

Geometric Transformation & Frequency Domain

H.M.R.M. Pushpakumara
200488E

Geometric Transformation



Left leg or Right leg?



Left leg or Right leg?

Flip Image Filter



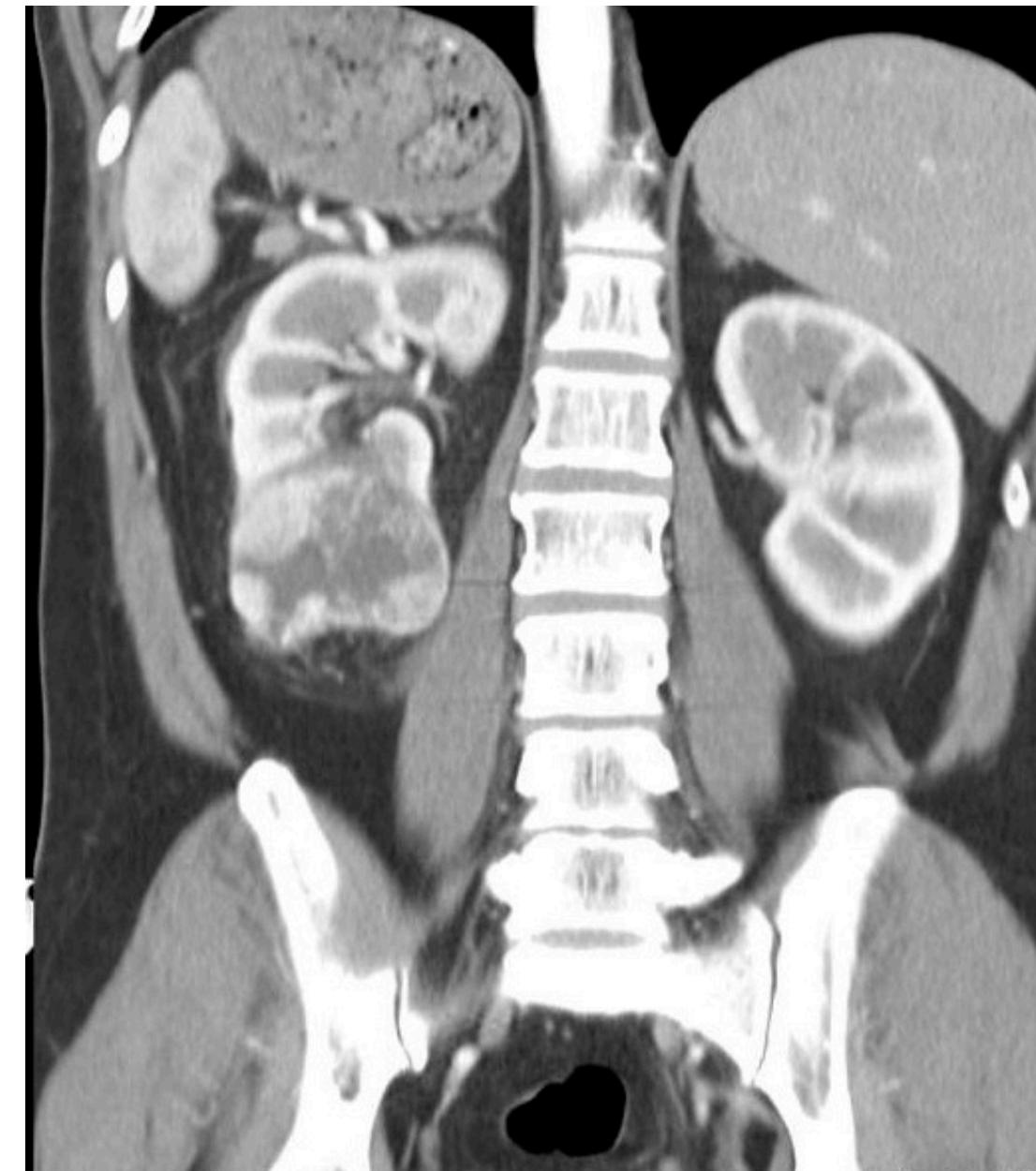
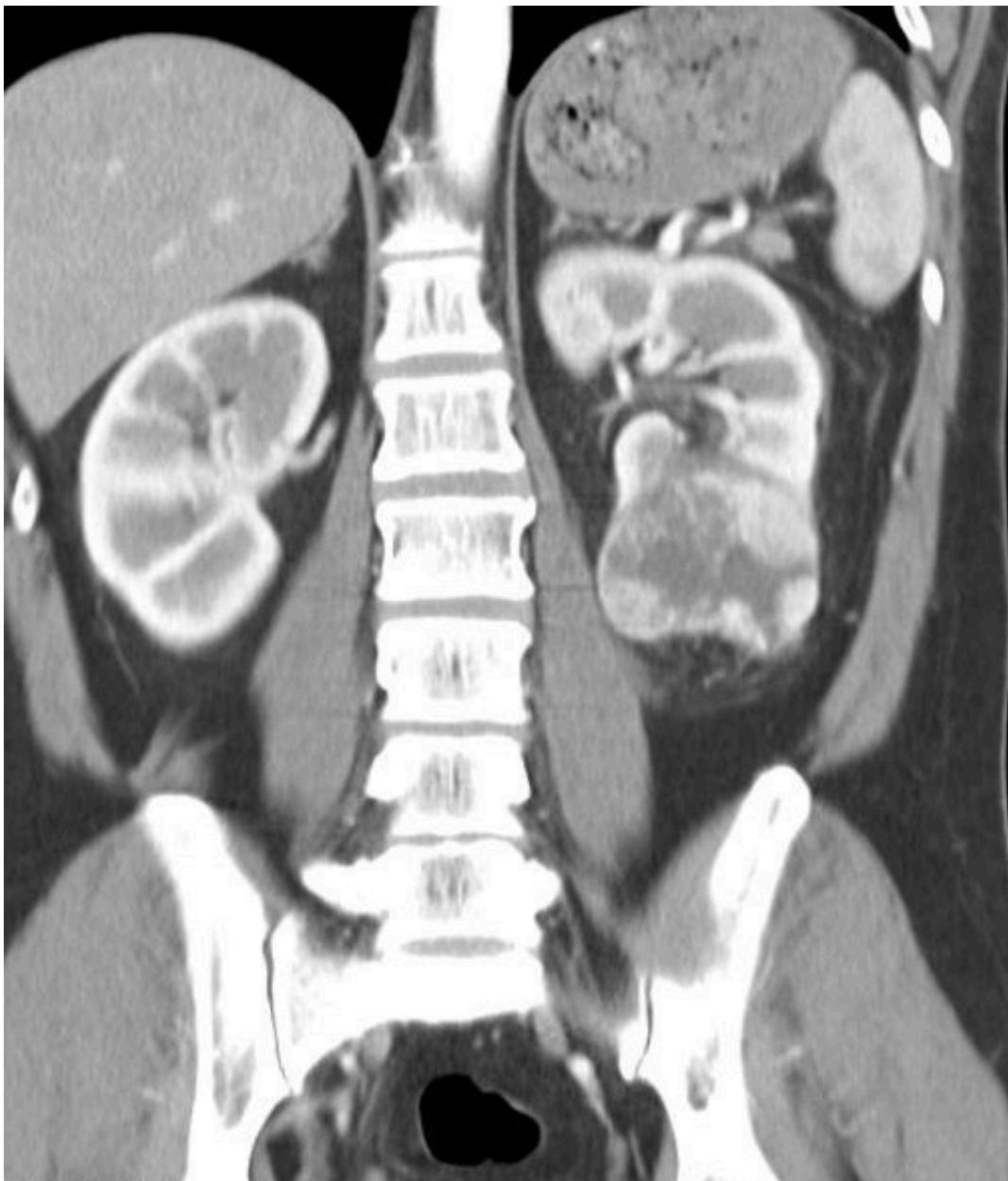
Left leg or Right leg?

Flip Image Filter

- This filter must be used with **EXTREME** caution

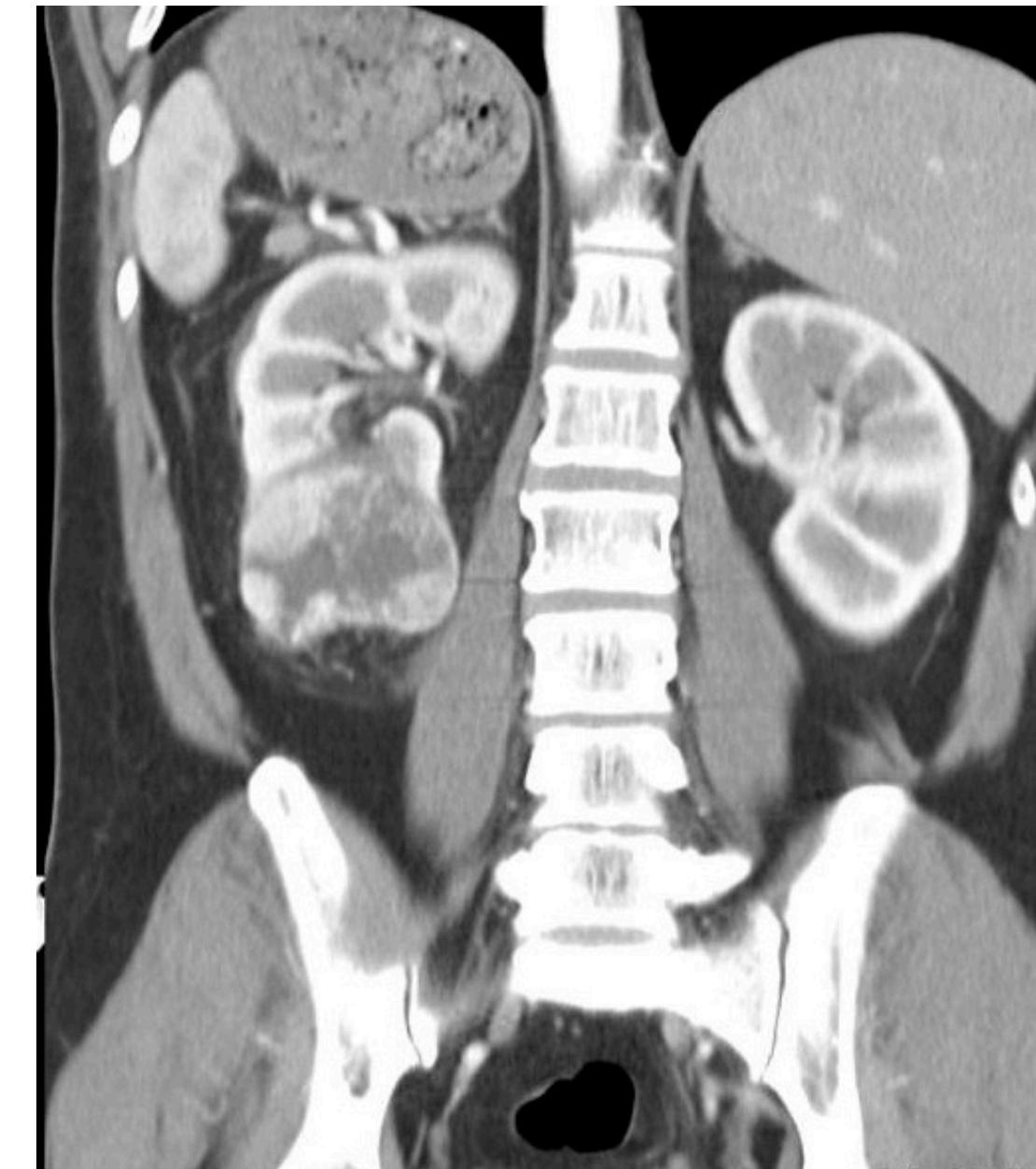
Flip Image Filter

- This filter must be used with **EXTREME** caution



Flip Image Filter

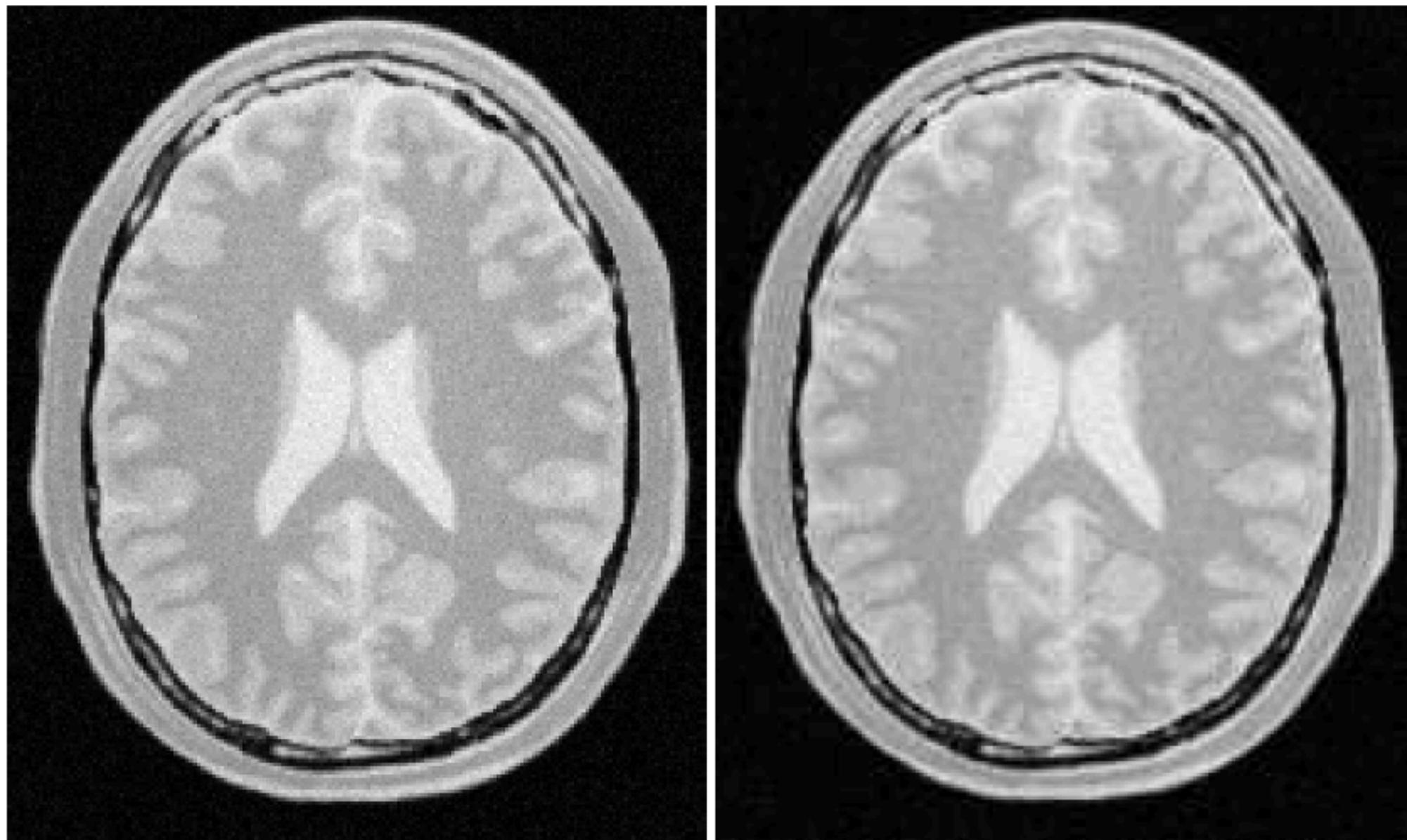
- This filter must be used with **EXTREME** caution



Wrong side surgery accounts for 2% of the reported medical errors in the United States. Trivial... but equally dangerous.

Flip Image Filter

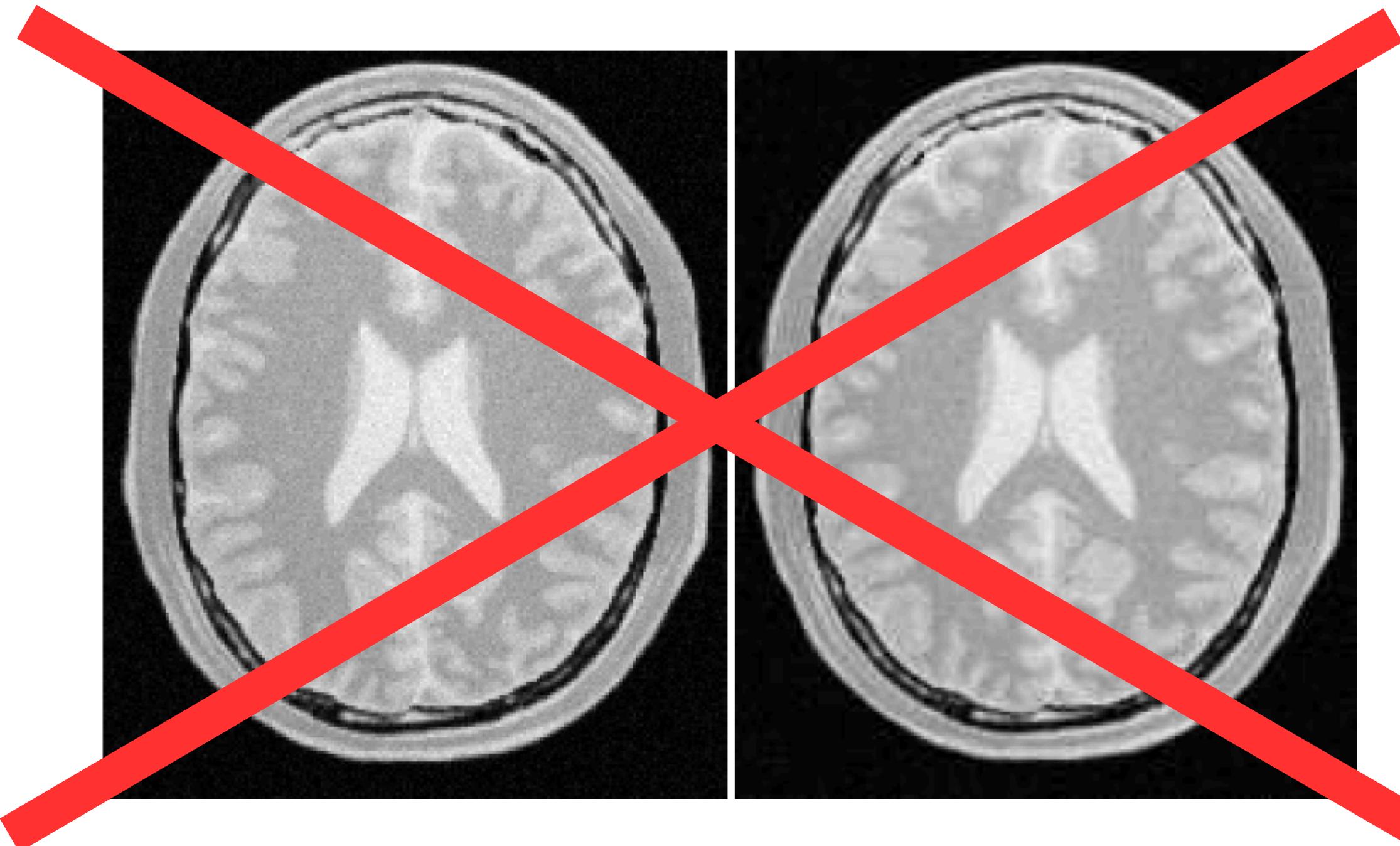
- Example that has given in the Software Guide



Can you observe any flip?

Flip Image Filter

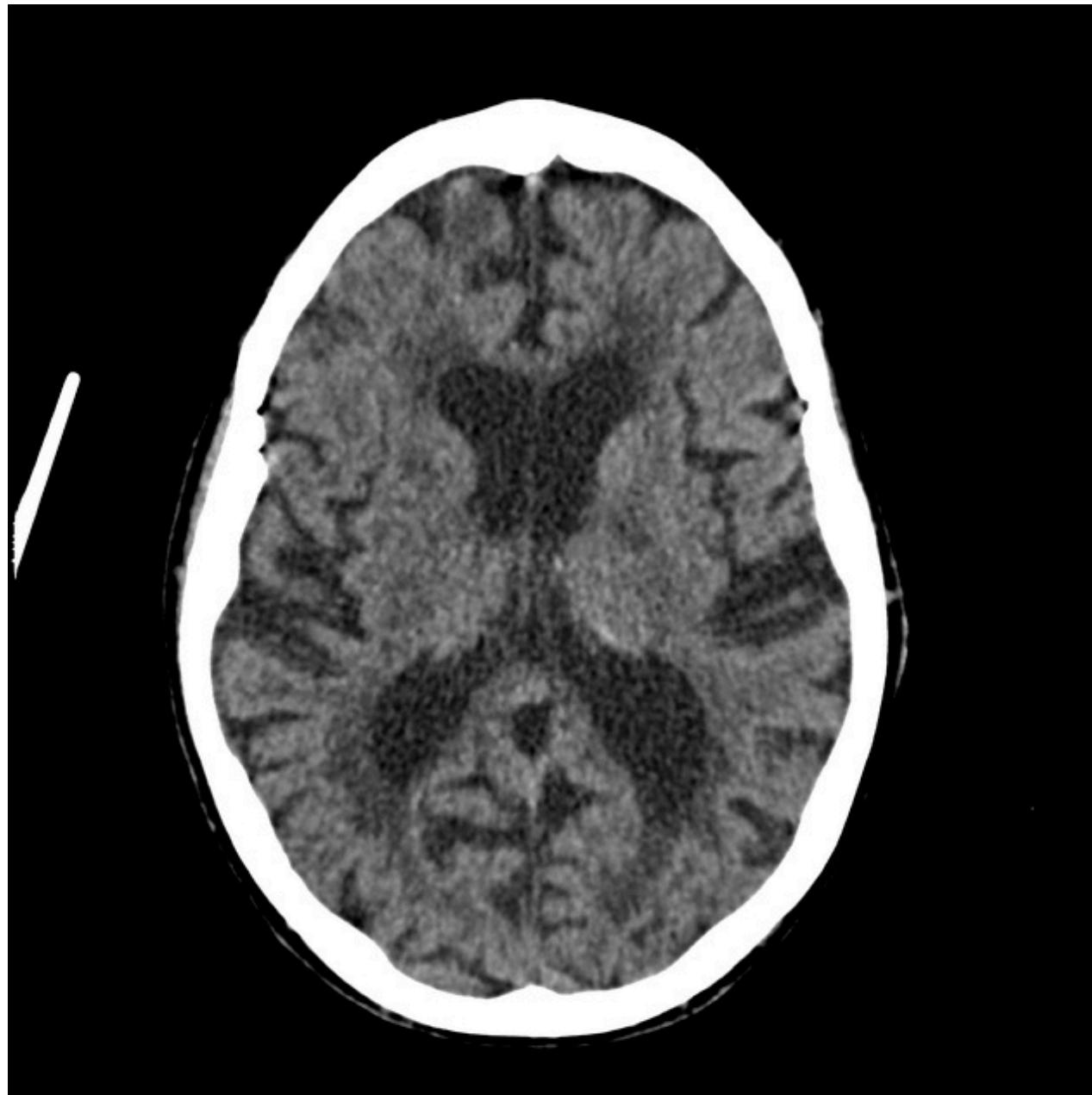
- Example that has given in the Software Guide



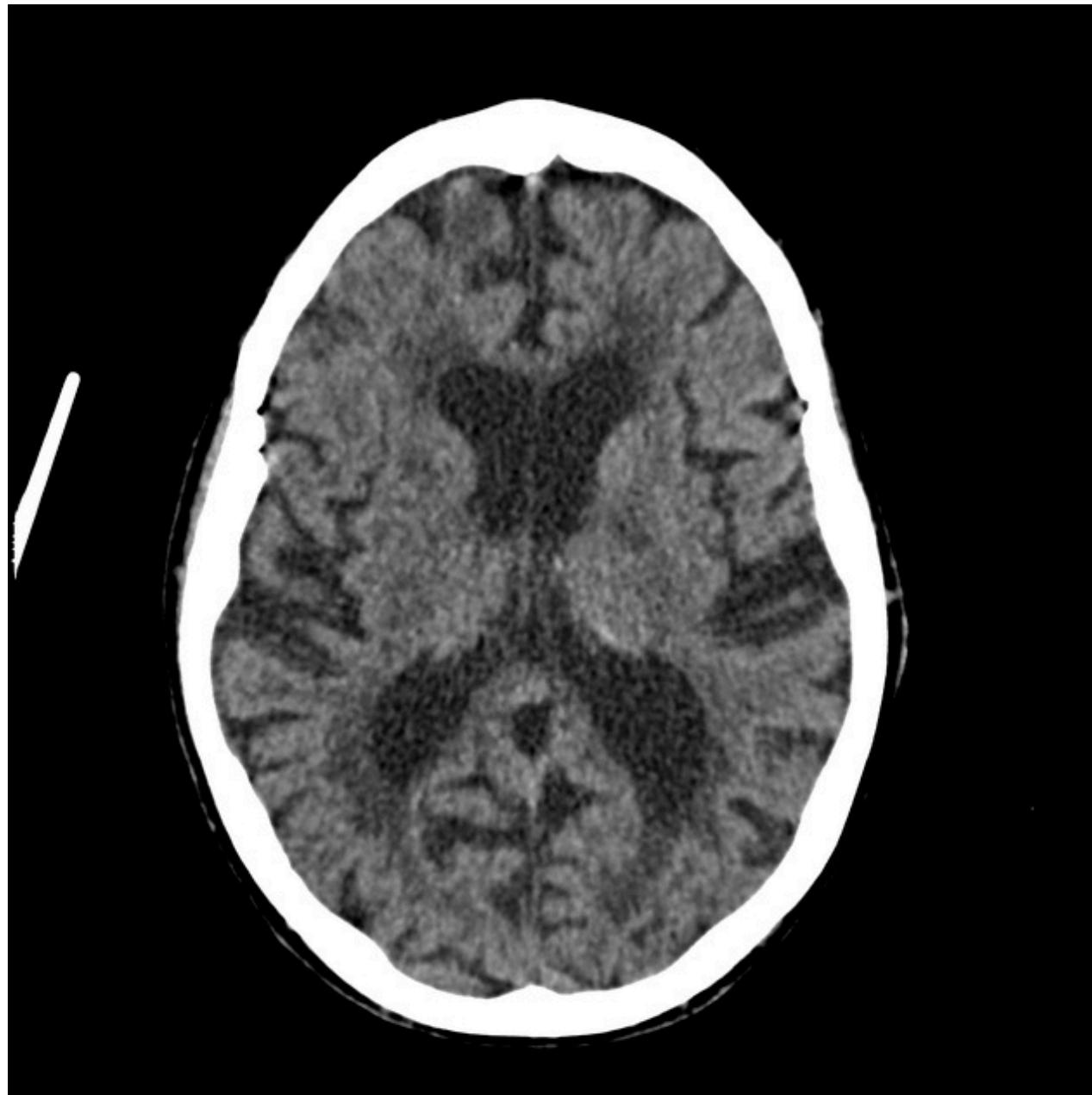
Can you observe any flip?

