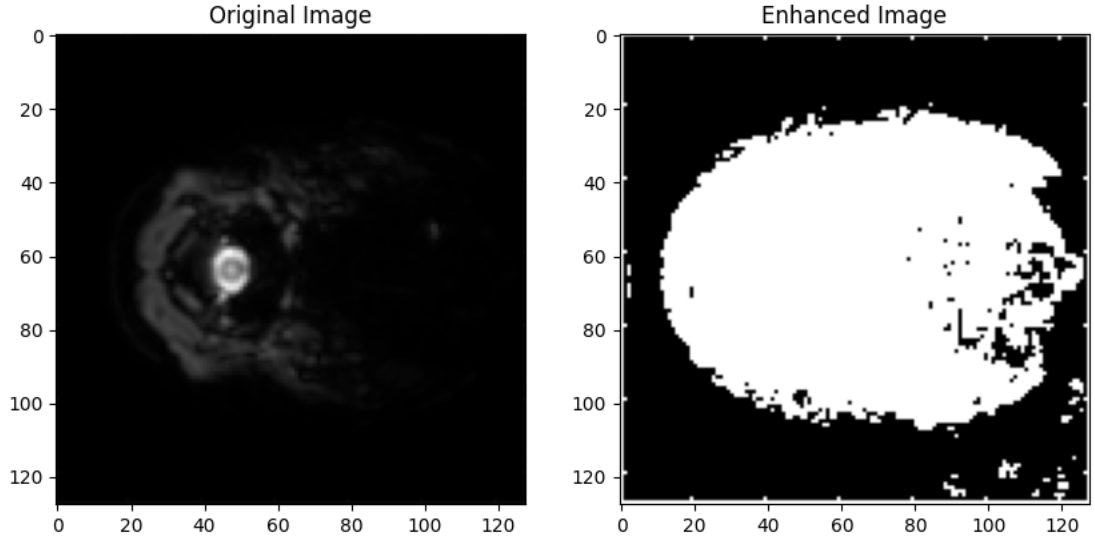
```
# Specify the path to your zip file
zip_file_path = '/content/fMRI dataset.zip'

# Unzip the file in the same directory
!unzip -q "/content/fMRI dataset.zip"

# List the contents of the current directory
!ls
```

```
     'fMRI dataset.zip'   model   sample_data
```

