

# Import Required Libraries

Import the necessary libraries, including OpenCV, NumPy, and Matplotlib.

```
In [34]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

# Load and Preprocess Image

Load the image and convert it to grayscale if necessary.

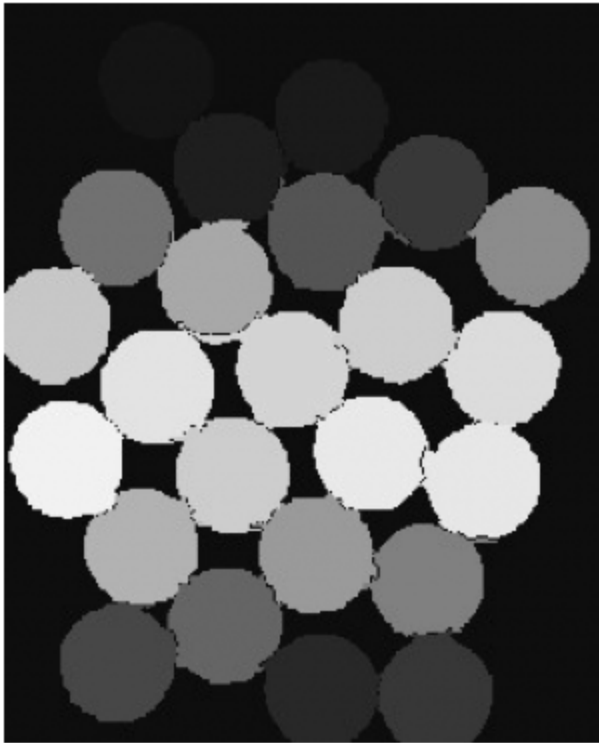
```
In [39]: # Load and Preprocess Image

# Load the image from file
base_path = r'C:\Users\danis\OneDrive\Documents\Repos\CV_PE2\Practicals\P3\data'
fig3 = cv2.imread( rf"{base_path}\fig3.jpg", cv2.IMREAD_GRAYSCALE) # Replace with

# Check if the image was loaded successfully
if fig3 is None:
    raise ValueError("Image not found or unable to load.")

# Display the original grayscale image
plt.imshow(fig3, cmap='gray')
plt.title('Original Grayscale Image')
plt.axis('off')
plt.show()
```

Original Grayscale Image

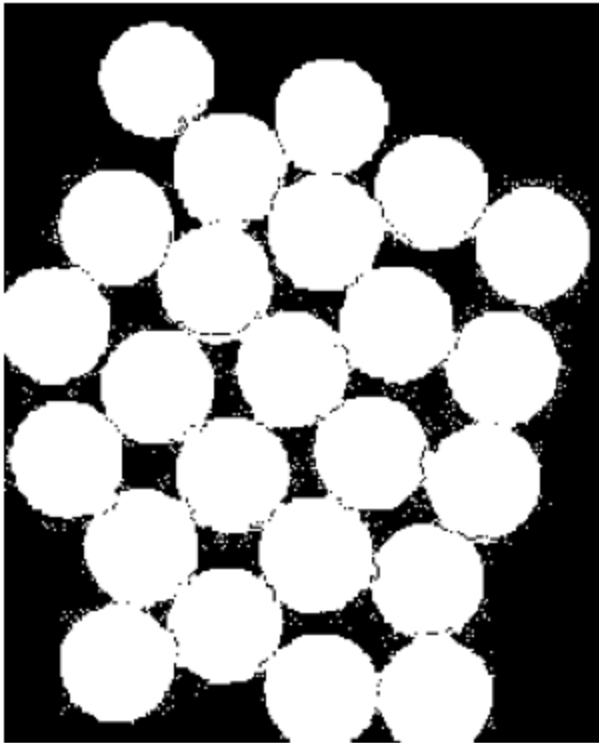


## Applying Binary Thresholding

```
In [47]: # Apply binary threshold
ret, binary_thresh = cv2.threshold(fig3, 18, 255, cv2.THRESH_BINARY)

