

CS 736, Spring 2025

## Medical Image Computing

# Deblurring of Medical Images using Deterministic Algorithms

Aadish Sethiya (210100001)  
Harshit Agarwal (210020054)

April 23, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
<b>3</b>	<b>Problem Description</b>	<b>3</b>
<b>4</b>	<b>Deblurring Algorithms</b>	<b>3</b>
4.1	Richardson-Lucy Deconvolution . . . . .	3
4.2	Wiener Deconvolution . . . . .	4
4.3	Total Variation Regularization . . . . .	4
<b>5</b>	<b>Mathematical Derivations</b>	<b>5</b>
5.1	Richardson-Lucy Deconvolution . . . . .	5
5.2	Derivation of Wiener Deconvolution . . . . .	6
5.3	Derivation of Total Variation Regularized Deblurring . . . . .	8
<b>6</b>	<b>Results</b>	<b>10</b>
6.1	Evaluation Metric: LPIPS . . . . .	10
6.1.1	Motivation . . . . .	11
6.1.2	How LPIPS Works . . . . .	11
6.1.3	Advantages of LPIPS . . . . .	11
6.1.4	Usage in This Report . . . . .	11
6.2	Wiener Deconvolution . . . . .	11
6.2.1	Brain MRI Image . . . . .	11
6.3	Richardson Lucy Deconvolution . . . . .	14
6.3.1	Brain MRI Image . . . . .	14
6.4	Total Variation . . . . .	16
6.4.1	Brain MRI Image . . . . .	16
<b>7</b>	<b>Effect of Parameters on Deblurring Performance</b>	<b>18</b>
7.1	Effect of Kernel Size . . . . .	18
7.2	Effect of Noise Level ( $\sigma = 1, 3, 5$ ) . . . . .	19
7.3	Effect of PSF Orientation (Angles: $0^\circ, 45^\circ, 90^\circ$ ) . . . . .	19
7.4	Comparison of Algorithms . . . . .	20
7.4.1	Kernel Size - 20, Angle - 45, Noise Sigma - 3 . . . . .	20
<b>8</b>	<b>Observations</b>	<b>21</b>
8.1	Visual Quality and Restoration . . . . .	21
8.2	Computation Time and Efficiency . . . . .	22
8.3	Overall Trade-offs . . . . .	22
<b>9</b>	<b>Conclusion</b>	<b>22</b>

# 1 Introduction

Medical imaging plays a crucial role in modern diagnostics. However, the quality of captured images often suffers from motion blur, instrument limitations, and noise. Image deblurring algorithms aim to restore the original scene from degraded observations. This report investigates three prominent deblurring methods for medical applications.

## 2 Background

Image degradation is typically modeled as the convolution of a latent sharp image with a point spread function (PSF), corrupted by noise. Various algorithms have been proposed to invert this process. Deterministic methods, which do not rely on probabilistic sampling, offer controllable and reproducible results, making them attractive for clinical use.

## 3 Problem Description

We consider the general problem of restoring a blurred image  $B$  given a known point spread function (PSF)  $P$ . The image formation process is commonly modeled as:

$$B = I * P + N$$

where  $I$  is the latent (original) image,  $*$  denotes convolution, and  $N$  represents additive noise (Gaussian or Poisson).

In this report, we explore three different deterministic algorithms to estimate  $I$ :

- **Richardson-Lucy Deconvolution:** Assumes Poisson noise and iteratively maximizes the likelihood using a multiplicative update rule.
- **Wiener Deconvolution:** Assumes Gaussian noise and performs inverse filtering with regularization in the frequency domain.
- **Total Variation (TV) Regularization:** Balances data fidelity with edge-preserving regularization to restore smooth, sharp images.

Each method approaches the deblurring problem differently, leveraging specific noise assumptions and regularization strategies.

## 4 Deblurring Algorithms

This section presents the algorithmic procedures used to perform image deblurring for each of the three methods implemented in this project.

### 4.1 Richardson-Lucy Deconvolution

The Richardson-Lucy (RL) algorithm is an iterative method based on maximizing the likelihood under Poisson noise. The update rule at iteration  $t$  is:

$$I^{(t+1)} = I^{(t)} \cdot \left( \frac{B}{I^{(t)} * P + \epsilon} * P^* \right)$$

where:

- $I^{(t)}$  is the estimate at iteration  $t$
- $B$  is the observed blurred image
- $P$  is the known PSF
- $P^*$  is the 180° rotated version of  $P$
- $\epsilon$  is a small constant to avoid division by zero

The algorithm is stopped after a fixed number of iterations or when convergence is achieved.

## 4.2 Wiener Deconvolution

The Wiener filter operates in the frequency domain and minimizes the mean square error. The deblurred image  $I$  is computed as:

$$\hat{I}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K} \cdot \hat{B}(u, v)$$

where:

- $\hat{I}(u, v), \hat{B}(u, v)$ : Fourier transforms of the restored and blurred images
- $H(u, v)$ : Fourier transform of the PSF
- $H^*(u, v)$ : complex conjugate of  $H(u, v)$
- $K$ : a constant representing the noise-to-signal power ratio

The inverse Fourier transform of  $\hat{I}(u, v)$  gives the deblurred image in spatial domain.

## 4.3 Total Variation Regularization

Total Variation (TV) deblurring solves the following optimization problem:

$$\min_I (\|I * P - B\|^2 + \lambda \cdot TV(I))$$

This is solved using iterative gradient descent. At each iteration, the update is:

$$I^{(t+1)} = I^{(t)} - \eta \cdot \left[ (I^{(t)} * P - B) * P^* + \lambda \cdot \nabla TV(I^{(t)}) \right]$$

where:

- $\eta$ : step size
- $\lambda$ : regularization weight
- $\nabla TV(I)$ : gradient of the total variation term, encouraging edge preservation

The process continues until convergence or a fixed number of iterations.

## 5 Mathematical Derivations

### 5.1 Richardson-Lucy Deconvolution

The Richardson-Lucy (RL) algorithm is derived by formulating the image deblurring problem as a maximum likelihood estimation under the assumption of Poisson noise. Below, we walk through the derivation step-by-step.

#### Step 1: Forward Image Formation Model

Let  $\mathbf{x}$  be the unknown true image (vectorized), and  $\mathbf{m}$  the observed blurred image. The imaging system is modeled by a known point spread function (PSF), represented as a linear operator  $H$ . The expected measurement is then:

$$\mathbf{E} = H\mathbf{x}, \quad \text{where } E_i = \sum_j H_{ij}x_j$$

Each observed pixel  $m_i$  is modeled as a Poisson-distributed random variable with mean  $E_i$ .

#### Step 2: Poisson Likelihood Model

Assuming independent Poisson noise, the probability of observing  $\mathbf{m}$  given  $\mathbf{E}$  is:

$$P(\mathbf{m} | \mathbf{E}) = \prod_i \frac{E_i^{m_i} e^{-E_i}}{m_i!}$$

#### Step 3: Log-Likelihood Function

To simplify optimization, we take the logarithm of the likelihood:

$$\log P(\mathbf{m} | \mathbf{x}) = \sum_i (m_i \log E_i - E_i - \log m_i!)$$

Since  $\log m_i!$  does not depend on  $\mathbf{x}$ , we define the objective function  $\alpha(\mathbf{x})$  as:

$$\alpha(\mathbf{x}) = \sum_i (m_i \log E_i - E_i)$$

#### Step 4: Compute Gradient

We differentiate  $\alpha(\mathbf{x})$  with respect to  $x_j$ . Since  $E_i = \sum_j H_{ij}x_j$ , we get:

$$\frac{\partial \alpha}{\partial x_j} = \sum_i \left( \frac{m_i}{E_i} - 1 \right) \frac{\partial E_i}{\partial x_j} = \sum_i H_{ij} \left( \frac{m_i}{E_i} - 1 \right)$$

The full gradient (vector form) is:

$$\nabla \alpha(\mathbf{x}) = H^\top \left( \frac{\mathbf{m}}{\mathbf{E}} - \mathbf{1} \right)$$

where division is element-wise.