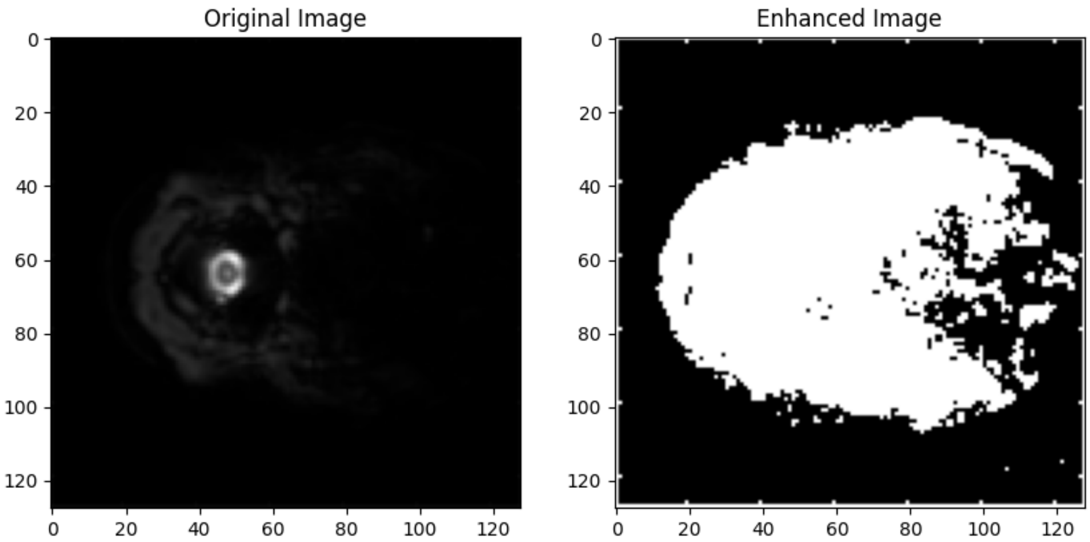
```
# Install necessary libraries
!pip install matplotlib scikit-image

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/afraid/Chanvol_181slice_1.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (0.19.3)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.11.4)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (3.2.1)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2023.12.9)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



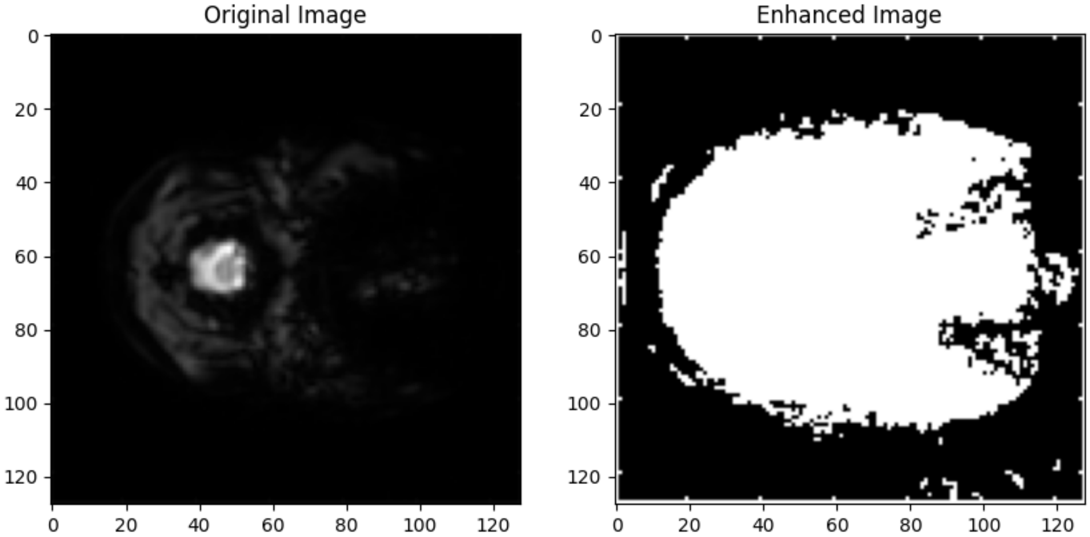
```
# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/calm/Chanvol_10slice_1.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```