# Display the binary threshold image
plt.imshow(binary_thresh, cmap='gray')
plt.title('Binary Threshold Image')
plt.axis('off')
plt.show()
```

Binary Threshold Image

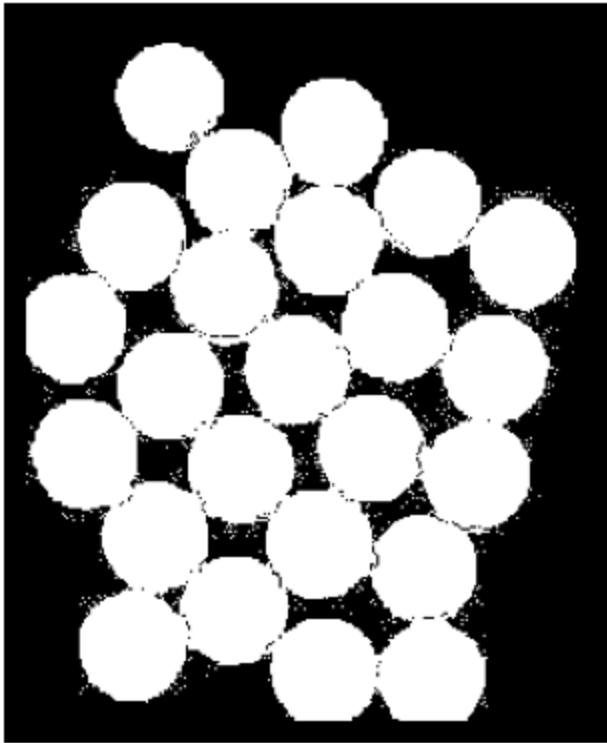


## Expanding Since the Borders are Close by

```
In [69]: # Expand the borders of the image by 10 pixels on each side and fill with 0s
expanded_image = cv2.copyMakeBorder(binary_thresh, 10, 10, 10, 10, cv2.BORDER_CONSTANT)

# Display the expanded image
plt.imshow(expanded_image, cmap='gray')
plt.title('Expanded Image with Borders')
plt.axis('off')
plt.show()
```

Expanded Image with Borders



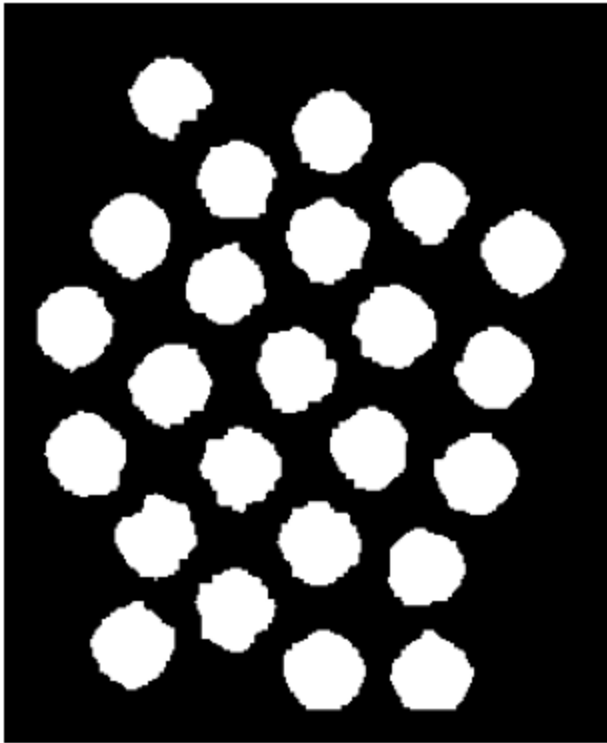
## Eroding

```
In [83]: # Define the kernel for erosion
kernel = np.ones((11, 11), np.uint8)

# Apply erosion
eroded = cv2.erode(expanded_image, kernel, iterations=1)

# Display the eroded image
plt.imshow(eroded, cmap='gray')
plt.title('Eroded Image')
plt.axis('off')
plt.show()
```

Eroded Image



```
In [84]: # Find contours in the eroded image
contours, _ = cv2.findContours(eroded, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Count the number of contours
ball_count = len(contours)

print(f"Number of balls: {ball_count}")
```

Number of balls: 24