Upload file:

In [ ]:

```
# To move all the .png files into the folder p1

from google.colab import drive
drive.mount('/content/drive')

import os
import shutil

# Define the source directory (current directory in this case)
source_dir = '/content/'

# Define the target directory
target_dir = '/content/images/'

# Ensure the target directory exists
if not os.path.exists(target_dir):
    os.makedirs(target_dir)

# Loop through all files in the source directory
for file_name in os.listdir(source_dir):
    if file_name.endswith('.png'):  # Check for ".png" extension
        source_file = os.path.join(source_dir, file_name)
        target_file = os.path.join(target_dir, file_name)

        # Move file
        shutil.move(source_file, target_file)
        print(f'Moved {file_name} to {target_dir}')
```

Preprocessing images and extracting features - Verison 1:

In [ ]:

```
# # Version 1.1
# import cv2
# import numpy as np
# import matplotlib.pyplot as plt
# from scipy.ndimage import label, find_objects
# from scipy.spatial.distance import cdist
# from skimage.filters import threshold_otsu

# """
# Calculating perimeter
# """
# def compute_perimeter(region_mask):
```

```
#     """Approximate the perimeter of the region using edge detection."""
#     from scipy.ndimage import binary_erosion
#     from numpy import logical_xor

#     structure = np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]])
#     eroded_image = binary_erosion(region_mask, structure)

#     boundary = logical_xor(region_mask, eroded_image)
#     return np.sum(boundary)

# """
# feature extraction
# """
# def process_image(image_path, image_id):
#     # Load image
#     image = plt.imread(image_path)  # (h, w, 3) or (h, w)
#     plt.figure()
#     plt.imshow(image)
#     plt.title('Original