) = \frac{H^*(u,v)P_s(u,v)}{|H(u,v)|^2 P_s(u,v) + P_n(u,v)}$$

Equation 2

# Wiener Filter

- Equation 1 shows the Wiener Filter function after dividing the numerator and denominator by $P_s(u,v)$
- If there is a strong signal to noise ratio, the term $\frac{P_n(u,v)}{P_s(u,v)}$ in equation 1 becomes very small, allowing the Wiener filter to become an inverse filter, $H^{-1}(u,v)$ for the point spread function.
- If there is additive white noise with no blurring signal, the Wiener Filter becomes equation 2 below where $\sigma_n$ represents the noise variance

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 P_s(u,v) + \frac{P_n(u,v)}{P_s(u,v)}}$$

Equation 1

$$W(u,v) = \frac{P_s(u,v)}{P_s(u,v) + \sigma_n}$$

Equation 2

# Richardson-Lucy Algorithm

- An iterative method to recover an image smoothed by a known point-spread function
- Robust in the presence of noise and blur
- Originally developed to solve positron tomography imaging problems
- The Bayes Theorem is shown in the equation below where
  - P(y|x) is the conditional probability of an event y, given event x
  - P(x) is the probability of an event x
  - P(x|y) is the conditional probability of an event x, given event y

$$P(x|y) = \frac{P(y|x)P(x)}{\int P(y|x)P(x)dx}$$

# Richardson-Lucy Algorithm

- The inverse relation of the Bayes' theorem permits the derivation of the iterative algorithm shown in equation 1 below
  - where i is the iteration number
  - g(y,x) is the known point spread function, describing the distribution of the blur or degradation
  - c(y) is the degraded image or convolution
  - $f_i(x)$ is the estimate of the image at the ith iteration
- In terms of convolutions, the equation can also be written as equation 2 below
- Detail is added to the image in subsequent iterations because of the algorithm's form
- Energy is conserved with every iteration

$$f_{i+1}(x) = \int \frac{g(y,x)c(y)dy}{\int g(y,z)f_i(z)\,dz} f_i(x)$$

Equation 1

$$f_{i+1}(x) = \left\{ \left[ \frac{c(x)}{f_i(x) \otimes g(x)} \right] \otimes g(-x) \right\} f_i(x),$$

Equation 2

# Richardson-Lucy Algorithm

- The inverse of equation 2 in the previous slide is shown in the equation below
- This equation allows for the direct calculation of the point spread function of an object

$$g_{i+1}^{k}(x) = \left\{ \left[ \frac{c(x)}{g_{i}^{k}\ (x) \otimes f^{k-1}(-x)} \right] \otimes f^{k-1}(-x) \right\} g_{i}^{k}(x),$$

# Inverse Filter, Wiener Filter, and Richardson-Lucy Algorithm comparison

| | Advantages | Disadvantages | Ideal Use |
|---|---|---|---|
| Inverse Filter | • Straightforward<br>• Fast implementation | • Noise sensitive<br>• Unable to deal with zero values | • When the noise level is minimal |
| Wiener Filter | • Apt to handle noise<br>• Versatile<br>• Works efficiently in the frequency domain | • Requires an accurate noise model<br>• Potential loss of fine details | • When noise is a significant factor |
| Richardson-Lucy Algorithm | • High quality restoration<br>• Noise suppression<br>• Nnon-negativity constraint | • Computationally intensive<br>• Relies on a well characterized point spread function | • Where fine detail is critical, such as microscopy and astronomy |

