```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

def apply_threshold(image):
    _, binary_image = cv2.threshold(image, 127, 255,
cv2.THRESH_BINARY)
    return binary_image

def apply_erosion(image, kernel_size=3, iterations=1):
    kernel = np.ones((kernel_size, kernel_size), np.uint8)
    return cv2.erode(image, kernel, iterations=iterations)

def apply_dilation(image, kernel_size=3, iterations=1):
    kernel = np.ones((kernel_size, kernel_size), np.uint8)
    return cv2.dilate(image, kernel, iterations=iterations)

def apply_opening(image, kernel_size=3):
    kernel = np.ones((kernel_size, kernel_size), np.uint8)
    return cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

def apply_closing(image, kernel_size=3):
    kernel = np.ones((kernel_size, kernel_size), np.uint8)
    return cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)

def apply_gradient(image, kernel_size=3):
    kernel = np.ones((kernel_size, kernel_size), np.uint8)
    return cv2.morphologyEx(image, cv2.MORPH_GRADIENT, kernel)

# Load image
gray_image = cv2.imread('download.jpeg', cv2.IMREAD_GRAYSCALE)
binary_image = apply_threshold(gray_image)

# Apply morphological operations
erosion = apply_erosion(binary_image)
dilation = apply_dilation(binary_image)
opening = apply_opening(binary_image)
closing = apply_closing(binary_image)
gradient = apply_gradient(binary_image)

# Display results
fig, axs = plt.subplots(2, 3, figsize=(12, 8))
axs[0, 0].imshow(gray_image, cmap='gray')
axs[0, 0].set_title("Original Image")
axs[0, 1].imshow(binary_image, cmap='gray')
axs[0, 1].set_title("Binary Image")
axs[0, 2].imshow(erosion, cmap='gray')
axs[0, 2].set_title("Erosion")
axs[1, 0].imshow(dilation, cmap='gray')
axs[1, 0].set_title("Dilation")
axs[1, 1].imshow(opening, cmap='gray')
axs[1, 1].set_title("Opening")
axs[1, 2].imshow(closing, cmap='gray')
axs[1, 2].set_title("Closing")
```

```
for ax in axs.flat:
    ax.axis("off")
plt.show()
```

```
Matplotlib is building the font cache; this may take a moment.
```