Image')
#     plt.colorbar()
#     plt.show()

#     # Convert to grayscale if it is a color image
#     if len(image.shape) == 3 and image.shape[2] == 3:
#         image = np.mean(image, axis=2)
#         print(image.shape)
#         plt.figure()
#         plt.imshow(image, cmap='gray')
#         plt.title('grayscale')
#         plt.colorbar()
#         plt.show()

#     # Normalize if dtype is uint8
#     if image.dtype == np.uint8:
#         image = image.astype(float) / 255.0

#     # Thresholding
#     threshold = 0.5
#     binary_image = (image > threshold).astype(np.uint8)
#     plt.figure()
#     plt.imshow(binary_image, cmap='gray')
#     plt.title(f'Binary Image (threshold = {threshold})')
#     plt.colorbar()
#     plt.show()

#     # Label connected components
#     labeled_image, num_labels = label(binary_image)

#     features = []
#     for region_label in range(1, num_labels + 1):
#         region_mask = (labeled_image == region_label)
#         region_indices = np.argwhere(region_mask)

#         # Extract features
#         area = np.sum(region_mask)
#         perimeter = compute_perimeter(region_mask)
#         centroid_y, centroid_x = region_indices.mean(axis=0)
#         features.append((image_id, area, perimeter, centroid_x, centroid_y))


#     # Summary statistics
#     summary = {
#         "image_id": image_id,
#         "cell_count": len(features),
#         "avg_size": np.mean([f[1] for f in features]) if features else 0,
#         "avg_perimeter": np.mean([f[2] for f in features]) if features else 0,
#     }

#     return features, summary


# # usage:
# features, summary = process_image('/content/images/25_01_2024_13.png', 1)
# print(summary)
# print(features)
# features, summary = process_image('/content/images/26_01_2024_00.png', 1)
# print(summary)
# print(features)
# features, summary = process_image('/content/images/26_01_2024_04.png', 1)
# print(summary)
# print(features)
```