# Evaluation Metrics Used

- PSNR (Peak Signal to Noise Ratio)
  - A ratio between highest possible power of a signal, or image, and the power of the noise that affects its representation
  - A higher value generally indicates that the reconstruction has a higher quality, but may have a higher quality for lower PSNR in some cases
  - Calculation is shown below, where
    - R is the maximal variation of the input image, 255 if it s format is an 8 bit unsigned integer
    - MSE is the Mean Squared Error,
      - Where $I_1$ is the original image
      - Where $I_2$ is the estimated image
      - M represents a vertical coordinate of a pixel in the image
      - N represents a horizontal coordinate of a pixel in the image
      - m and n represent the location of the pixel in a two-dimensional image

$$PSNR = 10\log_{10}(\frac{R^2}{MSE})$$

$$MSE = \frac{\sum_{M,N}(I_1(m,n) - I_2(m,n))^2}{M,N}$$

# Evaluation Metrics Used

- SSIM (Structural Similarity Index Measure)
  - Operates in the following stages
    - Luminance: calculated as shown in the second equation below, estimated using its mean intensity, where N is the number of pixels in $\vec{x}$
      - l($\vec{x}$, $\vec{y}$) is used to compare the luminance of the true and estimated images as represented by $\mu_x$ and $\mu_y$
    - Comparison and normalization of contrasts of the two images
      - Contrast, as shown as c($\vec{x}$, $\vec{y}$) in the first equation below, is defined as the estimate of the standard deviation of the image intensities, which is calculated using the third equation below
    - Structure comparison performed on the normalized signals
      - s($\vec{x}$, $\vec{y}$) shown in the first equation below is obtained by performing $\frac{\vec{x}-\mu_x}{\sigma_x}$ and $\frac{\vec{y}-\mu_y}{\sigma_y}$

$$SSIM(\vec{x},\vec{y}) = f(l(\vec{x},\vec{y}), c(\vec{x},\vec{y}), s(\vec{x},\vec{y}))$$

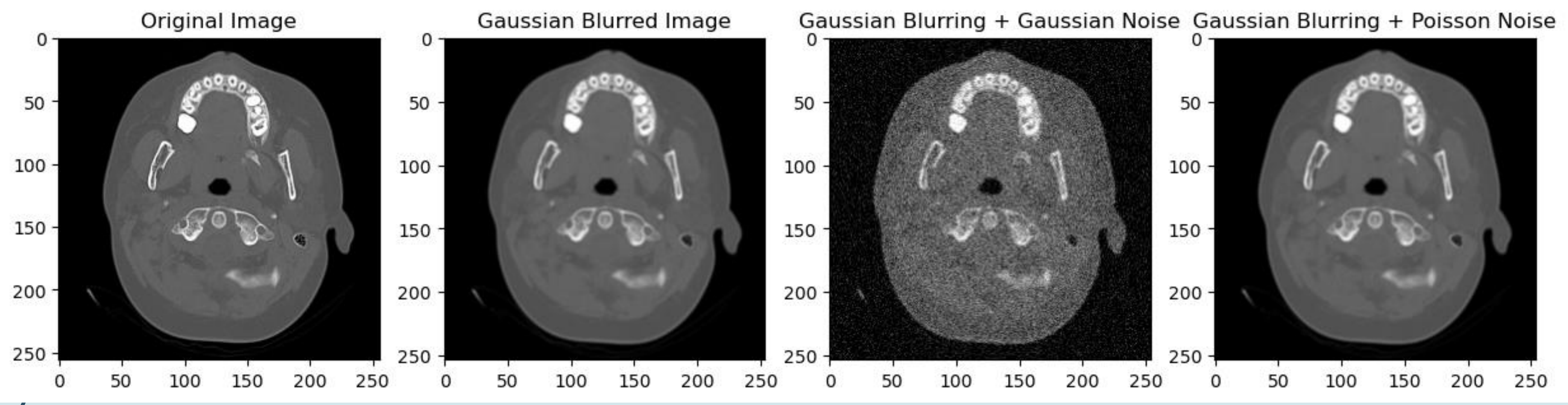
$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma_x = (\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x))^{1/2}$$