Don't trust even a Instruction manual

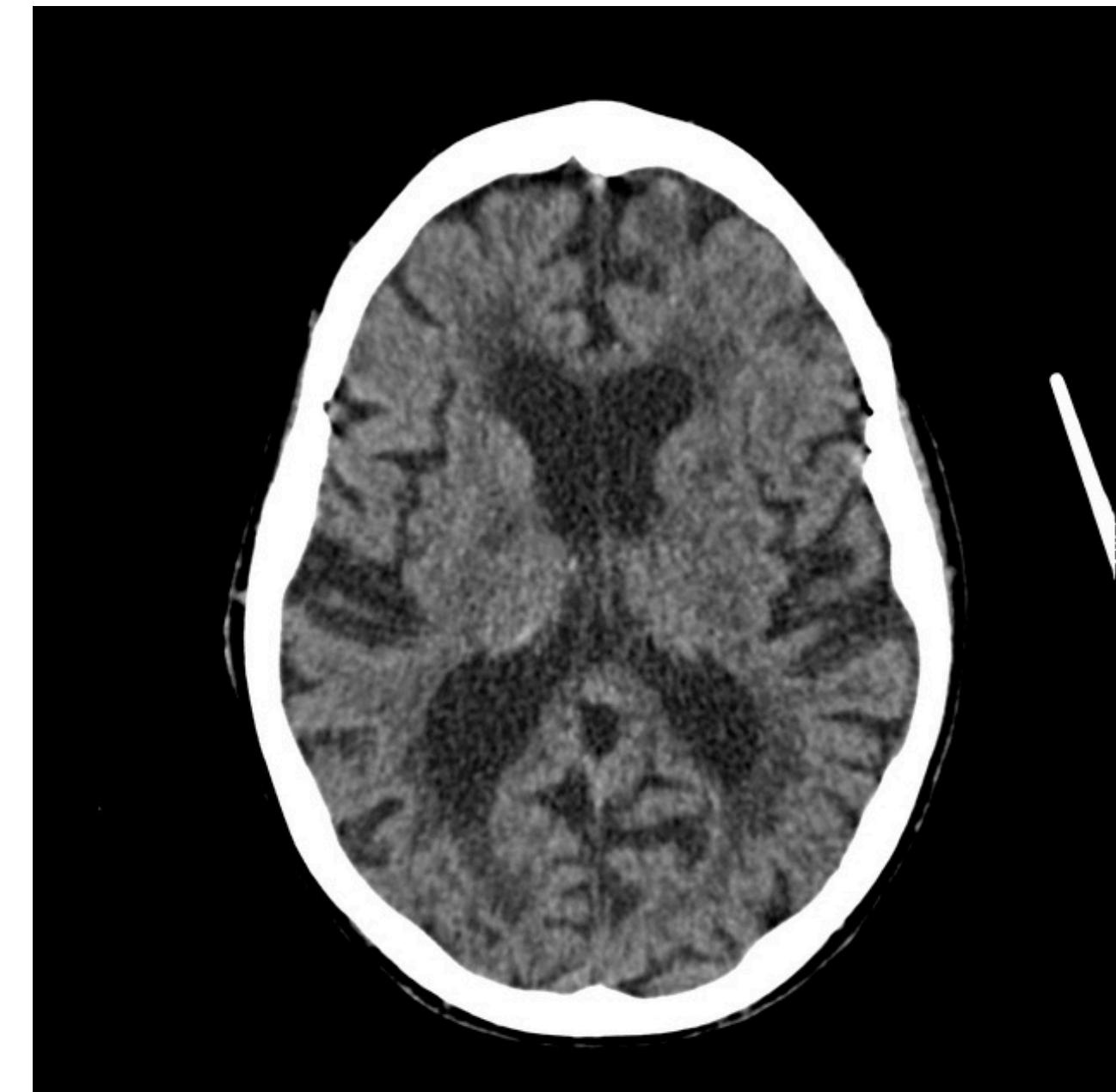
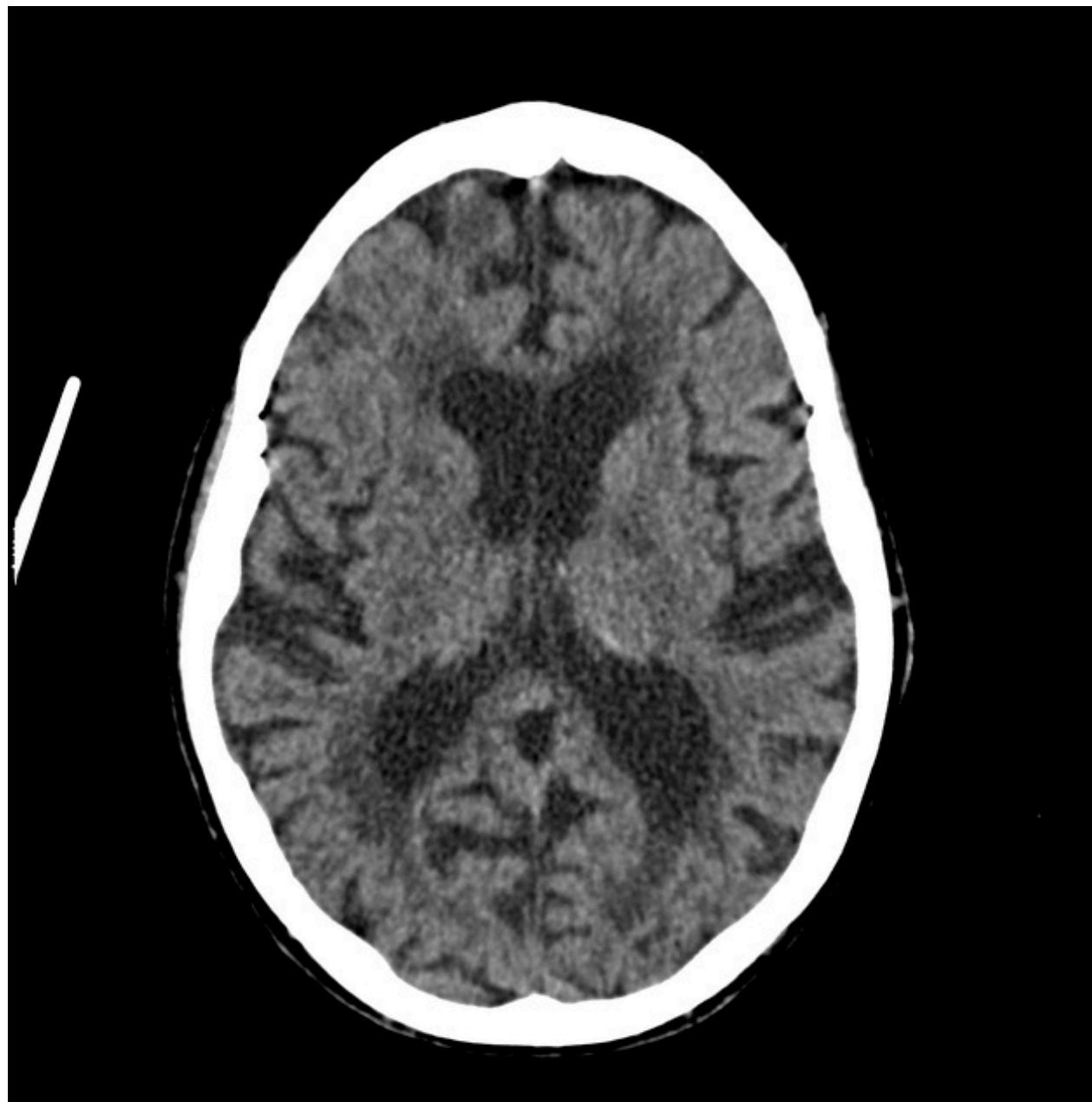
Flip Image Filter



Flip Image Filter



Flip Image Filter



Flip Image Filter

Applications

Flip Image Filter

Applications

Correcting Orientation:

- Patient Positioning Errors

Flip Image Filter

Applications

Correcting Orientation:

- Patient Positioning Errors

Data Analysis and Research

- Data Augmentation

Flip Image Filter

Applications

Correcting Orientation:

- Patient Positioning Errors

Data Analysis and Research

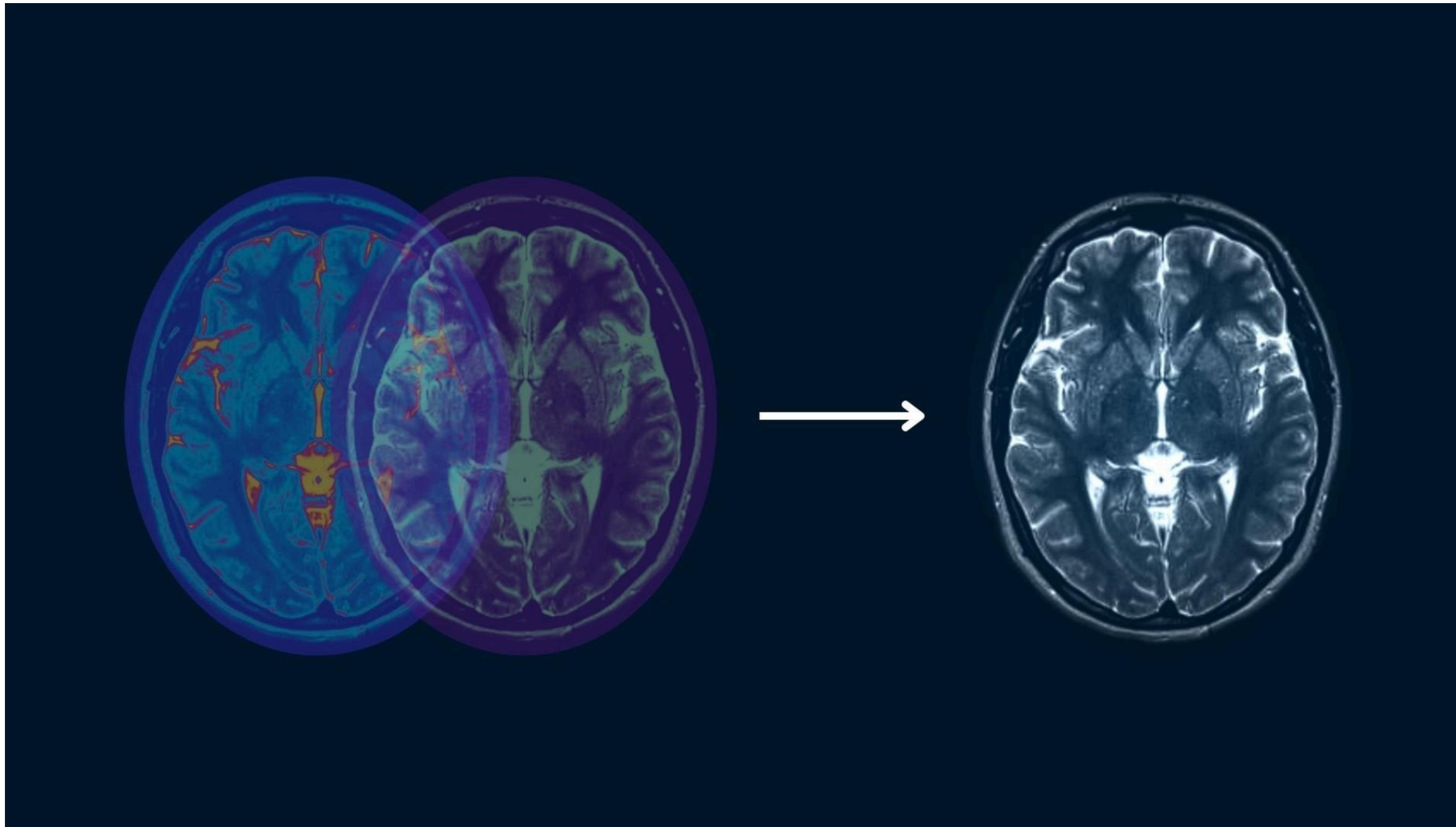
- Data Augmentation

Comparing Anatomical Structures:

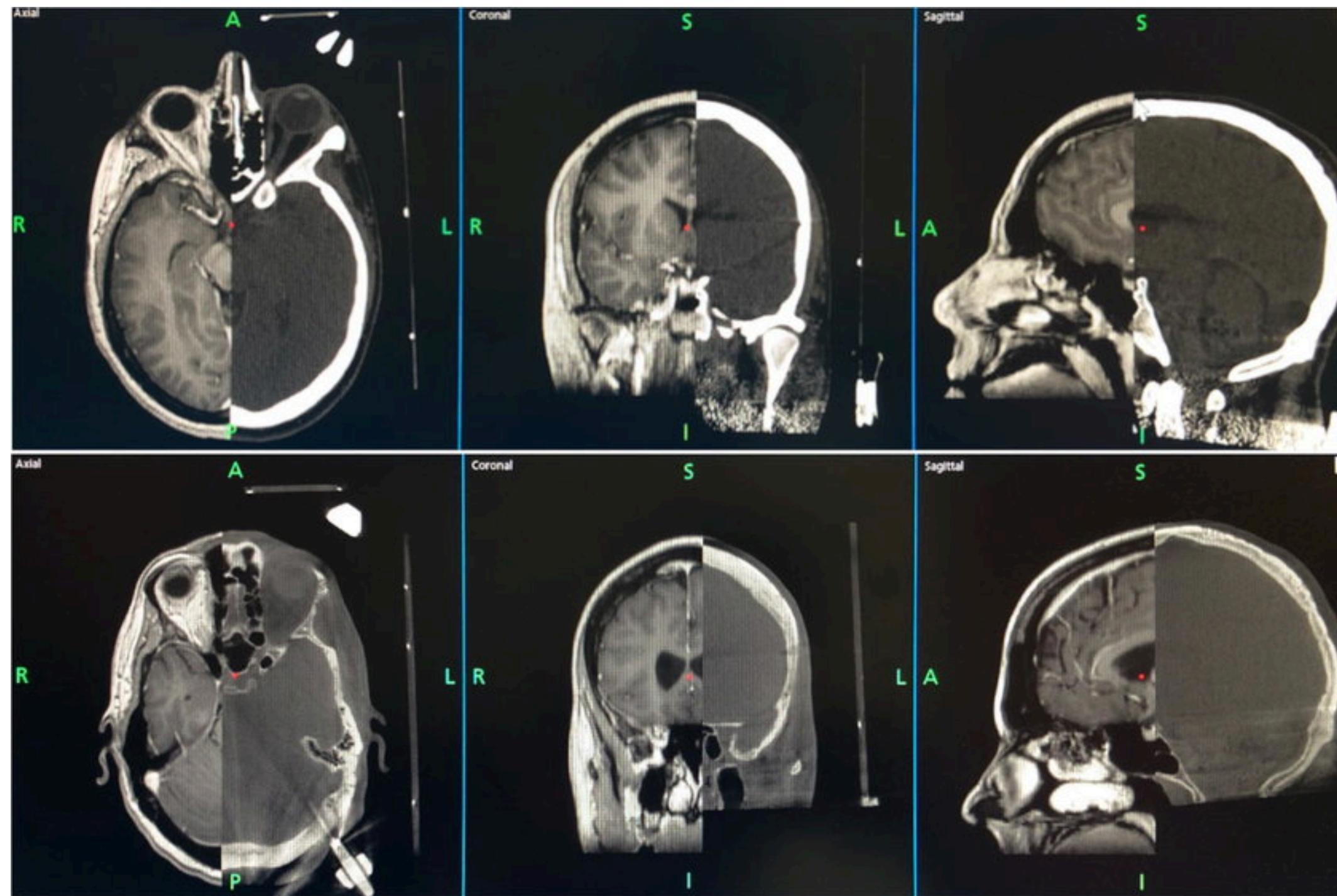
- Symmetry
- Reference Image Comparison

Do you know about “Image Registration” ?

Do you know about “Image Registration” ?



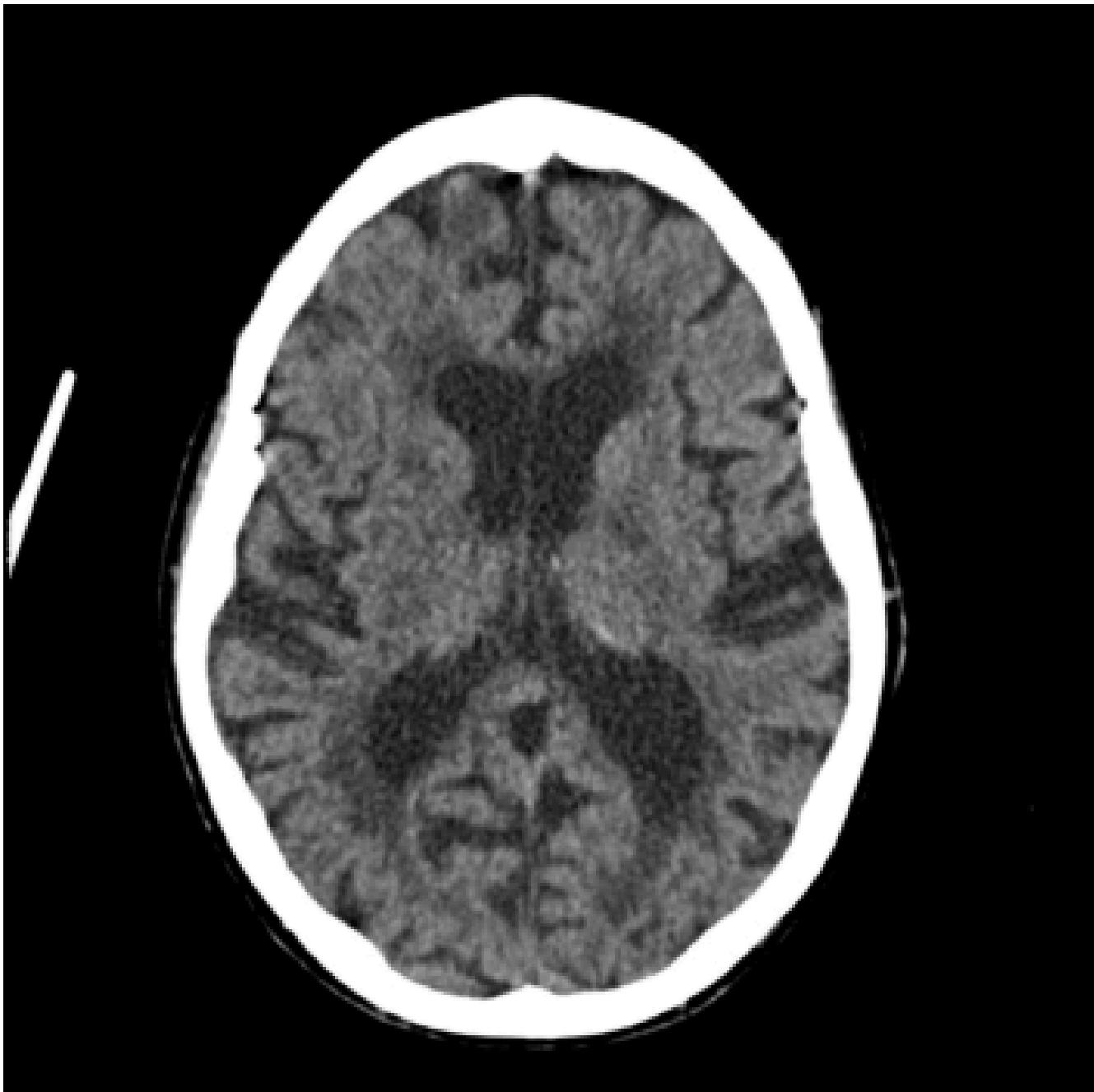
Do you know about “Image Registration” ?



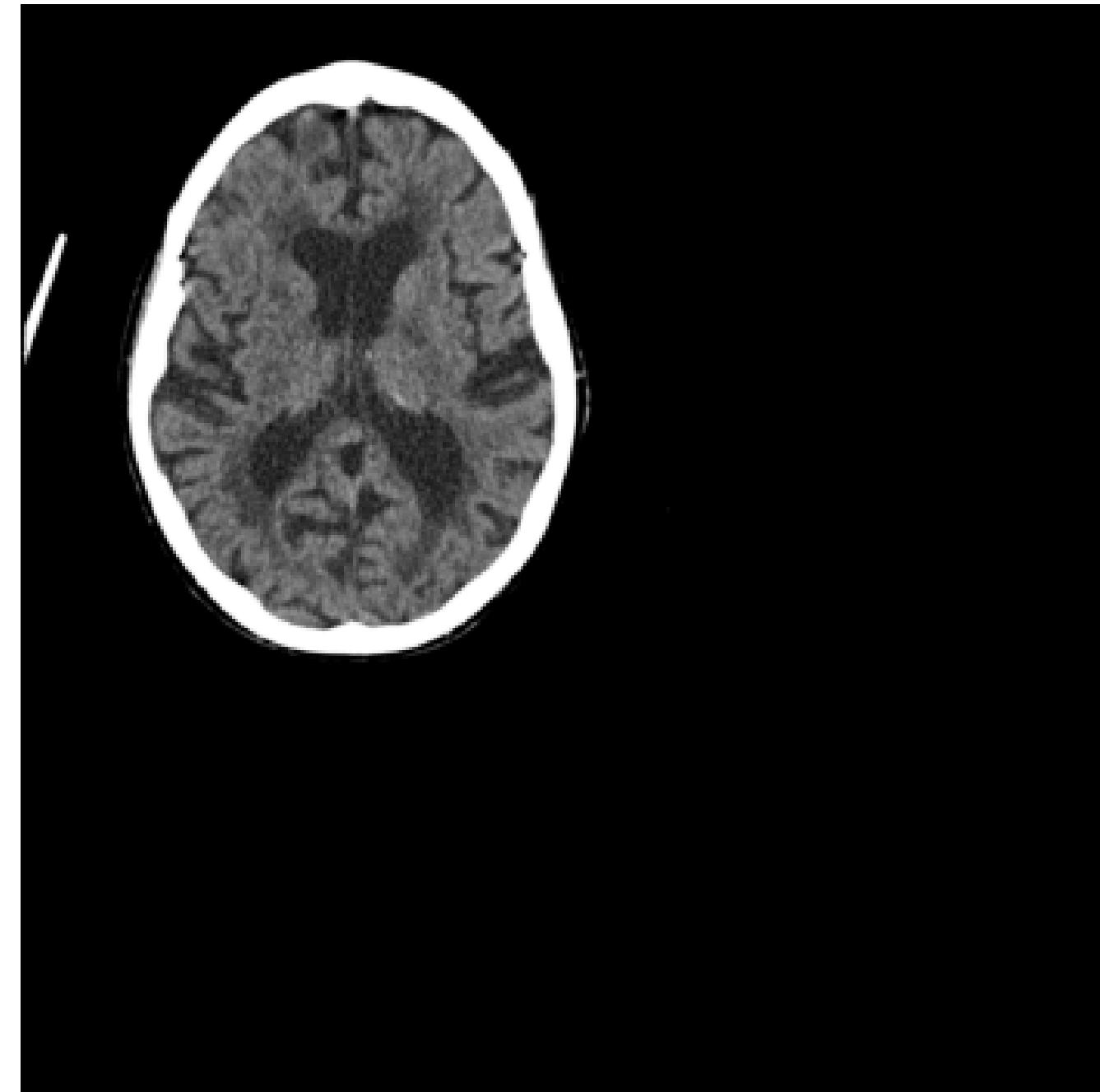
Resample Image Filter

Resample Image Filter

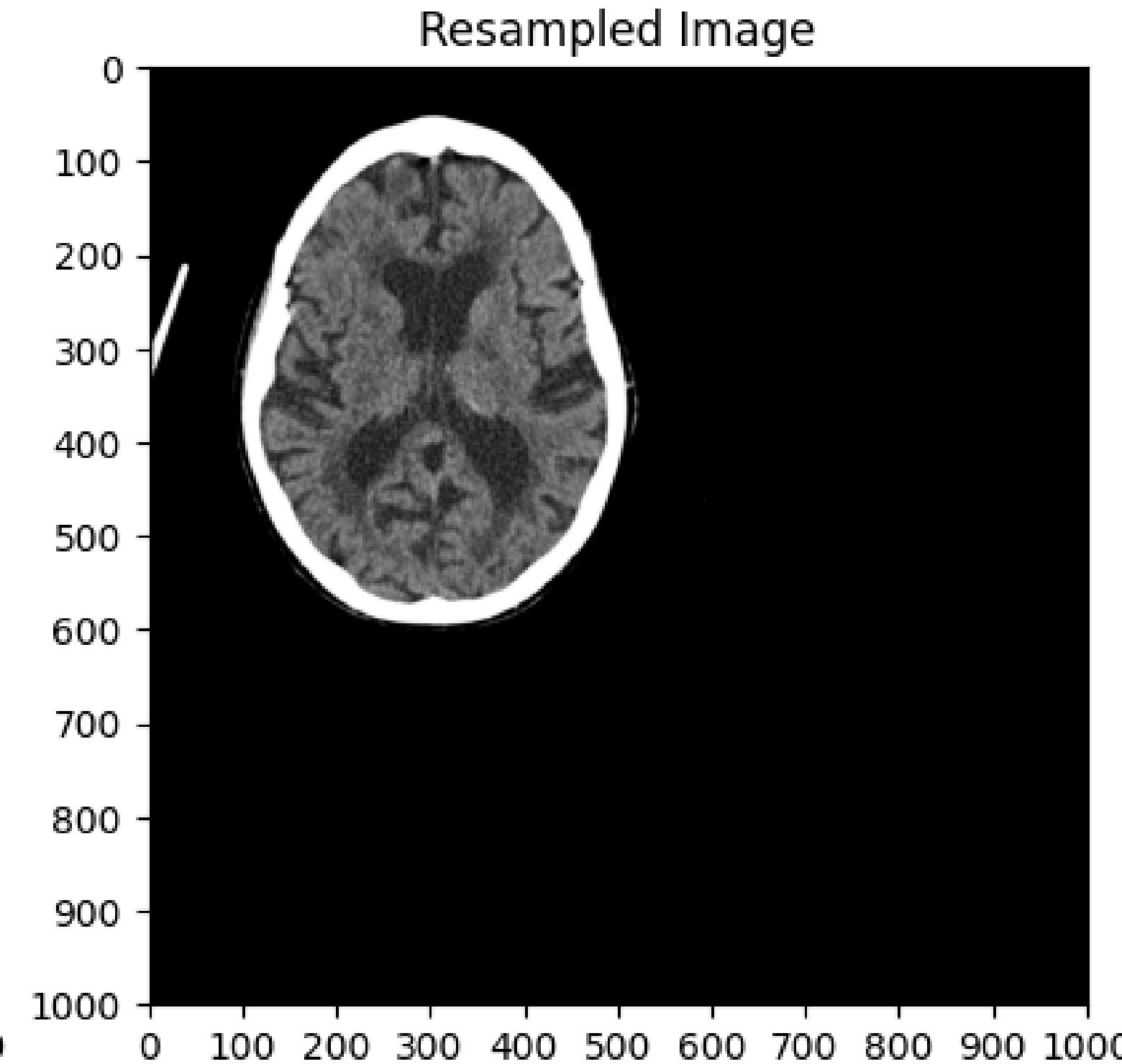
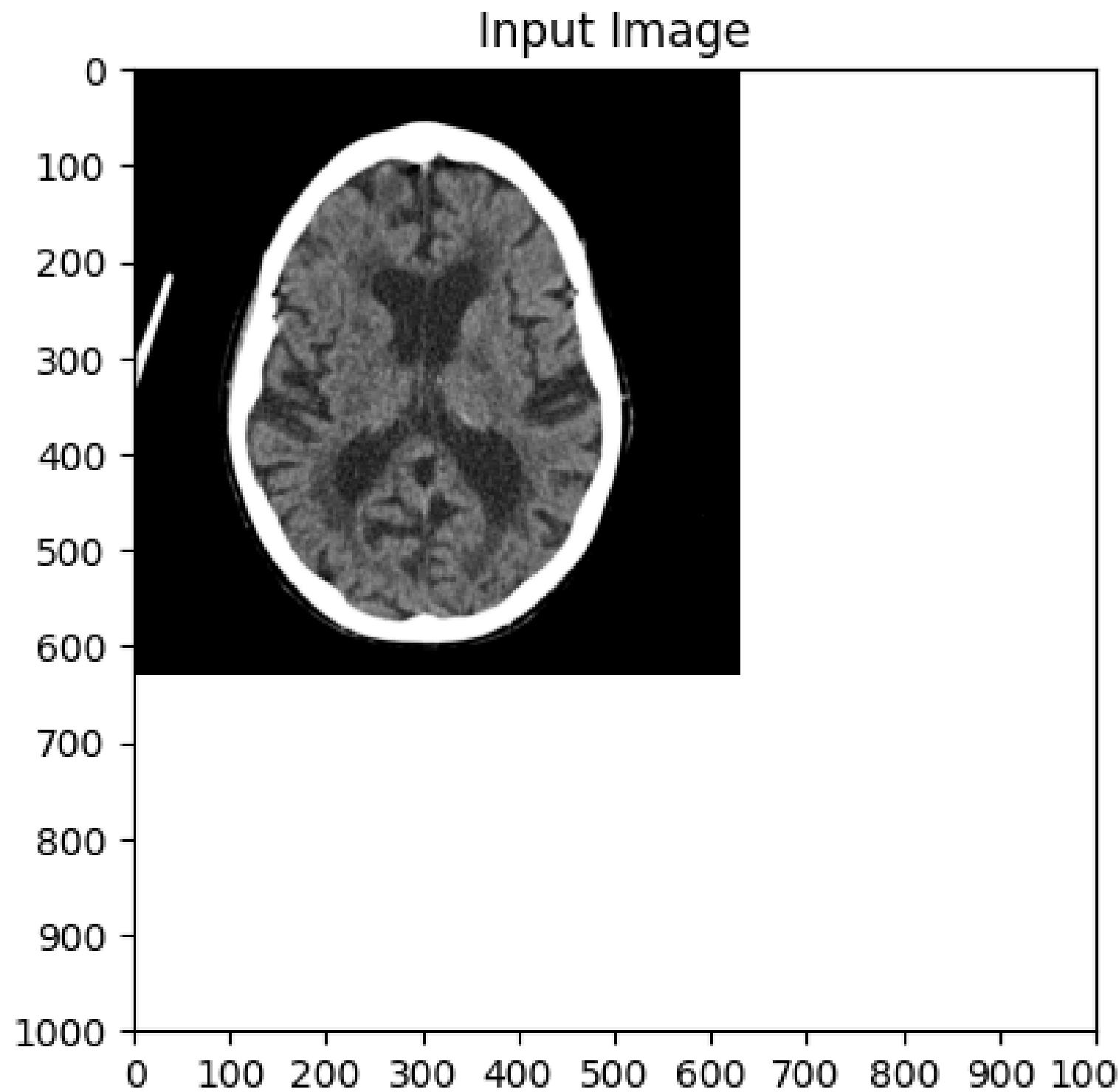
Input Image