### Step 5: Choosing a Scaled Gradient Step

To ensure positivity and derive a stable iterative update, we scale the gradient step by:

$$\lambda_j = \frac{x_j}{(H^\top \mathbf{1})_j}$$

This gives the update rule:

$$x_j^{\text{new}} = x_j + \lambda_j (\nabla \alpha(\mathbf{x}))_j = x_j \cdot \frac{(H^\top [\frac{\mathbf{m}}{\mathbf{E}}])_j}{(H^\top \mathbf{1})_j}$$

If  $H^\top \mathbf{1} = \mathbf{1}$  (i.e., columns of  $H$  sum to 1), then the update simplifies to:

$$x_j^{\text{new}} = x_j \cdot \left( H^\top \left[ \frac{\mathbf{m}}{\mathbf{E}} \right] \right)_j$$

### Step 6: Final Iterative Update (Vector Form)

Combining the terms into vector notation, we get the Richardson-Lucy iteration:

$$\boxed{\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} \cdot H^\top \left[ \frac{\mathbf{m}}{H\mathbf{x}^{(t)}} \right]}$$

where all operations are element-wise.

### Step 7: Convolution Form for Images

In practice, for 2D images where  $H$  corresponds to a convolution with a PSF  $P$ , and  $H^\top$  corresponds to convolution with the flipped PSF  $P^*$ , we write:

$$\boxed{I^{(t+1)} = I^{(t)} \cdot \left( \frac{B}{I^{(t)} * P + \epsilon} * P^* \right)}$$

Here:

- $B$  is the observed image
- $I^{(t)}$  is the estimate at iteration  $t$
- $*$  denotes convolution
- $P^*$  is  $P$  rotated by 180°
- $\epsilon$  is a small constant to avoid division by zero

This is the classic Richardson-Lucy deblurring algorithm used in image restoration under Poisson noise.

## 5.2 Derivation of Wiener Deconvolution

The Wiener deconvolution method seeks to recover an estimate of a signal that has been blurred and corrupted by additive noise. The approach is based on minimizing the mean square error between the true signal and the estimate in the frequency domain.

### Step 1: Define the Error Objective

We define the frequency-dependent mean square error  $\epsilon(f)$  between the true signal  $X(f)$  and its estimate  $\hat{X}(f)$ :

$$\epsilon(f) = \mathbb{E} \left[ |X(f) - \hat{X}(f)|^2 \right]$$

This expression quantifies the average squared difference between the actual and estimated signals at frequency  $f$ .

### Step 2: Plug in the Estimate

The estimate  $\hat{X}(f)$  is assumed to be of the form:

$$\hat{X}(f) = G(f)Y(f)$$

where:

- $G(f)$ : the filter we want to find (Wiener filter)
- $Y(f)$ : the observed signal in the frequency domain

Recall that the observed signal is:

$$Y(f) = H(f)X(f) + V(f)$$

Substituting into the error definition:

$$\epsilon(f) = \mathbb{E} \left[ |X(f) - G(f)(H(f)X(f) + V(f))|^2 \right]$$

### Step 3: Factor Out and Expand the Expression

Group terms:

$$\epsilon(f) = \mathbb{E} \left[ |[1 - G(f)H(f)]X(f) - G(f)V(f)|^2 \right]$$

Now expand the squared modulus expression:

$$\begin{aligned} \epsilon(f) &= [1 - G(f)H(f)][1 - G(f)H(f)]^* \mathbb{E}[|X(f)|^2] \\ &\quad - [1 - G(f)H(f)]G^*(f)\mathbb{E}[X(f)V^*(f)] \\ &\quad - G(f)[1 - G(f)H(f)]^*\mathbb{E}[X^*(f)V(f)] \\ &\quad + G(f)G^*(f)\mathbb{E}[|V(f)|^2] \end{aligned}$$

### Step 4: Assume Independence Between Signal and Noise

Assume that the signal and noise are uncorrelated:

$$\mathbb{E}[X(f)V^*(f)] = \mathbb{E}[X^*(f)V(f)] = 0$$

This simplifies the expression to:

$$\epsilon(f) = |1 - G(f)H(f)|^2 S(f) + |G(f)|^2 N(f)$$

where:

- $S(f) = \mathbb{E}[|X(f)|^2]$  is the power spectral density (PSD) of the signal.
- $N(f) = \mathbb{E}[|V(f)|^2]$  is the PSD of the noise.

### Step 5: Minimize the Error

To find the optimal  $G(f)$ , we minimize  $\epsilon(f)$  with respect to  $G(f)$  using the Wirtinger derivative:

$$\frac{d\epsilon(f)}{dG(f)} = 0$$

Differentiate the error expression:

$$\frac{d\epsilon(f)}{dG(f)} = G^*(f)N(f) - H(f)[1 - G(f)H(f)]^*S(f)$$

Set the derivative to zero:

$$G^*(f)N(f) - H(f)[1 - G(f)H(f)]^*S(f) = 0$$

### Step 6: Solve for the Wiener Filter

This equation can be solved algebraically to give the classic Wiener filter formula:

$$G(f) = \frac{H^*(f)S(f)}{|H(f)|^2S(f) + N(f)}$$

Here,  $H^*(f)$  is the complex conjugate of the frequency response of the blur kernel, and  $S(f)$  and  $N(f)$  are the PSDs of the signal and noise.

### Summary and Implementation Notes

- In practice,  $S(f)$  and  $N(f)$  are often unknown and approximated.
- A common simplification is assuming  $K = \frac{N(f)}{S(f)}$  is constant, giving:

$$G(f) = \frac{H^*(f)}{|H(f)|^2 + K}$$

- After filtering, we return to the spatial domain using the inverse Fourier transform:

$$\hat{x}(t) = \mathcal{F}^{-1}[G(f)Y(f)]$$

### 5.3 Derivation of Total Variation Regularized Deblurring

Total Variation (TV) regularization is a powerful method for restoring blurred images while preserving edges. It balances data fidelity with a regularization term that penalizes oscillations but allows for sharp discontinuities like edges.

## Step 1: Image Formation Model

As with the other methods, we model the observed image  $B$  as a blurred and noisy version of the latent image  $I$ :

$$B = I * P + N$$

where:

- $I$ : original (latent) image
- $P$ : known blur kernel (PSF)
- $*$ : convolution operator
- $N$ : additive noise

## Step 2: Optimization Objective

We aim to recover  $I$  by solving the following optimization problem:

$$\min_I \left( \frac{1}{2} \|I * P - B\|^2 + \lambda \cdot TV(I) \right)$$

Here: -  $\|I * P - B\|^2$  is the **data fidelity term** (L2 norm of the reconstruction error) -  $TV(I)$  is the **total variation** regularization term -  $\lambda$  controls the balance between fidelity and smoothness

## Step 3: Total Variation Definition

The **anisotropic total variation** of an image  $I$  is defined as:

$$TV(I) = \sum_{i,j} (|I_{i+1,j} - I_{i,j}| + |I_{i,j+1} - I_{i,j}|)$$

This penalizes the sum of absolute differences in horizontal and vertical directions.  
A smooth differentiable approximation is often used:

$$TV(I) \approx \sum_{i,j} \sqrt{(I_{i+1,j} - I_{i,j})^2 + (I_{i,j+1} - I_{i,j})^2 + \epsilon}$$

where  $\epsilon$  is a small constant to avoid division by zero during gradient computation.