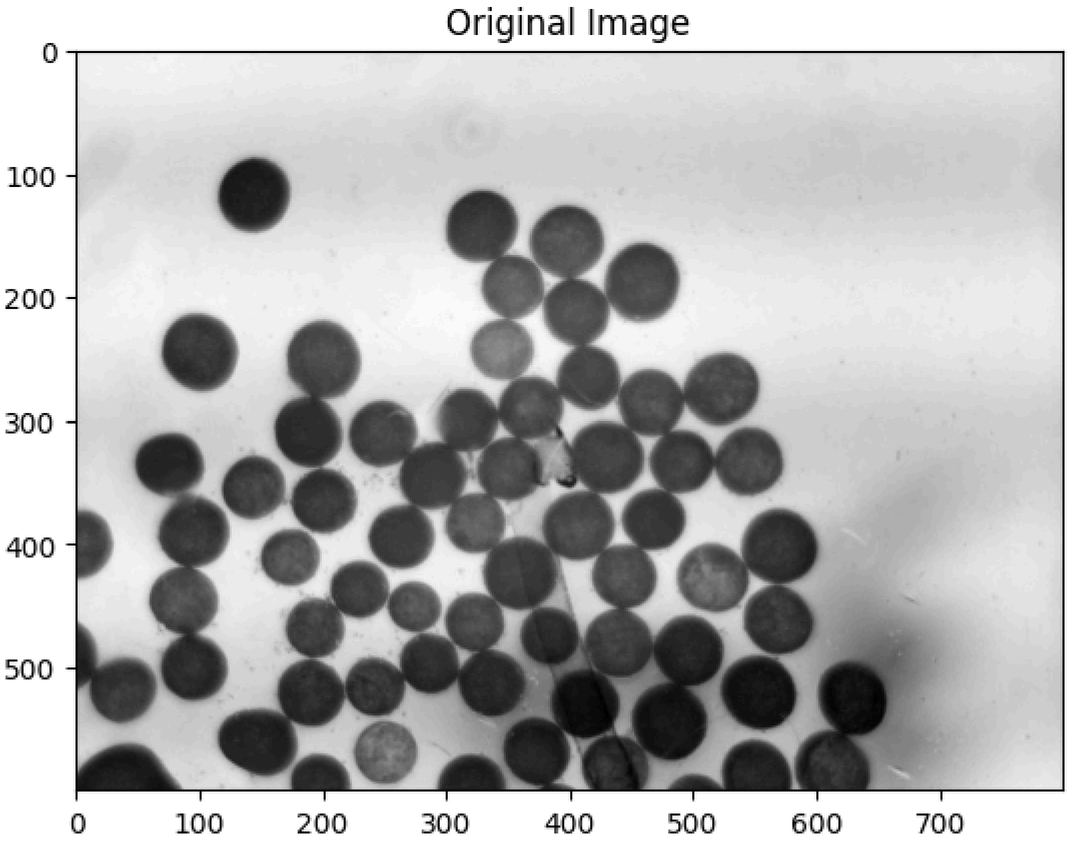
Preprocessing images and extracting features - Verison 2:

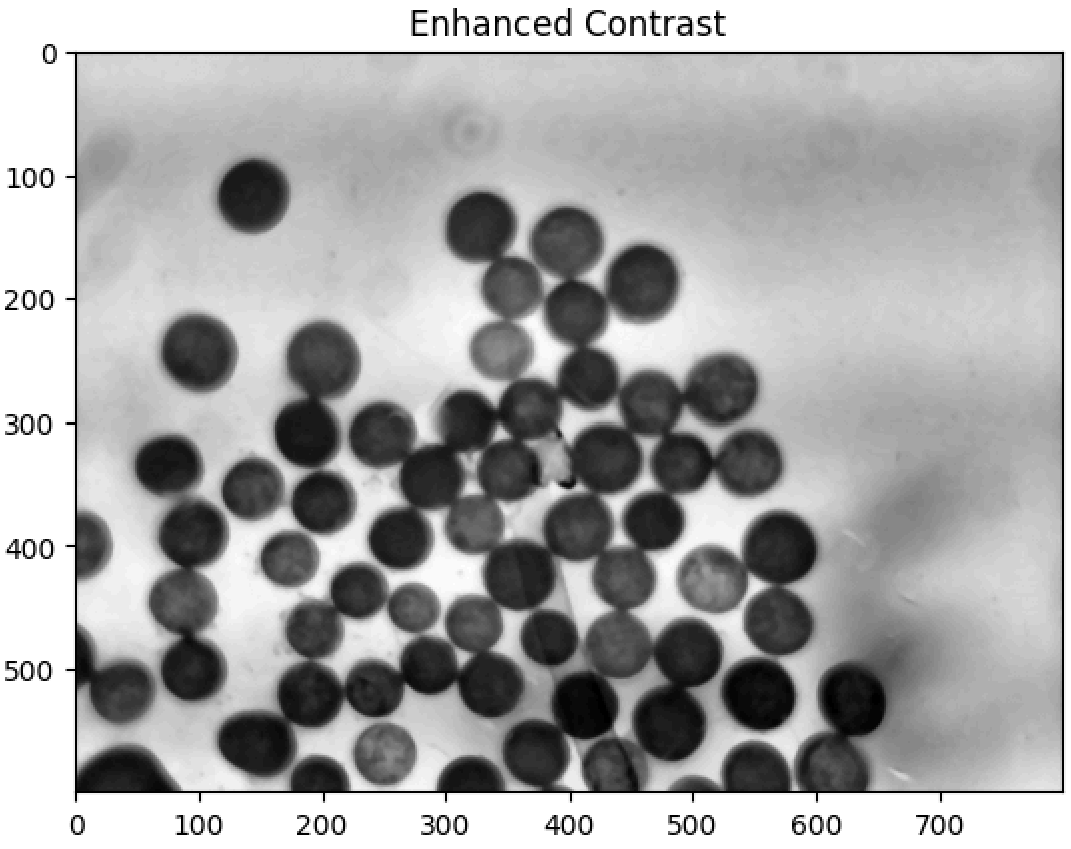
In [23]:

```
import os
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist

from scipy.ndimage import label, maximum_filter, binary_erosion
```

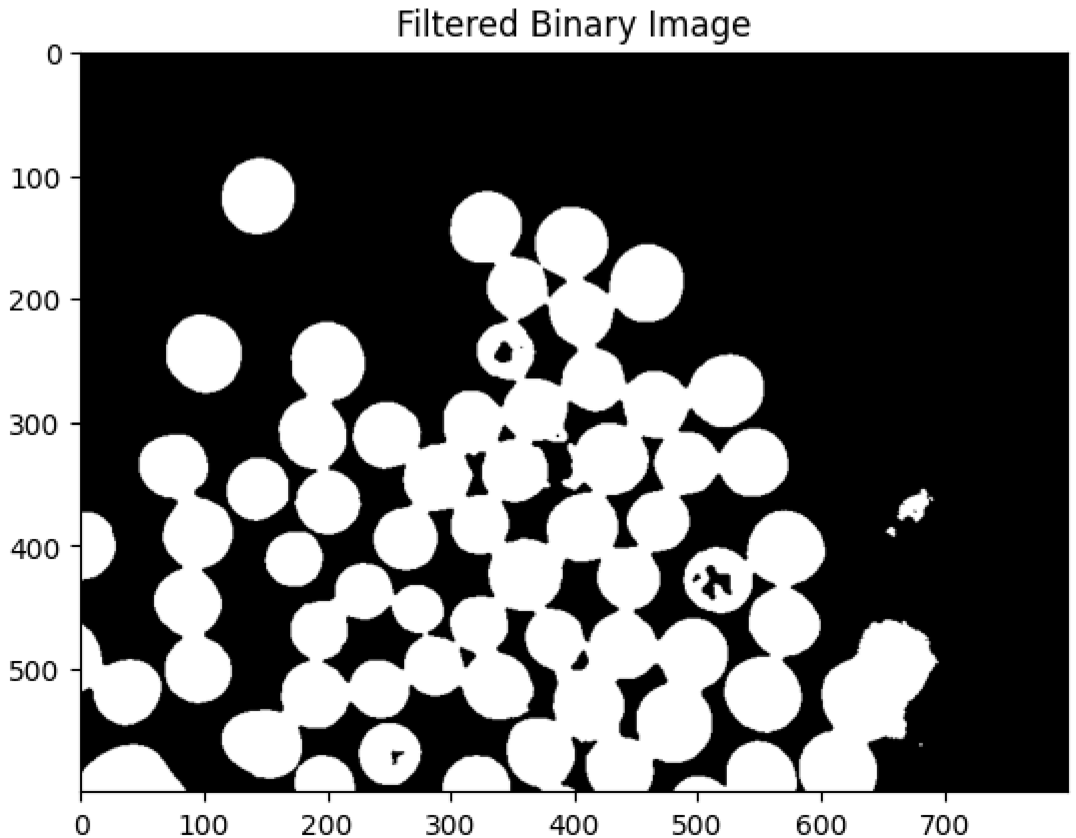
```python
from numpy import logical_xor

# DataFrame
cell_data_list = []
image_data_list = []


def compute_perimeter(region_mask):
    """Approximate the perimeter of the region using edge detection."""

    # Binary corrosion of regions using predefined structural elements
    structure = np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]])
    eroded_image = binary_erosion(region_mask, structure)
    # User logical XOR because direct subtraction is not applicable in Boolean types
    boundary = logical_xor(region_mask, eroded_image)
    return np.sum(boundary)

def compute_cell_centroids(watershed_labels, num_cells):
    """Compute centroid (X, Y) for each cell."""
    centroids = []
    for cell_label in range(1, num_cells + 1):
        region_mask = (watershed_labels == cell_label).astype(np.uint8)
        moments = cv2.moments(region_mask)
        if moments["m00"] != 0:  # Avoid division by zero
            cx = int(moments["m10"] / moments["m00"])  # X coordinate
            cy = int(moments["m01"] / moments["m00"])  # Y coordinate
            centroids.append({"label": cell_label, "x": cx, "y": cy})
    return centroids

def compute_local_density(centroids, radius=50):
    """Compute local cell density (number of neighbors within a given radius)."""