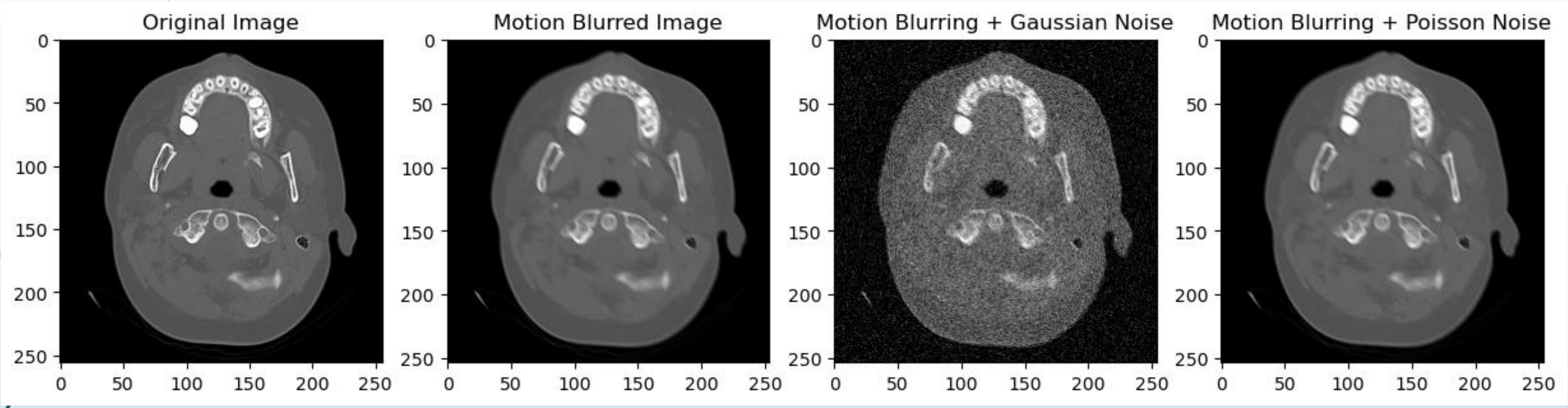
# Experiment

- We developed a Python program that takes an image and applies the different types of blurring and additive noise.
- The image used in our experiment is an image of a brain tumor taken from a Kaggle dataset of CT scan images
- Experiment of setup
  1. Original Image is taken
  2. Blur is added. (Gaussian Blur or Motion Blur)
  3. Additive noise is added to each blurred image (Gaussian Noise or Poisson Noise)
  4. Four images to unblur (Gaussian Blur + Gaussian Noise, Gaussian Blur + Poisson Noise, Motion Blur + Gaussian Noise, and Motion Blur + Poisson Noise)
  5. The three restoration methods used in the experiment get applied to each of the four blurred and noisy images
  6. PSNR and SSIM values are taken from each unblurred image