## Step 4: Gradient of the Objective Function

Let:

$$\mathcal{L}(I) = \frac{1}{2} \|I * P - B\|^2 + \lambda \cdot TV(I)$$

Then the gradient is:

$$\nabla \mathcal{L}(I) = (I * P - B) * P^T + \lambda \cdot \nabla TV(I)$$

Where: -  $P^T$  is the flipped version of  $P$  -  $\nabla TV(I)$  is the gradient of the total variation term, computed using finite differences

### Step 5: Iterative Gradient Descent Update

To minimize  $\mathcal{L}(I)$ , we apply iterative gradient descent:

$$I^{(t+1)} = I^{(t)} - \eta \cdot \nabla \mathcal{L}(I^{(t)})$$

where: -  $\eta$  is the learning rate (step size) -  $\nabla \mathcal{L}(I)$  is computed as described above

### Step 6: Implementation Notes

- Gradients are computed using forward finite differences.
- The convolution is implemented using FFT for speed.
- The update is repeated until convergence or a maximum number of iterations is reached.
- Values are optionally clipped to the valid image range after each update.

### Summary

Total Variation deblurring solves the following optimization:

$$\min_I \left( \frac{1}{2} \|I * P - B\|^2 + \lambda \cdot TV(I) \right)$$

It is effective at restoring blurry images while preserving sharp edges, making it especially valuable in medical imaging applications.

## 6 Results

We conducted deblurring experiments on three different medical images:

- X-Ray Lungs
- Chest CT Scan
- Brain MRI

Each of these images was synthetically blurred using a known kernel (PSF) and corrupted with additive Gaussian noise. We applied three different deblurring algorithms to each image:

- Richardson-Lucy Deconvolution
- Wiener Deconvolution
- Total Variation Regularized Deblurring

The results were evaluated both visually and quantitatively using LPIPS (Learned Perceptual Image Patch Similarity), which measures perceptual similarity.

### 6.1 Evaluation Metric: LPIPS

To quantitatively evaluate the quality of the deblurred images, we used the **LPIPS (Learned Perceptual Image Patch Similarity)** metric. Unlike traditional metrics such as MSE (Mean Squared Error) or PSNR (Peak Signal-to-Noise Ratio), which measure pixel-wise differences, LPIPS aims to assess perceptual similarity in a way that aligns more closely with human visual perception.

### 6.1.1 Motivation

Medical image clarity and structural fidelity are critical for diagnosis, but standard distortion metrics often fail to capture visual quality as perceived by clinicians. LPIPS addresses this by comparing deep feature representations of image patches extracted from a neural network trained on perceptual tasks.

### 6.1.2 How LPIPS Works

- LPIPS computes the difference between features extracted from a pretrained deep network (e.g., AlexNet, VGG) at multiple layers.
- These feature activations are *normalized* and their  $L_2$  distance is calculated across the network's layers.
- A learned linear weighting is applied to each layer's contribution, producing a scalar distance score.

Lower LPIPS values indicate higher perceptual similarity between the deblurred image and the ground truth (original) image.

### 6.1.3 Advantages of LPIPS

- Correlates better with human judgments of image quality compared to PSNR/SSIM.
- Sensitive to structural distortions and textural artifacts.
- Particularly useful for assessing medical images where perceptual clarity is essential.

### 6.1.4 Usage in This Report

We used the LPIPS metric to evaluate all deblurred images across different algorithms and parameter settings. It provided consistent and meaningful assessments that matched qualitative visual comparisons. The LPIPS scores are reported alongside visual results for each method to quantify perceptual improvements.