Original Image
Enhanced Image

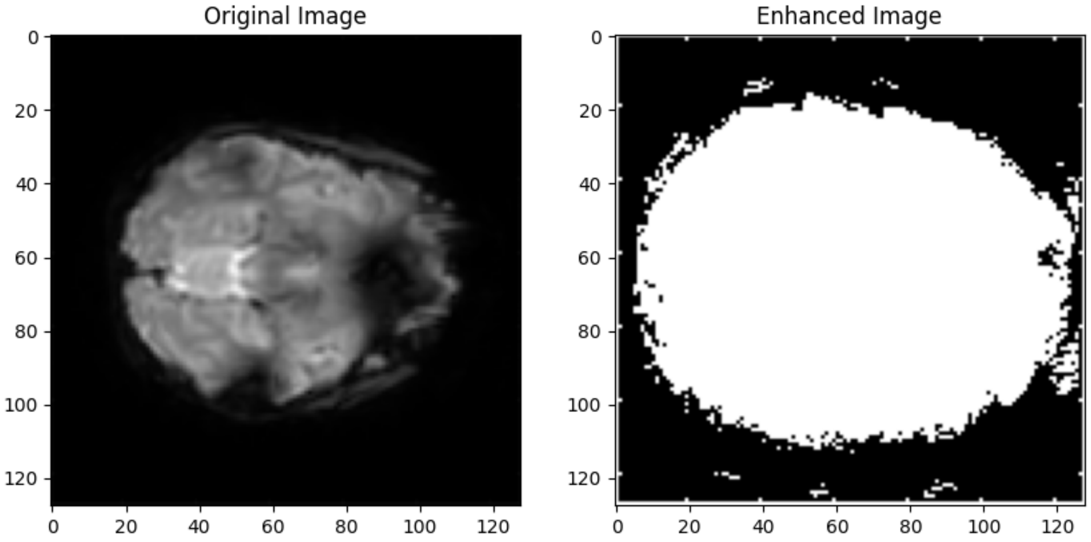
```
# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/delighted/Chanvol_101slice_3.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```


Original Image
Enhanced Image

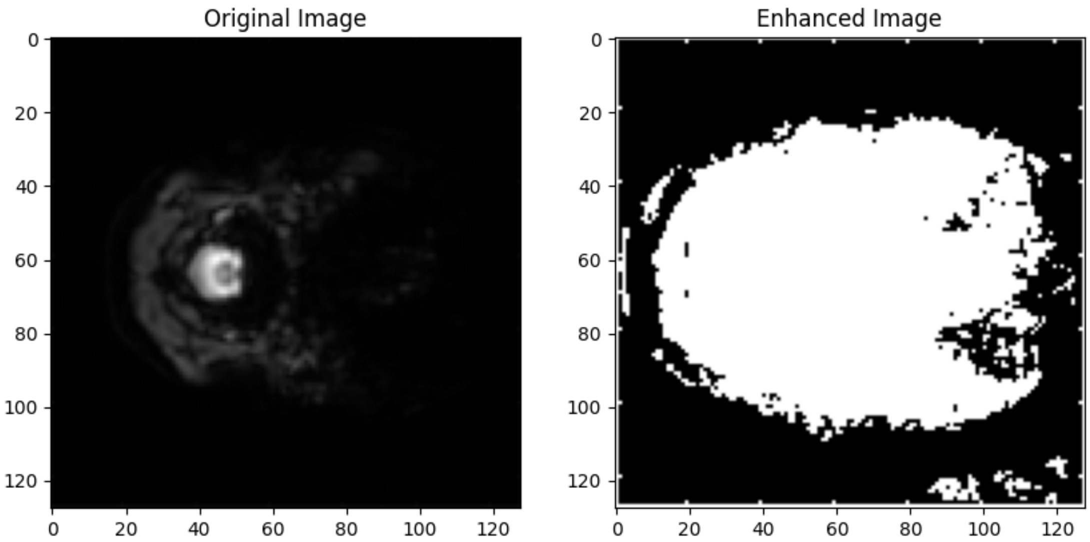
```
# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/depressed/Chanvol_121slice_17.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