# Applying Gaussian Blur and Additive Noises

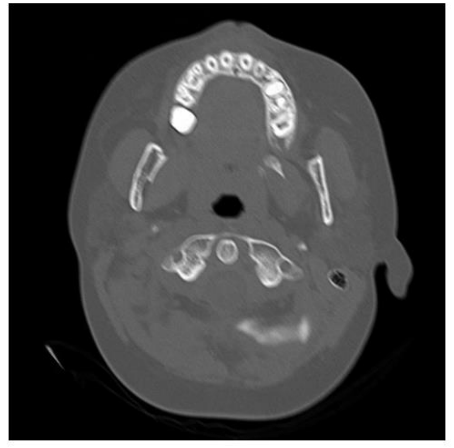


# Applying Motion Blur and Additive Noises

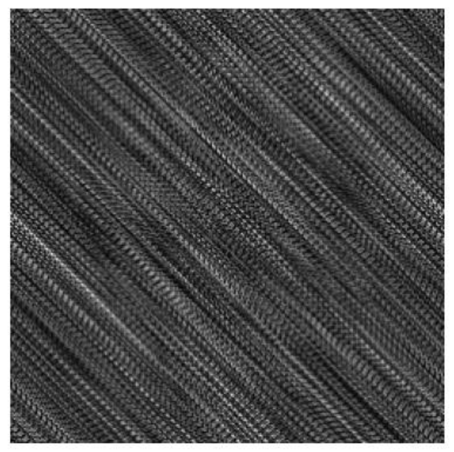


# Results – Inverse Filter on Motion Blur

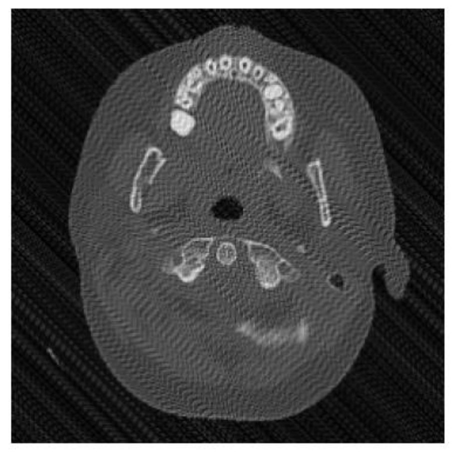
Motion Blur + Gaussian Noise



Motion Blur + Poisson Noise



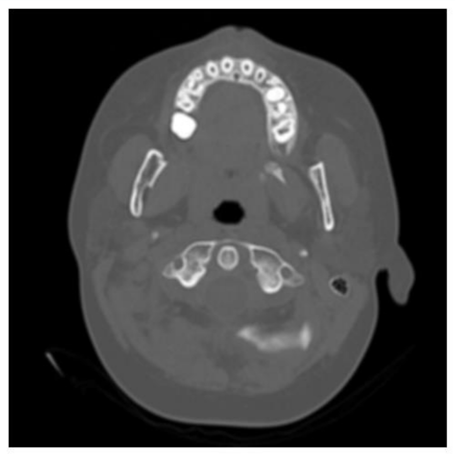
Direct Inverse Filter Applied to top image



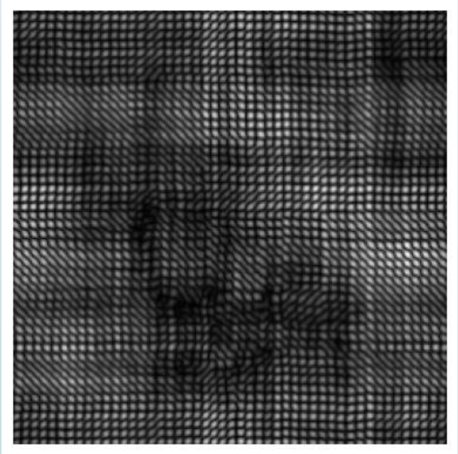
Direct Inverse Filter Applied to top image

# Results – Inverse Filter on Gaussian Blur

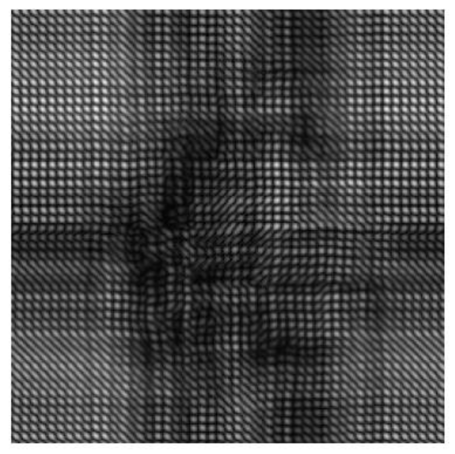
Gaussian Blur + Gaussian Noise



Gaussian Blur + Poisson Noise



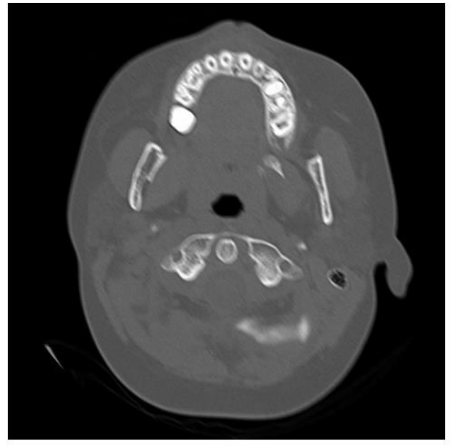
Direct Inverse Filter Applied to top image