## 6.2 Wiener Deconvolution

### 6.2.1 Brain MRI Image

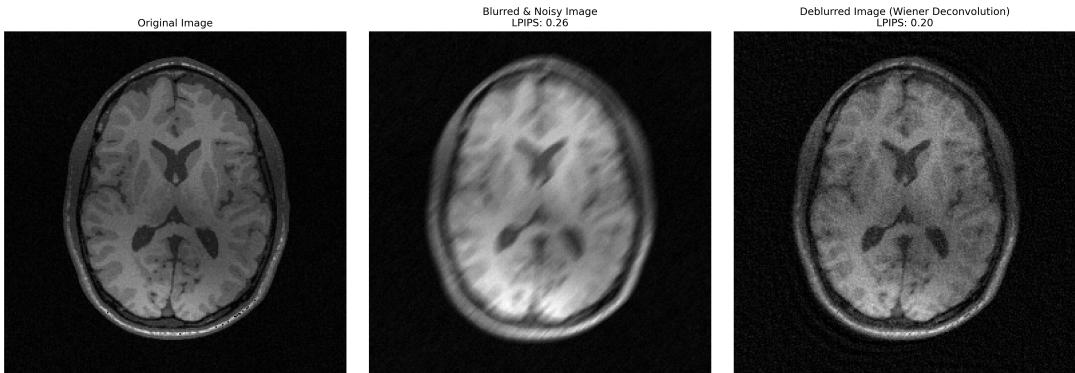


Figure 1: Kernel Size - 10, Angle - 45, Noise Sigma - 3

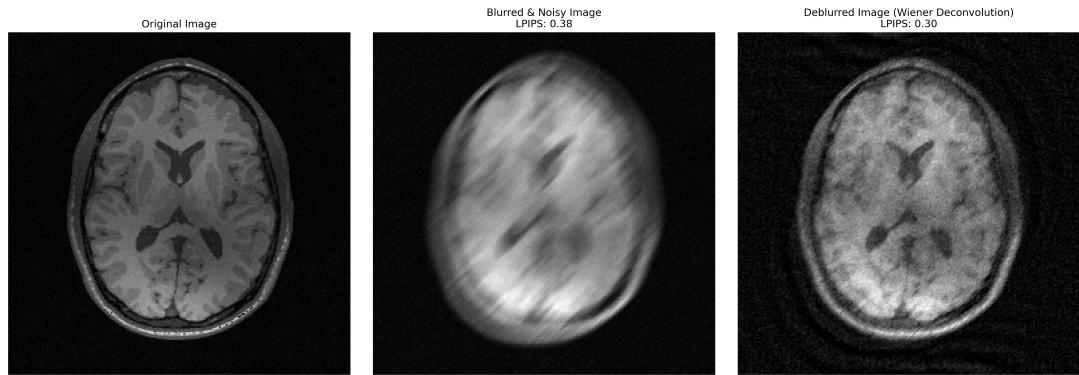


Figure 2: Kernel Size - 20, Angle - 45, Noise Sigma - 3

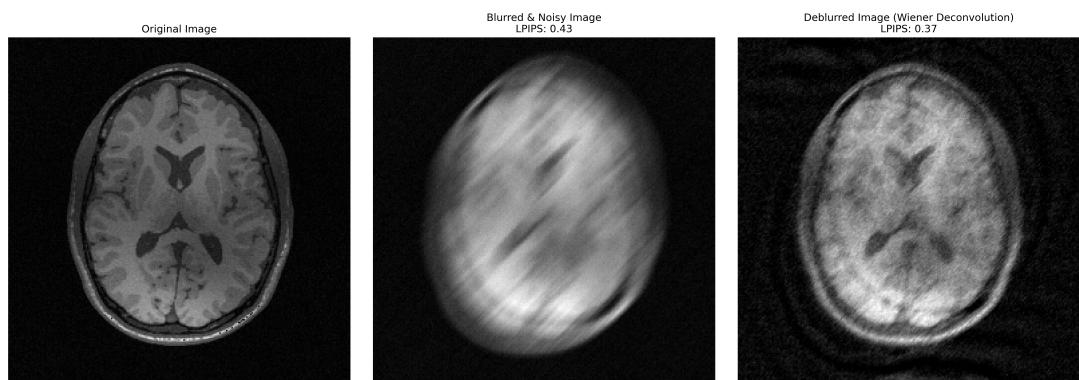


Figure 3: Kernel Size - 30, Angle - 45, Noise Sigma - 3

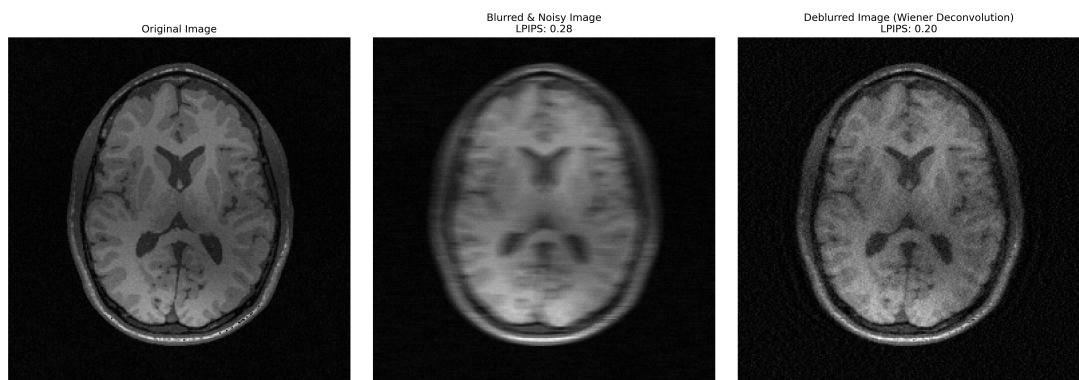


Figure 4: Kernel Size - 10, Angle - 0, Noise Sigma - 3

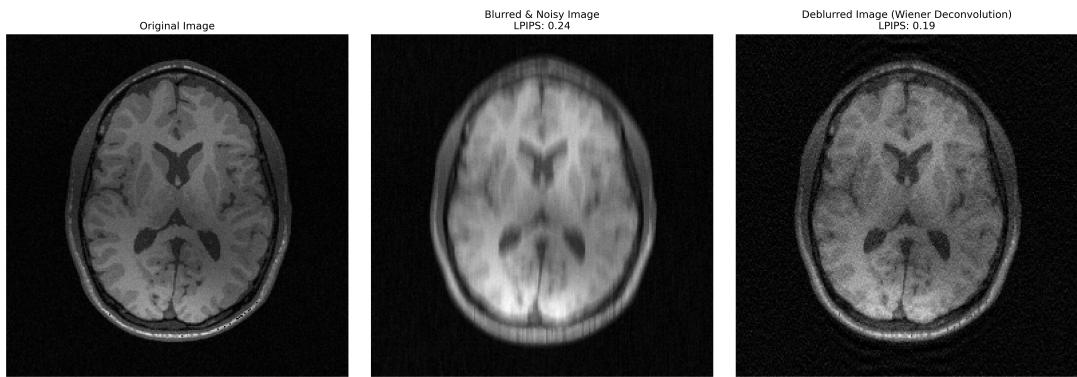


Figure 5: Kernel Size - 10, Angle - 90, Noise Sigma - 3

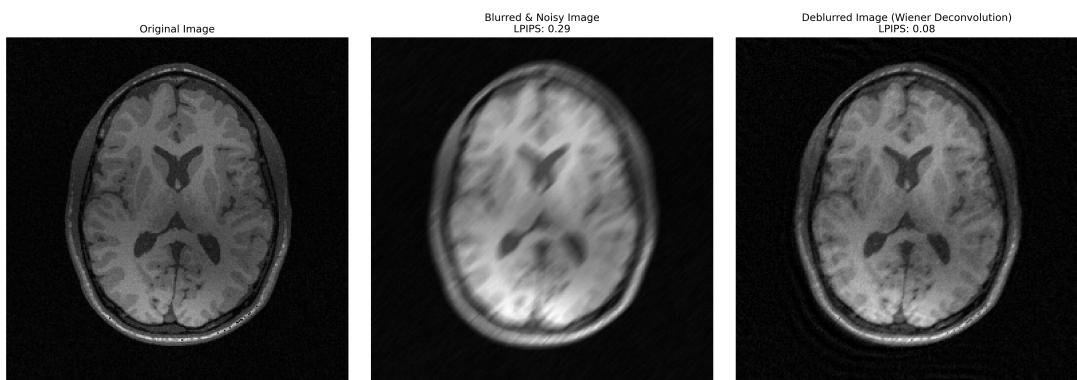


Figure 6: Kernel Size - 10, Angle - 45, Noise Sigma - 1

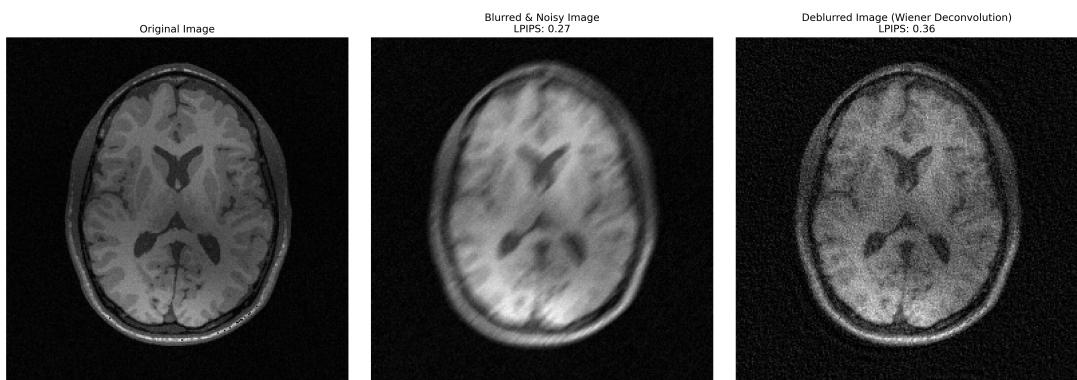


Figure 7: Kernel Size - 10, Angle - 45, Noise Sigma - 5

### 6.3 Richardson Lucy Deconvolution

#### 6.3.1 Brain MRI Image

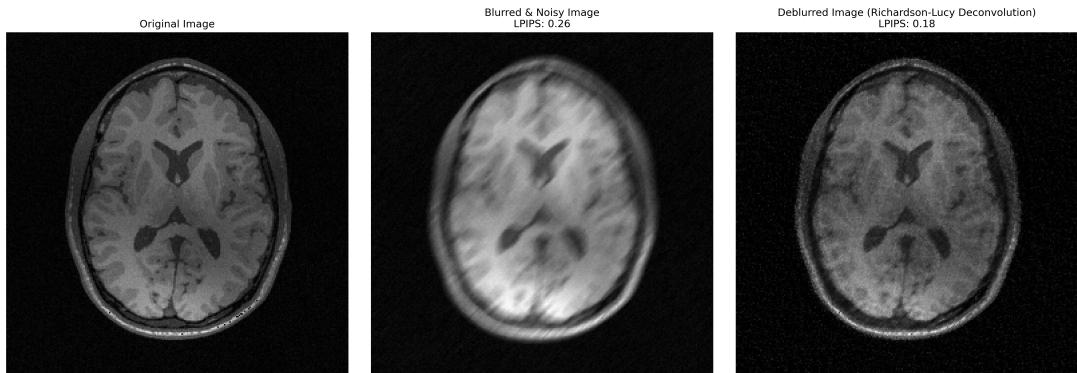


Figure 8: Kernel Size - 10, Angle - 45, Noise Sigma - 3

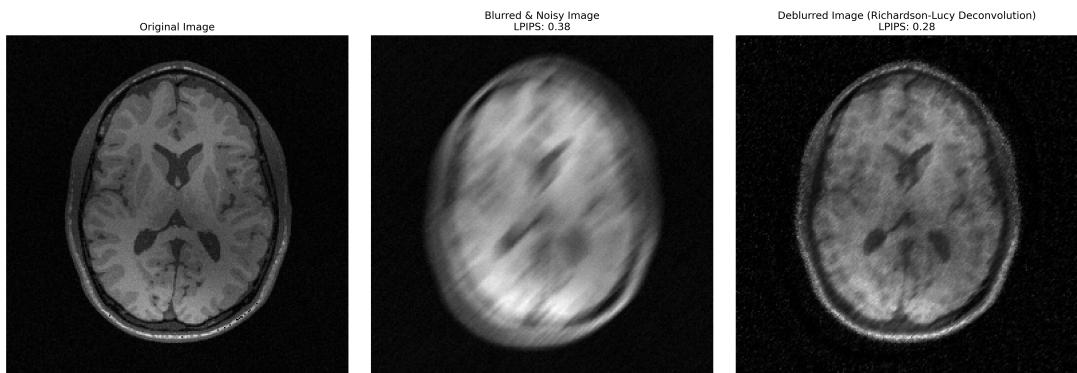


Figure 9: Kernel Size - 20, Angle - 45, Noise Sigma - 3

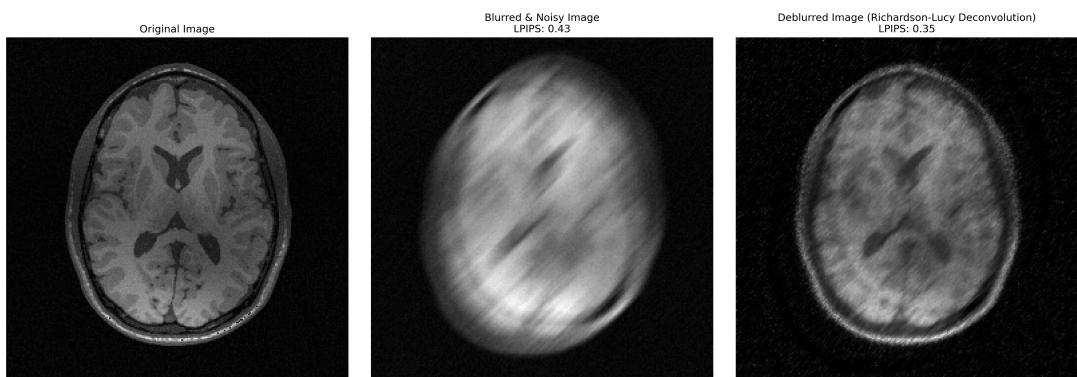