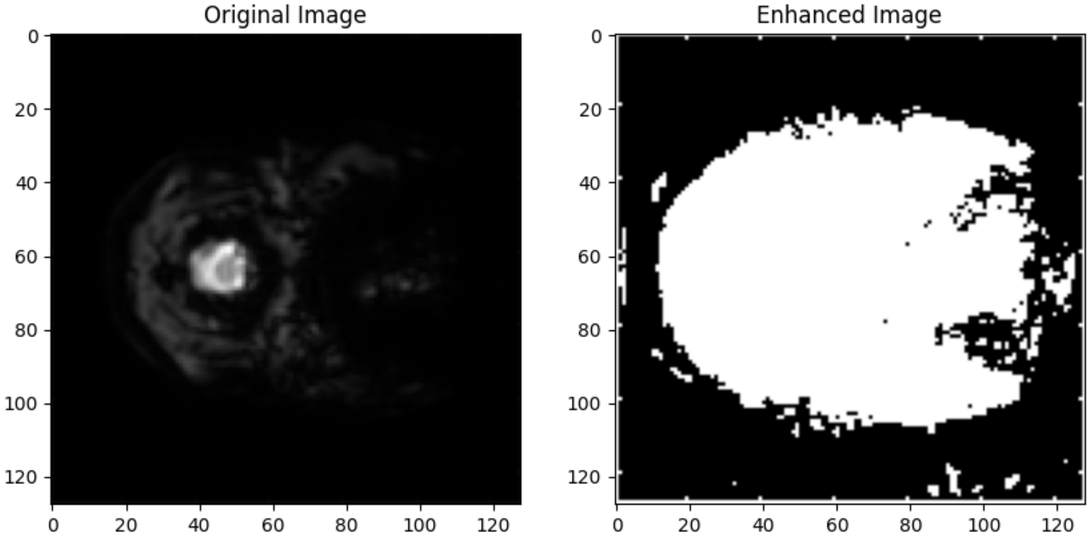
```
# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/depressed/Chanvol_122slice_2.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



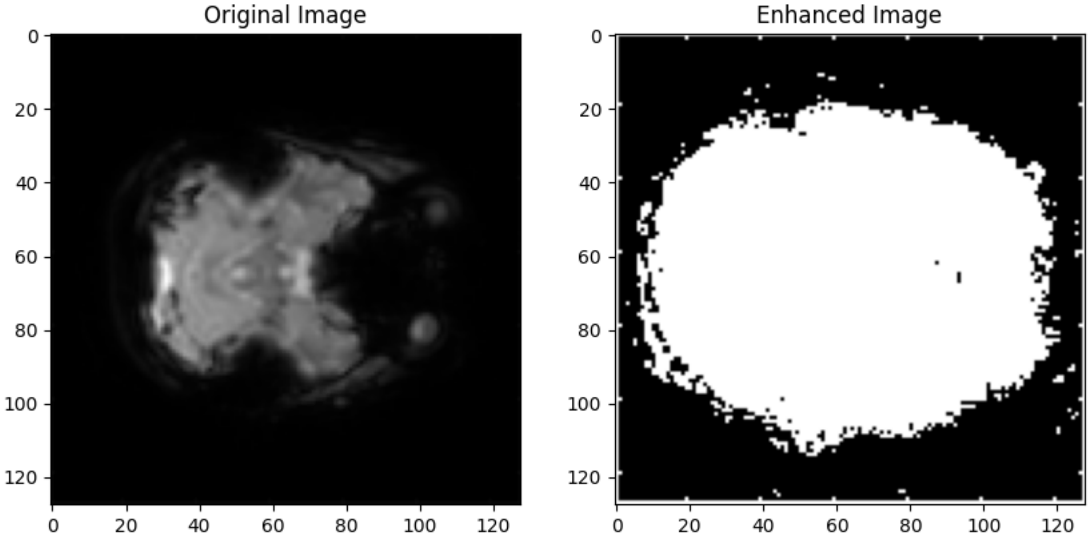
```
# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/depressed/Chanvol_62slice_3.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Visualize the original and enhanced images
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