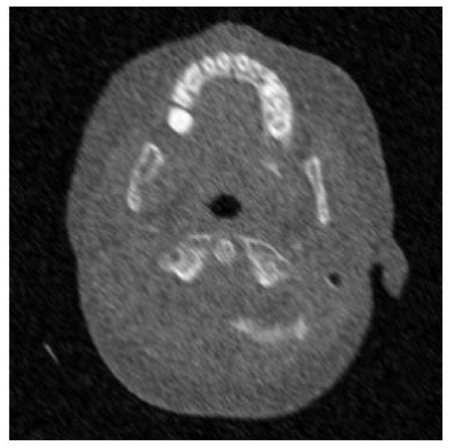
Direct Inverse Filter Applied to top image

# Results – Wiener Filter on Motion Blur

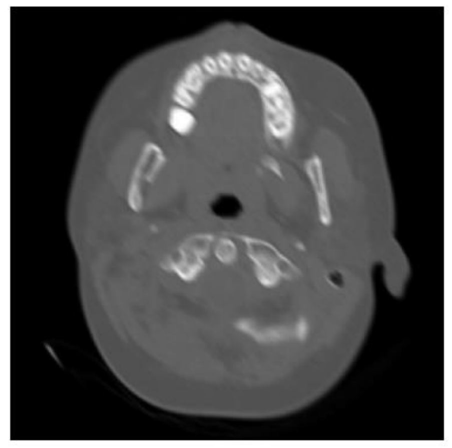
Motion Blur + Gaussian Noise



Motion Blur + Poisson Noise



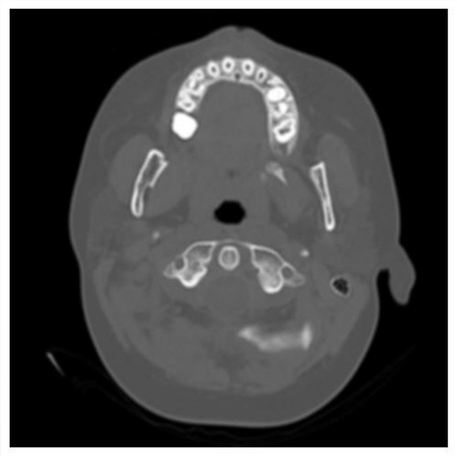
Wiener Filter Applied to top image



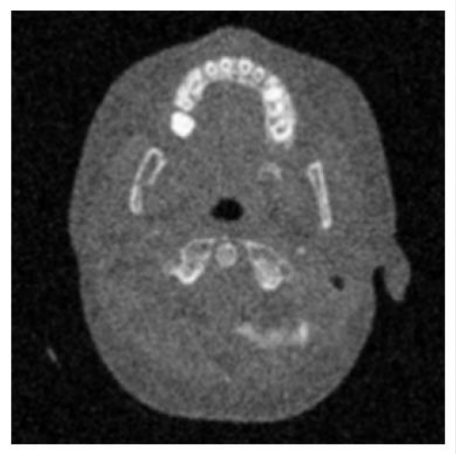
Wiener Filter Applied to top image

# Results – Wiener Filter on Gaussian Blur

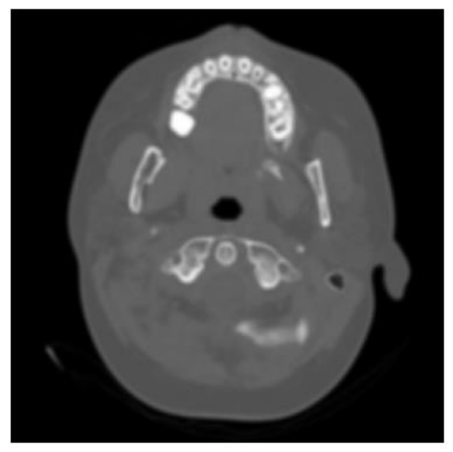
Gaussian Blur + Gaussian Noise



Gaussian Blur + Poisson Noise



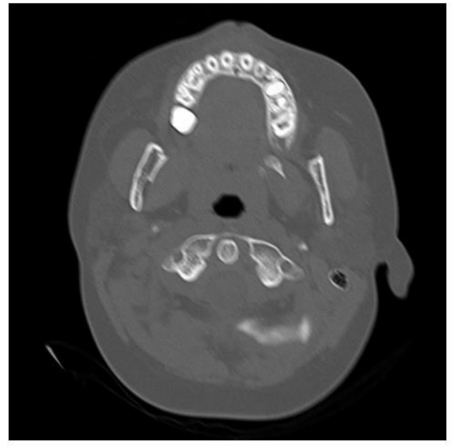
Wiener Filter Applied to top image



Wiener Filter Applied to top image

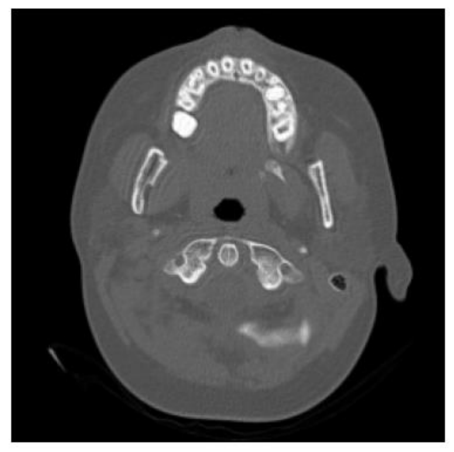
# Results – RL Algorithm on Motion Blur

Motion Blur + Gaussian Noise