Figure 10: Kernel Size - 30, Angle - 45, Noise Sigma - 3

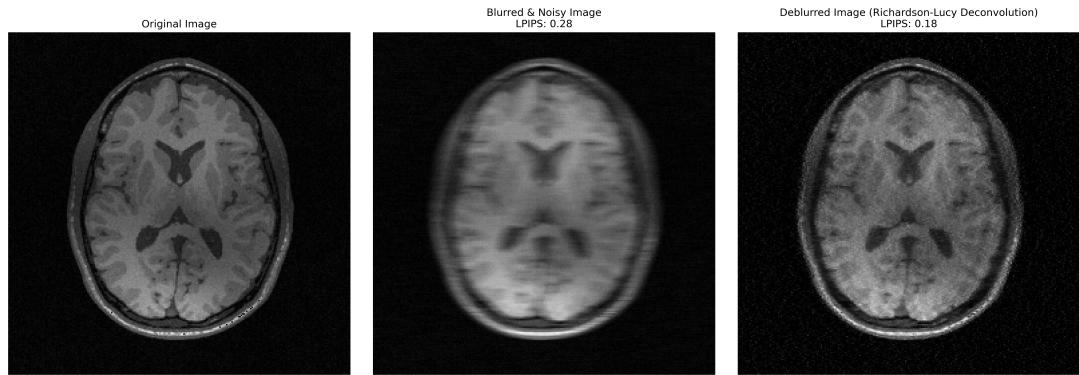


Figure 11: Kernel Size - 10, Angle - 0, Noise Sigma - 3

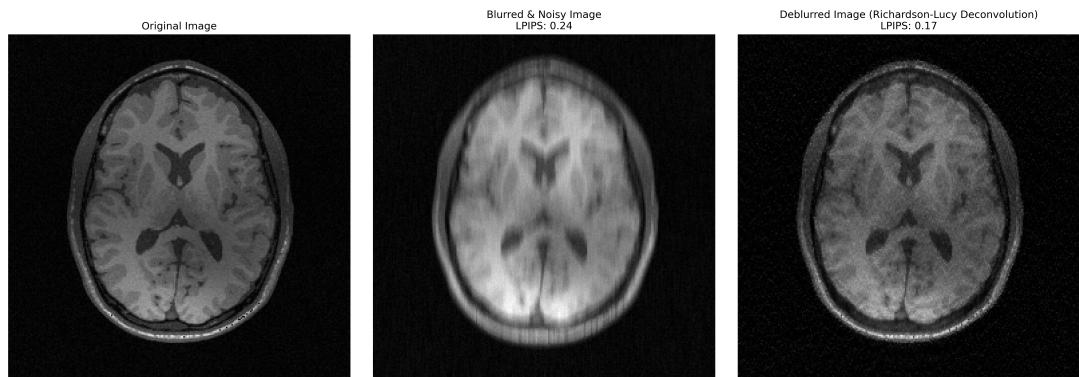


Figure 12: Kernel Size - 10, Angle - 90, Noise Sigma - 3

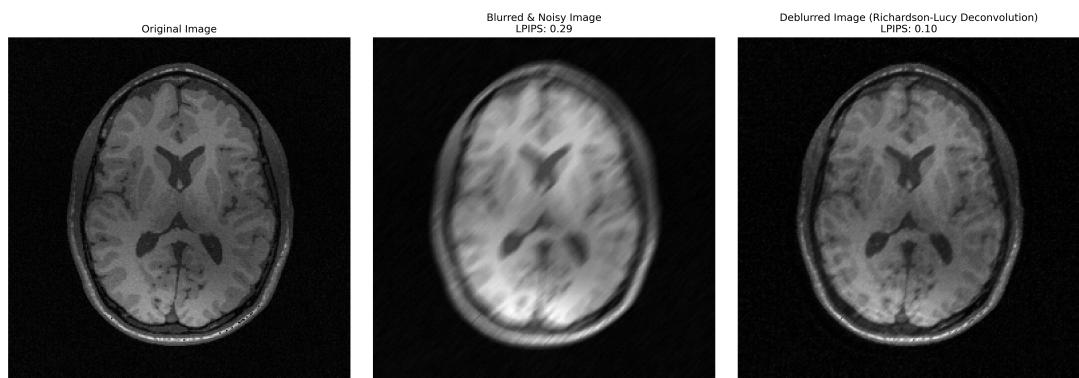


Figure 13: Kernel Size - 10, Angle - 45, Noise Sigma - 1

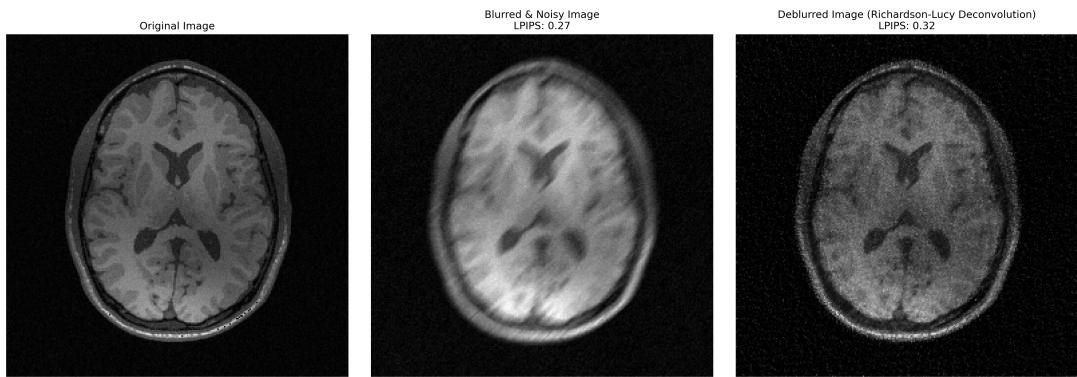


Figure 14: Kernel Size - 10, Angle - 45, Noise Sigma - 5

## 6.4 Total Variation

### 6.4.1 Brain MRI Image

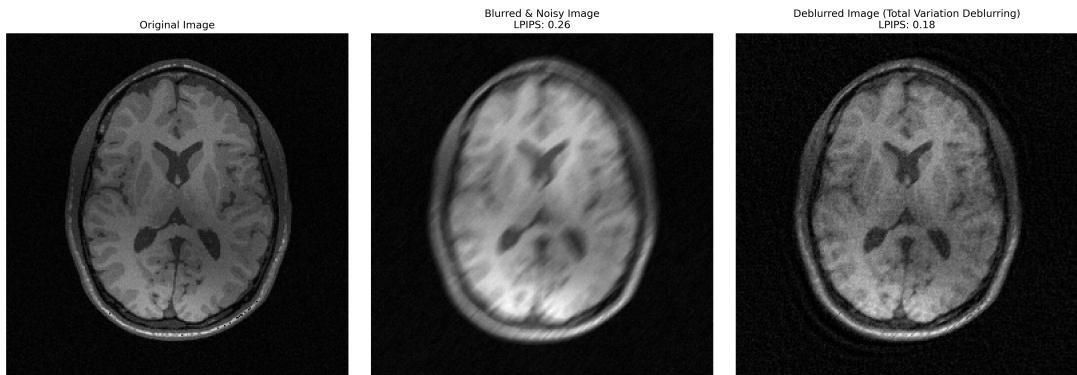


Figure 15: Kernel Size - 10, Angle - 45, Noise Sigma - 3

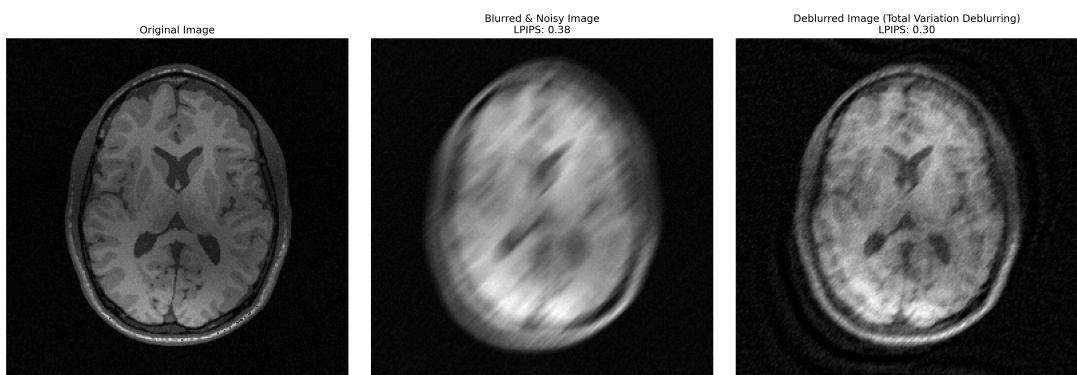


Figure 16: Kernel Size - 20, Angle - 45, Noise Sigma - 3

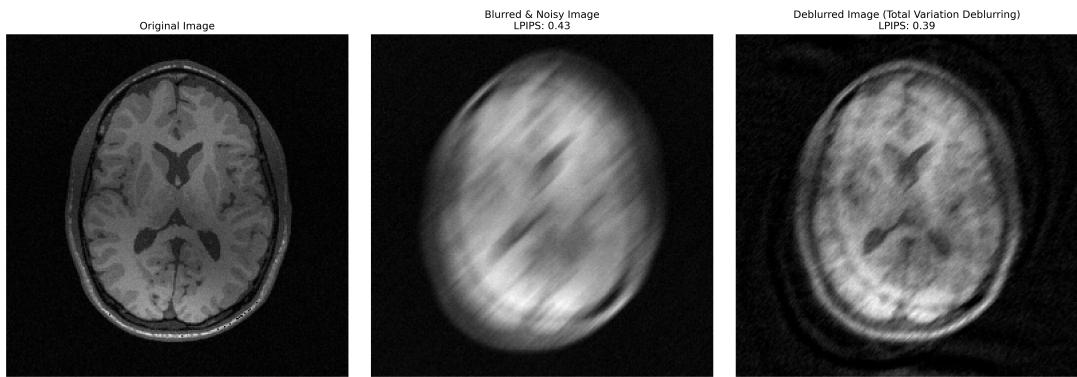


Figure 17: Kernel Size - 30, Angle - 45, Noise Sigma - 3

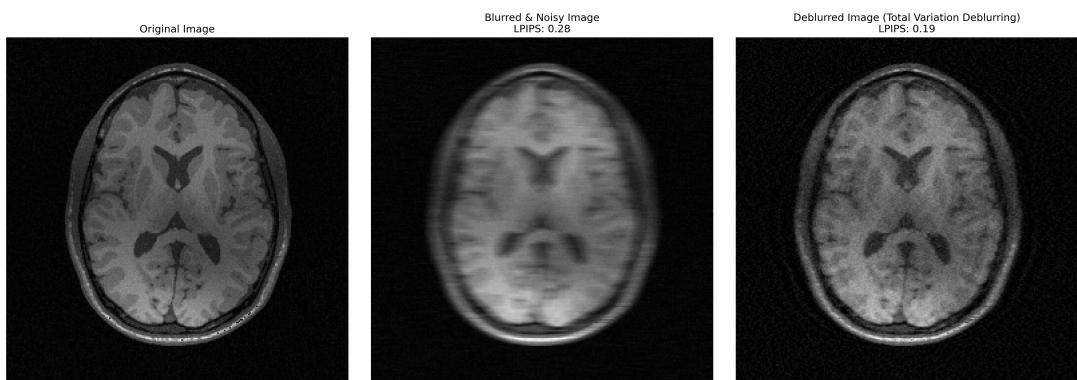


Figure 18: Kernel Size - 10, Angle - 0, Noise Sigma - 3

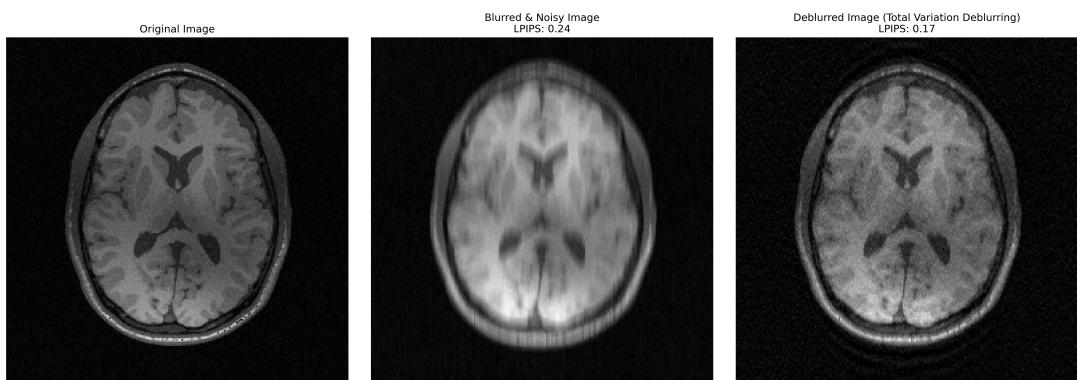


Figure 19: Kernel Size - 10, Angle - 90, Noise Sigma - 3

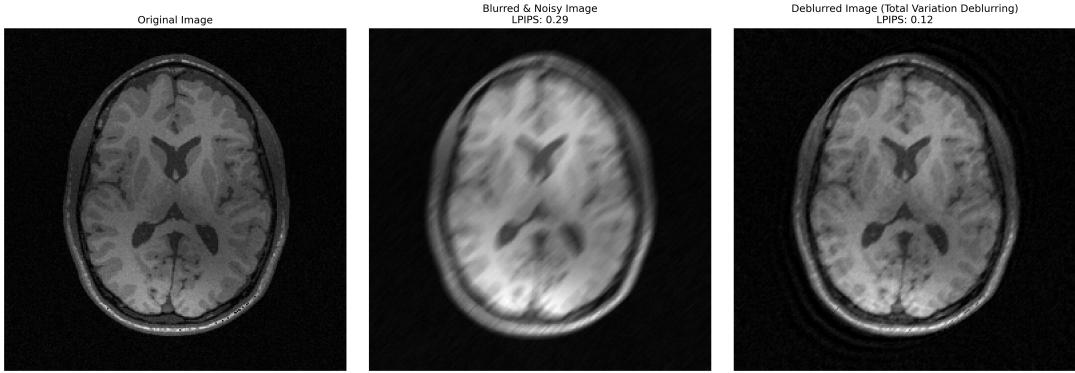


Figure 20: Kernel Size - 10, Angle - 45, Noise Sigma - 1

