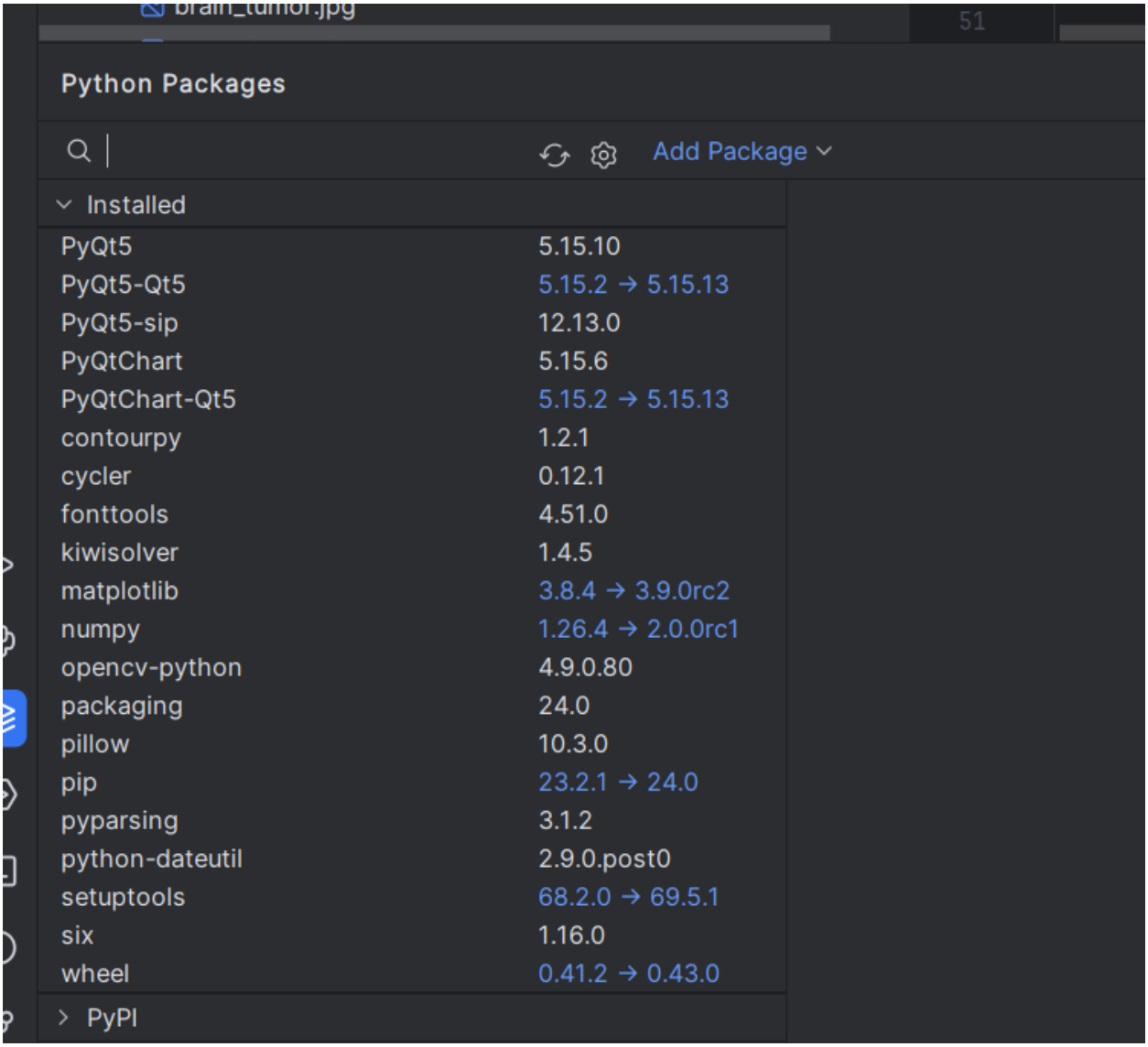
Step 1:


brain_tumor.jpg
51
Python Packages
Add Package
Installed
PyQt5 5.15.10
PyQt5-Qt5 5.15.2 → 5.15.13
PyQt5-sip 12.13.0
PyQtChart 5.15.6
PyQtChart-Qt5 5.15.2 → 5.15.13
contourpy 1.2.1
cycler 0.12.1
fonttools 4.51.0
kiwisolver 1.4.5
matplotlib 3.8.4 → 3.9.0rc2
numpy 1.26.4 → 2.0.0rc1
opencv-python 4.9.0.80
packaging 24.0
pillow 10.3.0
pip 23.2.1 → 24.0
pyparsing 3.1.2
python-dateutil 2.9.0.post0
setuptools 68.2.0 → 69.5.1
six 1.16.0
wheel 0.41.2 → 0.43.0
PyPI

Step 2:

```
image = cv2.imread('tumor2.jpg')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred_image = cv2.GaussianBlur(gray_image, ksize: (5, 5), sigmaX: 0)
enhanced_image = cv2.equalizeHist(blurred_image)