Motion Blur + Poisson Noise

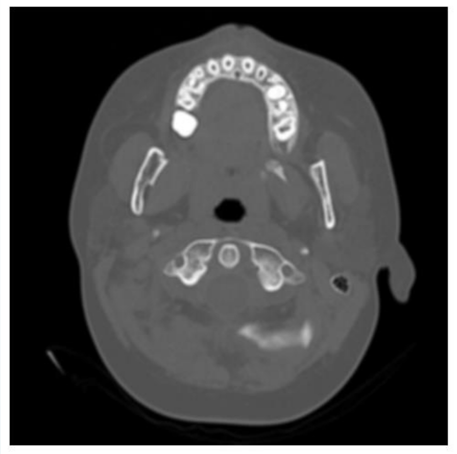
RL Algorithm Applied to top image



RL Algorithm Applied to top image

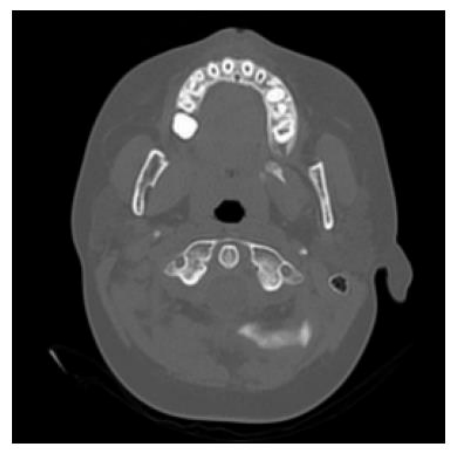
# Results – RL Algorithm on Gaussian Blur

Gaussian Blur + Gaussian Noise



Gaussian Blur + Poisson Noise

RL Algorithm Applied to top image



RL Algorithm Applied to top image

# PSNR Values

| | Motion Blur + Gaussian Noise | Motion Blur + Poisson Noise | Gaussian Blur + Gaussian Noise | Gaussian Blur + Poisson Noise |
|---|---|---|---|---|
| Inverse Filter | 10.5024 | 10.4136 | 8.34143 | 10.5493 |
| Wiener Filter | 10.38124 | 10.38124 | 10.38123 | 10.38124 |
| RL Algorithm | 10.414 | 10.414 | 10.414 | 10.414 |