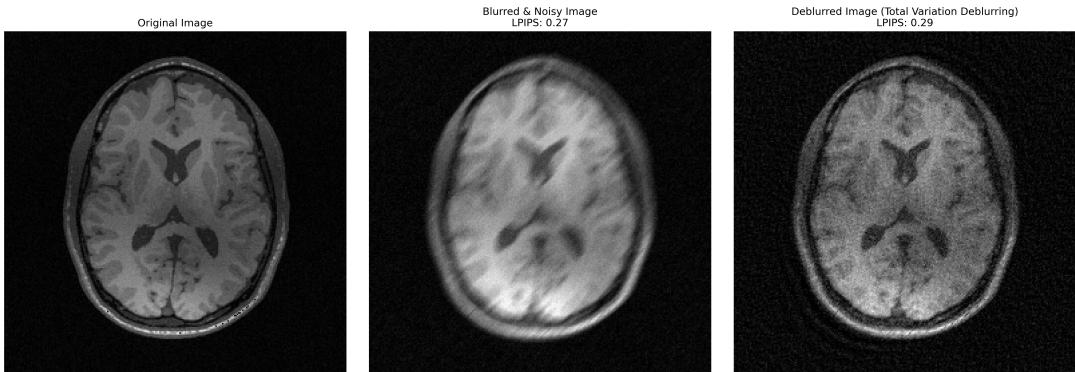


Figure 21: Kernel Size - 10, Angle - 45, Noise Sigma - 5

## 7 Effect of Parameters on Deblurring Performance

To understand the behavior of each deblurring algorithm under different image degradation settings, we analyzed the influence of the blur kernel size, the noise standard deviation, and the angle of the Point Spread Function (PSF). Each parameter was varied independently to assess its effect on the restoration quality.

### 7.1 Effect of Kernel Size

Three kernel sizes were tested:  $k = 10, 20, 30$ .

- **Wiener Deconvolution:**

- Performance degrades significantly as kernel size increases due to poor frequency domain inversion in the presence of strong low-pass filtering.
- Amplifies noise when the PSF becomes too broad.

- **Richardson-Lucy:**

- Handles larger kernels better due to its iterative nature.

- However, requires more iterations and is more sensitive to noise as kernel size increases.

- **Total Variation:**

- More resilient to larger kernels.
- Maintains edge sharpness even as blur grows, though at the cost of increased smoothness.