    if not centroids:  # Check if centroids list is empty
        return  # If empty, skip density calculation

    positions = np.array([[cell["x"], cell["y"]] for cell in centroids])

    if positions.shape[0] < 2:
        for cell in centroids:
            cell["local_density"] = 0  # No neighbors, density = 0
        return

    distances = cdist(positions, positions)  # Compute pairwise distances
    local_densities = np.sum(distances < radius, axis=1) - 1  # Count neighbors (excluding self)

    for i, cell in enumerate(centroids):
        cell["local_density"] = local_densities[i]
```

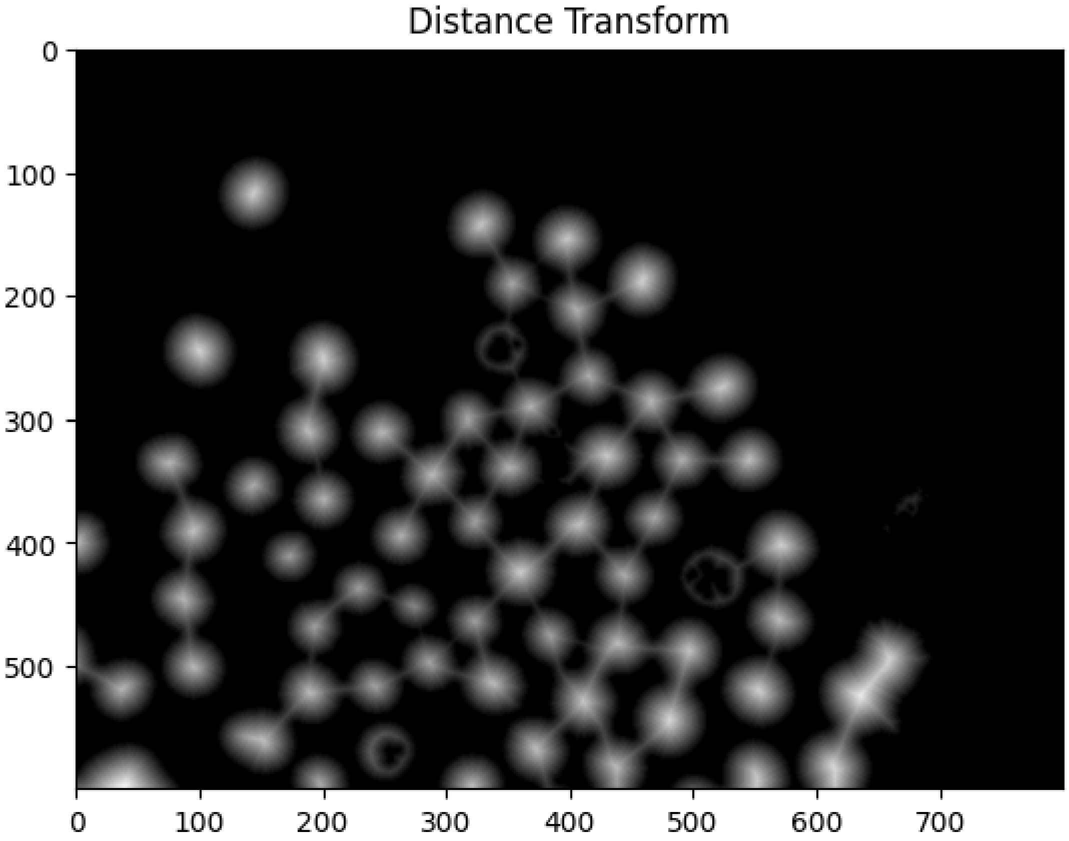
```python
def distance_tranform(num_labels, stats, filtered_binary, labels, area_t):
    # Preserve small connected domains
    # Retain only small connected components
    for i in range(1, num_labels):  # Skip background
        area = stats[i, cv2.CC_STAT_AREA]
        if area < area_t:  # set maximum area threshold
            filtered_binary[labels == i] = 255

    plt.figure()
    plt.imshow(filtered_binary, cmap='gray')
    plt.title('Filtered Binary Image')
    plt.show()

    # 3. Distance Transform
    dist_transform = cv2.distanceTransform(filtered_binary, cv2.DIST_L2, 5)
    plt.figure()
    plt.imshow(dist_transform, cmap='gray')
    plt.title('Distance Transform')
    plt.show()

    # 4. Detect Local Maxima as Seeds
    local_max = maximum_filter(dist_transform, size=30)  # adjust window size
    maxima = (dist_transform == local_max) & (dist_transform > 0.3 * dist_transform.max())  # Enhanced threshold filtering
    labeled_maxima, num_cells = label(maxima)  # Marked seed point

    return labeled_maxima, num_cells


def count_cells_from_distance_transform(image_path):
    """Main function to process image and extract cell features."""