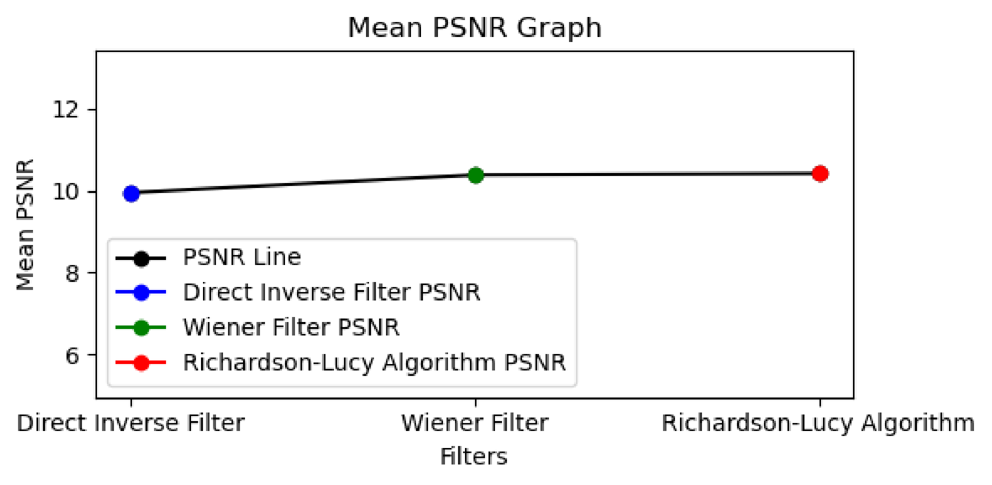
# SSIM Values

| | Motion Blur + Gaussian Noise | Motion Blur + Poisson Noise | Gaussian Blur + Gaussian Noise | Gaussian Blur + Poisson Noise |
|---|---|---|---|---|
| Inverse Filter | 0.30247 | 0.30247 | 0.01423 | 0.01423 |
| Wiener Filter | 0.36136 | 0.36136 | 0.36136 | 0.36136 |
| RL Algorithm | 0.36461 | 0.36461 | 0.36459 | 0.36459 |

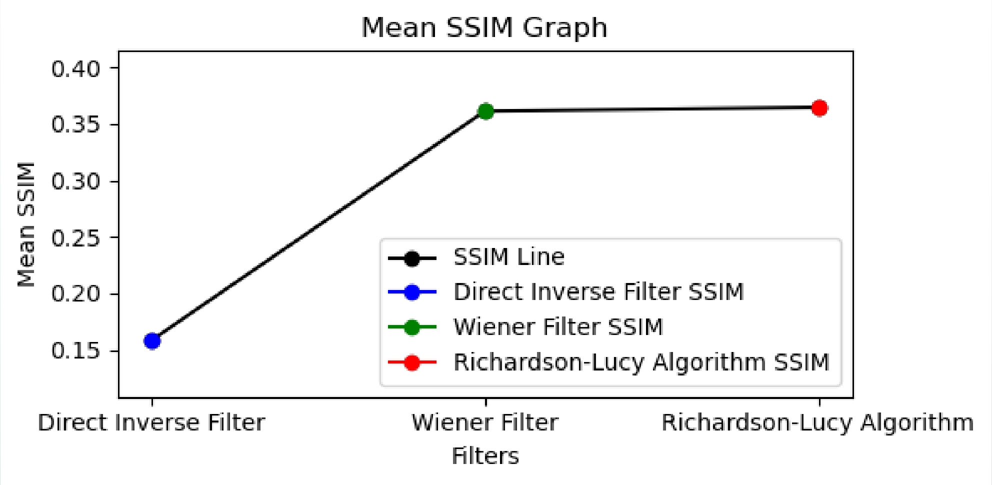
# Mean PSNR and SSIM Values

| | PSNR | SSIM |
|---|---|---|
| Inverse Filter | 9.949 | 0.1584 |
| Wiener Filter | 10.3812 | 0.3614 |
| RL Algorithm | 10.4139 | 0.3646 |

# Mean Graphs




Mean PSNR Graph




Mean SSIM Graph

# Conclusion Based on Results

- Based on the results and data, we determined that the Richardson-Lucy Algorithm was the best option out of the three methods to restore medical images, due to an iterative method that is more suitable to handle noise and blur
- The Wiener Filter also restored images very well, as it had slightly less PSNR and SSIM values than the Richardson-Lucy Algorithm.
- We also found out that the PSNR evaluation is not always an accurate measure, as some restored images using the Direct Inverse Filter does not look clear enough to human perception as to those that were restored using Wiener Filter or the Richardson-Lucy Algorithm