## 7.2 Effect of Noise Level ( $\sigma = 1, 3, 5$ )

Higher noise levels were simulated by adding Gaussian noise to the blurred image.

- **Wiener Deconvolution:**

- Unstable with increasing noise
- Amplifies noise, causing artifacts and detail loss.

- **Richardson-Lucy:**

- Sensitive to noise — noise gets amplified with iterations.
- Requires additional denoising mechanisms or early stopping.

- **Total Variation:**

- Performs best under high noise, due to the total variation term penalizing spurious detail.
- Excellent at preserving structure while removing noise.

## 7.3 Effect of PSF Orientation (Angles: $0^\circ, 45^\circ, 90^\circ$ )

The PSF was rotated to simulate motion in different directions.

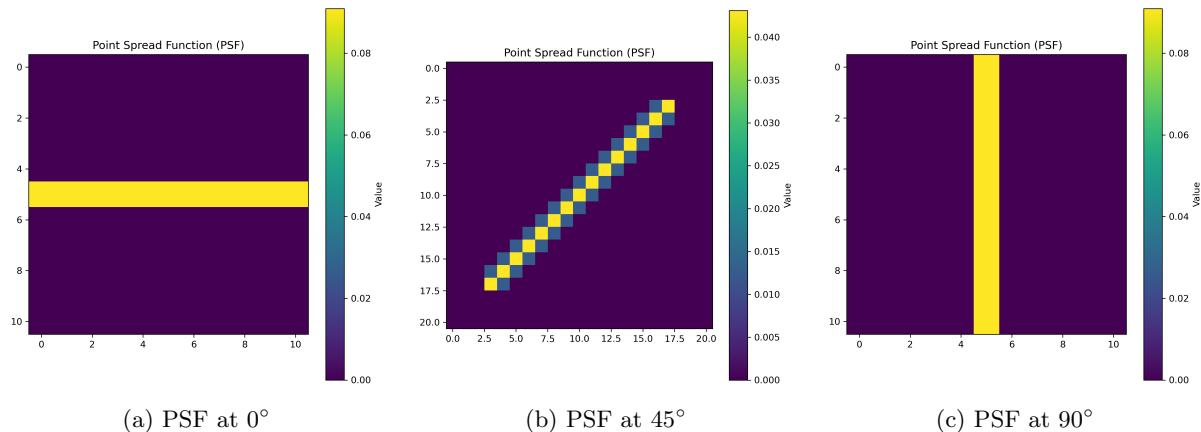


Figure 22: Visualization of Point Spread Functions (PSFs) at different motion blur angles.

- **Wiener Deconvolution:**

- Isotropic behavior in Fourier space causes it to perform similarly regardless of orientation.
- Slight directional sensitivity due to numerical artifacts.