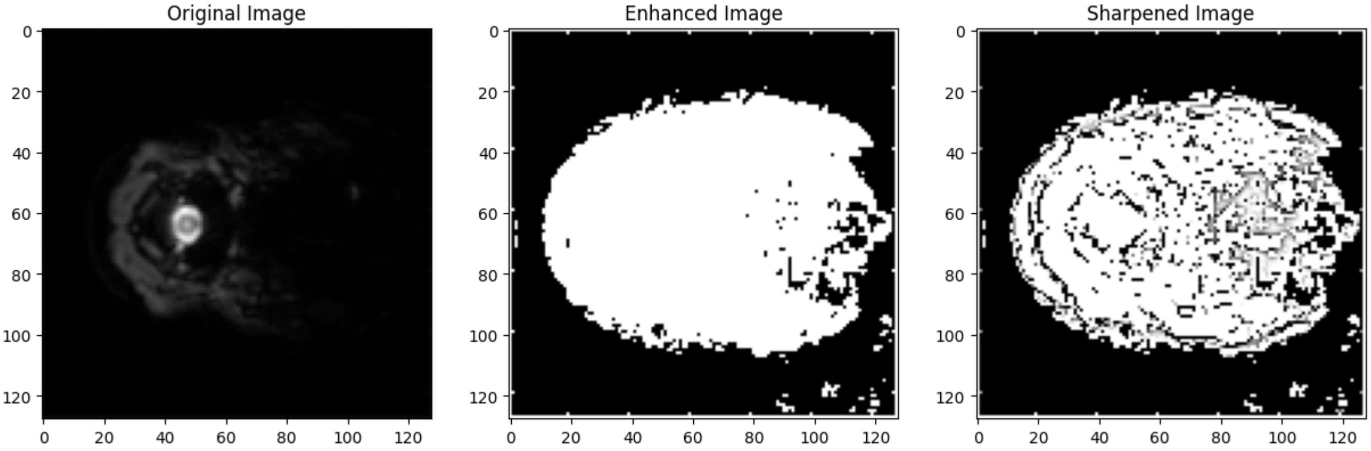
```
    # Load the fMRI PNG image (replace 'your_image.png' with your file name)
    image_path = '/content/model/train/excited/Chanvol_161slice_14.png'
    img = Image.open(image_path)

    # Convert the image to a NumPy array
    img_array = np.array(img)

    # Apply contrast stretching for enhancement
    enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

    # Visualize the original and enhanced images
    plt.figure(figsize=(10, 5))

    plt.subplot(1, 2, 1)
    plt.imshow(img_array, cmap='gray')
    plt.title('Original Image')

    plt.subplot(1, 2, 2)
    plt.imshow(enhanced_img, cmap='gray')
    plt.title('Enhanced Image')

    plt.show()
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