# References


- [1] E. J. Van Beek and E. A. Hoffman, "Functional imaging: Ct and mri," Clinics in Chest Medicine, vol. 29, no. 1, p. 195–216, Feb. 2008. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/ articles/PMC2435287/
- [2] J. Nakashima and H. Duong, "Radiology, image production and evaluation," StatPearls- NCBI Bookshelf, Jul. 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK553145/
- [3] W. Pijal, I. Pineda, and M. E. Morocho-Cayamcela, "Compar ative study of image degradation and restoration techniques," Information and Communication Technologies, pp. 253–265, 10 2022.
- [4] Z. Yang, H. Chen, Z. Qian, Y. Yi, H. Zhang, D. Zhao, B. Wei, and Y. Xu, "All-in-one medical image restoration via task-adaptive routing," 2024. [Online]. Available: https: //arxiv.org/abs/2405.19769
- [5] H. Zhao, Z. Liu, J. Tang, B. Gao, Q. Qin, J. Li, Y. Zhou, P. Yao, Y. Xi, Y. Lin, H. Qian, and H. Wu, "Energy efficient high-fidelity image reconstruction with memristor arrays for medical diagnosis," Nature Communications, vol. 14, no. 1, Apr. 2023. [Online]. Available: https://www.nature.com/ articles/s41467-023-38021-7
- [6] H.-M. Zhang and B. Dong, "A review on deep learning in med ical image reconstruction," Journal of the Operations Research Society of China, vol. 8, no. 2, p. 311–340, Jan. 2020. [Online]. Available: http://dx.doi.org/10.1007/s40305-019-00287-4
- [7] M. Lata, S. Ghosh, F. Bobi, and M. Yousuf, "Novel method to assess motion blur kernel parameters and comparative study of restoration techniques using different image layouts," 2016 In ternational Conference on Informatics, Electronics and Vision (ICIEV), pp. 367–372, 05 2016.
- [8] A. Sheta, "Restoration of medical images using genetic al gorithms," 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–8, 10 2017.

# References


- [9] R. Altaie, "Restoration for blurred noisy images based on guided f iltering and inverse filter," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, p. 1265, 04 2021.
- [10] T. Stathaki, "Digital image processing: Image restoration- inverse filtering," https://www.commsp.ee.ic.ac.uk/~tania/ teaching/DIP%202014/Image%20Restoration%20Pres.pdf [Ac cessed: (10/27/2024)].
- [11] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, April 2004.
- [12] M. D. Rudi Rottenfusser, Erin Wilson, "Education in mi croscopy and digital imaging," https://zeiss-campus.magnet. fsu.edu/articles/basics/psf.html [Accessed: (10/27/2024)].
- [13] H. I. Academy, "Point spread function," https://svi.nl/ Point-Spread-Function-(PSF) [Accessed: (10/27/2024)].
- [14] MathWorks, "Deblurring images using the lucy-richardson algorithm," https://www.mathworks.com/help/images/ deblurring-images-using-the-lucy-richardson-algorithm.html [Accessed: (10/27/2024)].
- [15] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," pp. 2366–2369, 2010.
- [16] D. Poobathy and R. Chezian, "Edge detection operators: Peak signal to noise ratio based comparison," International Journal of Image, Graphics and Signal Processing, vol. 6, pp. 55–61, 09 2014.
- [17] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, "Structural similarity index (ssim) revisited: A data-driven approach," Expert Systems with Applications, vol. 189, p. 116087, 2022. [Online]. Available: https://www. sciencedirect.com/science/article/pii/S0957417421014238
- [18] K. Seshadrinathan, H. Sheikh, Z. Wang, and A. Bovik, "In formation theoretic approaches to image quality assessment," Handbook of Image and Video Processing, 12 2005