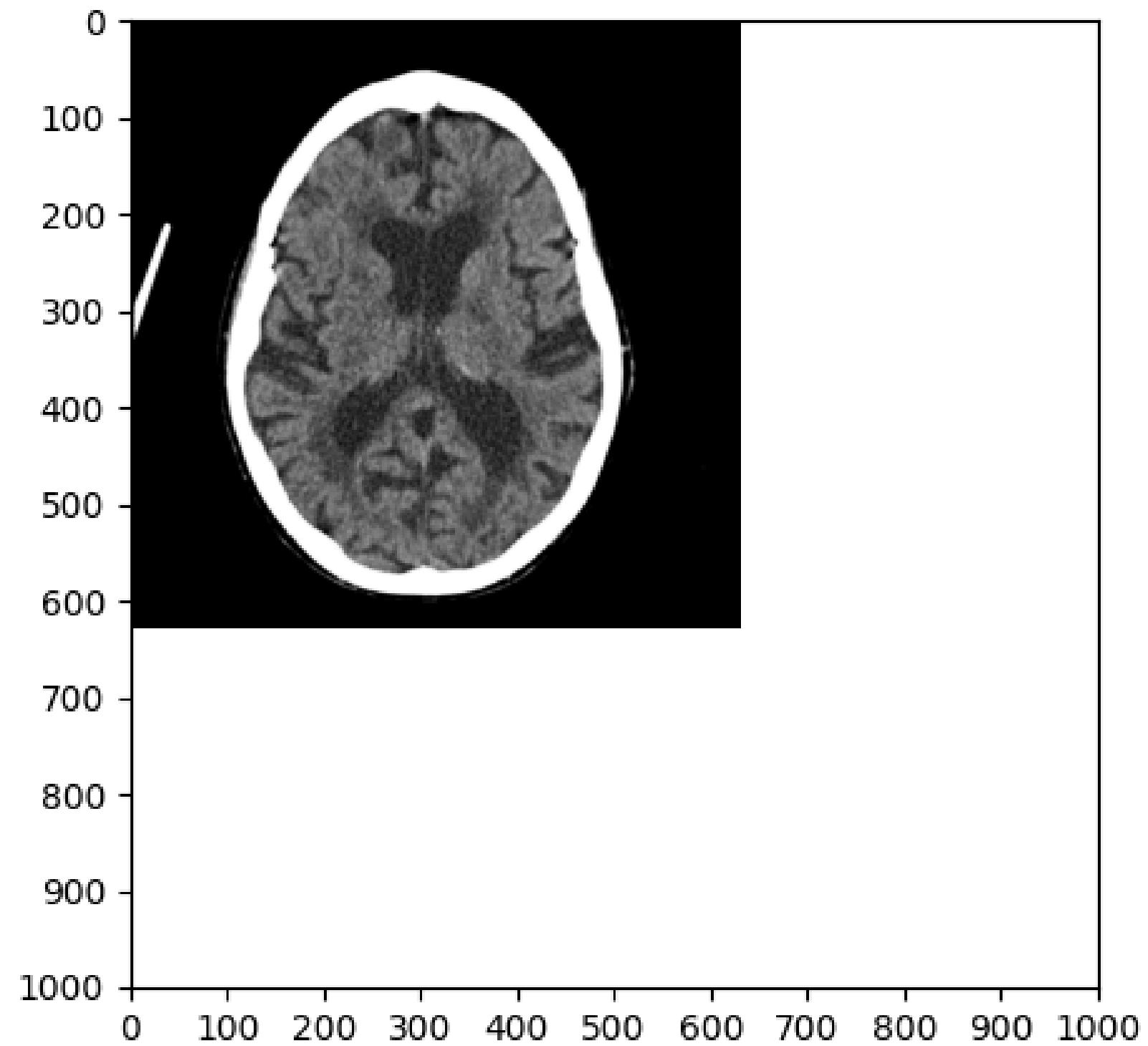
Resampled Image



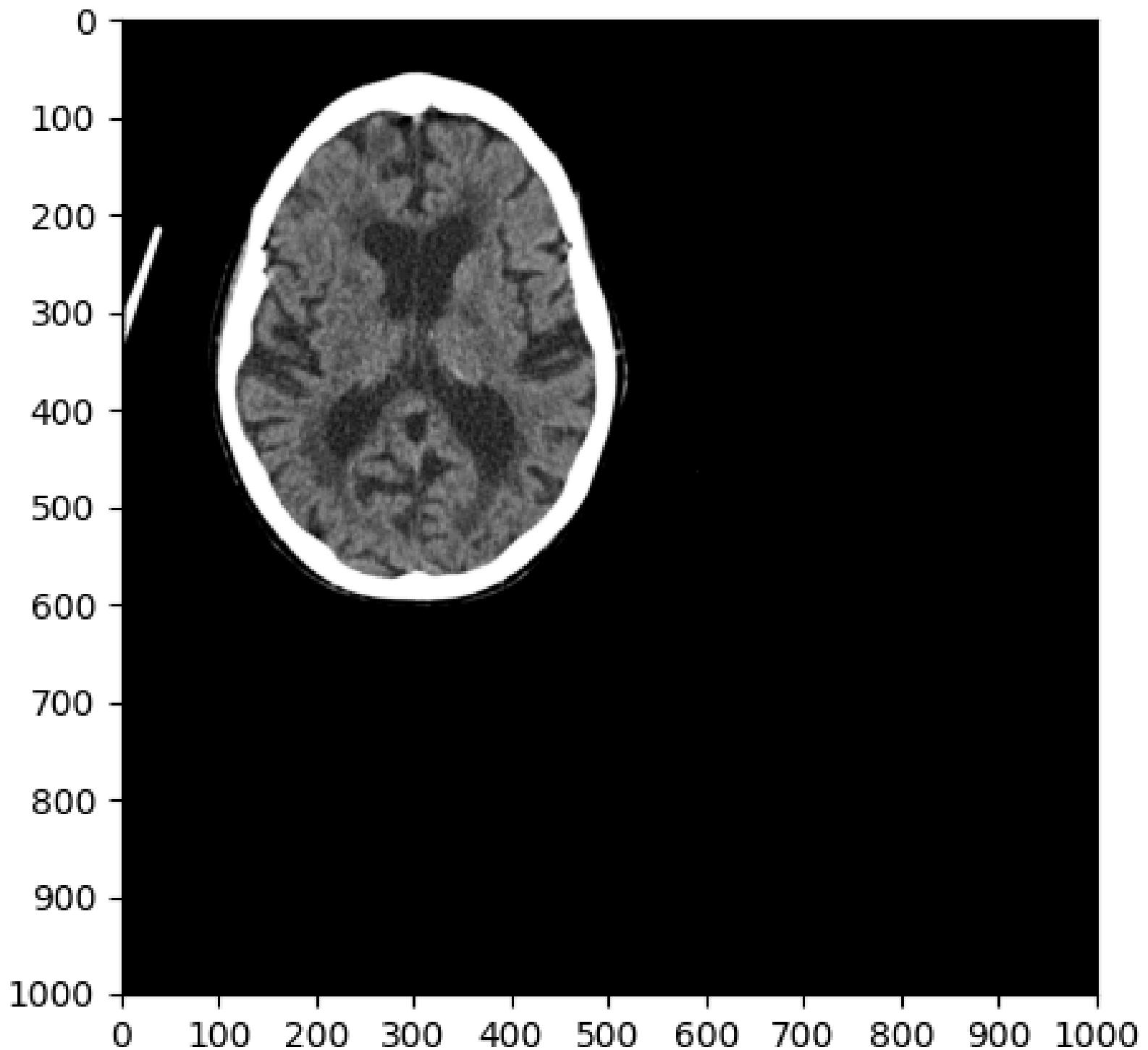
Resample Image Filter



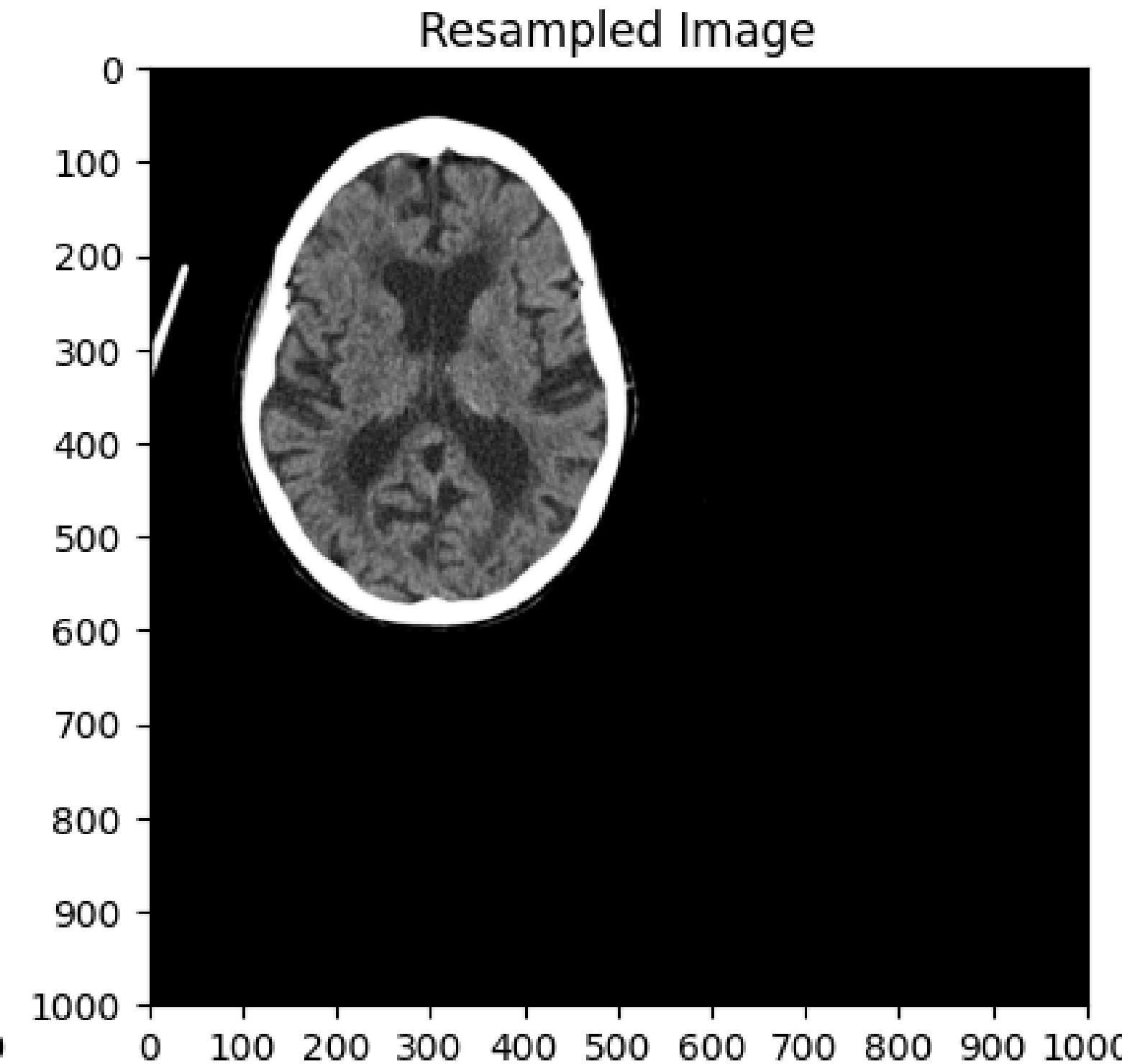
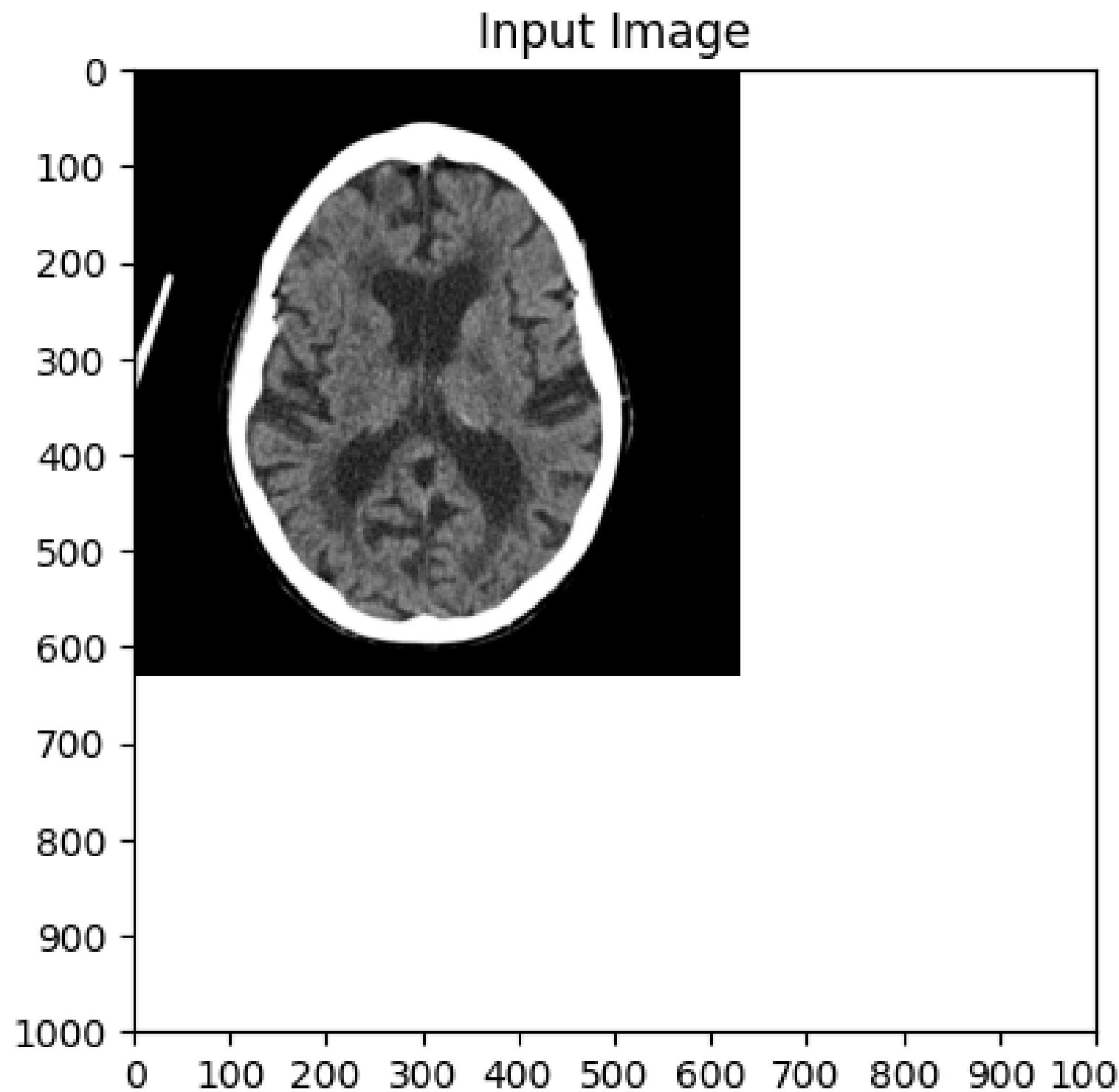
Resample Image Filter



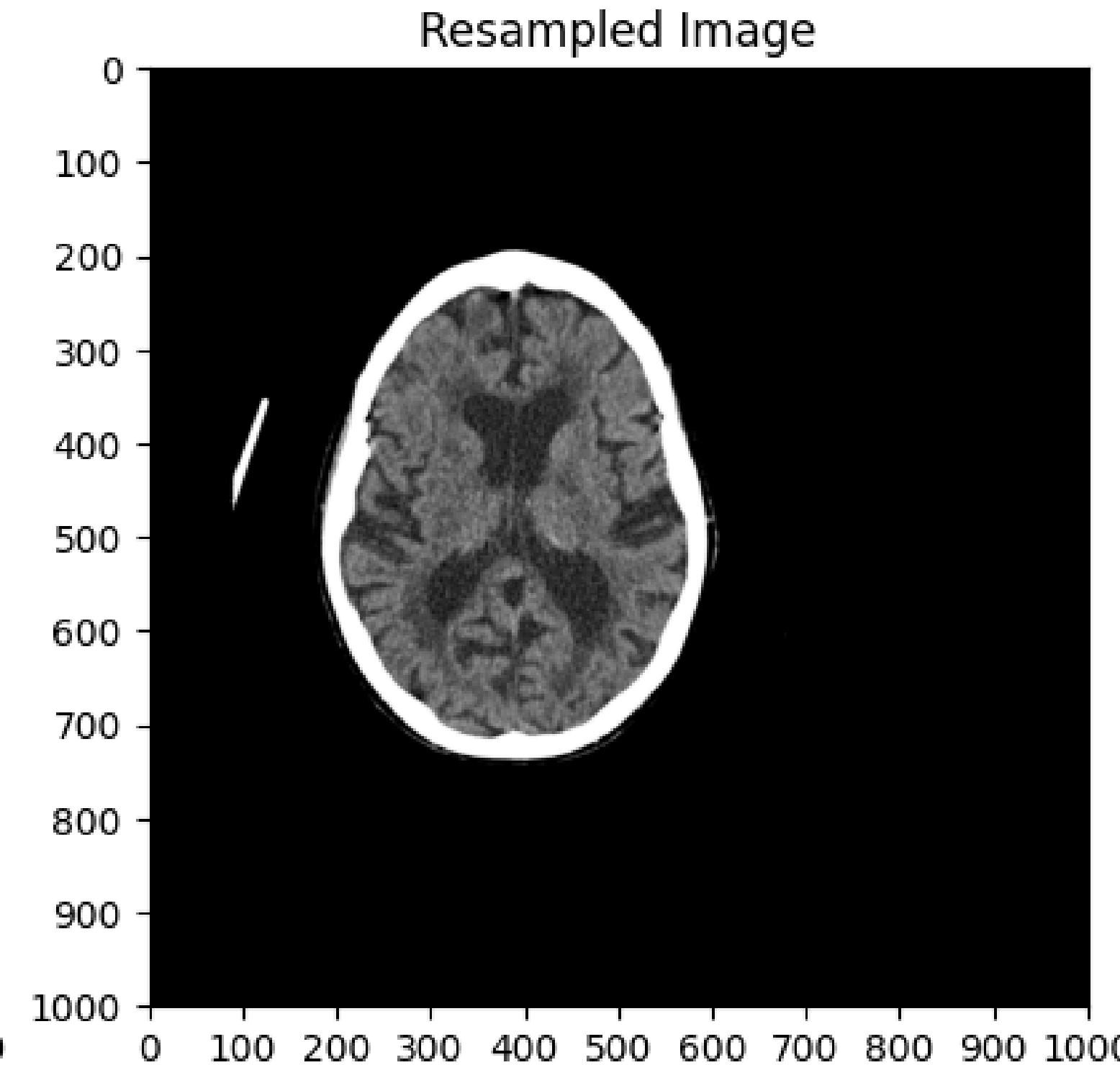
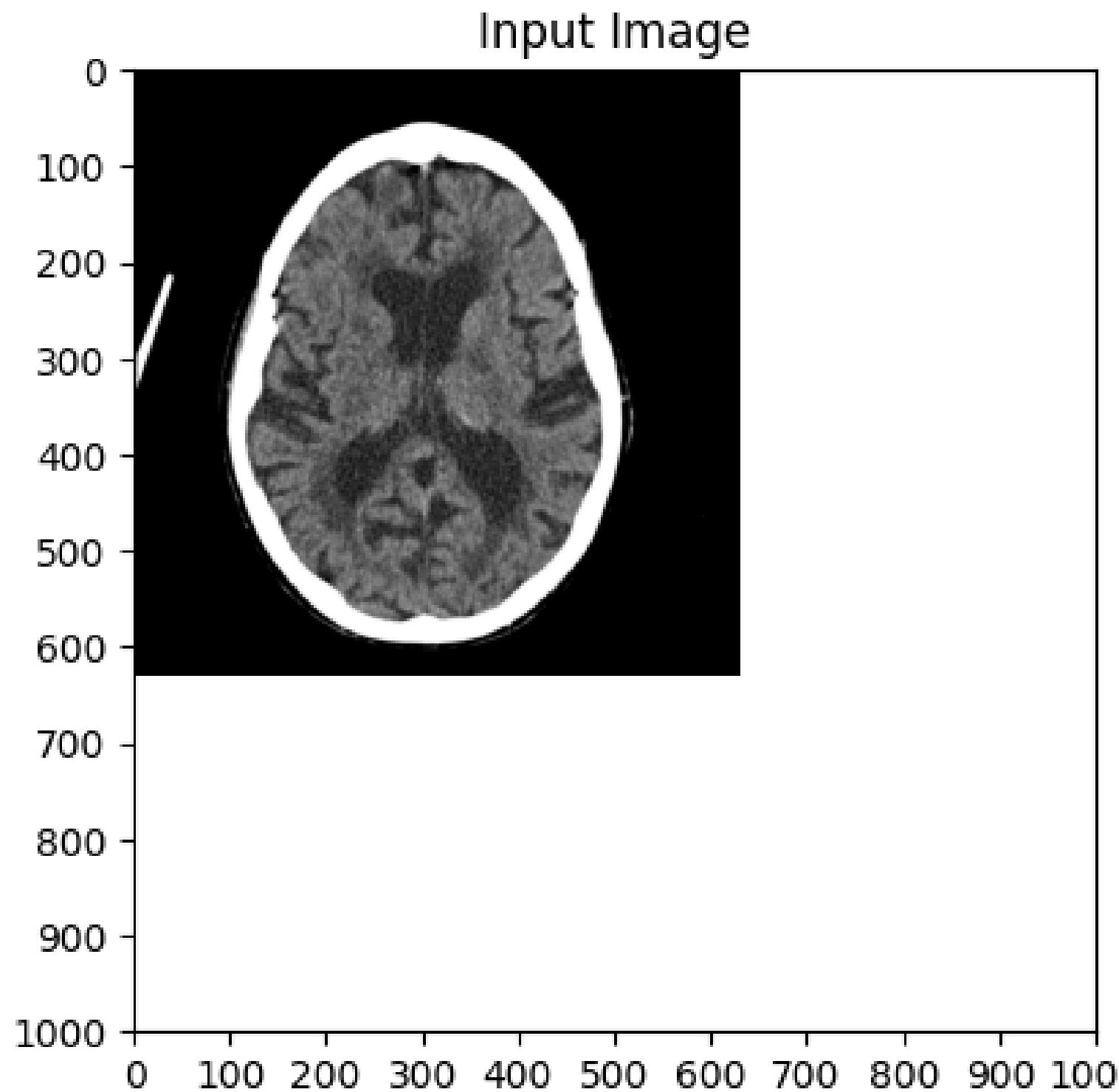
Resample Image Filter



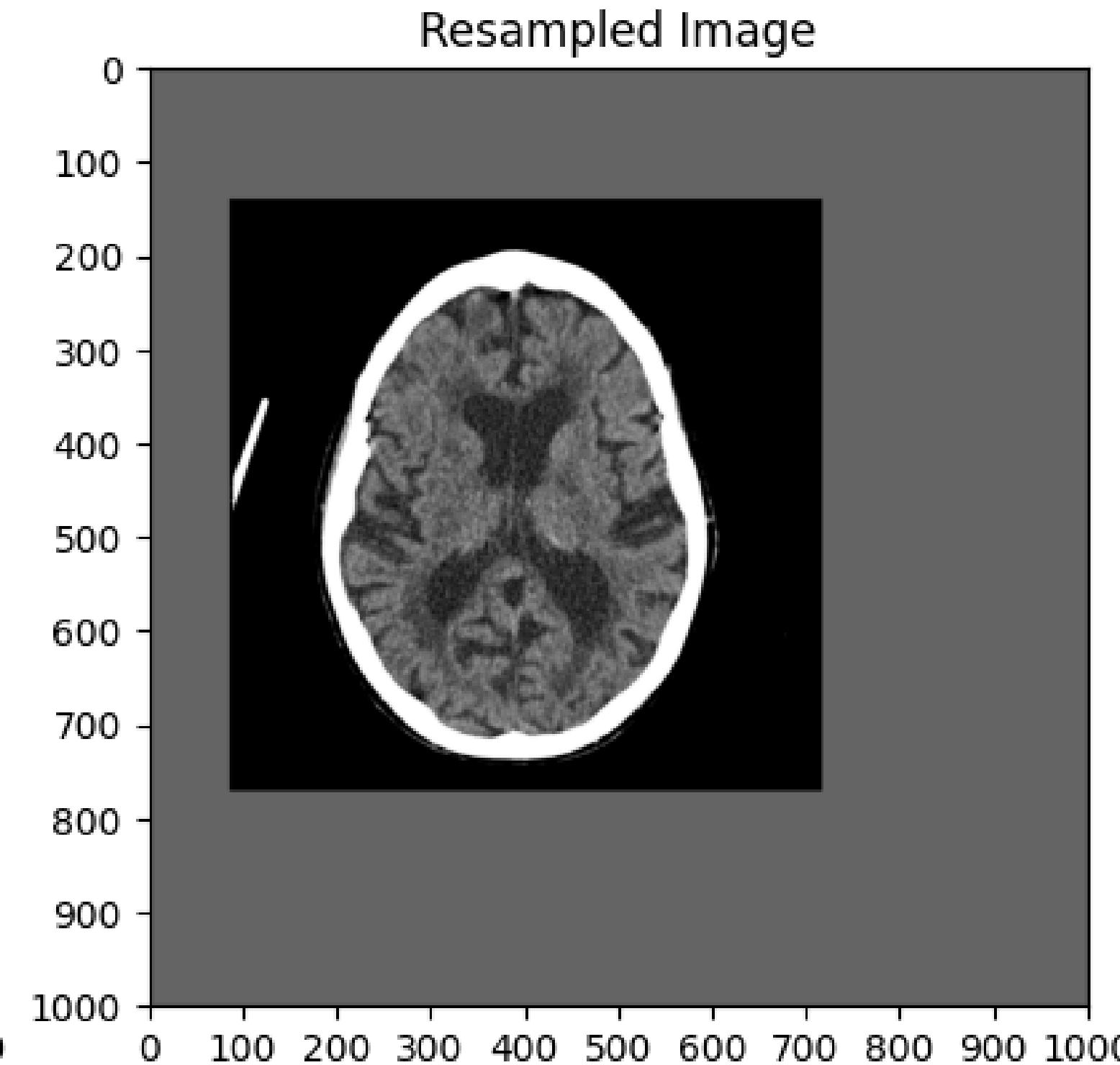
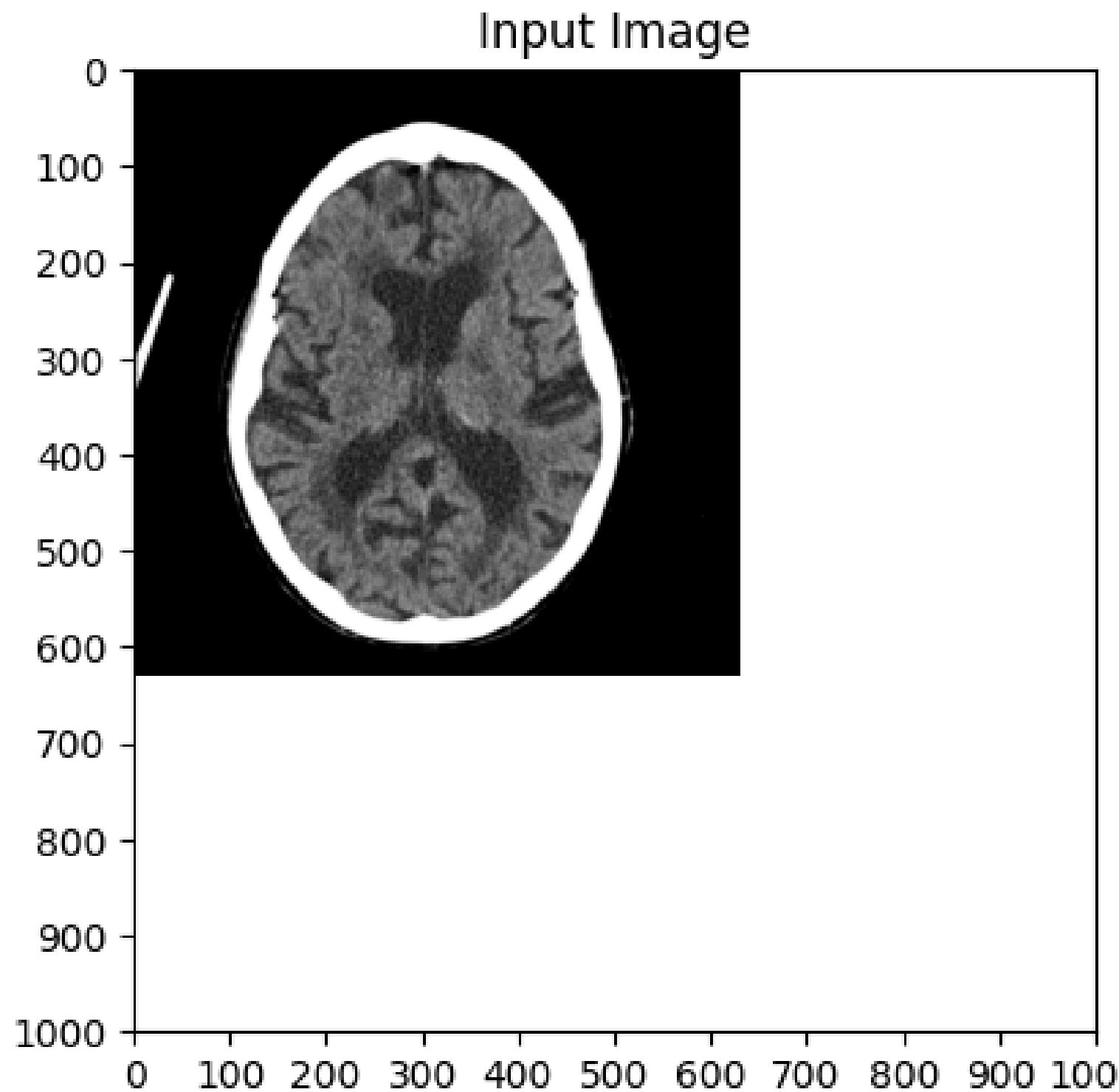
Resample Image Filter



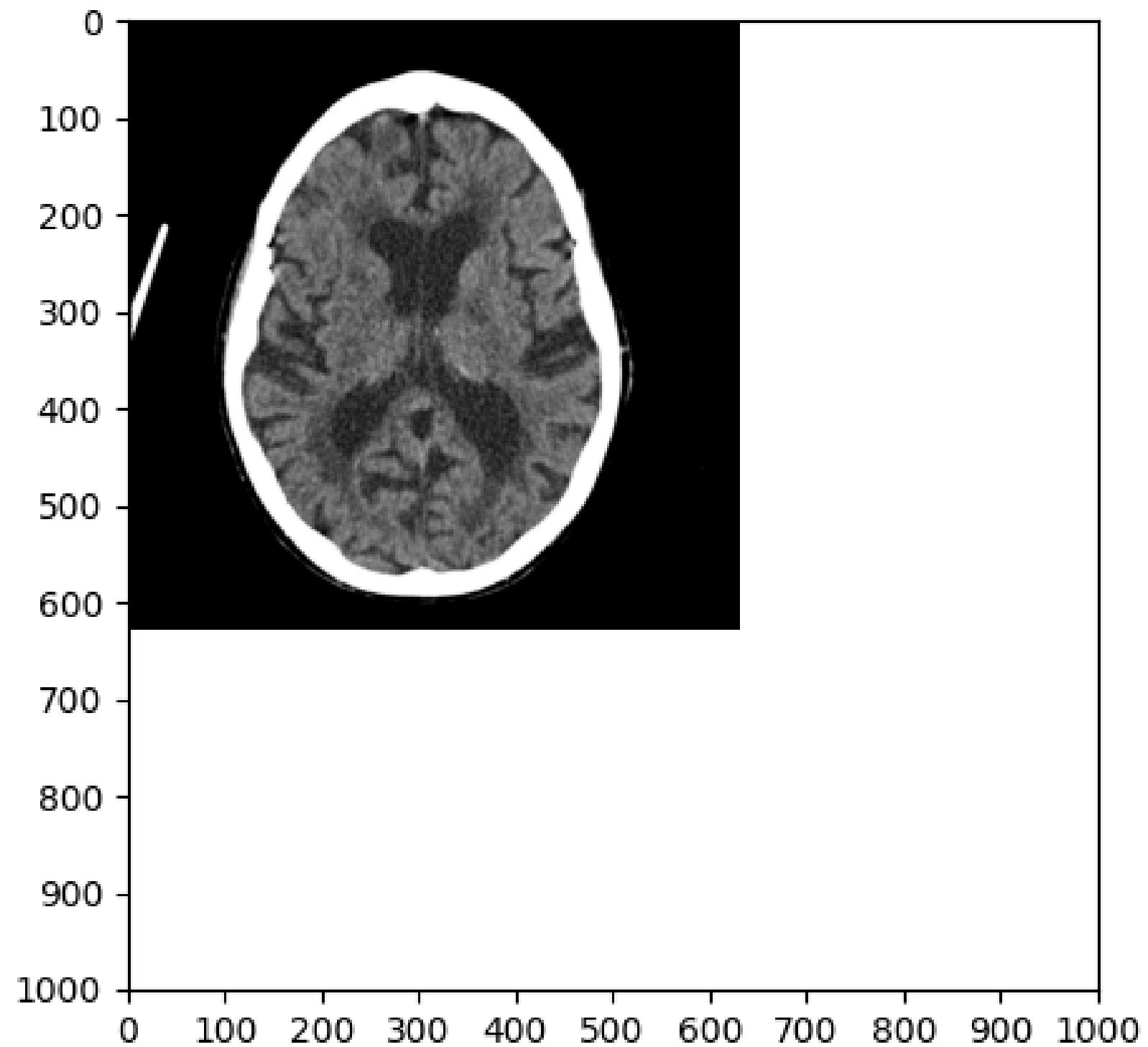
Resample Image Filter (with translation)



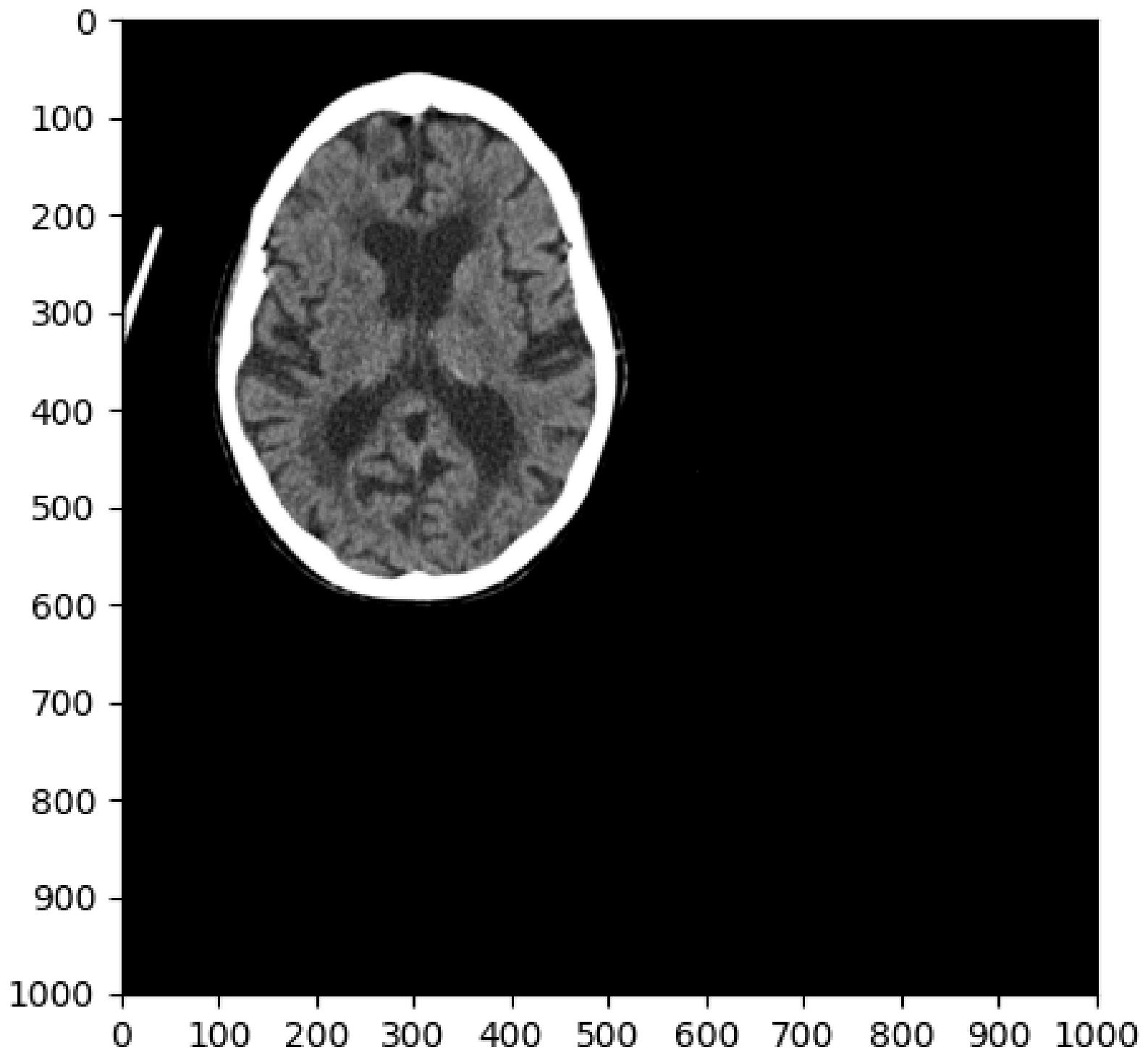
Resample Image Filter (Default pixel value)