edges = cv2.Canny(blurred_image, 50, 50)

cv2.namedWindow( winname: "Edge detection", cv2.WINDOW_NORMAL)
edge = cv2.resize(edges, dsize: (1280, 960))
cv2.imshow( winname: "Edge detection", edge)
```

Step 3:

```
image =cv2.imread('brain_mri.jpg')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred_image = cv2.GaussianBlur(gray_image, ksize: (5,5), sigmaX: 0)
enhanced_image = cv2.equalizeHist(blurred_image)
_, binary_image = cv2.threshold(gray_image, thresh: 100, maxval: 255, cv2.THRESH_BINARY)

edges = cv2.Canny(blurred_image, 50, 50)
hist = cv2.calcHist( images: [gray_image], channels: [0], mask: None, histSize: [256], ranges: [0, 2
```

Step 4+5+6:

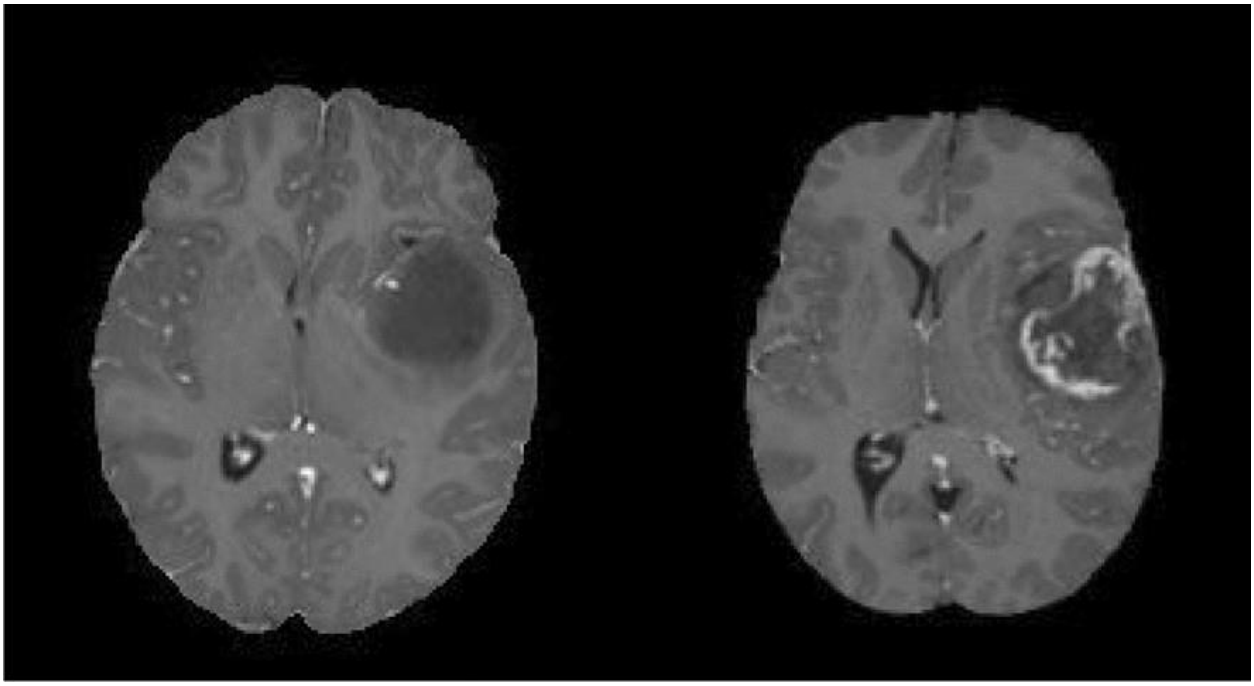
```python
class ImageProcessor(QMainWindow):