- **Richardson-Lucy:**

- Performs well across orientations, but artifacts become more directional as PSF angle increases.
- Sharper recovery along the blur direction.

- **Total Variation:**

- Handles directional blur consistently.
- Slight bias in smoothing depending on gradient direction, but more isotropic than Richardson-Lucy.

## 7.4 Comparison of Algorithms

### 7.4.1 Kernel Size - 20, Angle - 45, Noise Sigma - 3

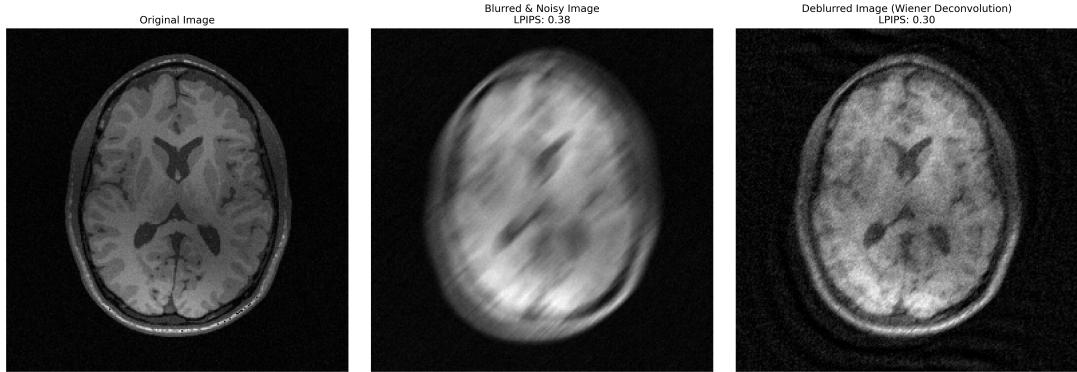


Figure 23: Wiener Deconvolution

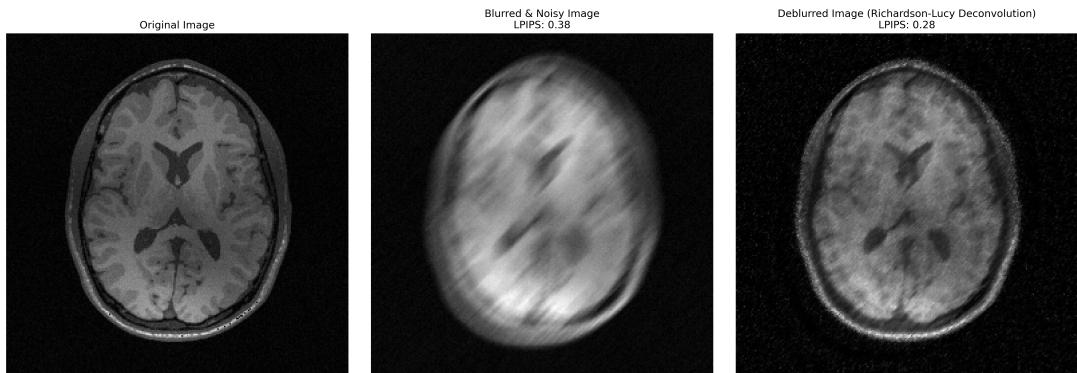


Figure 24: Richardson Lucy Deconvolution

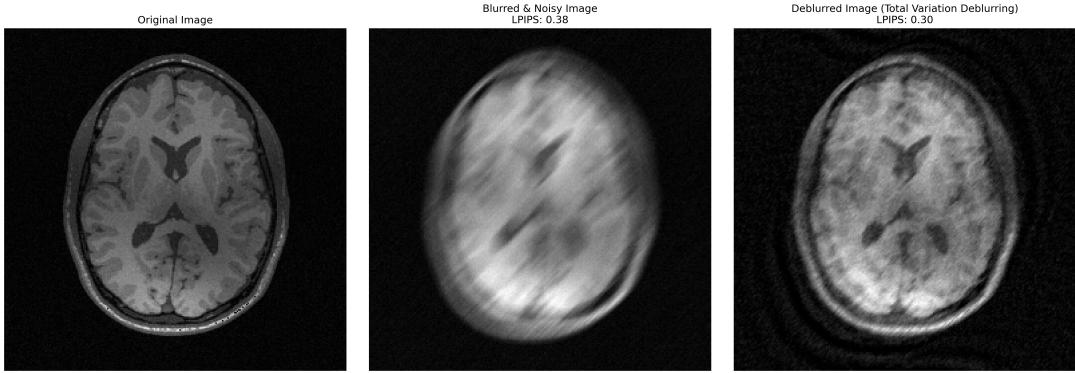


Figure 25: Total Variation Deconvolution

## 8 Observations

This section summarizes visual and algorithmic differences observed when applying Wiener Deconvolution, Richardson-Lucy Deconvolution, and Total Variation Regularization to the same blurred and noisy MRI brain image. All methods were tested under consistent degradation and parameter settings.

### 8.1 Visual Quality and Restoration

- **Wiener Deconvolution:**

- The restored image shows limited recovery of fine anatomical details.
- Although it reduces some of the blurring, edge sharpness is not fully recovered.
- Ringing artifacts and background texture distortions are more visible.
- LPIPS: 0.30 — indicating moderate perceptual improvement over the blurred input.

- **Richardson-Lucy Deconvolution:**

- Delivers the sharpest visual output among all three methods.
- Recovers intricate structures such as gyri, sulci, and ventricles more clearly.
- However, it amplifies high-frequency noise, especially in homogeneous background regions.
- LPIPS: 0.28 — lowest among the three, reflecting higher perceptual similarity.

- **Total Variation Deblurring:**

- Achieves a good balance between edge preservation and noise suppression.
- Smooths out noise effectively while preserving critical structural edges.
- May oversmooth low-contrast regions, suppressing weak texture details.
- LPIPS: 0.30 — same as Wiener, but with qualitatively better edge integrity.

## 8.2 Computation Time and Efficiency

- **Total Variation** took significantly longer to converge than both Wiener and Richardson-Lucy due to its iterative optimization with gradient descent and regularization.
- **Wiener** is the fastest as it is computed in a single pass using frequency-domain filtering.
- **Richardson-Lucy** lies between the two, with reasonable runtime for moderate iterations.

## 8.3 Overall Trade-offs

- **Wiener** is ideal when computational speed and noise suppression are priorities.
- **Richardson-Lucy** is preferable when detail enhancement is critical, at the cost of higher noise.
- **Total Variation** offers the best compromise for medical imaging, with excellent edge preservation and acceptable runtime when quality is paramount.

# 9 Conclusion

In this project, we explored and compared three deterministic algorithms for deblurring medical images: Wiener Deconvolution, Richardson-Lucy Deconvolution, and Total Variation Regularization. Each method was implemented from scratch and evaluated under various degradation settings, including different blur kernel sizes, noise levels, and motion angles.

Our experiments demonstrate that:

- **Wiener Deconvolution** is computationally efficient and performs well under low noise, but struggles with strong blur and fails to preserve sharp features.
- **Richardson-Lucy Deconvolution** offers superior detail recovery and edge sharpness but is sensitive to noise and may amplify artifacts with excessive iterations.
- **Total Variation Regularization** provides a robust balance, effectively suppressing noise while preserving structure, although at a higher computational cost.

Overall, the choice of algorithm depends on the specific application needs—speed and simplicity versus accuracy and edge preservation. For diagnostic medical imaging, where precision and clarity are paramount, Total Variation Regularization emerges as a strong candidate. Future work may explore adaptive parameter tuning and hybrid approaches that combine the strengths of these methods.