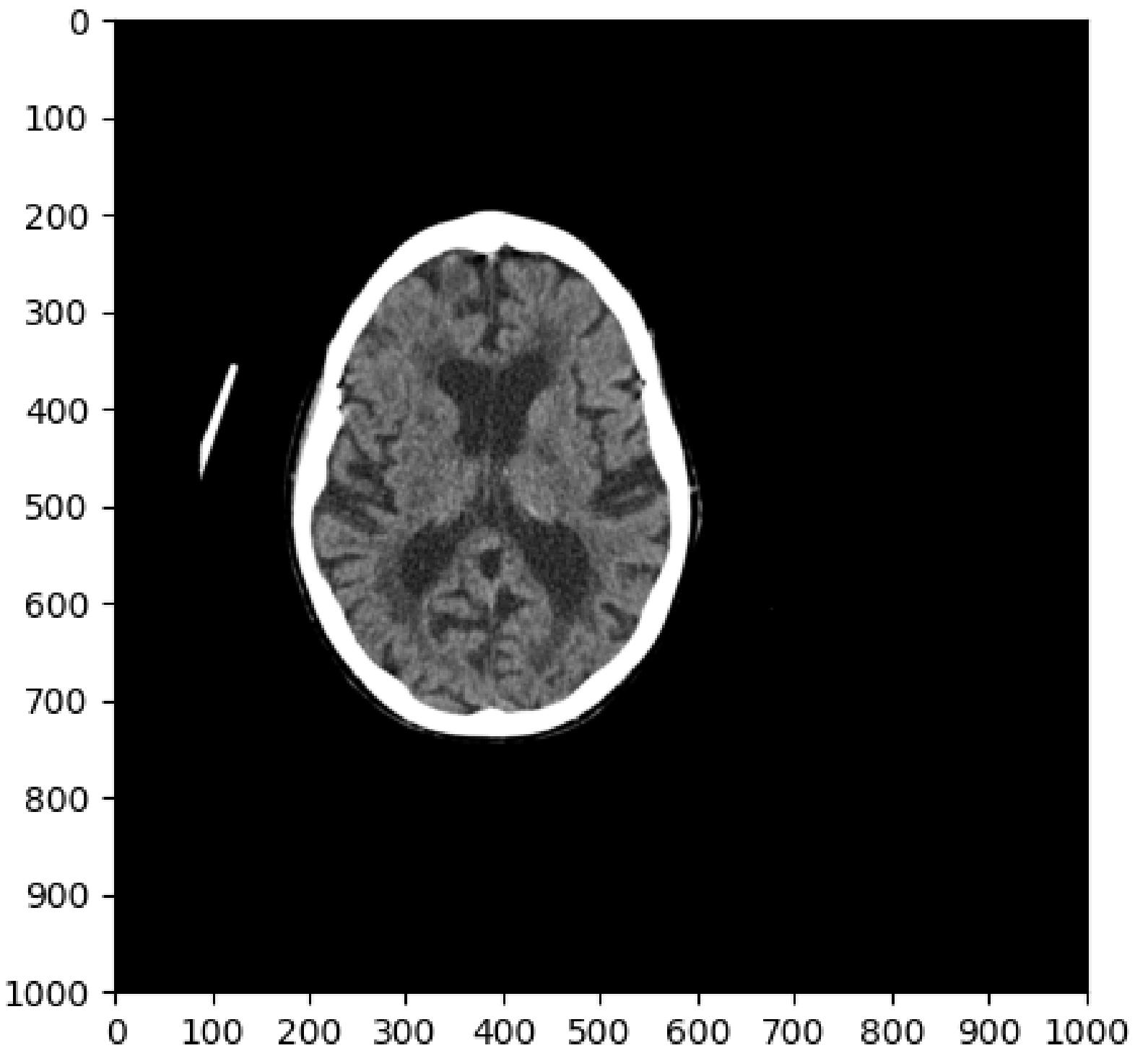
Resample Image Filter



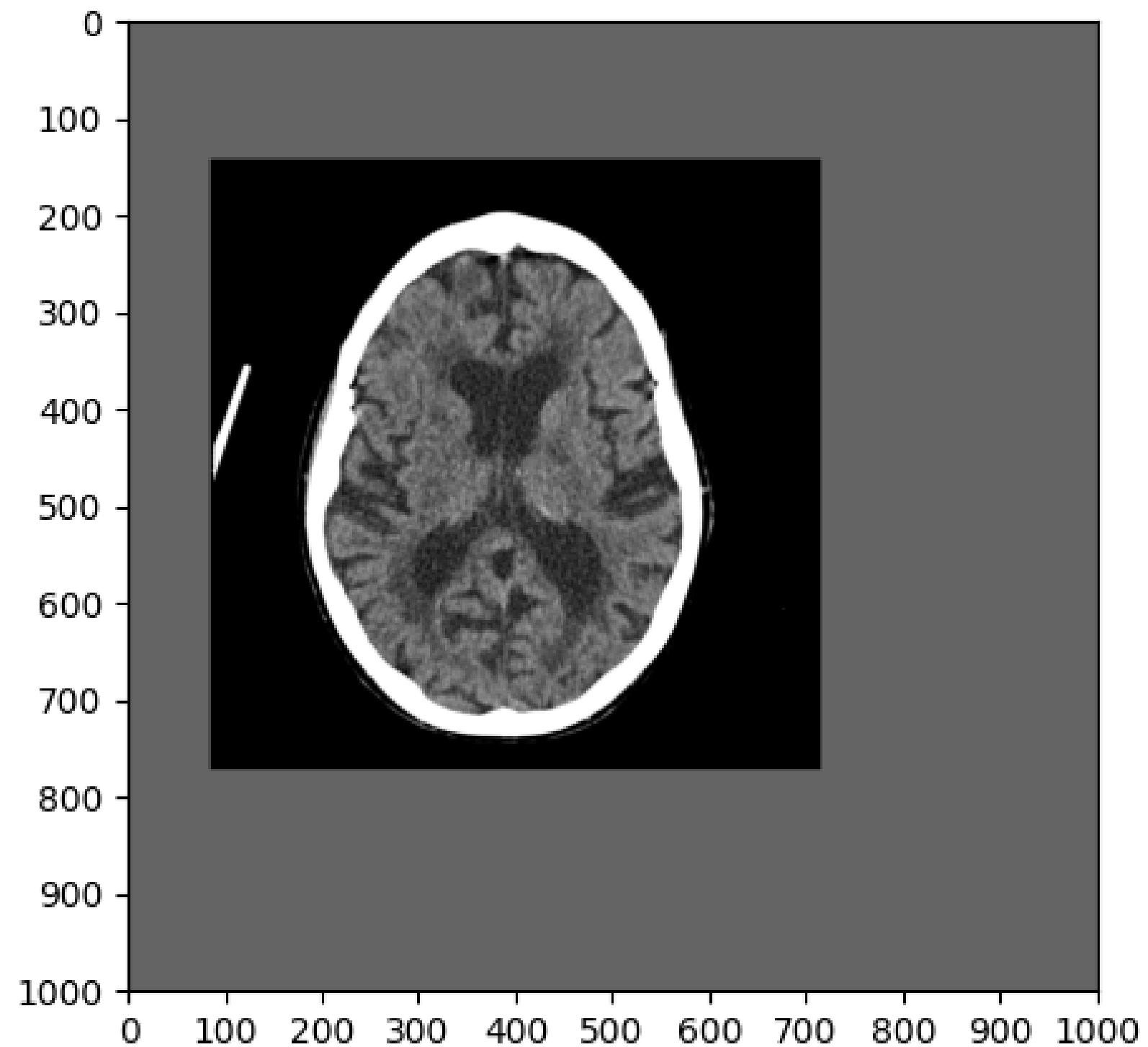
Resample Image Filter



Resample Image Filter

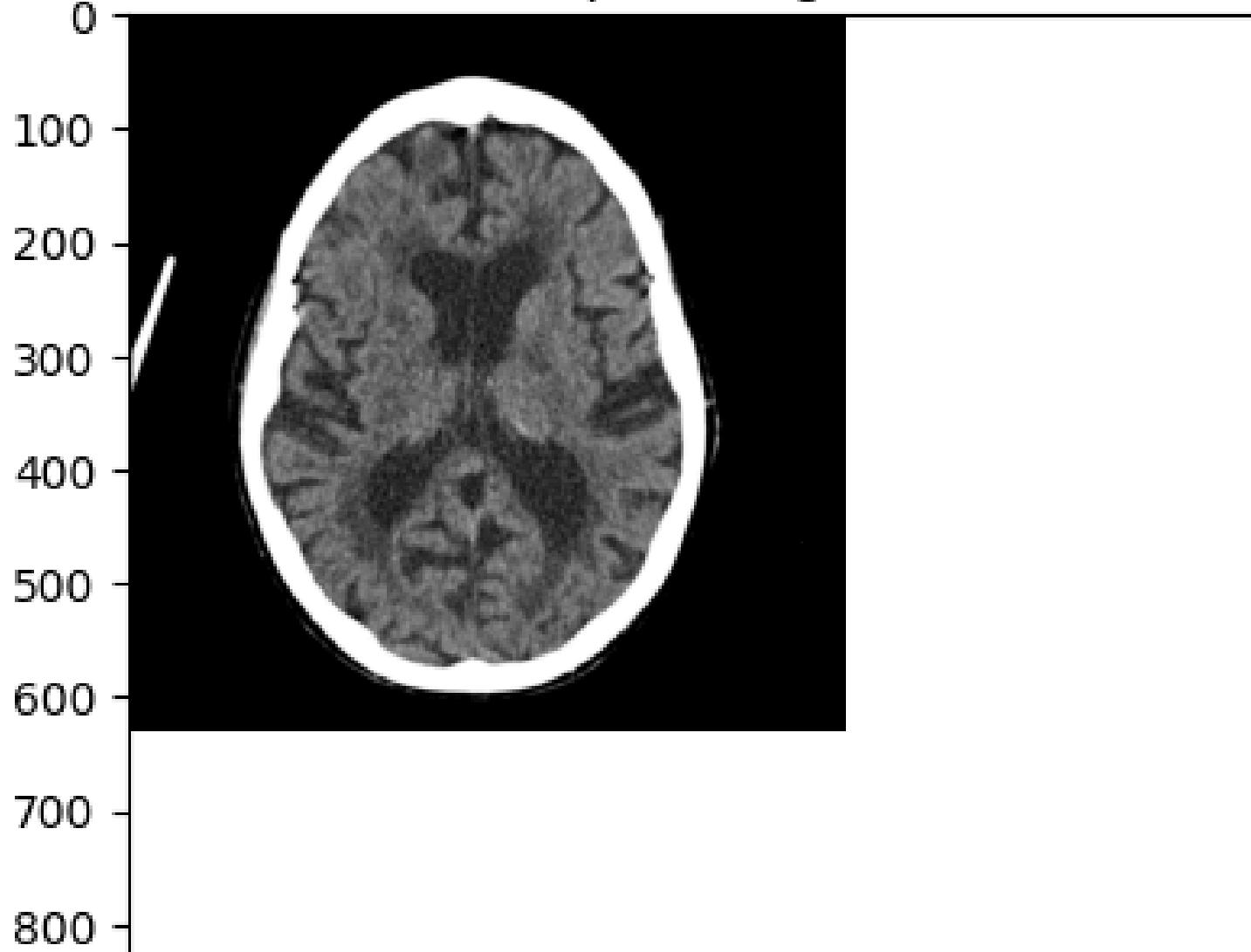


Resample Image Filter

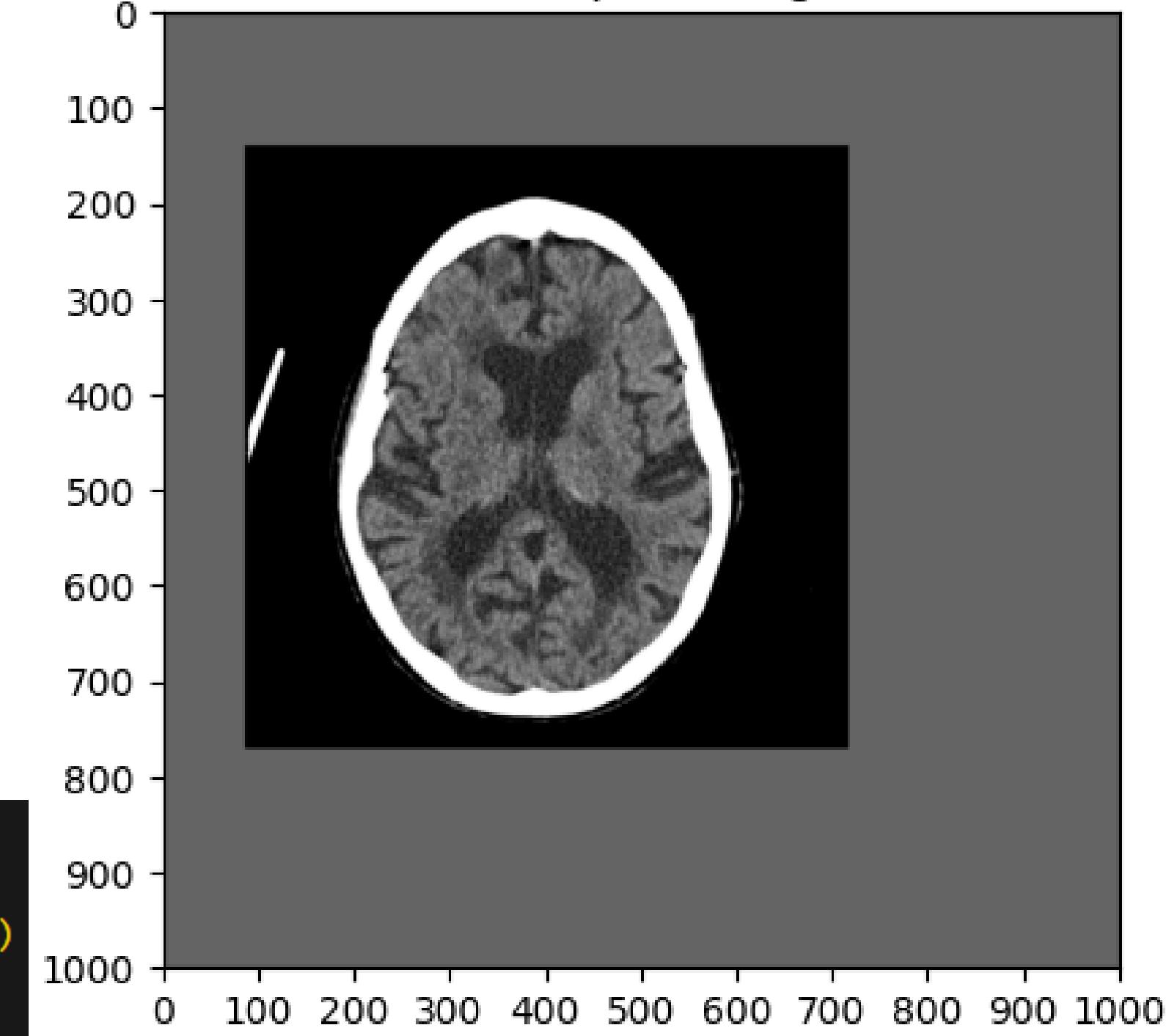


Importance of Spacing and Origin

Input Image



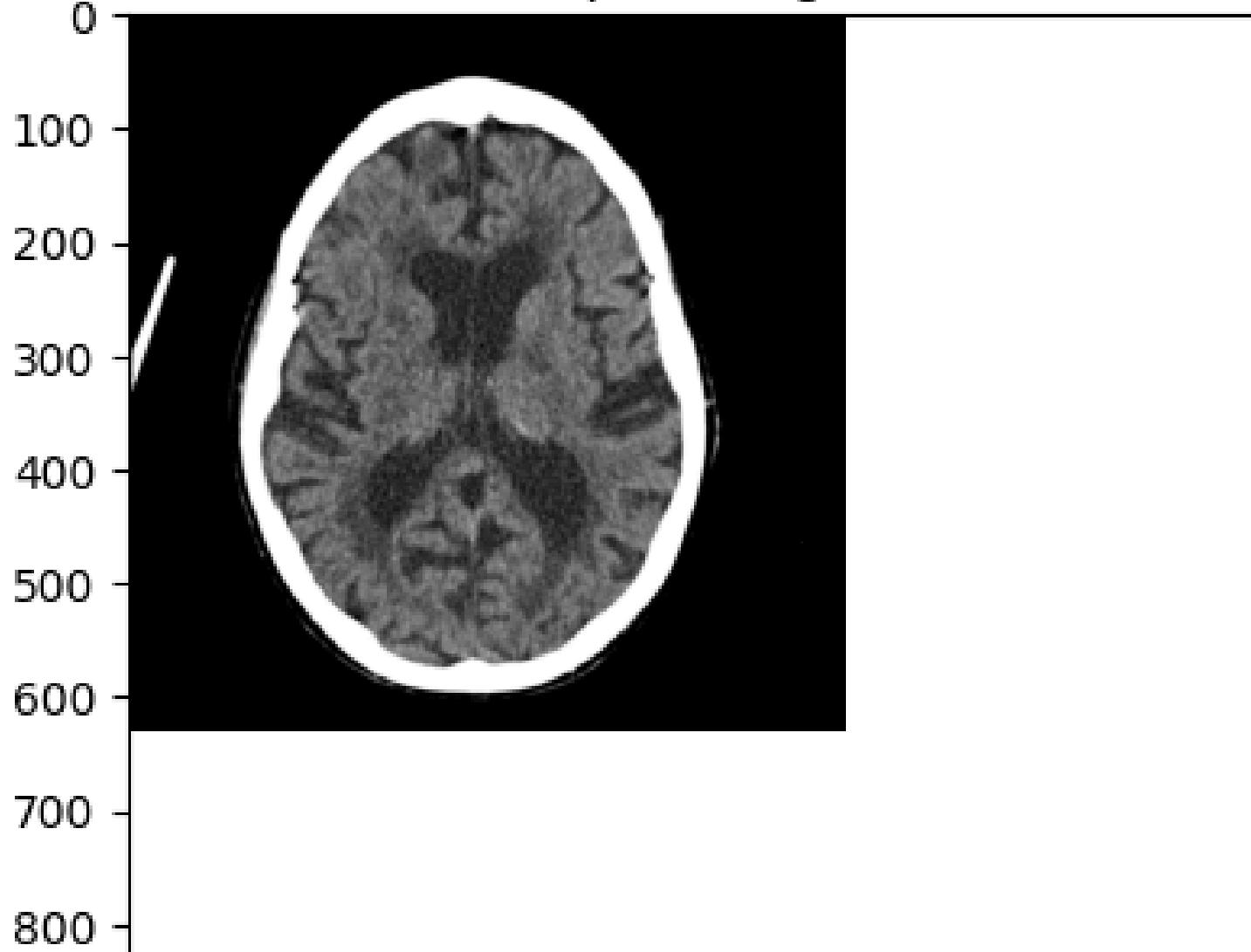
Resampled Image



```
# Perform translation
translation = [-30.0, -50.0]
translation_transform = itk.TranslationTransform[itk.D, 2].New()
translation_transform.Translate(translation)
resample_filter.SetTransform(translation_transform)
resample_filter.Update()
```

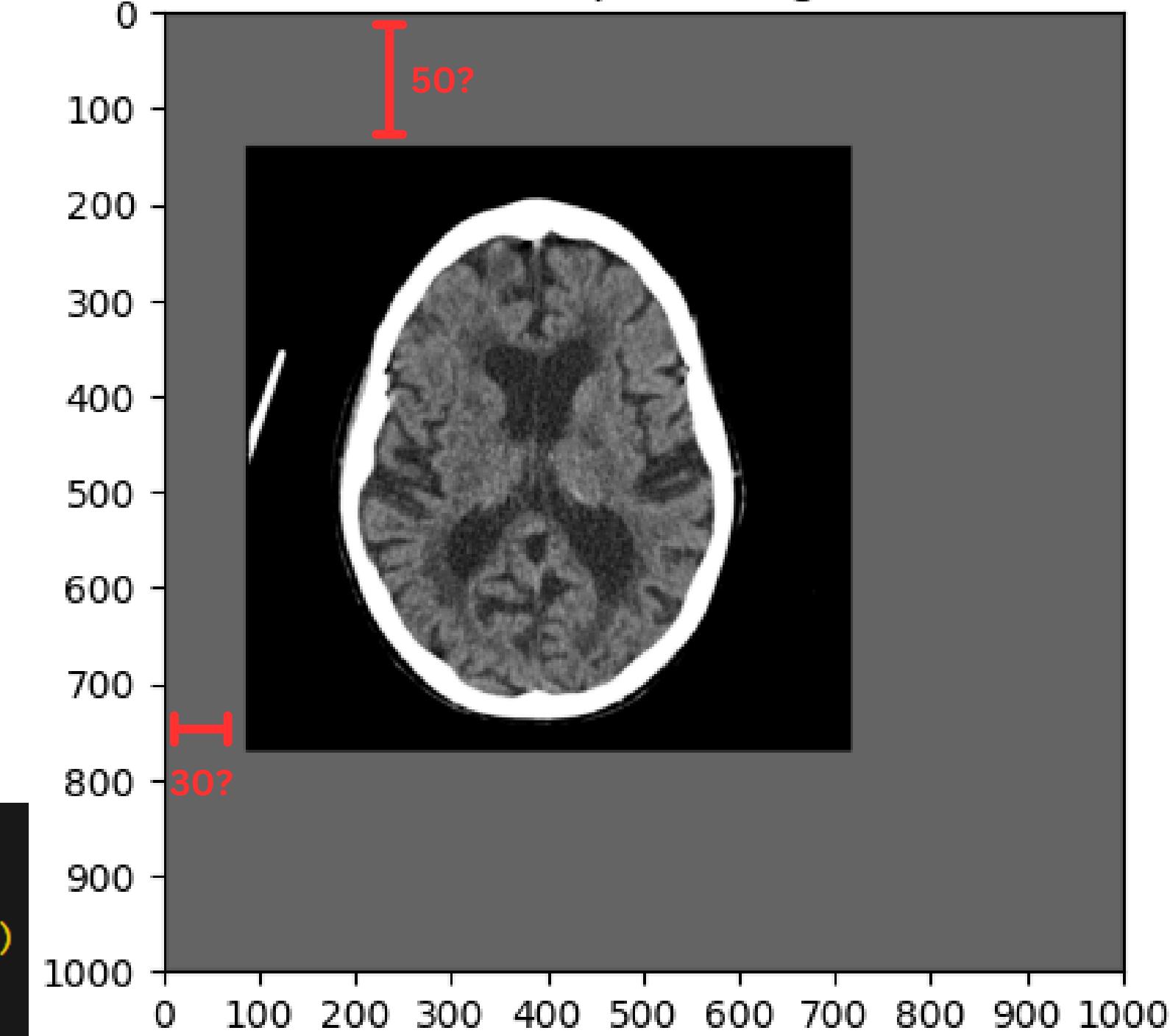
Importance of Spacing and Origin

Input Image



```
# Perform translation
translation = [-30.0, -50.0]
translation_transform = itk.TranslationTransform[itk.D, 2].New()
translation_transform.Translate(translation)
resample_filter.SetTransform(translation_transform)
resample_filter.Update()
```

Resampled Image



Importance of Spacing and Origin

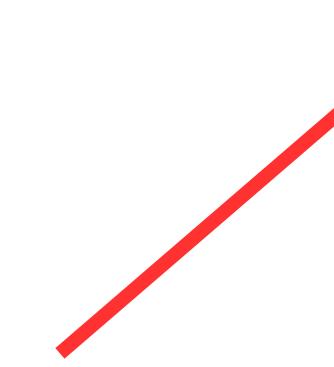
Importance of Spacing and Origin

$$P[i] = I[i] \times S[i] + O[i]$$

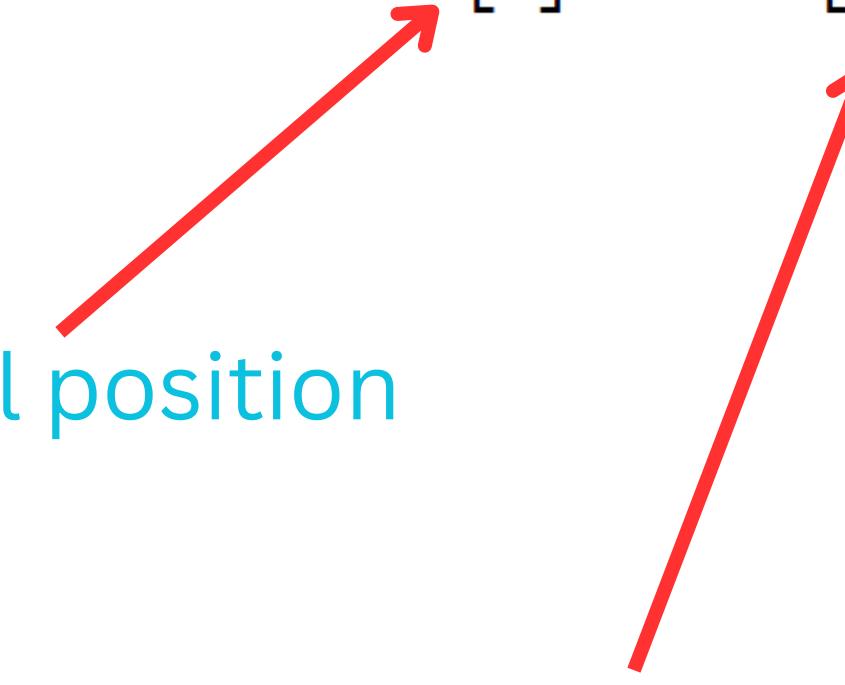
Importance of Spacing and Origin

$$P[i] = I[i] \times S[i] + O[i]$$

Spatial position



Importance of Spacing and Origin

$$P[i] = I[i] \times S[i] + O[i]$$


The diagram illustrates the components of the equation $P[i] = I[i] \times S[i] + O[i]$. Two red arrows point from labels to specific terms: one arrow points from the label "Spatial position" to the term $S[i]$, and another arrow points from the label "Pixel index" to the term $I[i]$.