    def __init__(self):
        self.open_button = QPushButton("Open Image")
        self.open_button.clicked.connect(self.open_image)

        self.threshold_slider = QSlider(Qt.Horizontal)
        self.threshold_slider.setMinimum(0)
        self.threshold_slider.setMaximum(255)
        self.threshold_slider.setValue(127)
        self.threshold_slider.setTickInterval(10)
        self.threshold_slider.valueChanged.connect(self.process_image)

        self.plot_button = QPushButton("Plot Histogram")
        self.plot_button.clicked.connect(self.plot_histogram)

        layout = QVBoxLayout()
        layout.addWidget(self.image_label)
        layout.addWidget(self.open_button)
        layout.addWidget(QLabel("Threshold:"))
        layout.addWidget(self.threshold_slider)
        layout.addWidget(self.plot_button)

        central_widget = QWidget()
```

```python
2 usages
def process_image(self):
    if self.image is not None:
        gray = cv2.cvtColor(self.image, cv2.COLOR_BGR2GRAY)
        blurred = cv2.GaussianBlur(gray, ksize: (5, 5), sigmaX: 0)
        _, binary_image = cv2.threshold(blurred, self.threshold_slider.value(), maxval: 255, cv2.THRESH_BINARY)
        self.display_image(binary_image)

1 usage
def display_image(self, image):
    h, w = image.shape
    q_image = QImage(image.data, w, h, w, QImage.Format_Grayscale8)
    pixmap = QPixmap.fromImage(q_image)
    self.image_label.setPixmap(pixmap.scaled(self.image_label.size(), Qt.KeepAspectRatio))

1 usage
```

