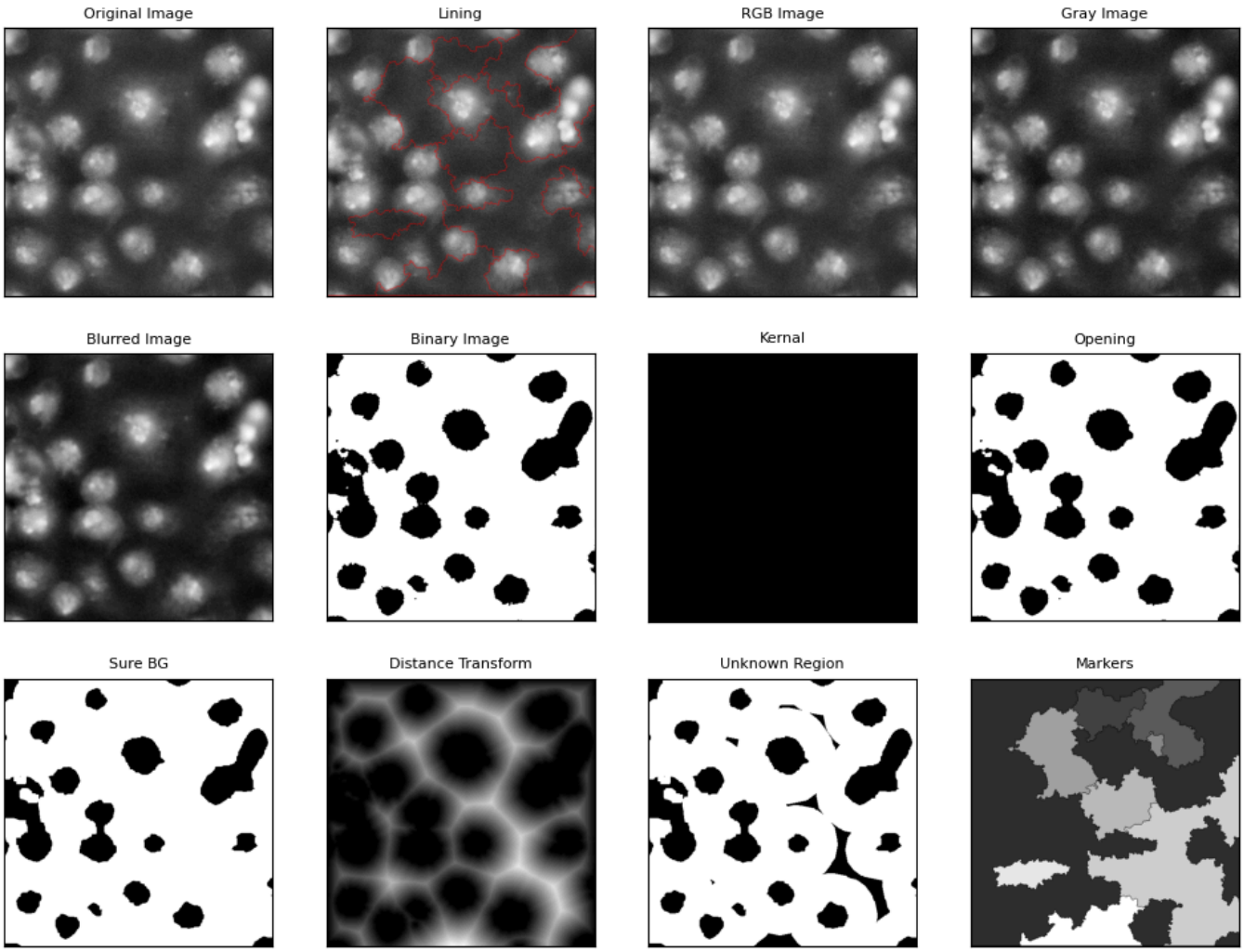
CSC_3141_Assignment_07_S19355

1.

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

# Load the image
img = cv2.imread(r'cell_segmentation.jpg')
image = img.copy()
imgRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply Gaussian blur
blurred = cv2.GaussianBlur(gray, (5, 5), 0)

# Apply Otsu's thresholding
_, binary = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)

# Remove noise
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(binary, cv2.MORPH_OPEN, kernel, iterations=2)

# Sure background area
sure_bg = cv2.dilate(opening, kernel, iterations=3)

# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening, cv2.DIST_L2, 5)
_, sure_fg = cv2.threshold(dist_transform, 0.7 * dist_transform.max(),
255, 0)

# Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg, sure_fg)
```

```
# Marker labelling
_, markers = cv2.connectedComponents(sure_fg)

# Add one to all labels so that sure background is not 0, but 1
markers = markers + 1

# Mark the region of unknown with zero
markers[unknown == 255] = 0

# Apply the watershed algorithm
markers = cv2.watershed(image, markers)
image[markers == -1] = [255, 0, 0]

# Count the number of cells
# The unique markers correspond to different cells
unique_markers = len(np.unique(markers)) - 2  # Subtract 2 for background
and boundary

# Display the results
plt.figure(figsize=(12, 9))

titles = ['Original Image', 'Lining', 'RGB Image', 'Gray Image', 'Blurred
Image', 'Binary Image', 'Kernal', 'Opening', 'Sure BG', 'Distance
Transform', 'Unknown Region', 'Markers']
images = [img, image, imgRGB, gray, blurred, binary, kernel, opening,
sure_bg, dist_transform, unknown, markers]

for i in range(len(titles)):
    plt.subplot(3, 4, i+1)
    plt.title(titles[i], fontsize = 8)
    plt.xticks([]), plt.yticks([])
    plt.imshow(images[i], 'gray')

print(f'Number of cells: {unique_markers}')
```

```
Number of cells: 9
```



2.

- Noise Reduction : Improves accuracy by reducing false markers.
- Smoothing : Enhances object boundaries for better segmentation.
- Edge Detection : Sharpens boundaries, helping watershed to detect precise edges.
- Gradient Computation : Provides better marker placement by highlighting intensity changes.
- Morphological Operations : Refines shapes, separating touching objects effectively.

3.

- marker-oriented Watershed : Makes use of markers to avoid over-segmentation, which is important when handling objects.
- Distance Transform : By locating an object's center, it can assist in separating overlapping objects.
- Pre-processing : For improved outcomes, combine with methods like thresholding, morphological operations, and Gaussian blurring.

<u>Uses in Medical Diagnosis and Treatment</u>

- ★ Tumor Segmentation : Provides precise boundaries for more effective treatment planning.
- ★ Cell counting : Assists in cell analysis by separating individual cells in microscopy images.
- ★ Organ Segmentation : Helps distinguish between organs in scans for accurate diagnosis.
- ★ Lesion Detection : Recognizes and classifies lesions to enable precise monitoring and diagnosis.