Importance of Spacing and Origin

$$P[i] = I[i] \times S[i] + O[i]$$

Spatial position

Pixel index

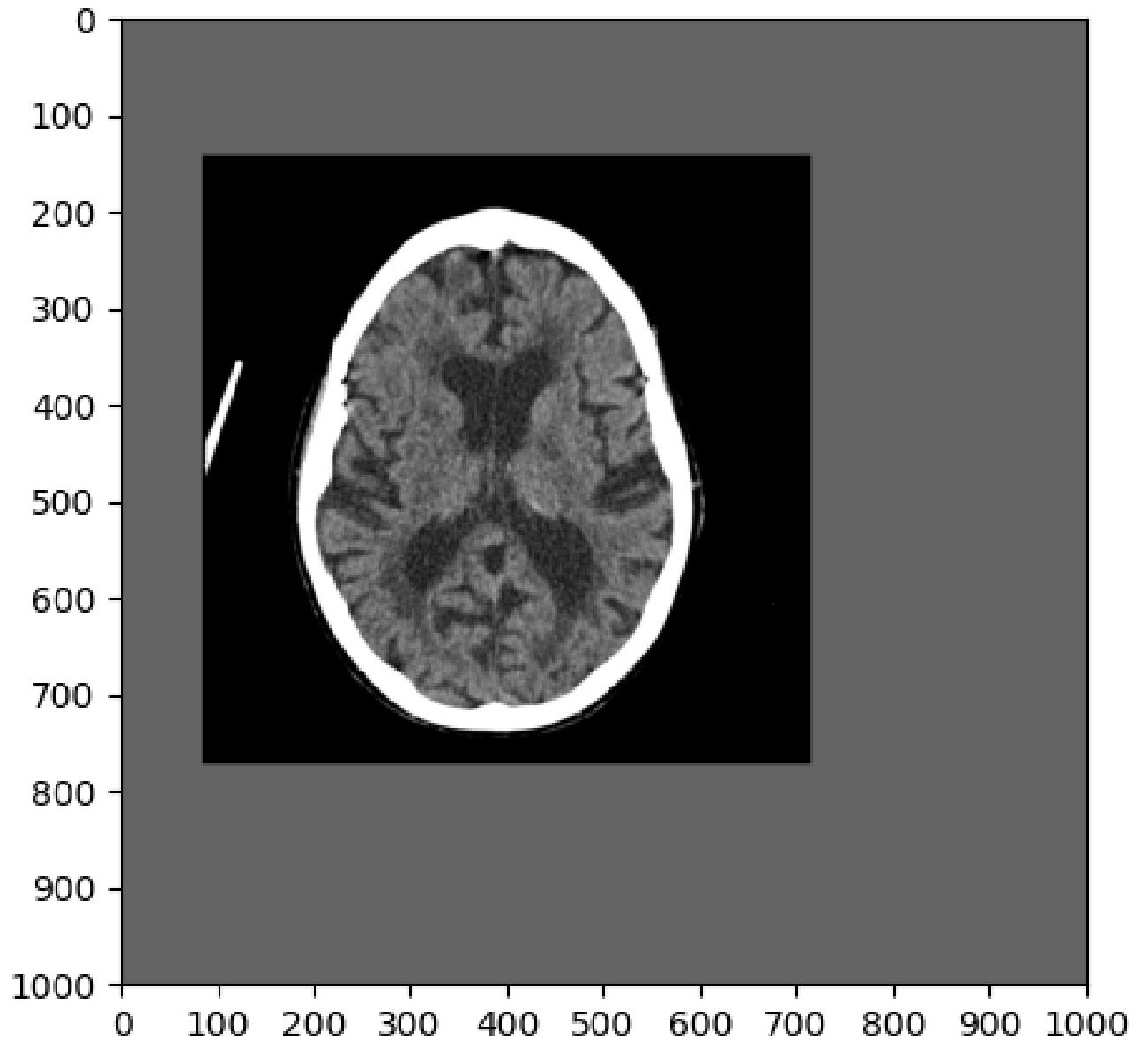
Pixel spacing

Importance of Spacing and Origin

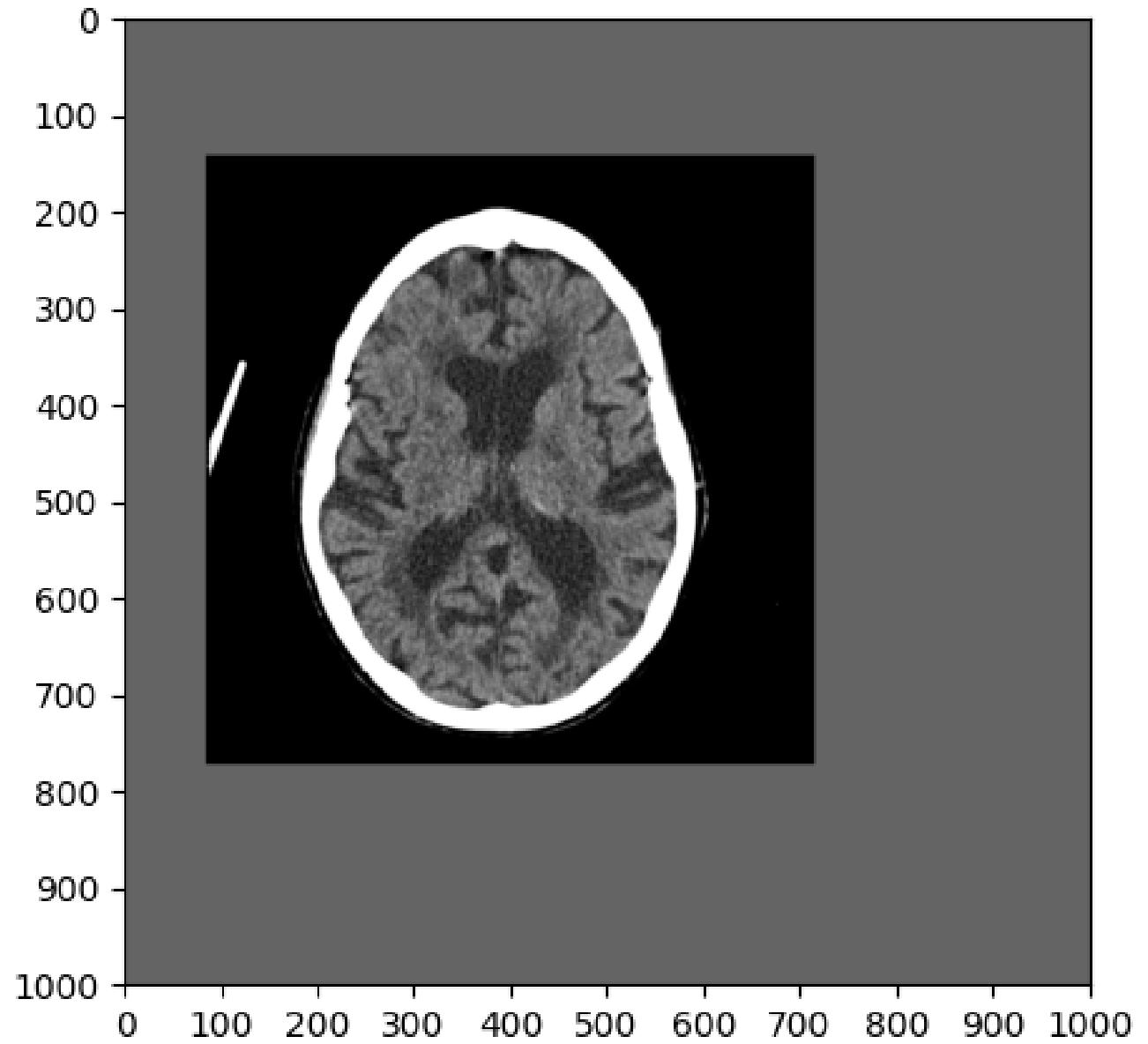
$$P[i] = I[i] \times S[i] + O[i]$$

The diagram illustrates the formula $P[i] = I[i] \times S[i] + O[i]$ by showing four red arrows pointing towards the terms in the equation. The first arrow points from the left towards the term $I[i]$. The second arrow points upwards towards the term $S[i]$. The third arrow points downwards towards the term $O[i]$. The fourth arrow points from the right towards the term $I[i]$. The labels for these components are: "Spatial position" (cyan text) next to the arrow pointing up; "Pixel index" (cyan text) next to the arrow pointing left; "Pixel spacing" (cyan text) next to the arrow pointing down; and "Origin" (cyan text) next to the arrow pointing right.

Importance of Spacing and Origin

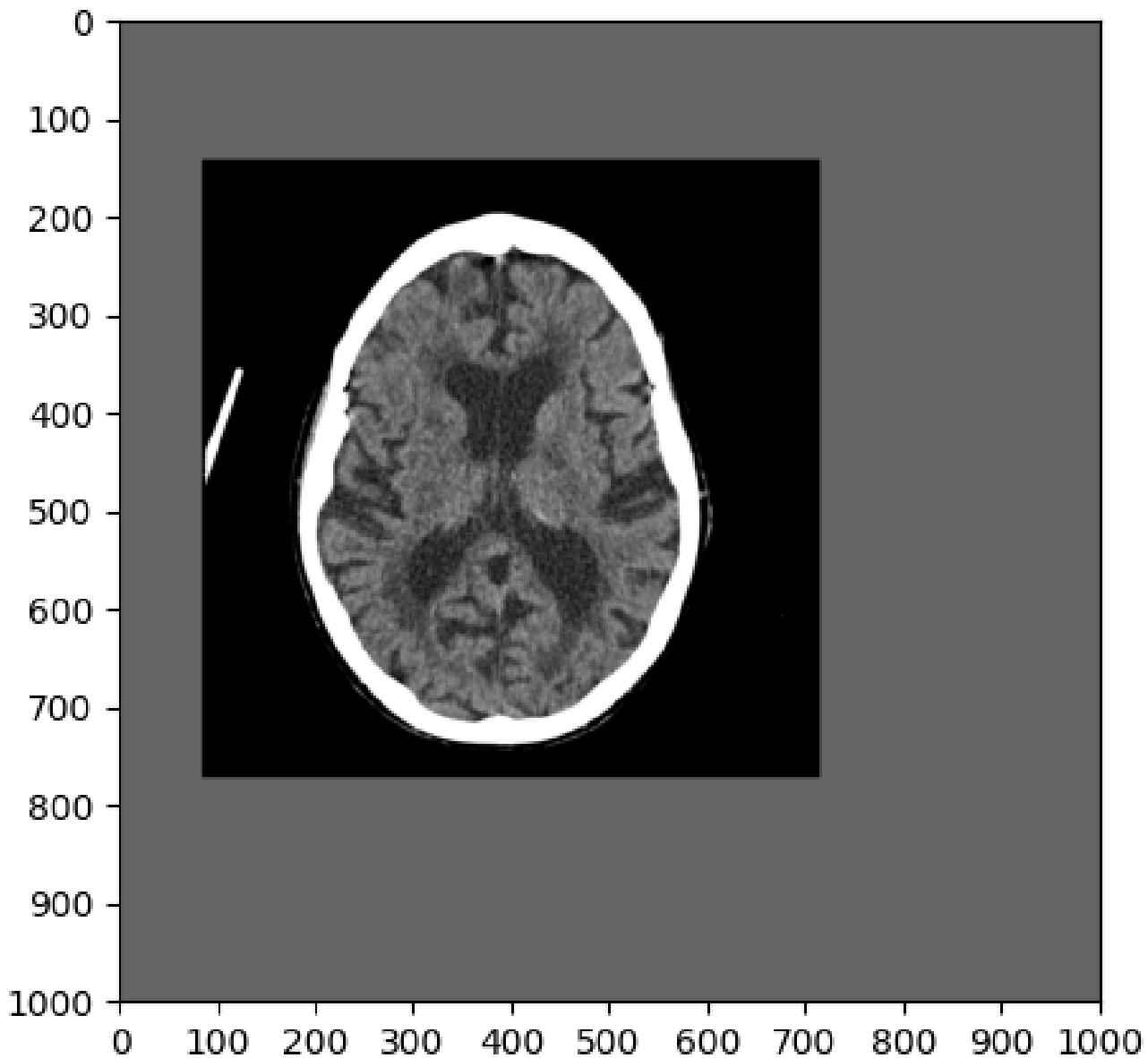


Importance of Spacing and Origin



```
Input image size: itkSize2 ([630, 630])
Input image spacing: itkVectorD2 ([0.352778, 0.352778]) ← S[i]
Input image origin: itkPointD2 ([0, 0]) ← O[i]
Input image direction: itkMatrixD22 ([[1.0, 0.0], [0.0, 1.0]])
```

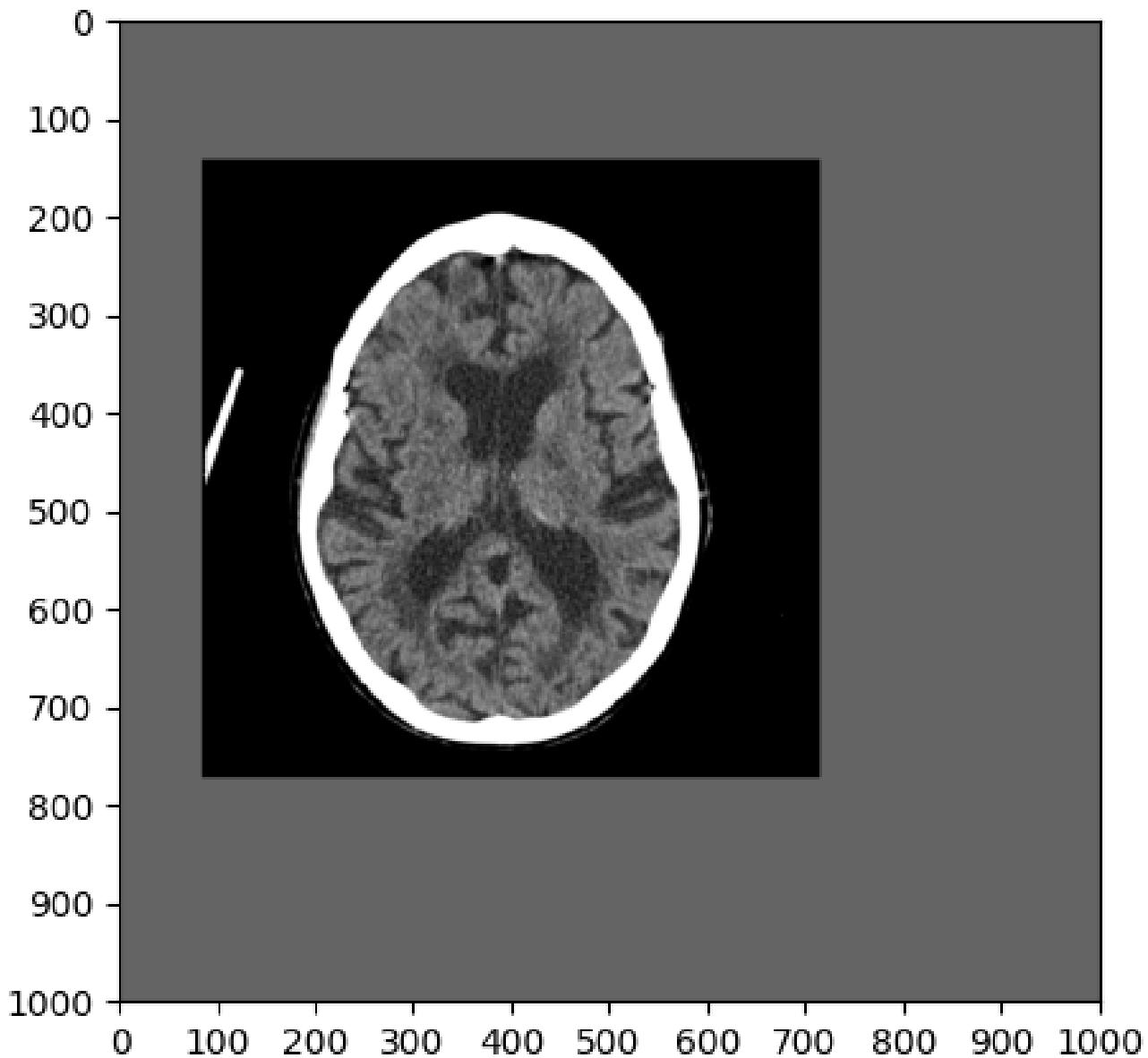
Importance of Spacing and Origin



```
# Perform translation
translation = [-30.0, -50.0] ← P[i]
translation_transform = itk::TranslationTransform[itk.D, 2]::New()
translation_transform::Translate(translation)
resample_filter.SetTransform(translation_transform)
resample_filter.Update()
```

```
Input image size: itkSize2 ([630, 630])
Input image spacing: itkVectorD2 ([0.352778, 0.352778]) ← S[i]
Input image origin: itkPointD2 ([0, 0]) ← O[i]
Input image direction: itkMatrixD22 ([[1.0, 0.0], [0.0, 1.0]])
```

Importance of Spacing and Origin

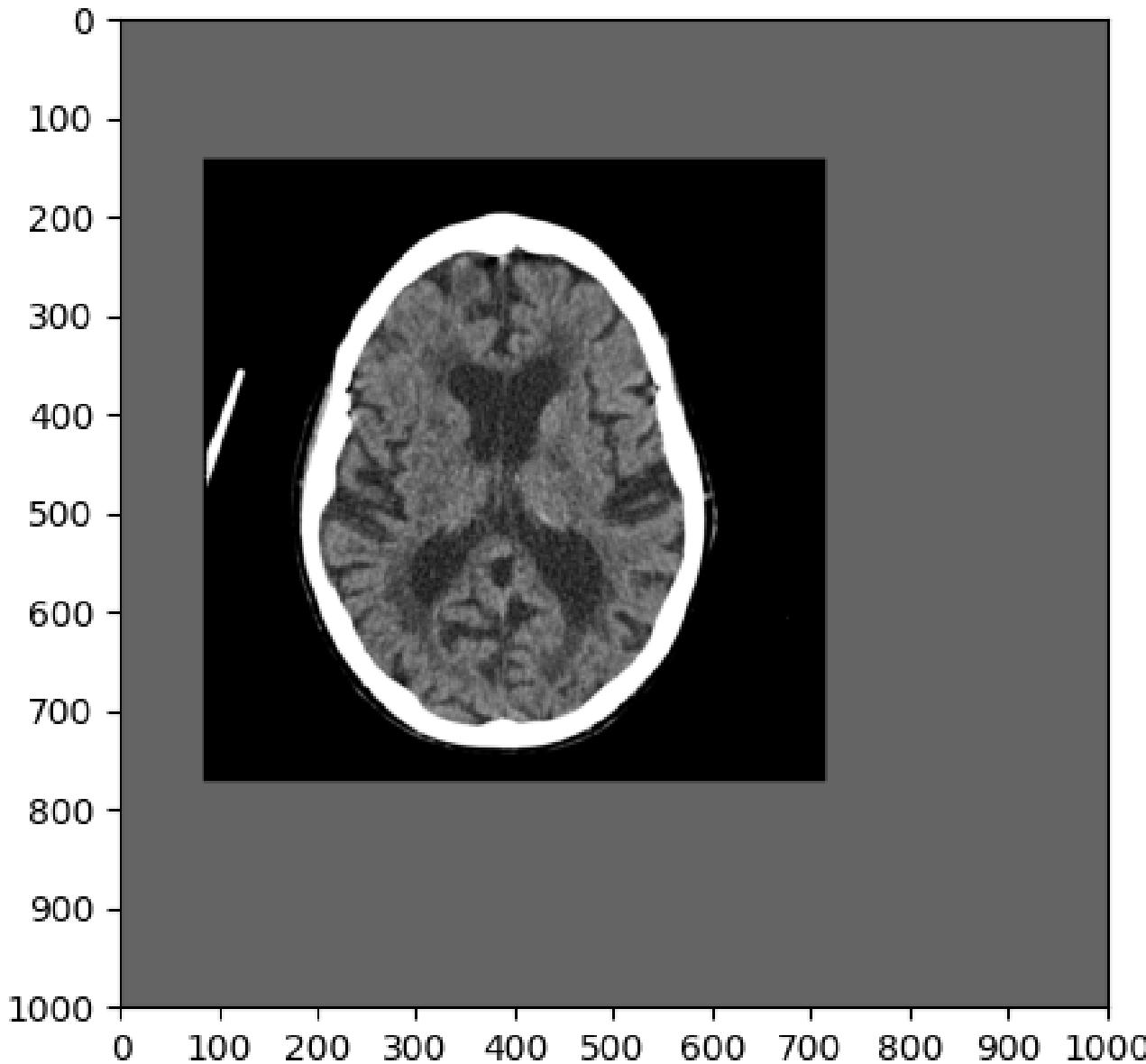


```
# Perform translation
translation = [-30.0, -50.0] ← P[i]
translation_transform = itk::TranslationTransform[itk.D, 2]::New()
translation_transform::Translate(translation)
resample_filter.SetTransform(translation_transform)
resample_filter.Update()
```

I[i] = ?

```
Input image size: itkSize2 ([630, 630])
Input image spacing: itkVectorD2 ([0.352778, 0.352778]) ← S[i]
Input image origin: itkPointD2 ([0, 0]) ← O[i]
Input image direction: itkMatrixD22 ([[1.0, 0.0], [0.0, 1.0]])
```

Importance of Spacing and Origin



```
# Perform translation
translation = [-30.0, -50.0] ← P[i]
translation_transform = itk::TranslationTransform[itk::D, 2]::New()
translation_transform::Translate(translation)
resample_filter.SetTransform(translation_transform)
resample_filter.Update()
```

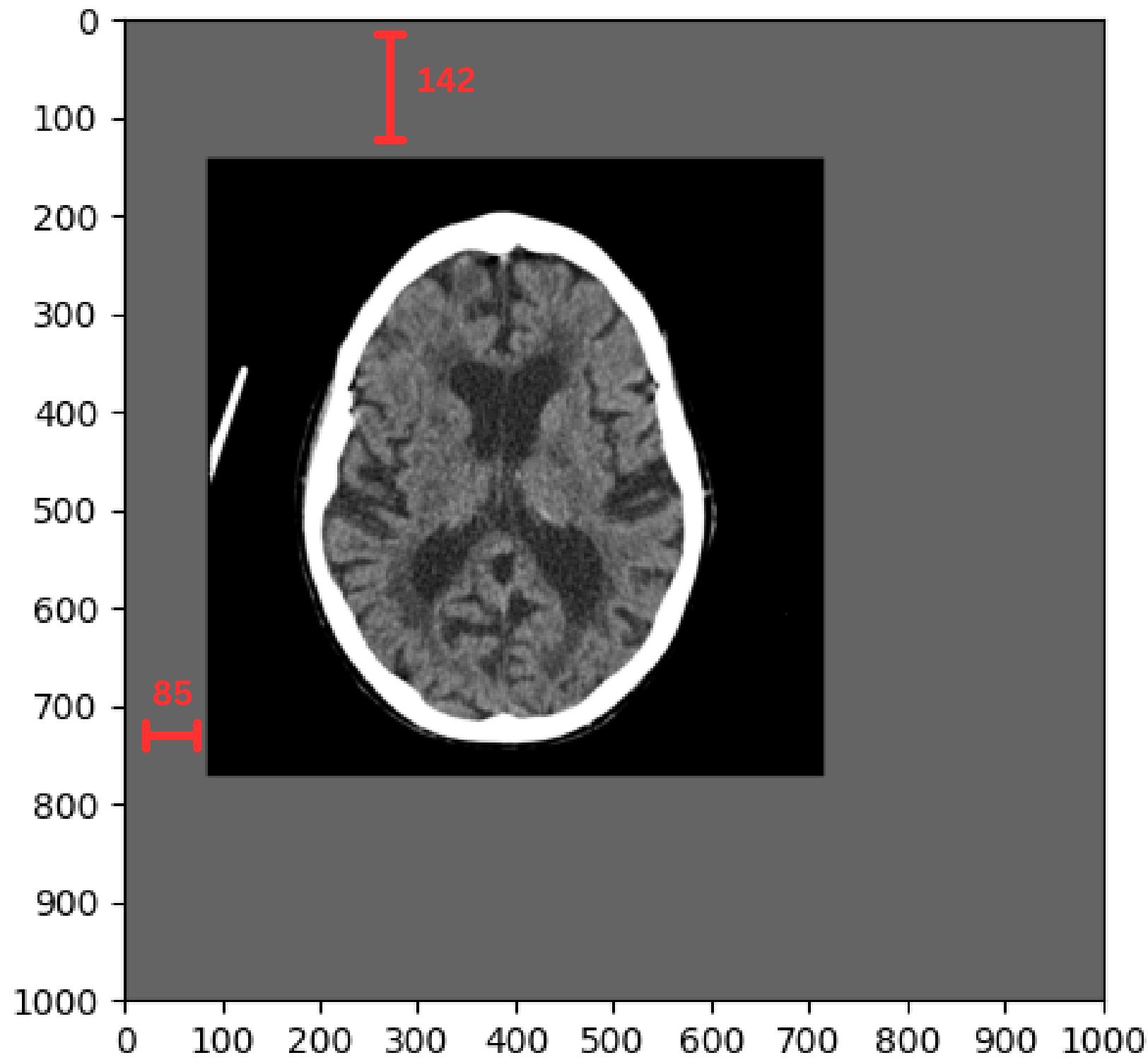
$$I[i] = ?$$

$$[30 \ 50] = I[i] [0.3528 \ 0.3528] + [0 \ 0]$$

$$I[i] = [85 \ 142]$$

```
Input image size: itkSize2 ([630, 630])
Input image spacing: itkVectorD2 ([0.352778, 0.352778]) ← S[i]
Input image origin: itkPointD2 ([0, 0]) ← O[i]
Input image direction: itkMatrixD22 ([[1.0, 0.0], [0.0, 1.0]])
```

Importance of Spacing and Origin



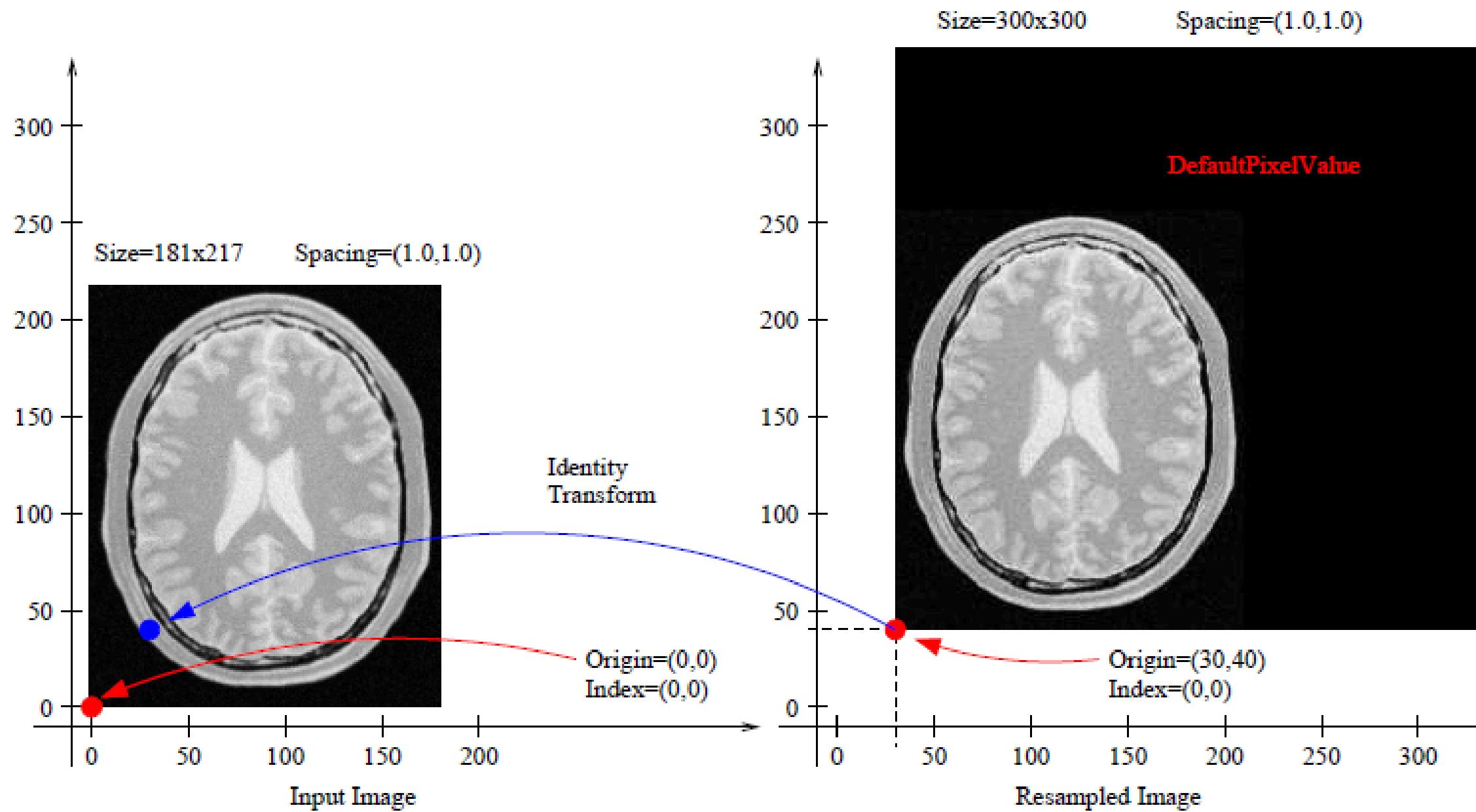
Importance of Spacing and Origin

Specify a non-zero origin

```
resample_filter.SetOutputOrigin([-30,-40])  
  
# Execute resampling with identity transform  
resample_filter.Update()  
output_image = resample_filter.GetOutput()
```