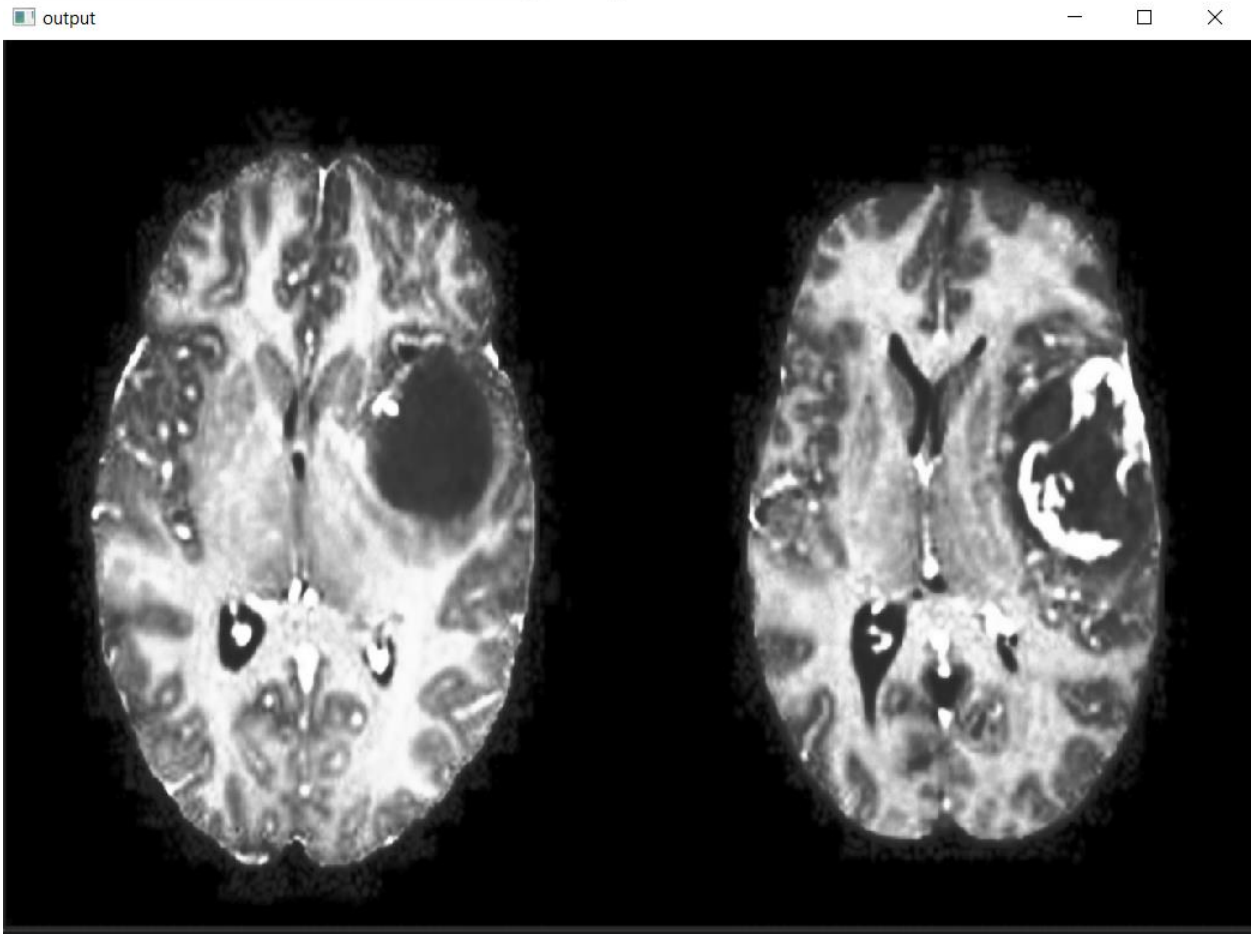
```
1 usage
def plot_histogram(self):
    if self.image is not None:
        gray = cv2.cvtColor(self.image, cv2.COLOR_BGR2GRAY)
        hist = cv2.calcHist( images: [gray], channels: [0], mask: None, histSize: [256], ranges: [0, 256])
        plt.plot( *args: hist, color='black')
        plt.xlabel('Pixel Intensity')
        plt.ylabel('Frequency')
        plt.title('Histogram')
        plt.show()
```

```
    app = QApplication(sys.argv)
    window = ImageProcessor()
    window.setGeometry(100, 100, 800, 600)
    window.show()

    image = cv2.imread('tumor2.jpg')
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred_image = cv2.GaussianBlur(gray_image, ksize: (5, 5), sigmaX: 0)