    # Load Image
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    image_name = image_path.split('/')[-1]
    plt.figure()
    plt.imshow(image, cmap='gray')
    plt.title('Original Image')
    plt.show()

    # 1. Preprocessing
    # Noise Reduction
    image_filtered = cv2.medianBlur(image, 5)

    # Contrast Enhancement
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    enhanced_image = clahe.apply(image_filtered)
    plt.figure()
    plt.imshow(enhanced_image, cmap='gray')
    plt.title('Enhanced Contrast')
    plt.show()

    # Threshold Segmentation
    _, binary_image = cv2.threshold(enhanced_image, 127, 255, cv2.THRESH_BINARY_INV)

    # Morphological Operations - Noise Removal
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    binary_image = cv2.morphologyEx(binary_image, cv2.MORPH_OPEN, kernel)

    # Connected domain analysis to remove large areas
    num_labels, labels, stats, _ = cv2.connectedComponentsWithStats(binary_image)

    filtered_binary = np.zeros_like(binary_image)
    labeled_maxima, num_cells = distance_tranform(num_labels, stats, filtered_binary, labels, 120000)

    area_t = 50000
    while num_cells < 10 and area_t >= 10000:
      print(f"num_cells < 10: Reapplying area filtering with stricter threshold {area_t}")

      # Creates a new binary image to store the filtered region
      filtered_binary = np.zeros_like(binary_image)
      labeled_maxima, num_cells = distance_tranform(num_labels, stats, filtered_binary, labels, area_t)

      area_t -= 10000

    # Watershed Algorithm for Segmentation
    markers = np.zeros_like(filtered_binary, dtype=np.int32)

    # Iterate through each seed point, assigning unique tags
    seed_indices = np.argwhere(labeled_maxima)  # Get the seed point coordinates
    for idx, (y, x) in enumerate(seed_indices, start=1):
        markers[y, x] = idx

    watershed_labels = cv2.watershed(cv2.cvtColor(filtered_binary, cv2.COLOR_GRAY2BGR), markers)

    # 2. Compute Features for Each Cell
    cell_features = []
    centroids = compute_cell_centroids(watershed_labels, num_cells)
    compute_local_density(centroids, radius=50)

    for centroid in centroids:
        region_mask = (watershed_labels == centroid["label"]).astype(np.uint8)
        area = np.sum(region_mask)  # area
        perimeter = compute_perimeter(region_mask)  # perimeter

        cell_features.append({
            "Image Name": image_name,
            "Cell ID": centroid["label"],
            "X Position": centroid["x"],