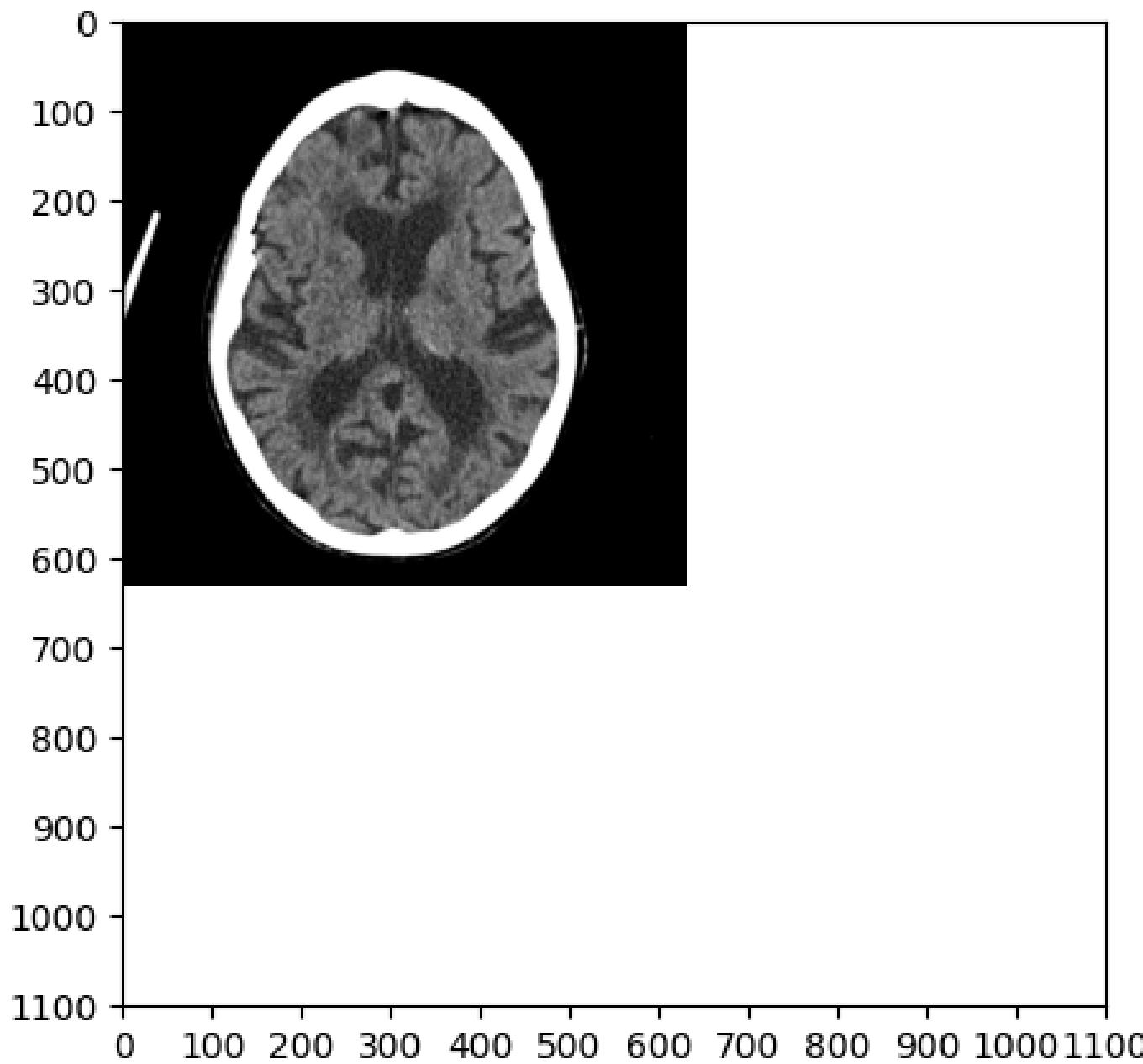
Importance of Spacing and Origin

Specify a non-zero origin



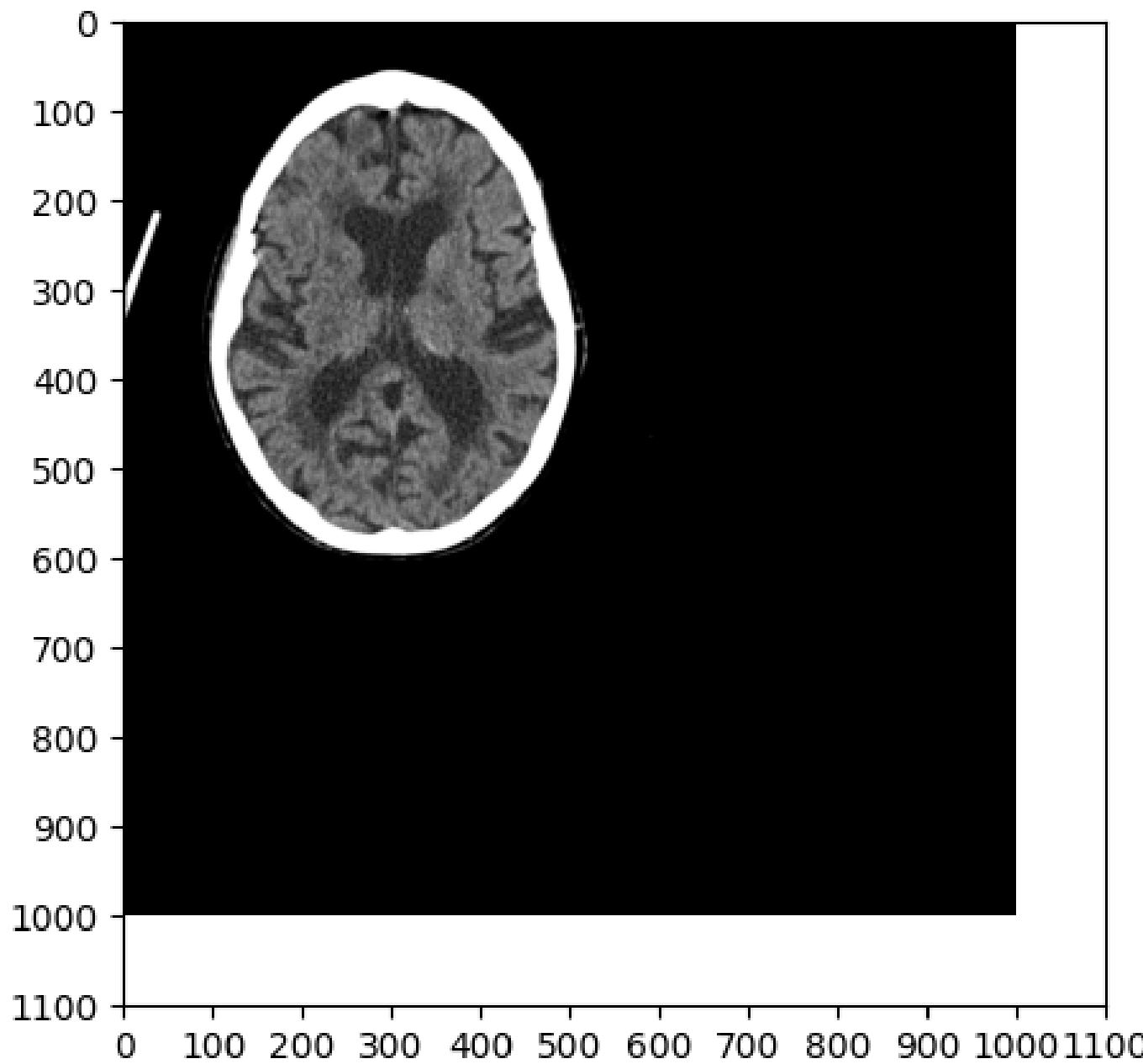
Importance of Spacing and Origin

Specify a non-zero origin



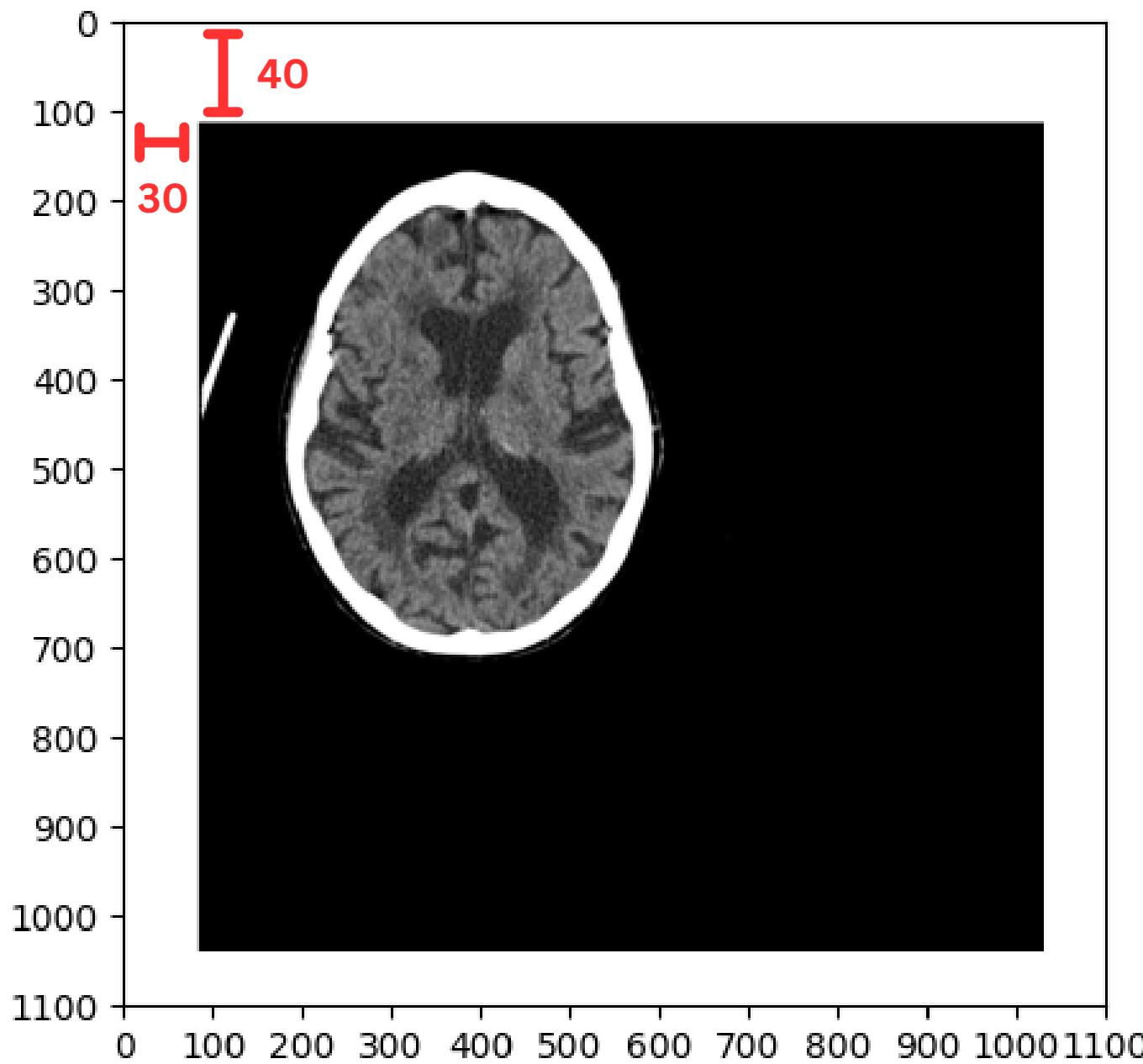
Importance of Spacing and Origin

Specify a non-zero origin



Importance of Spacing and Origin

Specify a non-zero origin



Rotating an Image

Rotating an Image

Rotations are performed around the **origin of physical coordinates**
not the image origin nor the image center.

Rotating an Image

Rotations are performed around the **origin of physical coordinates**
not the image origin nor the image center.

Hence, the process requires three steps

Rotating an Image

Rotations are performed around the **origin of physical coordinates**
not the image origin nor the image center.

Hence, the process requires three steps

1. The rotating point of the image should be moved to origin of the coordinate system.

Rotating an Image

Rotations are performed around the **origin of physical coordinates**
not the image origin nor the image center.

Hence, the process requires three steps

1. The rotating point of the image should be moved to origin of the coordinate system.
2. Rotate the Image (using Rotate2D).

Rotating an Image

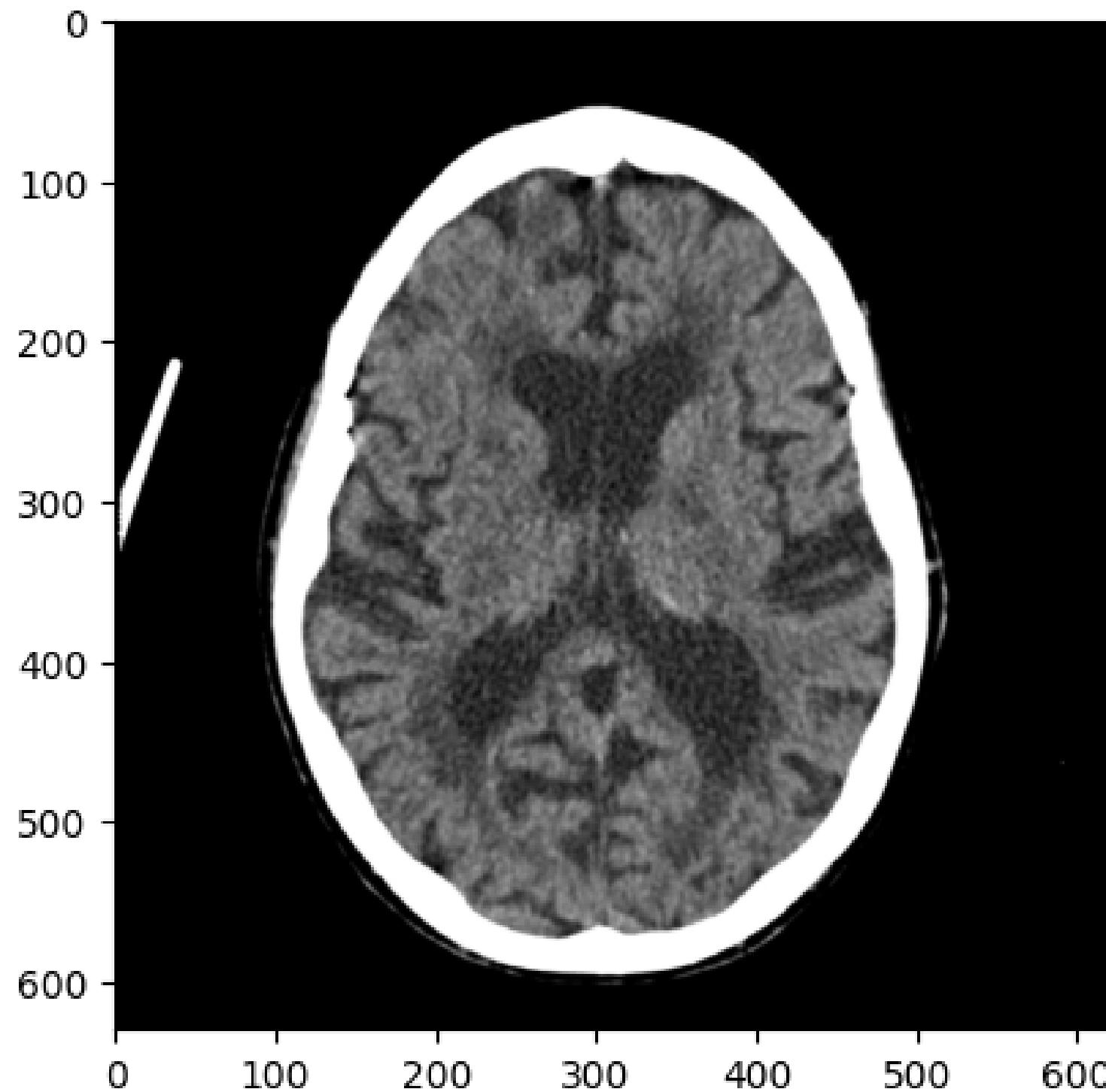
Rotations are performed around the **origin of physical coordinates**
not the image origin nor the image center.

Hence, the process requires three steps

1. The rotating point of the image should be moved to origin of the coordinate system.
2. Rotate the Image (using Rotate2D).
3. Translate image origin back in to previous location.

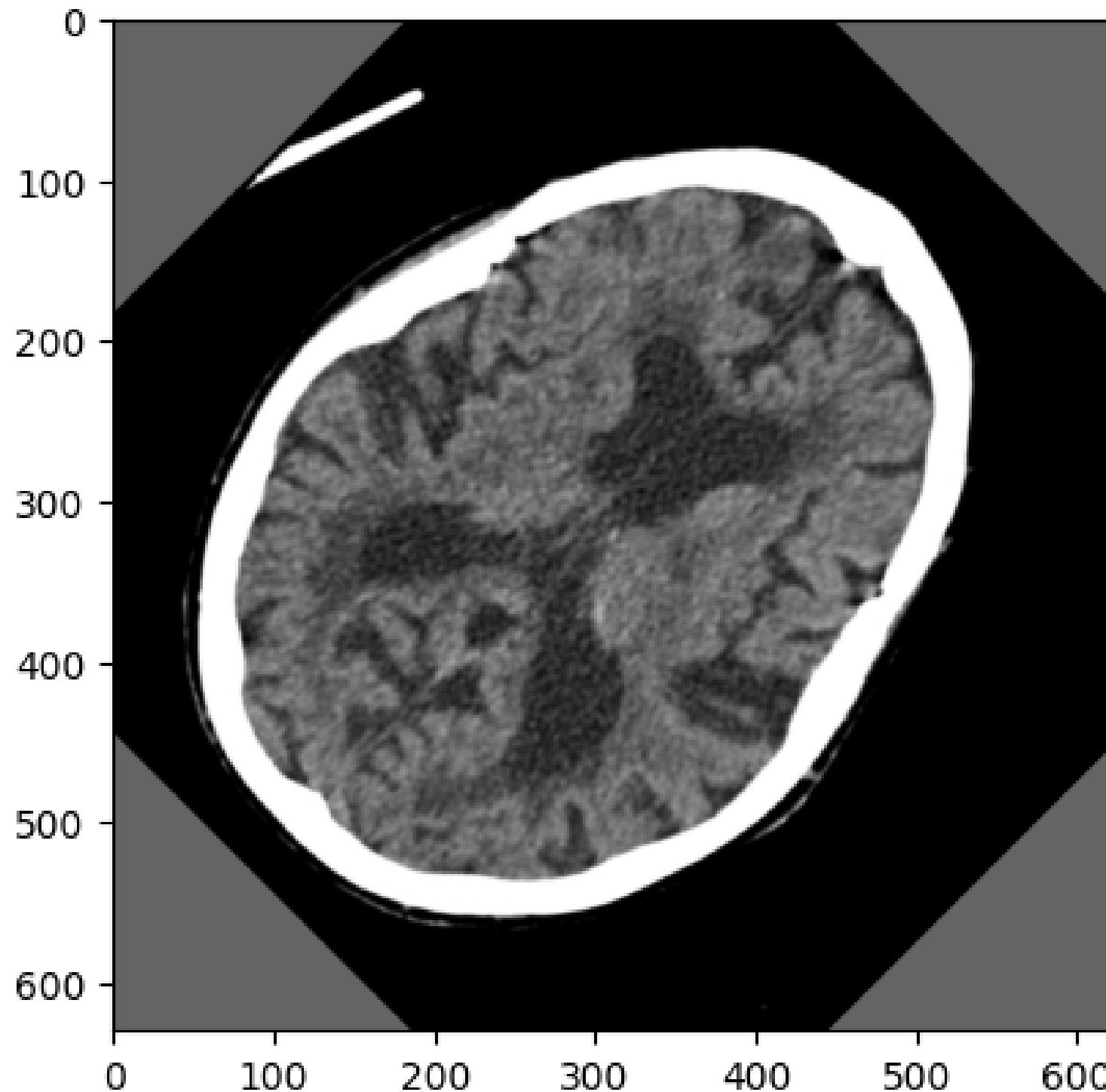
Rotating an Image

Original Image

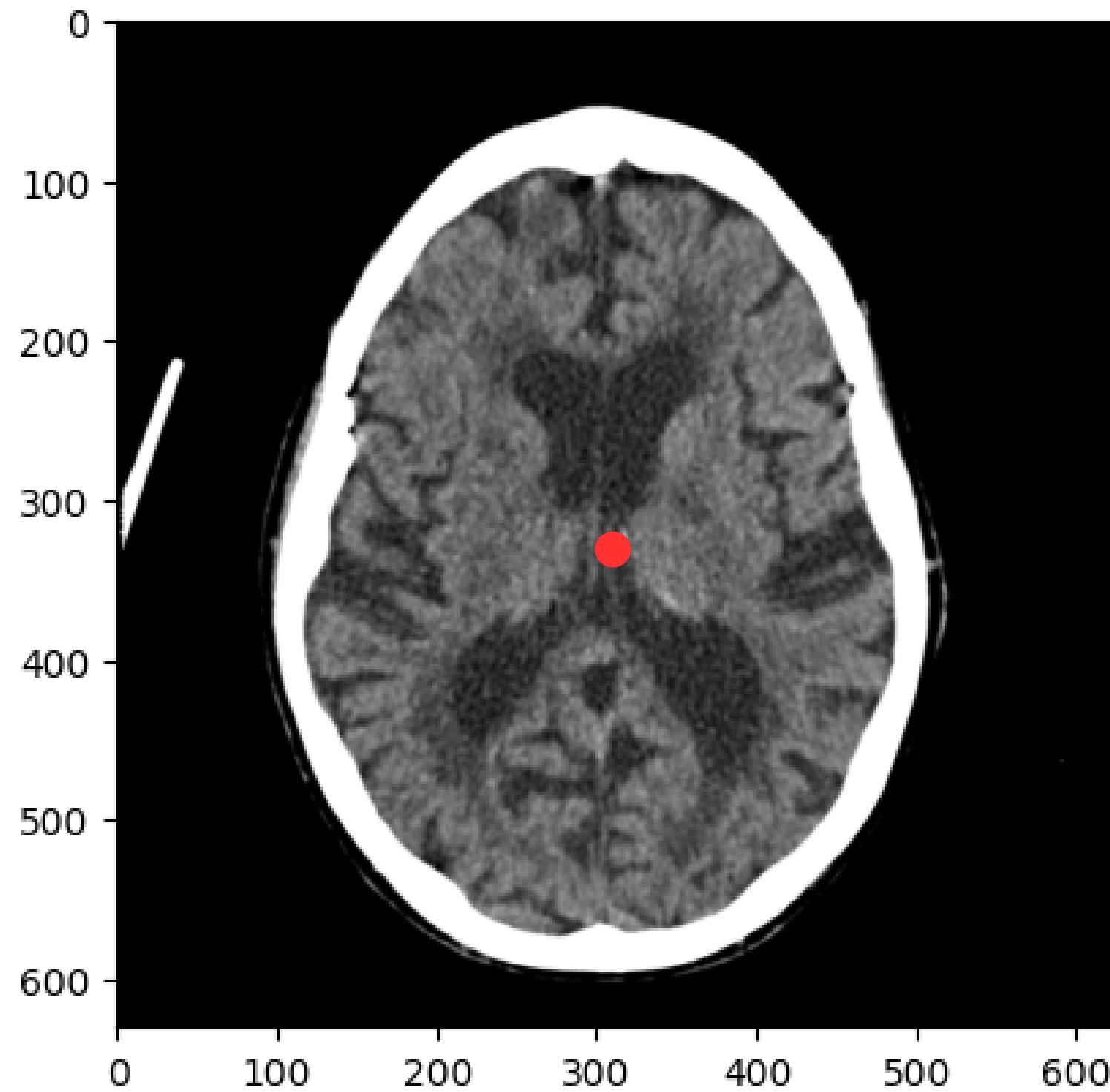


Rotating an Image

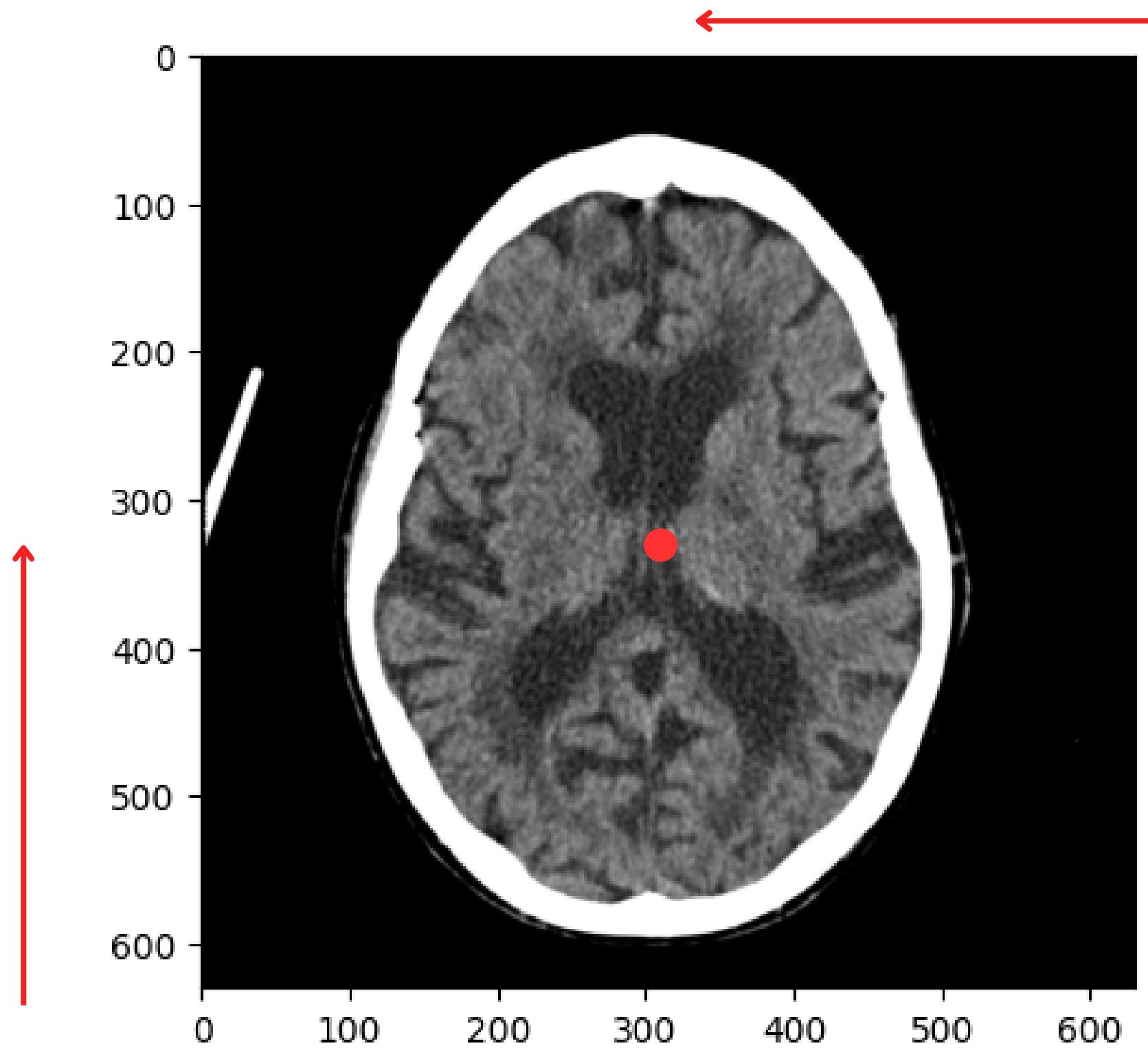
Rotated Image



Rotating an Image

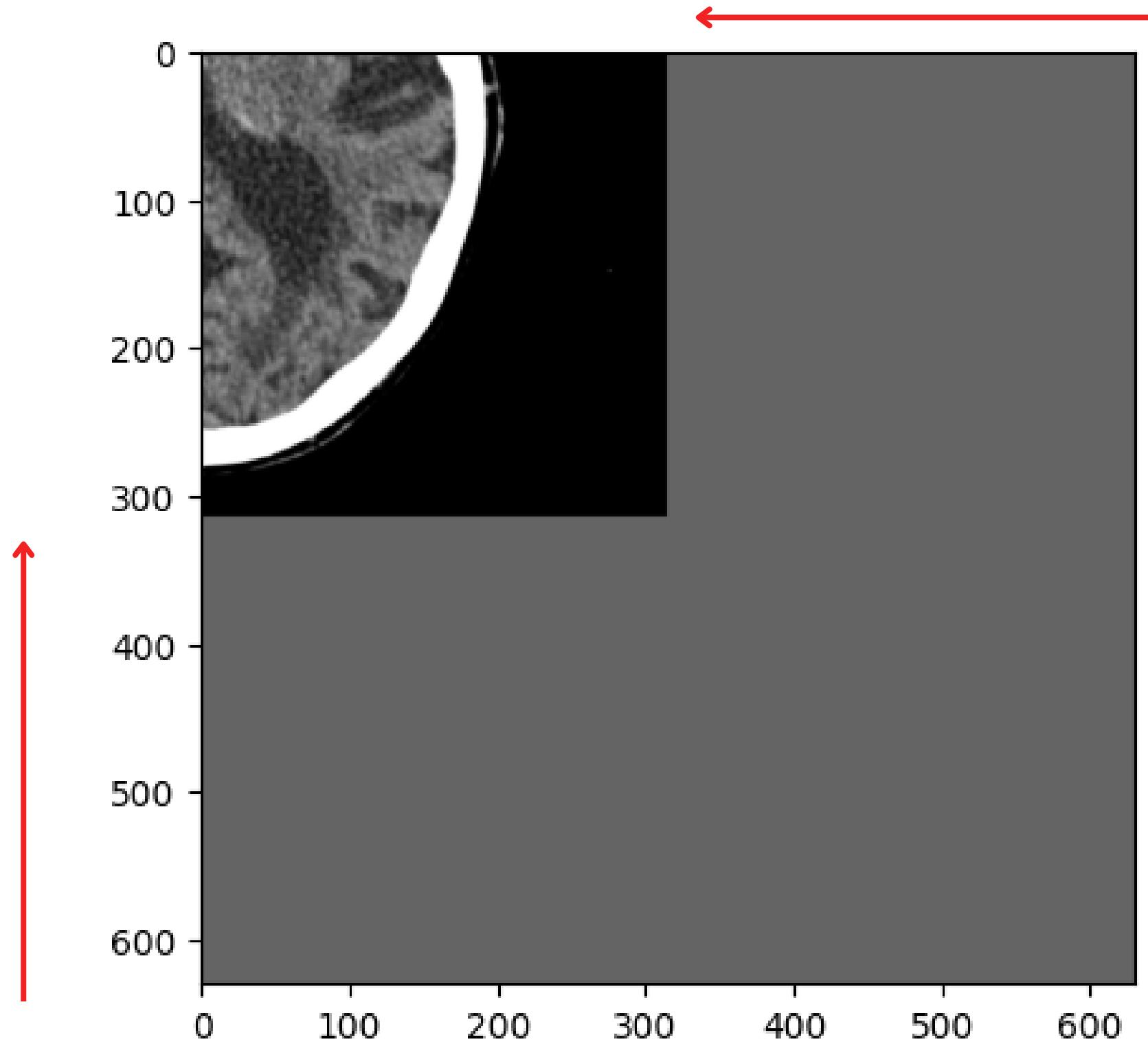


Rotating an Image



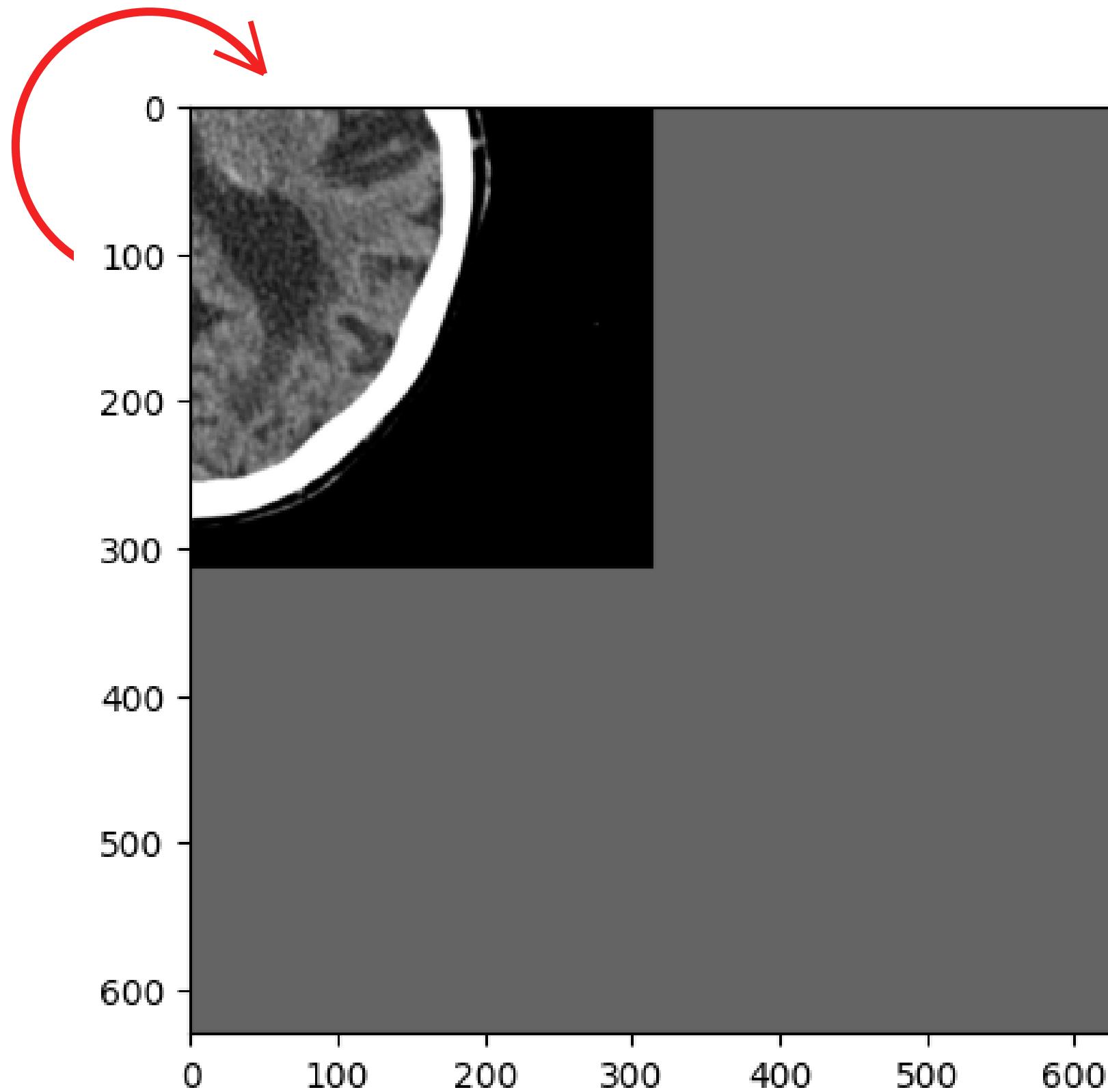
Rotating an Image

Step 1:



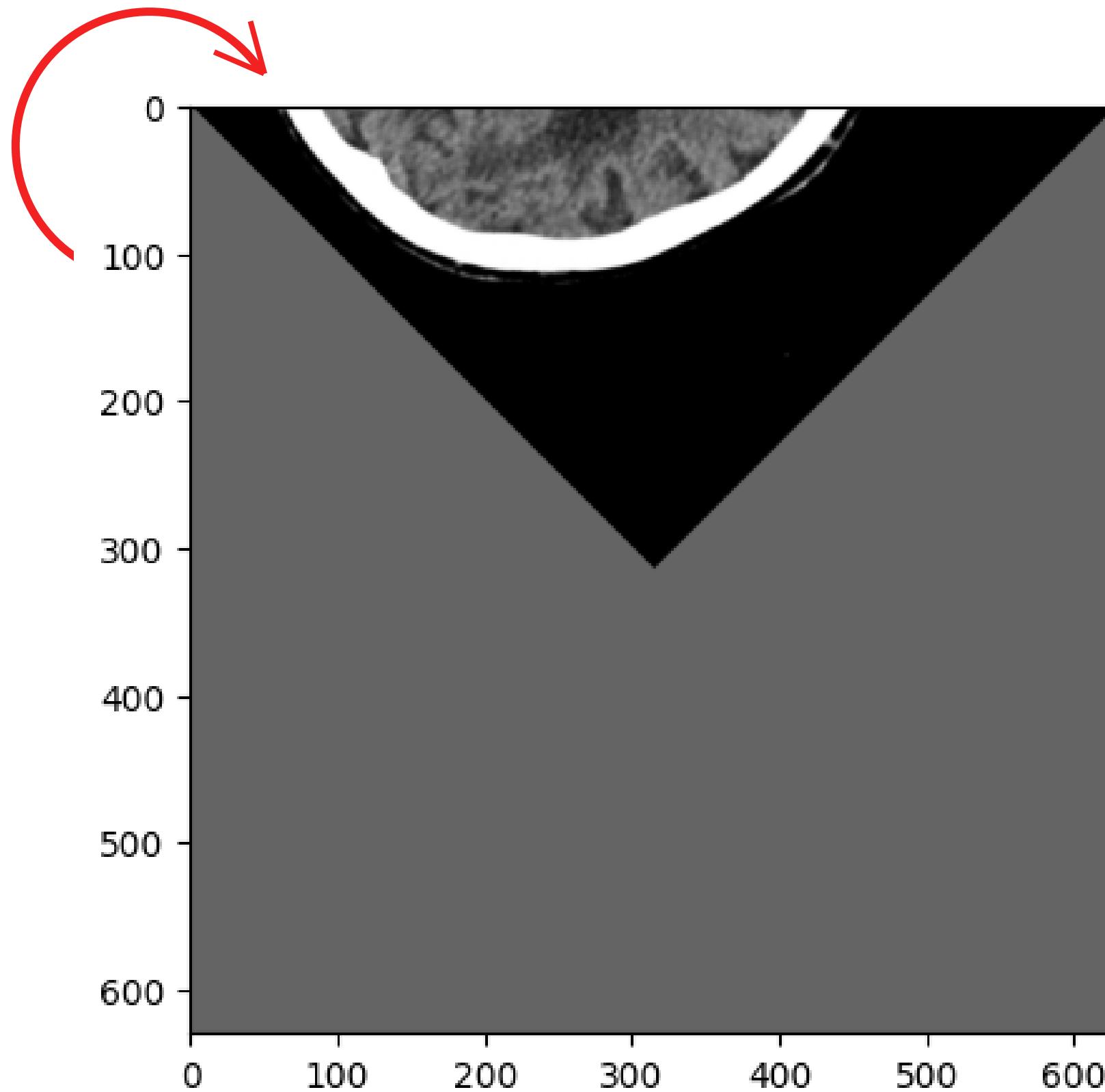
Rotating an Image

Step 1:



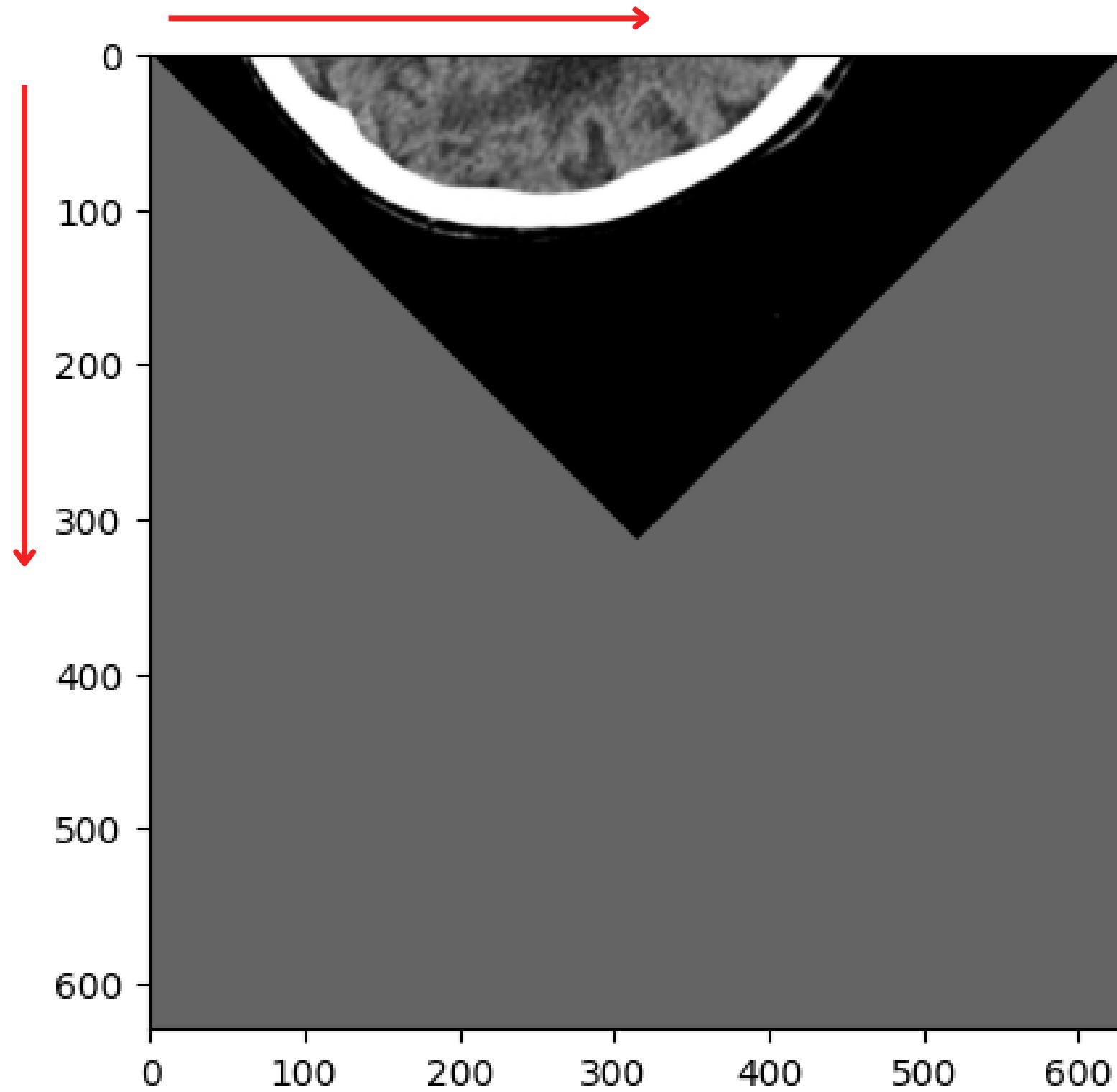
Rotating an Image

Step 2:



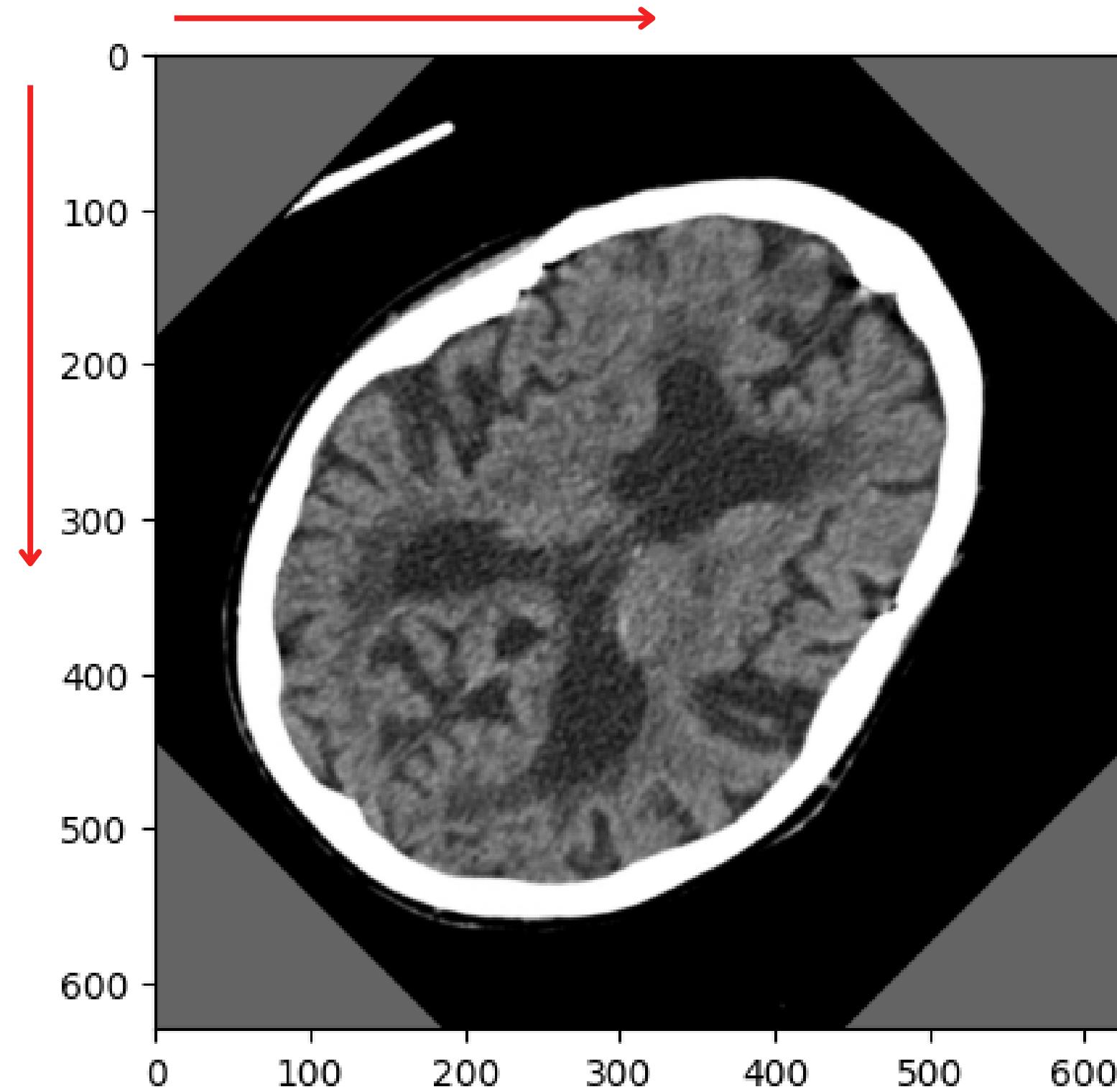
Rotating an Image

Step 2:

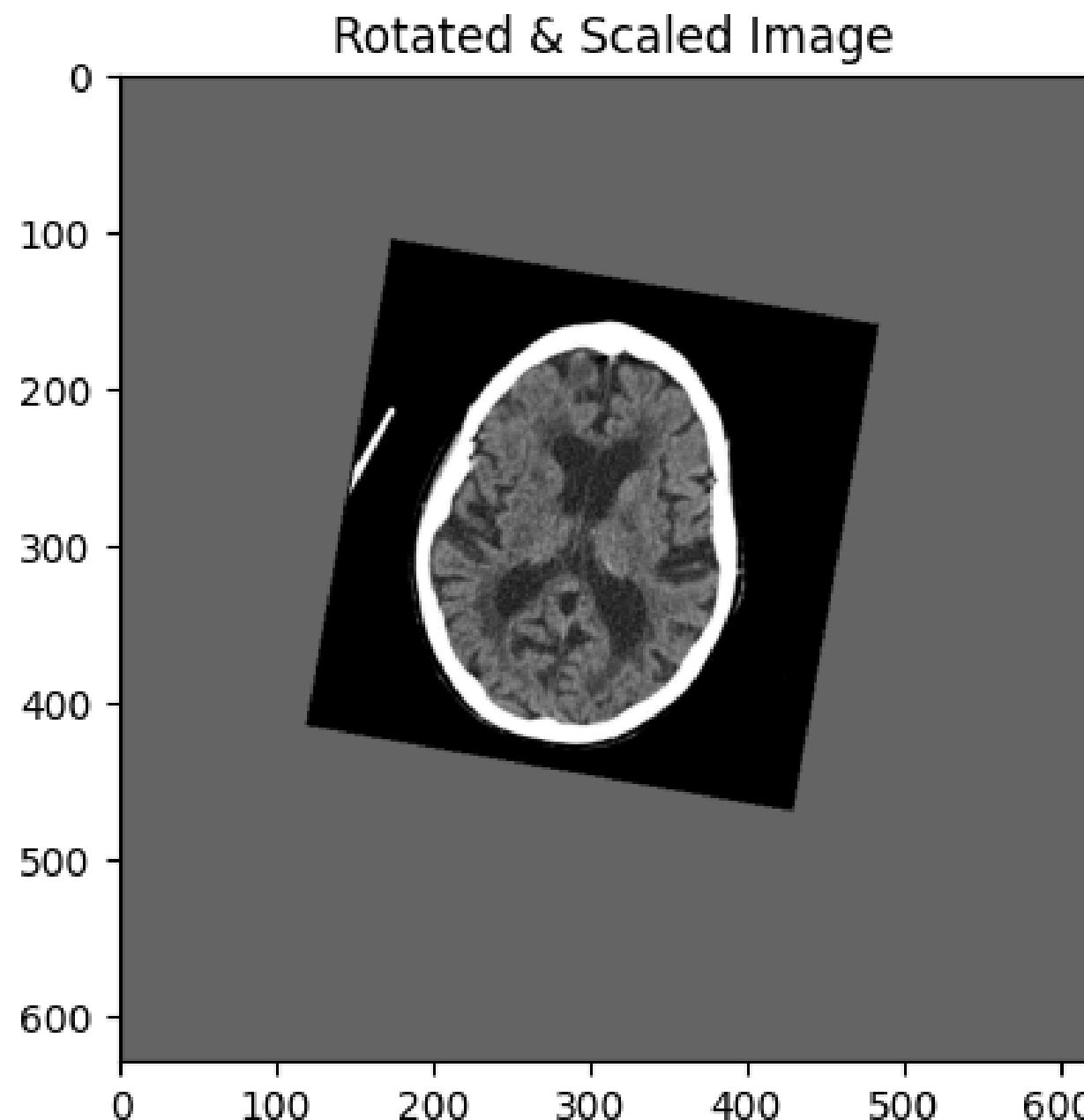
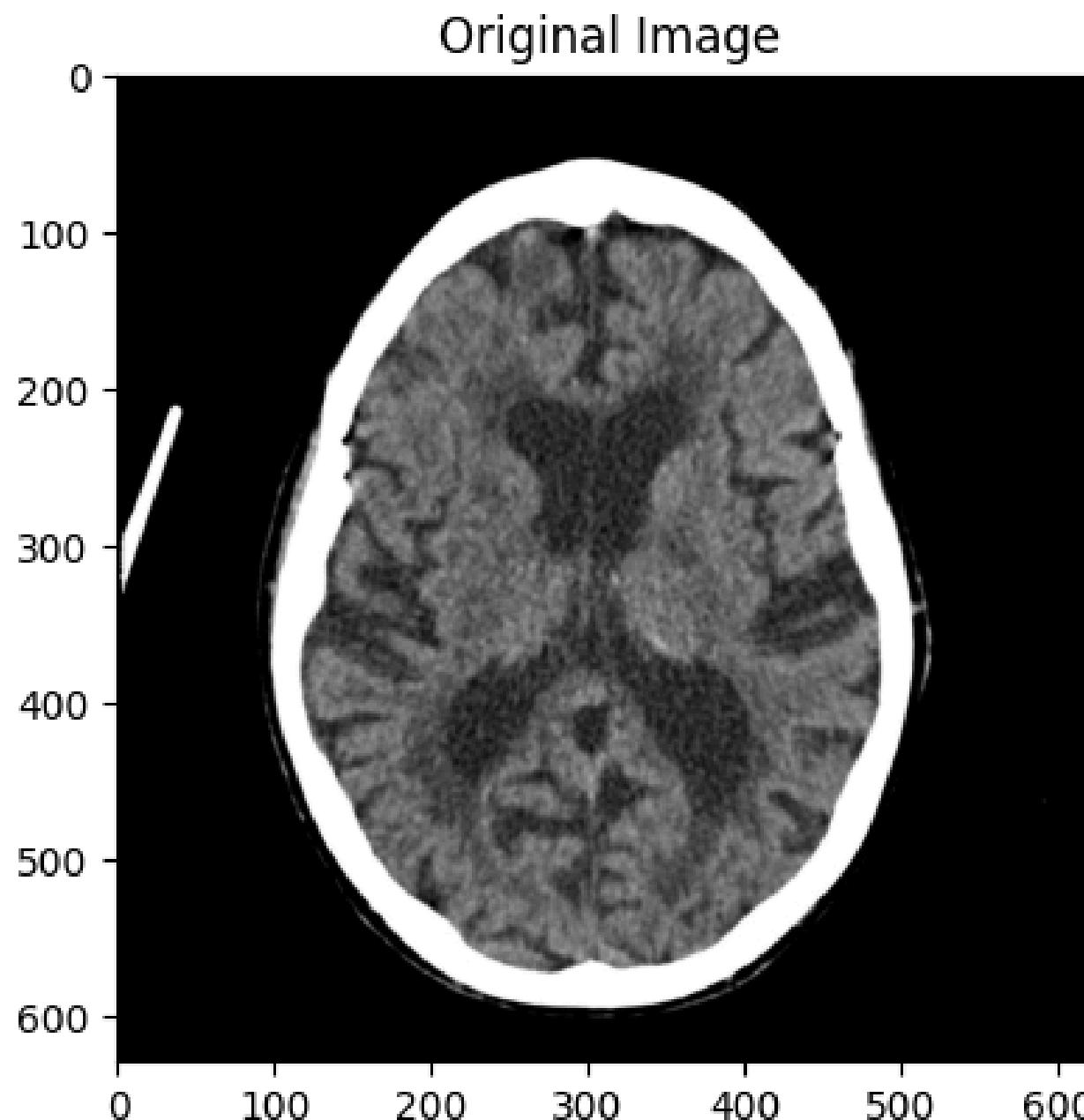


Rotating an Image

Step 3:



Rotating and Scaling



(Scaling factor = 2)

Rotating and Scaling

```
# Create the similarity transform
TransformType = itk.Similarity2DTransform[itk.D]
transform = TransformType.New()
```

Rotating and Scaling

```
# Create the similarity transform
TransformType = itk.Similarity2DTransform[itk.D]
transform = TransformType.New()
```

- Rotation
- Translation
- Scaling

Rotating and Scaling

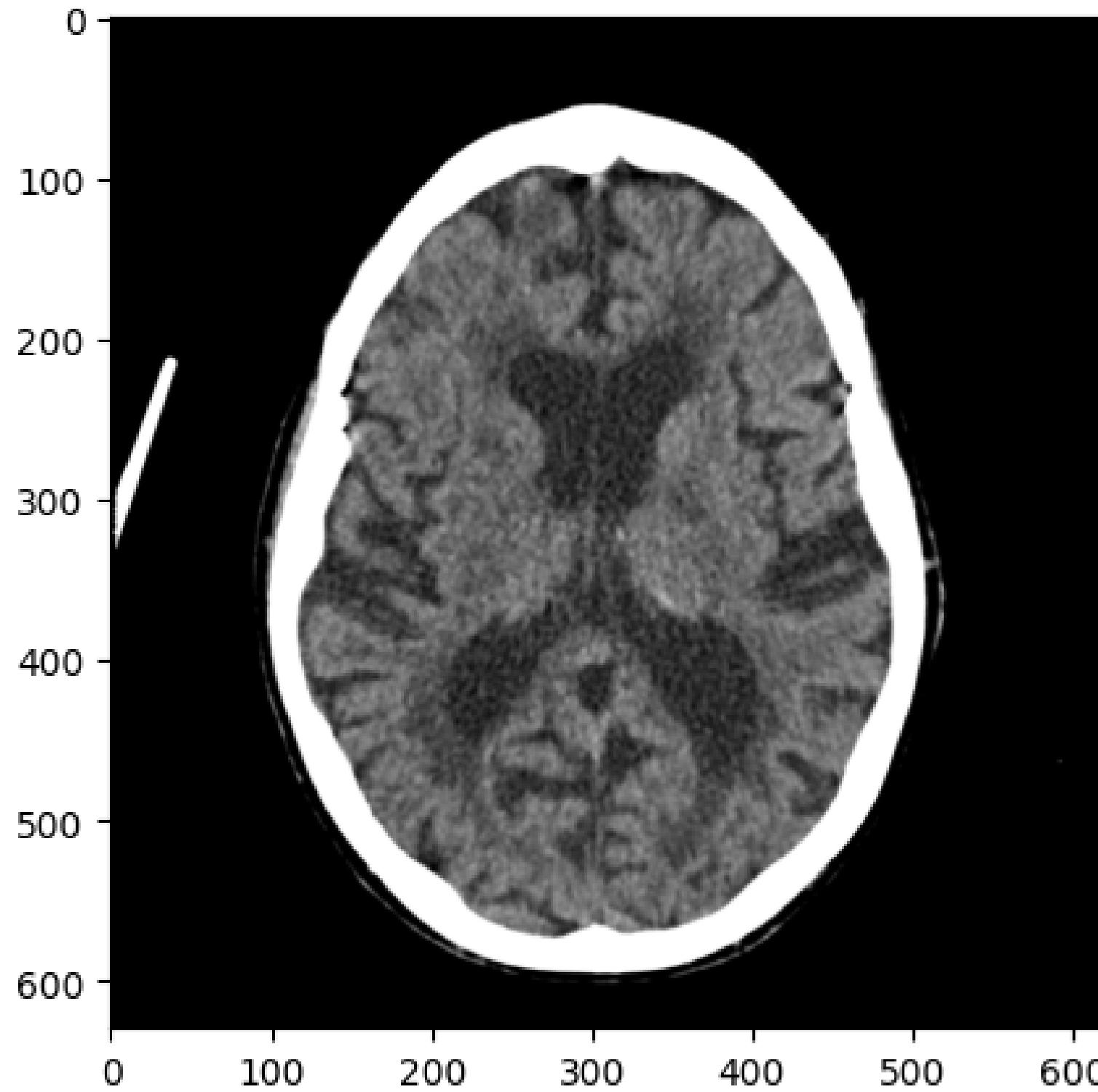
```
# Create the similarity transform
TransformType = itk.Similarity2DTransform[itk.D]
transform = TransformType.New()
```

- Rotation
- Translation
- Scaling

```
# Convert angle to radians
degrees_to_radians = np.pi / 180.0
angle = degrees * degrees_to_radians
```