    enhanced_image = cv2.equalizeHist(blurred_image)
    edges = cv2.Canny(blurred_image, 50, 50)

    cv2.namedWindow( winname: "Edge detection", cv2.WINDOW_NORMAL)
    edge = cv2.resize(edges, dsize: (1280, 960))
    cv2.imshow( winname: "Edge detection", edge)

    cv2.namedWindow( winname: "output", cv2.WINDOW_NORMAL)
    output = cv2.resize(enhanced_image, dsize: (1280, 960))
    cv2.imshow( winname: "output", output)

    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Original Image:

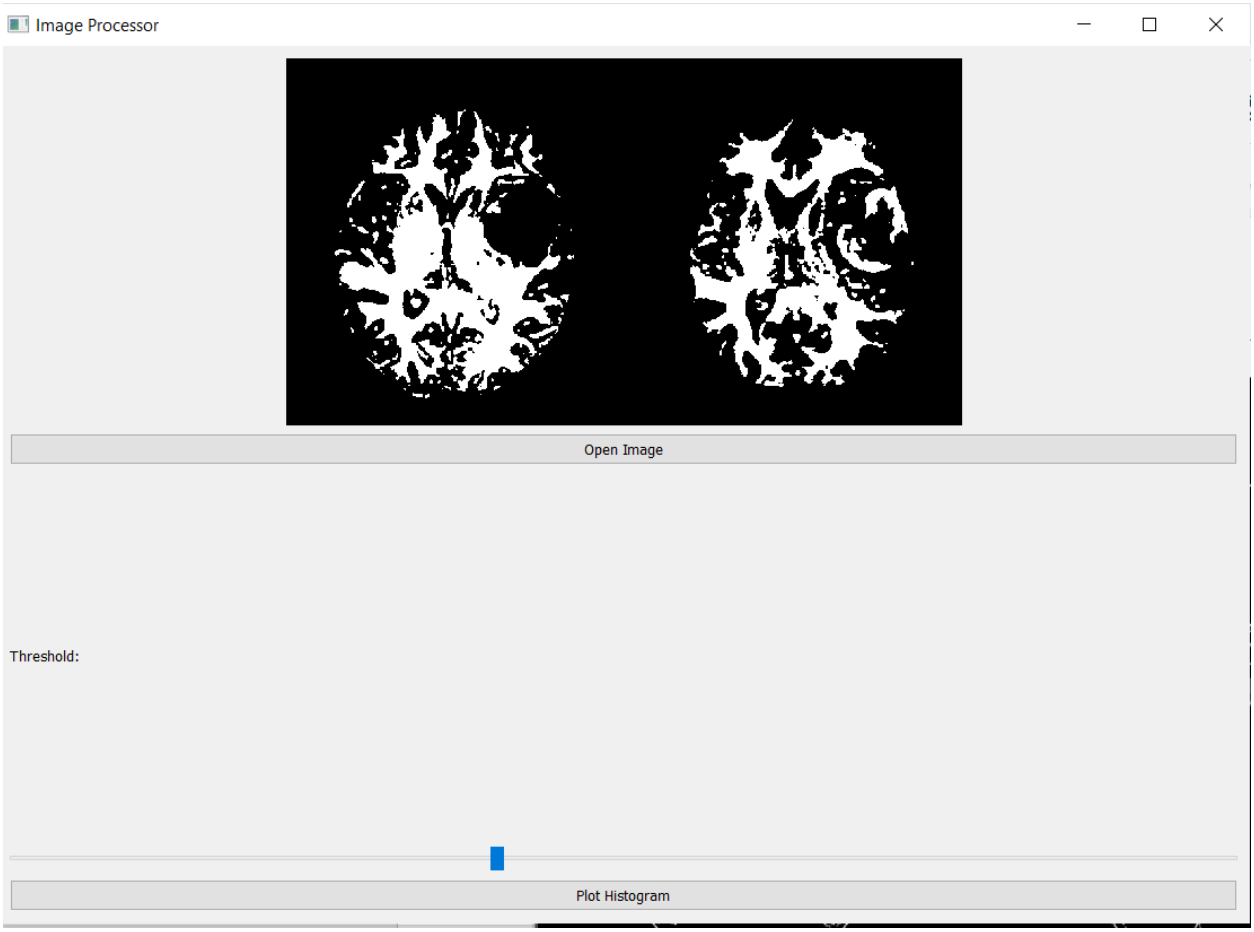


Enhanced Image:



Edge Detection Image:

Image Threshold and Histogram



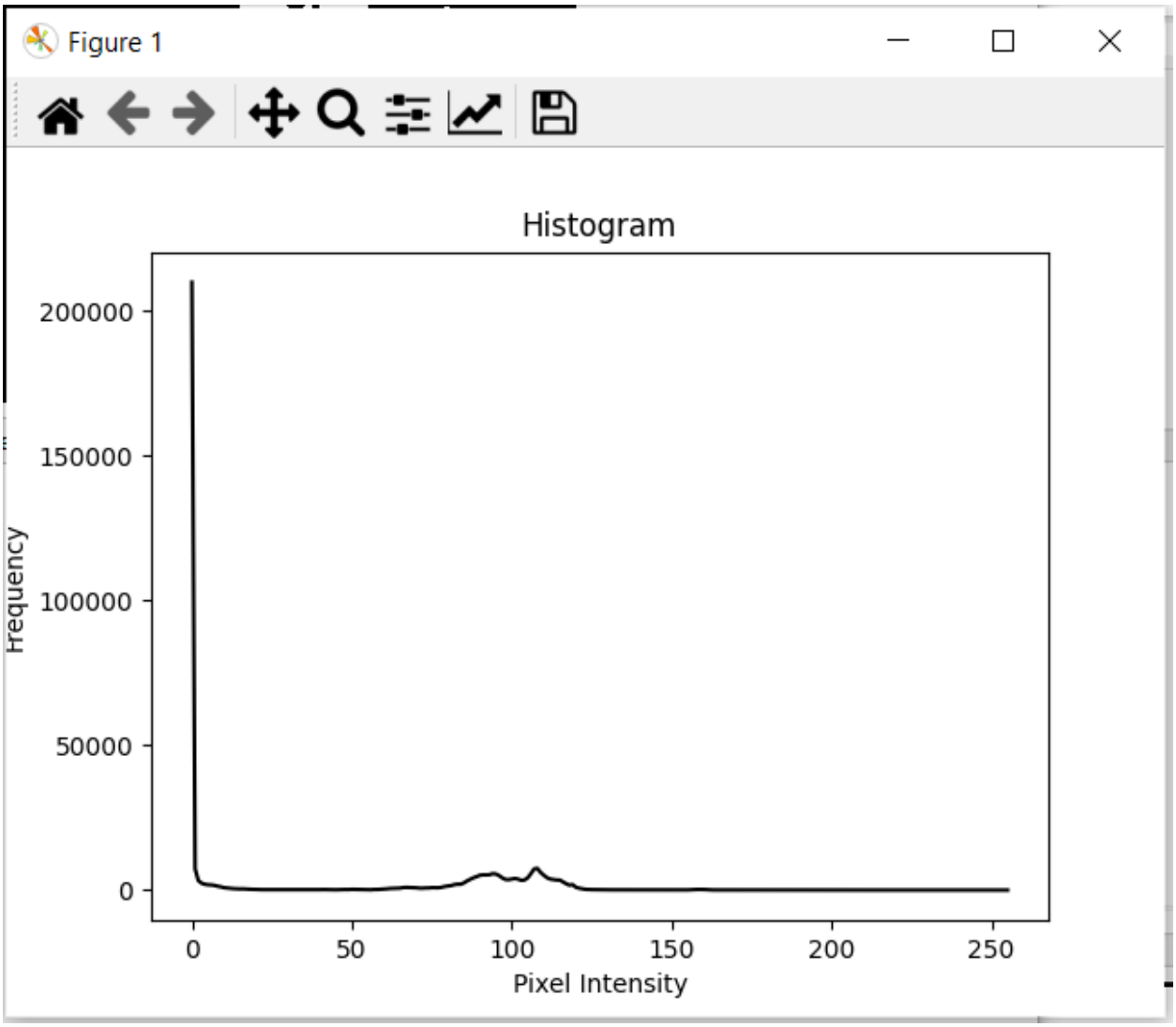


Figure 1
Histogram
200000
150000
100000
50000
0
Frequency
0
50
100
150
200
250
Pixel Intensity