```
# Install necessary libraries
!pip install matplotlib scikit-image scipy

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter, convolve

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/afraid/Chanvol_181slice_1.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Function to apply sharpening filter
def sharpen_image(image, alpha=1.5):
    blurred = gaussian_filter(image, sigma=1)
    sharpened = image + alpha * (image - blurred)
    return np.clip(sharpened, 0, 255)

# Sharpen the enhanced image
sharpened_img = sharpen_image(enhanced_img)

# Visualize the original, enhanced, and sharpened images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.subplot(1, 3, 3)
plt.imshow(sharpened_img, cmap='gray')
plt.title('Sharpened Image')

plt.show()
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (0.19.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (3.2.1)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2023.12.9)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



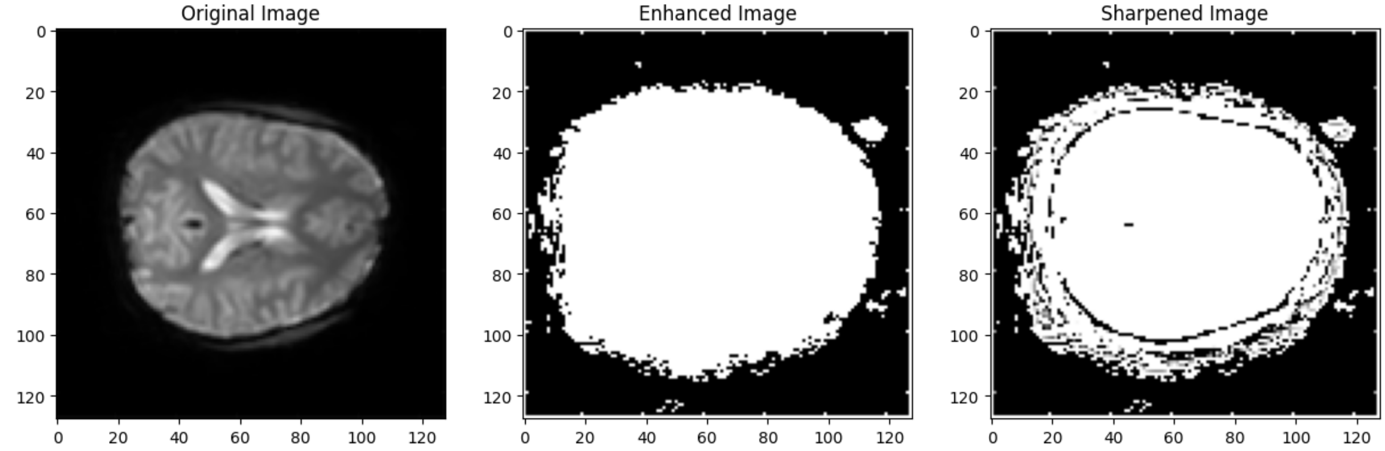
Original Image

Enhanced Image

Sharpened Image

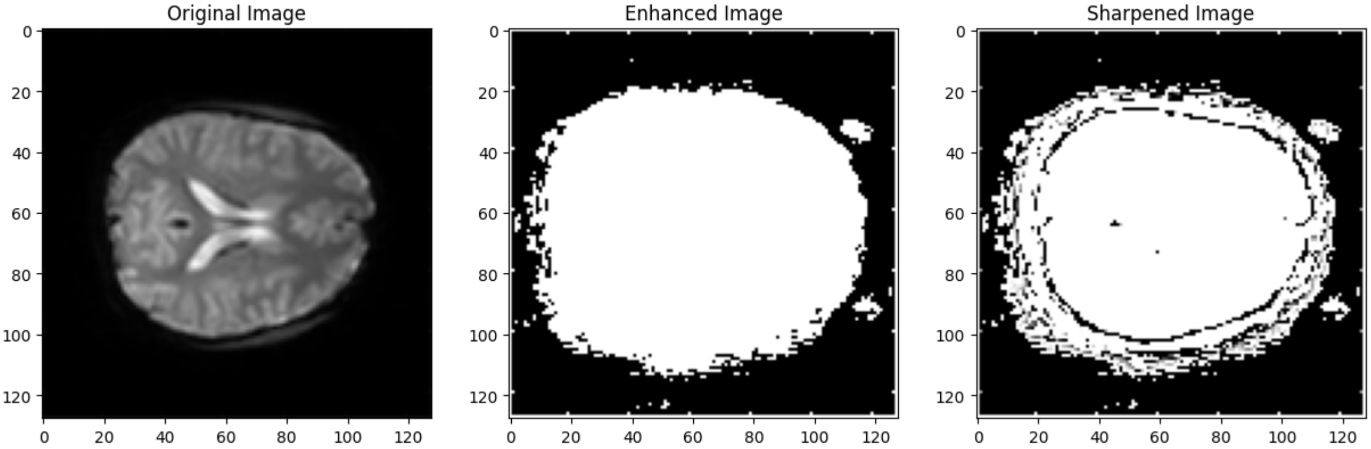
```
# Install necessary libraries
!pip install matplotlib scikit-image scipy

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter, convolve

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/calm/Chanvol_10slice_23.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Function to apply sharpening filter
def sharpen_image(image, alpha=1.5):
    blurred = gaussian_filter(image, sigma=1)
    sharpened = image + alpha * (image - blurred)
    return np.clip(sharpened, 0, 255)

# Sharpen the enhanced image
sharpened_img = sharpen_image(enhanced_img)

# Visualize the original, enhanced, and sharpened images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.subplot(1, 3, 3)
plt.imshow(sharpened_img, cmap='gray')
plt.title('Sharpened Image')

plt.show()
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (0.19.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (3.2.1)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2023.12.9)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



```
# Install necessary libraries
!pip install matplotlib scikit-image scipy

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter, convolve

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/delighted/Chanvol_101slice_23.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Function to apply sharpening filter
def sharpen_image(image, alpha=1.5):
    blurred = gaussian_filter(image, sigma=1)
    sharpened = image + alpha * (image - blurred)
    return np.clip(sharpened, 0, 255)

# Sharpen the enhanced image
sharpened_img = sharpen_image(enhanced_img)

# Visualize the original, enhanced, and sharpened images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.subplot(1, 3, 3)
plt.imshow(sharpened_img, cmap='gray')
plt.title('Sharpened Image')

plt.show()
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (0.19.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (3.2.1)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (2023.12.9)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image) (1.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



```
# Install necessary libraries
!pip install matplotlib scikit-image scipy

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter, convolve

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/depressed/Chanvol_121slice_14.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Function to apply sharpening filter
def sharpen_image(image, alpha=1.5):
    blurred = gaussian_filter(image, sigma=1)
    sharpened = image + alpha * (image - blurred)
    return np.clip(sharpened, 0, 255)

# Sharpen the enhanced image
sharpened_img = sharpen_image(enhanced_img)

# Visualize the original, enhanced, and sharpened images
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.imshow(img_array, cmap='gray')
plt.title('Original Image')

plt.subplot(1, 3, 2)
plt.imshow(enhanced_img, cmap='gray')
plt.title('Enhanced Image')

plt.subplot(1, 3, 3)
plt.imshow(sharpened_img, cmap='gray')
plt.title('Sharpened Image')

plt.show()
```

```
# Install necessary libraries
!pip install matplotlib scikit-image scipy

# Import libraries
import matplotlib.pyplot as plt
from skimage import exposure
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter, convolve

# Load the fMRI PNG image (replace 'your_image.png' with your file name)
image_path = '/content/model/train/depressed/Chanvol_121slice_25.png'
img = Image.open(image_path)

# Convert the image to a NumPy array
img_array = np.array(img)

# Apply contrast stretching for enhancement
enhanced_img = exposure.rescale_intensity(img_array, in_range=(0, 255), out_range=(0, 255))

# Function to apply sharpening filter
```