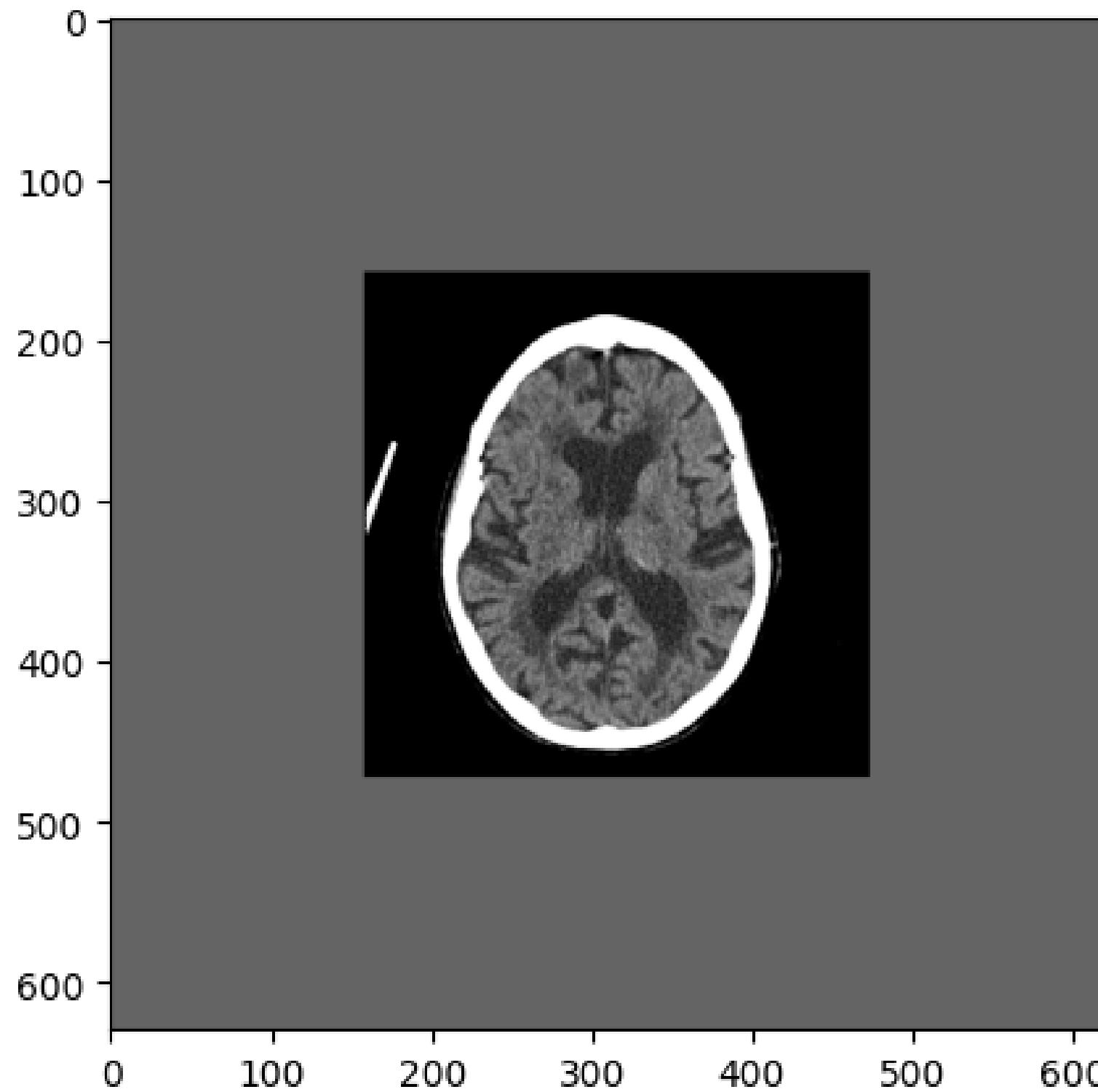
Rotating and Scaling

Original Image



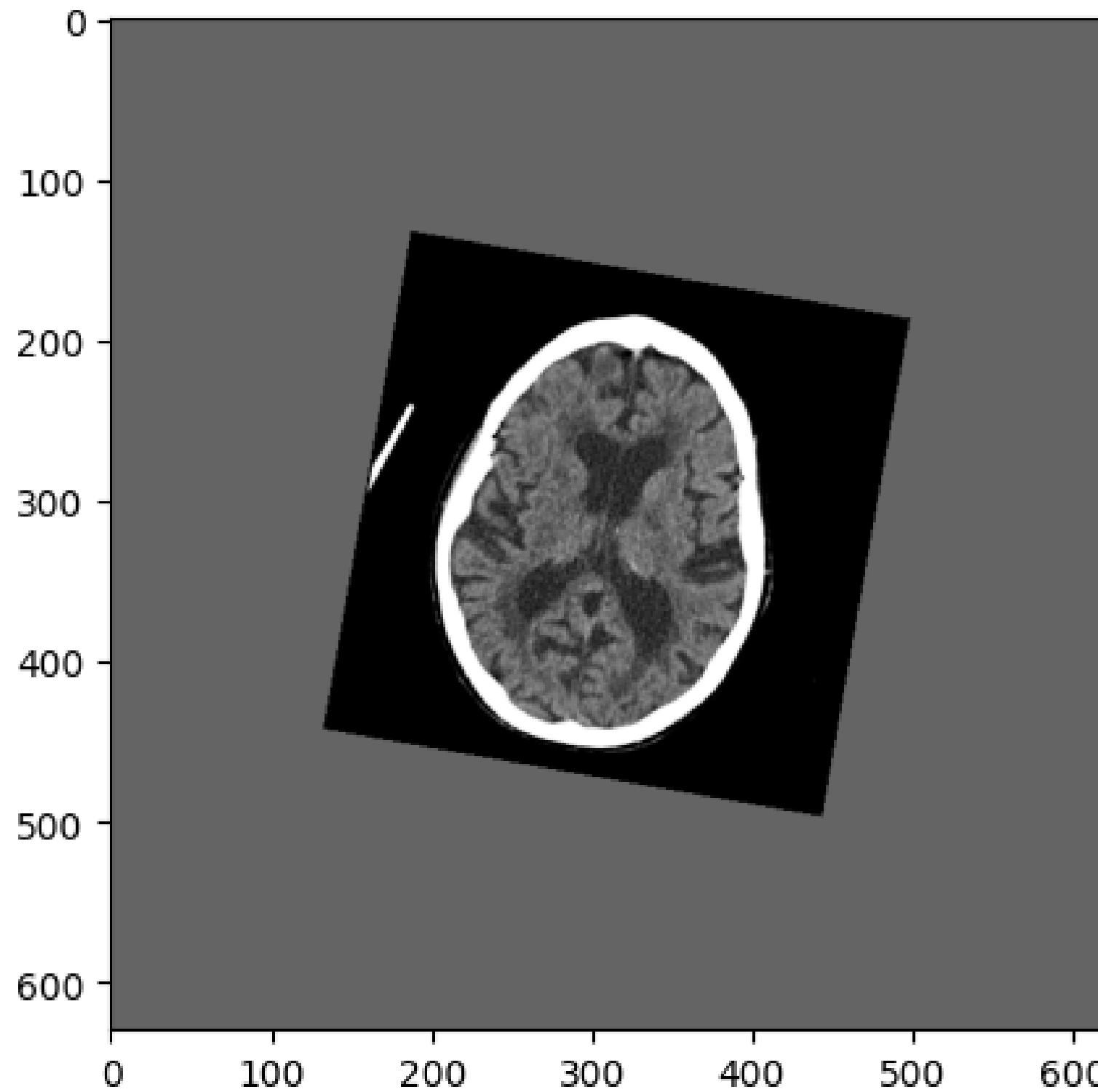
Rotating and Scaling

Scaled Image



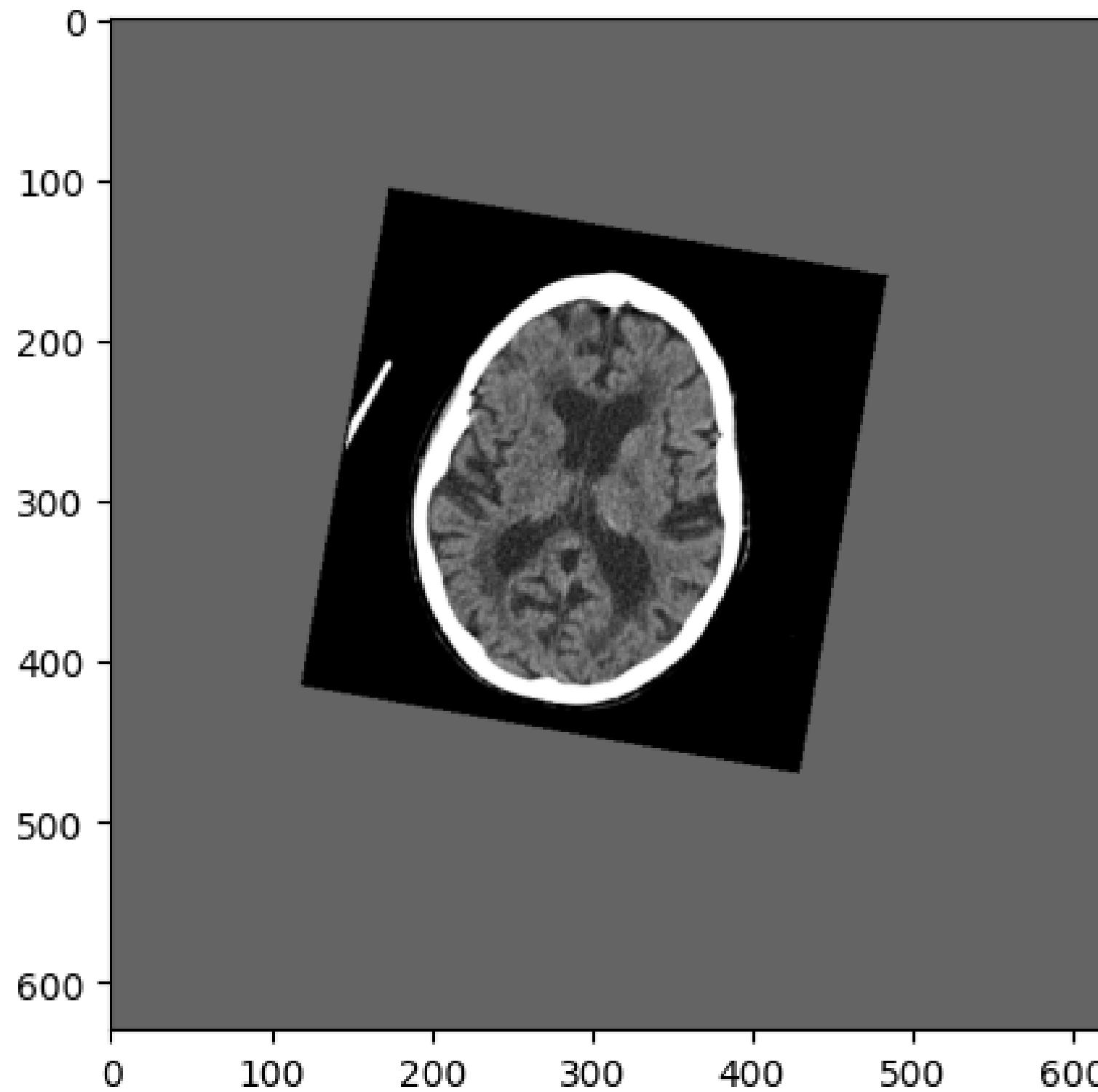
Rotating and Scaling

Rotated Image

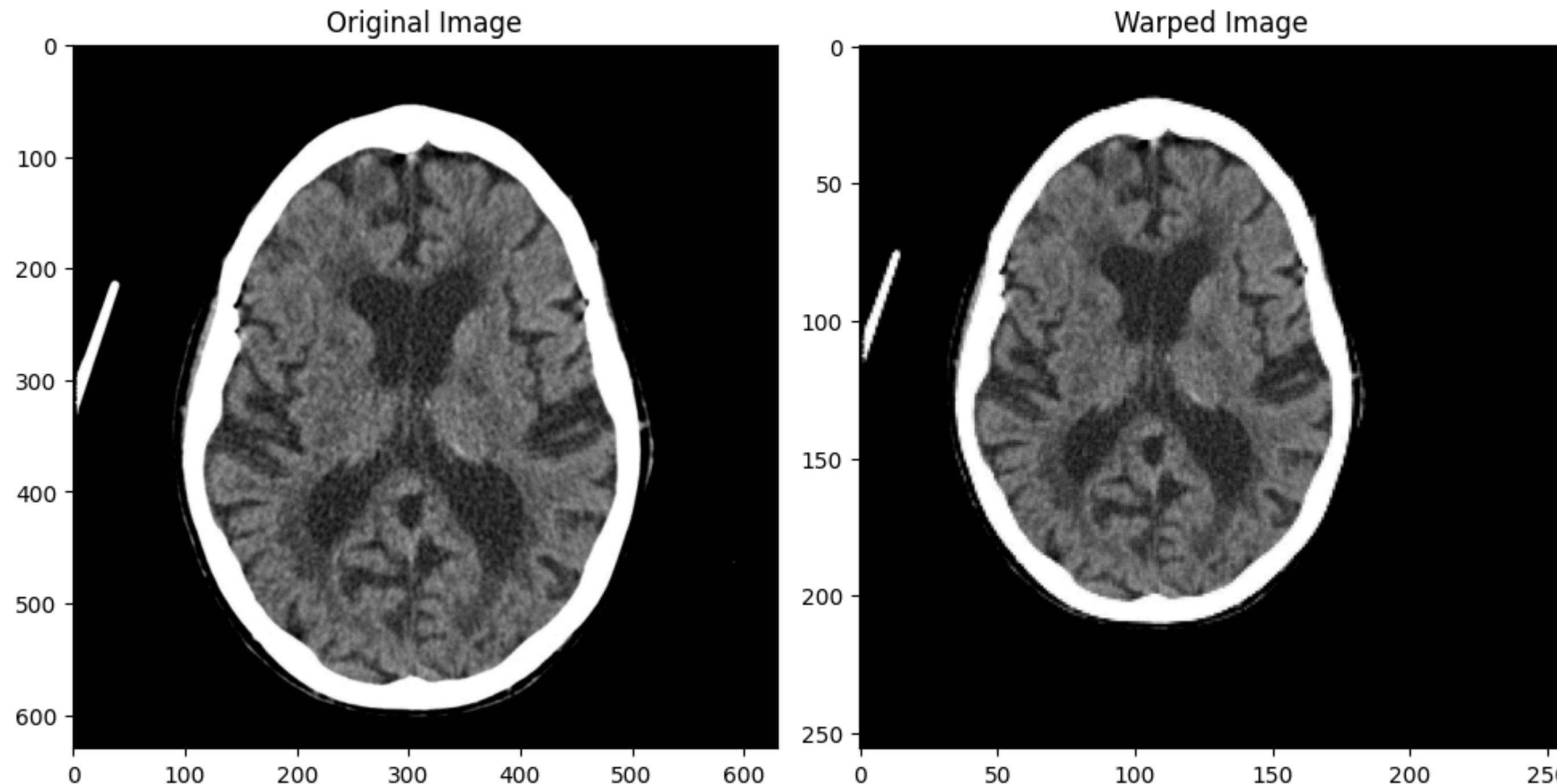


Rotating and Scaling

Translated Image



Resampling using a deformation field



Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDimensions = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDimensions = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDimensions = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDimensions = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDimensions = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

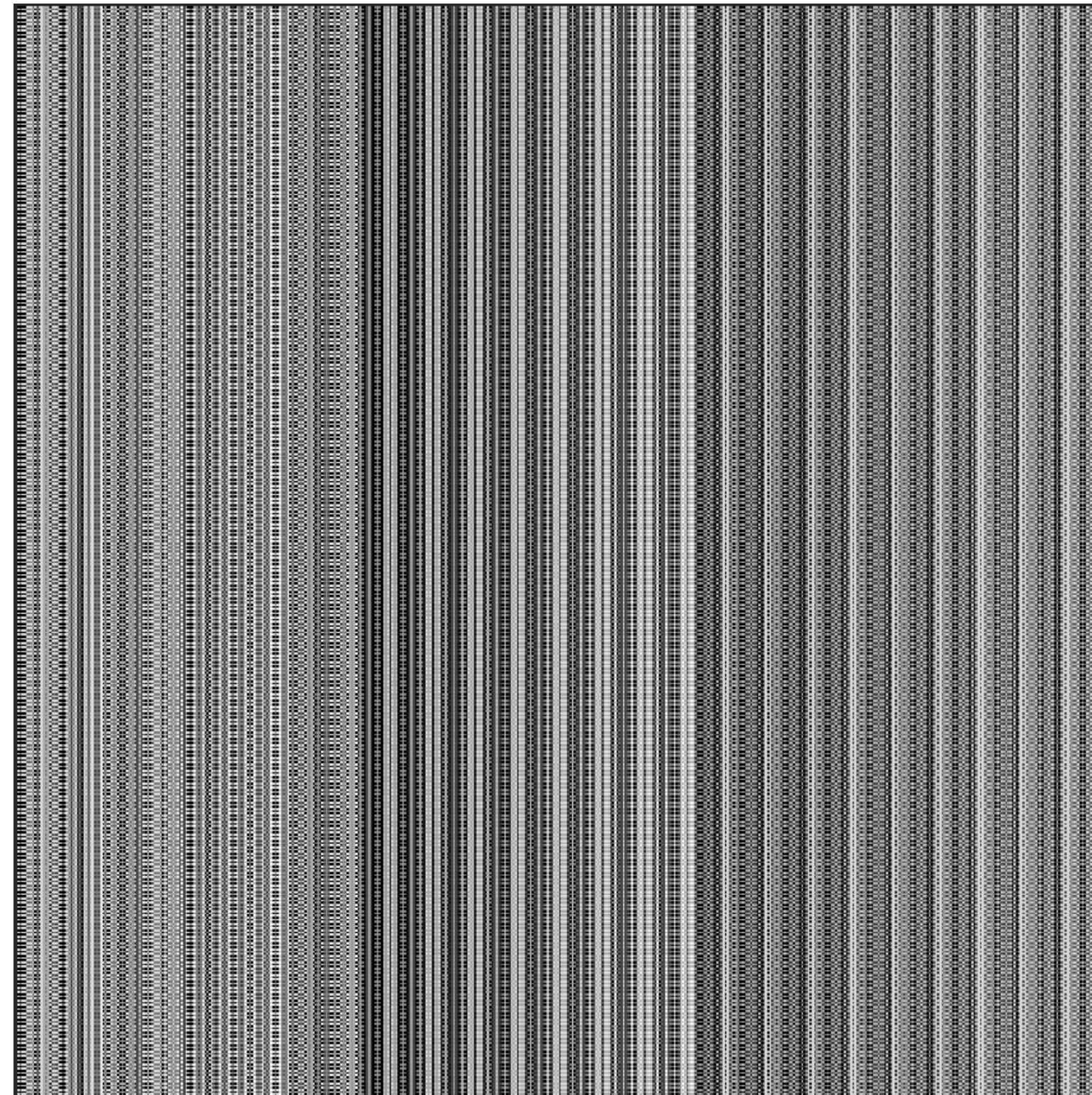
```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

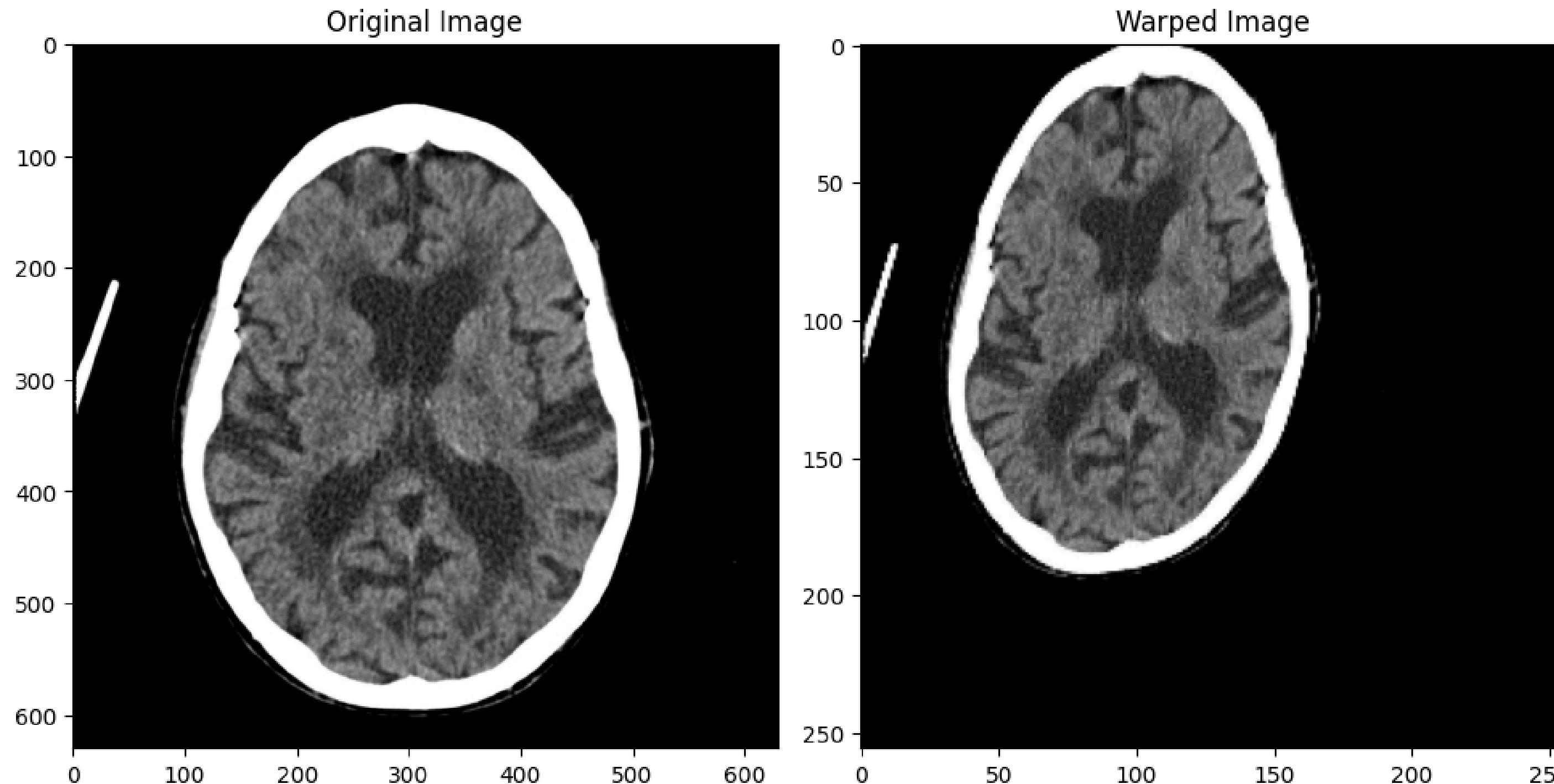
```
☰ displacement_field.mha × +  
1 objectType = Image  
2 NDims = 2  
3 BinaryData = True  
4 BinaryDataByteOrderMSB = False  
5 CompressedData = False  
6 TransformMatrix = 1 0 0 1  
7 Offset = 0 0  
8 CenterOfRotation = 0 0  
9 DimSize = 256 256  
10 ElementSpacing = 1 1  
11 ElementNumberOfChannels = 2  
12 ElementType = MET_FLOAT  
13 ElementDataFile = displacement_field.raw  
14
```

Resampling using a deformation field

[displacement_field.raw](#)



Resampling using a deformation field



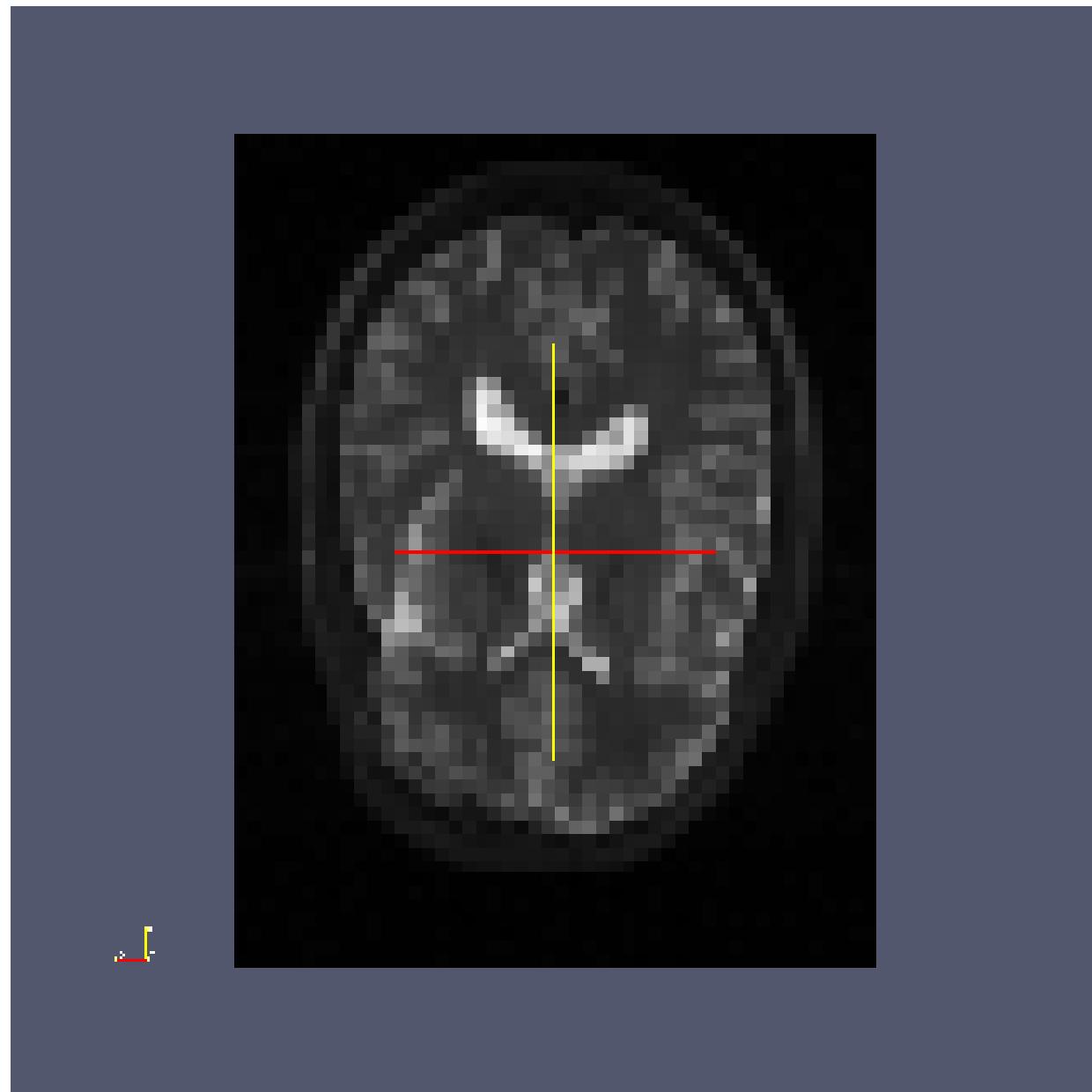
Frequency Domain

Frequency Domain

Unfortunately, no python classes for this

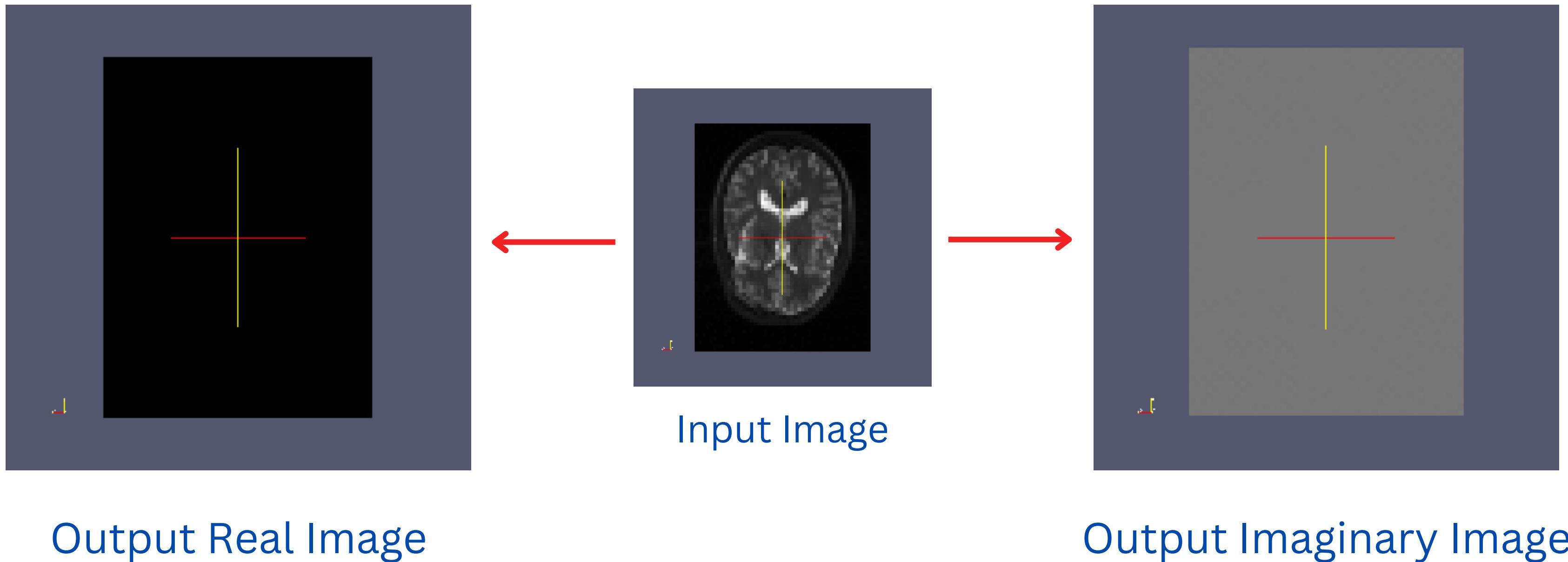


Computing FFT

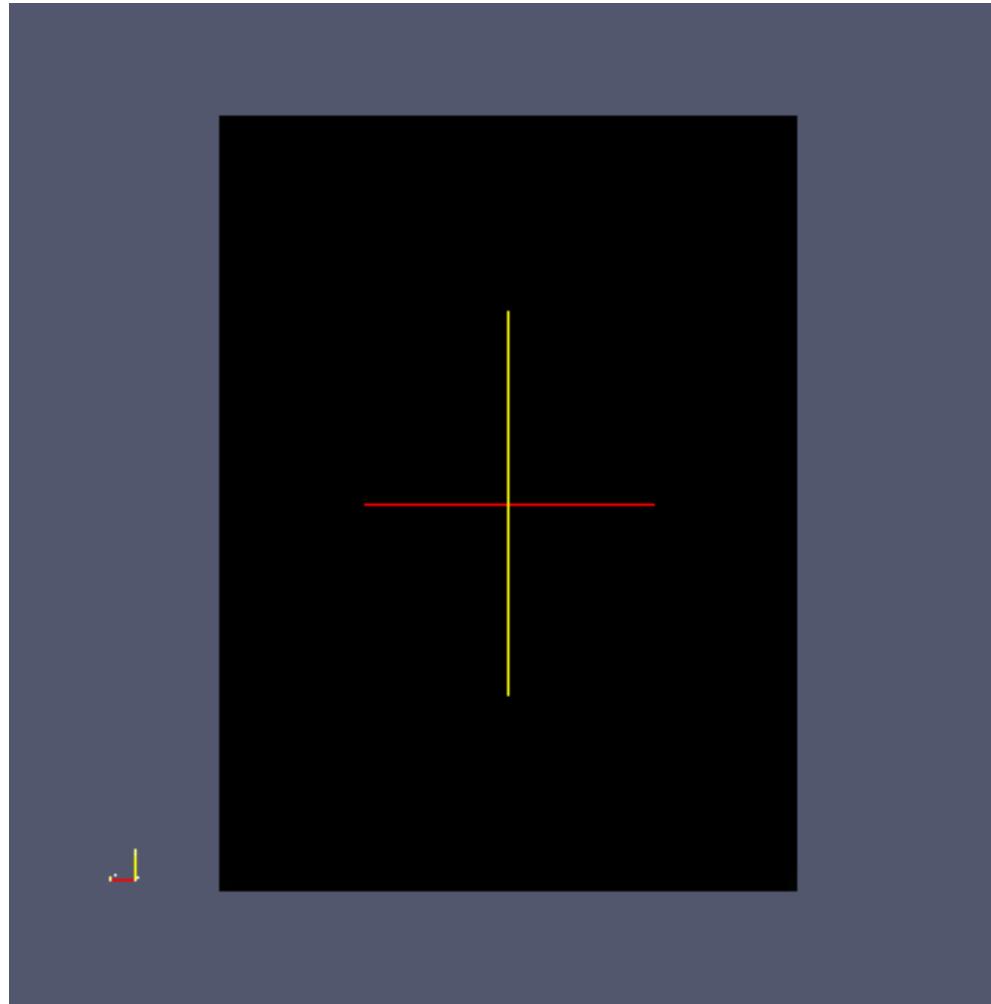


Input Image

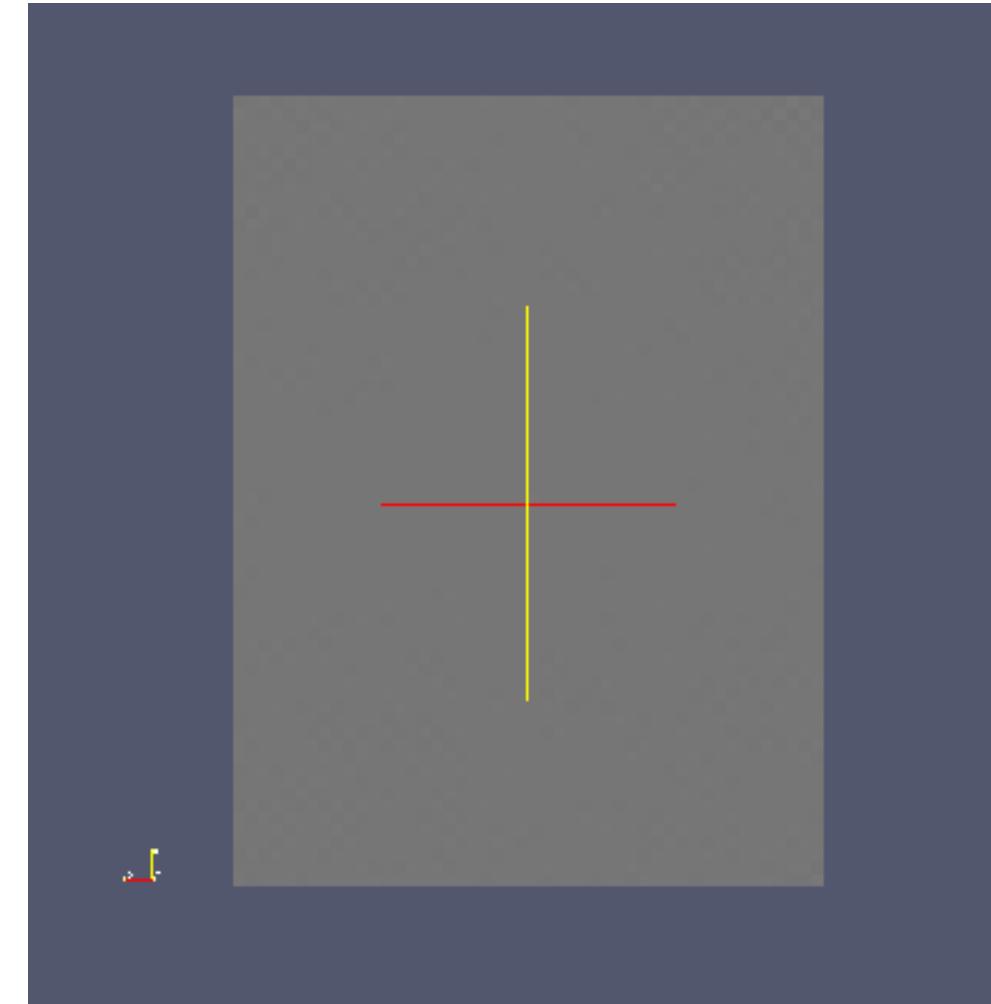
Computing FFT



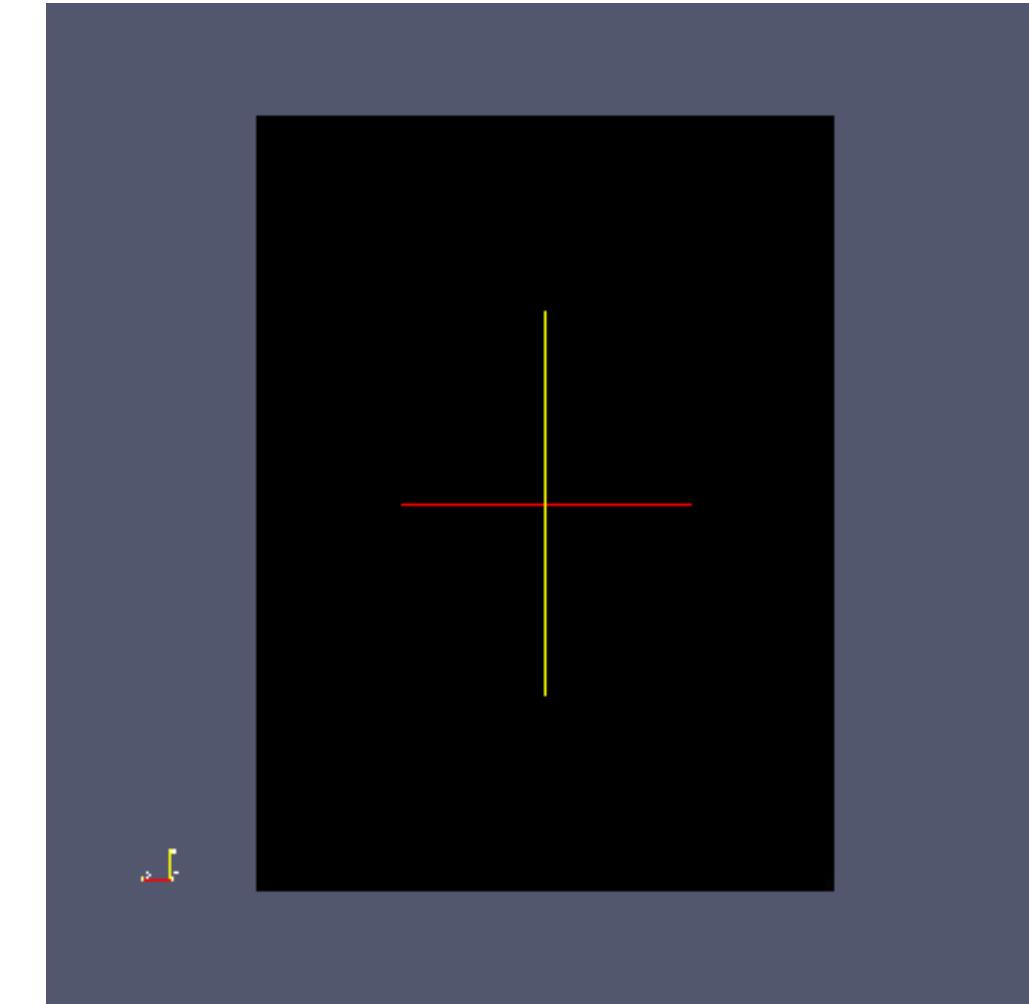
Computing FFT



Output Real Image



Output Imaginary Image



Output modulus Image

Computing FFT

```
#include "itkImage.h"
#include "itkVnlForwardFFTImageFilter.h"
#include "itkComplexToRealImageFilter.h"
#include "itkComplexToImaginaryImageFilter.h"
```

Summery

Geometric Transformation

- Flip Image
- Resample Image
- Resample and Translation
- Space and Origin
- Rotating Image
- Deformation Field

Frequency Domain

- FFT