            "Y Position": centroid["y"],
            "Area": area,
            "Perimeter": perimeter,
            "Local density": centroid["local_density"]
        })

    # Append cell data to list
    cell_data_list.extend(cell_features)


    # 3. Compute Image-Level Summary Statistics
    avg_area = np.mean([cell["Area"] for cell in cell_features])
    avg_perimeter = np.mean([cell["Perimeter"] for cell in cell_features])
    avg_local_density = np.mean([cell["Local density"] for cell in cell_features])


    image_summary = {
        "Image Name": image_name,
        "Total Cells": num_cells,
        "Average Area": avg_area,
        "Average Perimeter": avg_perimeter,
        "Average Local Density": avg_local_density
    }

    image_data_list.append(image_summary)

    # 4. Draw final image with labels
    image_colored = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
    for cell_label in range(1, num_cells + 1):
        coords = np.argwhere(labeled_maxima == cell_label)
        if len(coords) == 0:  # Prevent invalid seed points
            continue
        y, x = coords[0]  # Select a coordinate for the seed point
        cv2.circle(image_colored, (x, y), 5, (0, 255, 0), -1)  # draw seed point
        cv2.putText(image_colored, str(cell_label), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)

    plt.figure(figsize=(10, 10))
    plt.imshow(cv2.cvtColor(image_colored, cv2.COLOR_BGR2RGB))
    plt.title(f'Total Cells Detected: {num_cells}')
    plt.show()
    # Print each cell's features
    for cell in cell_features:
      print(f"Cell {cell['Cell ID']}: Area = {cell['Area']}, Perimeter = {cell['Perimeter']}, "
                    f"Position = ({cell['X Position']}, {cell['Y Position']}), Local Density = {cell['Local density']}")
    return num_cells, cell_features


# Examples:
num_cells, cell_features = count_cells_from_distance_transform('/content/images/25_01_2024_13.png')
num_cells, cell_features = count_cells_from_distance_transform('/content/images/25_01_2024_16.png')
num_cells, cell_features = count_cells_from_distance_transform('/content/images/26_01_2024_04.png')
```

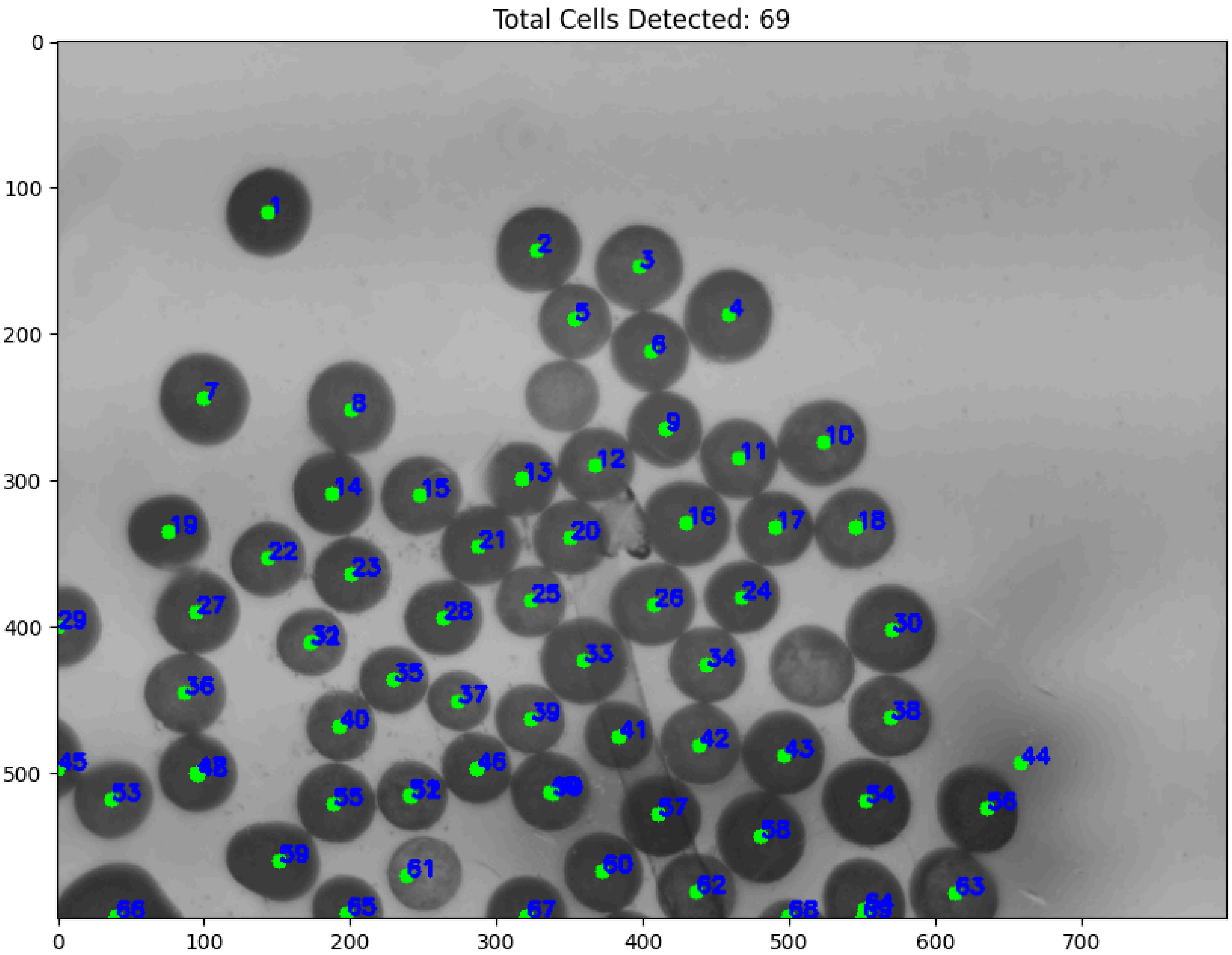




Enhanced Contrast
0
100
200
300
400
500
0
100
200
300
400
500
600
700

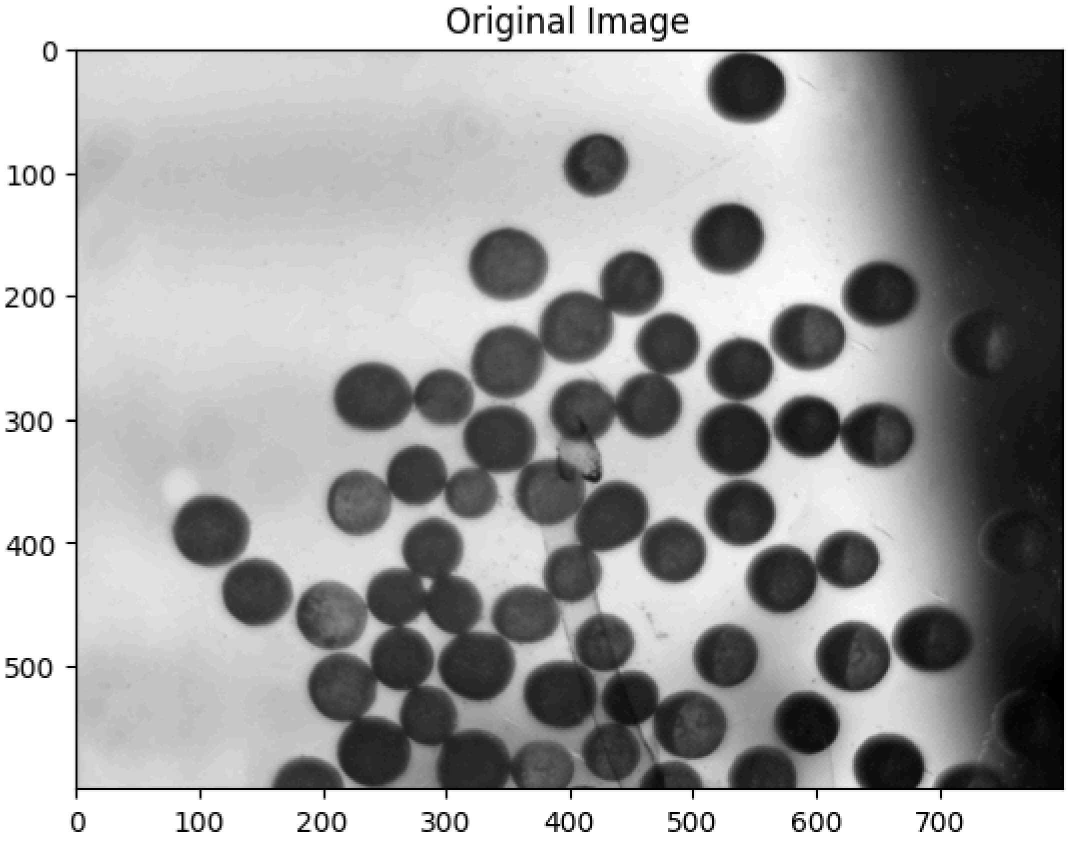

Filtered Binary Image
0
100
200
300
400
500
0
100
200
300
400
500
600
700

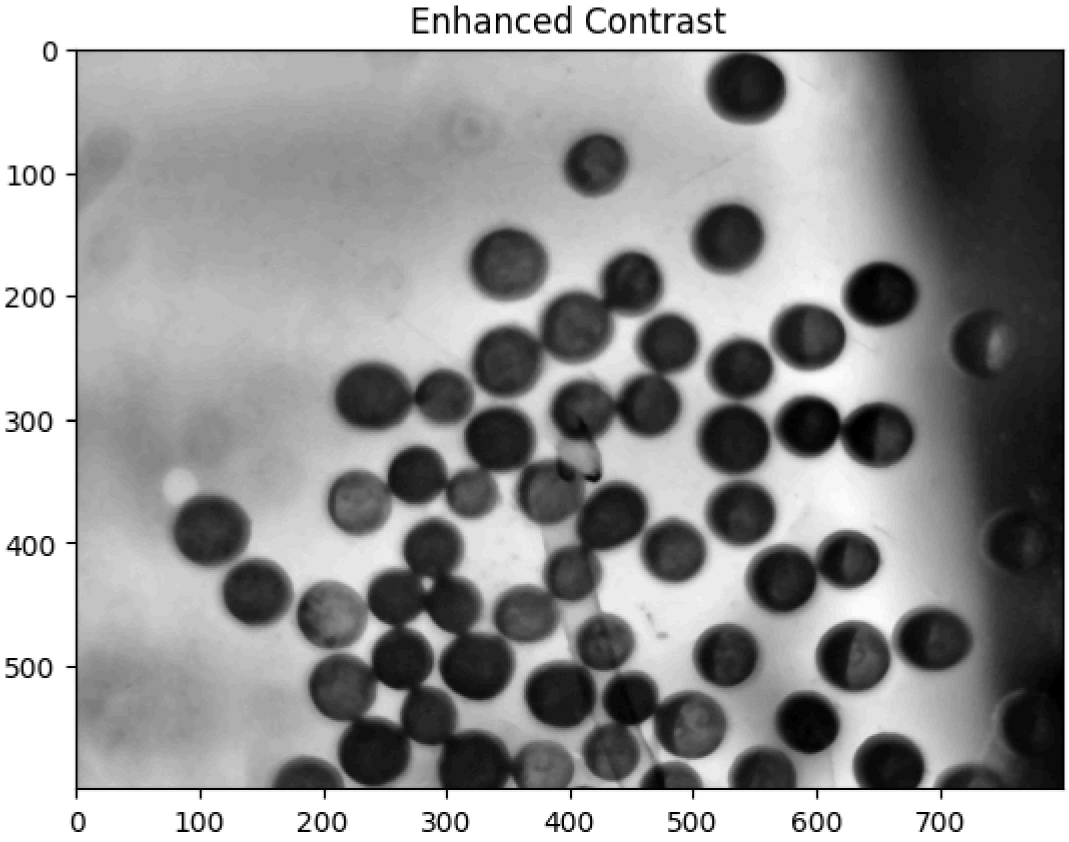

Distance Transform
0
100
200
300
400
500
0
100
200
300
400
500
600
700

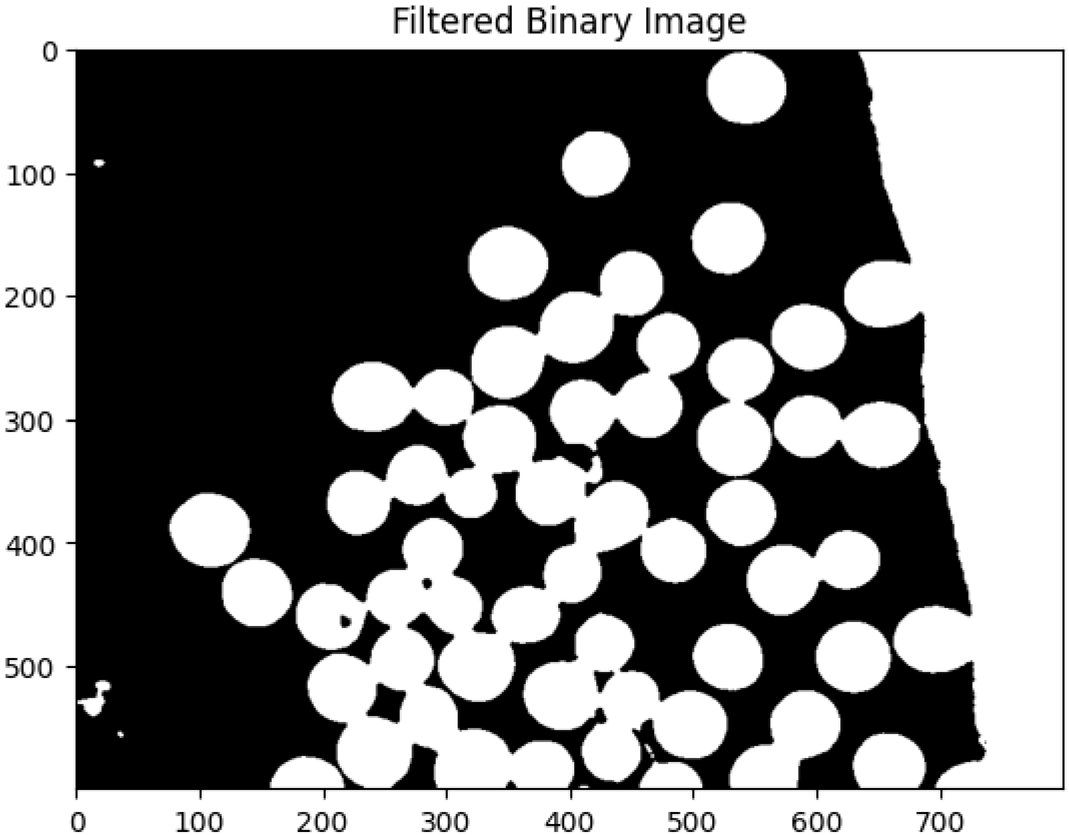


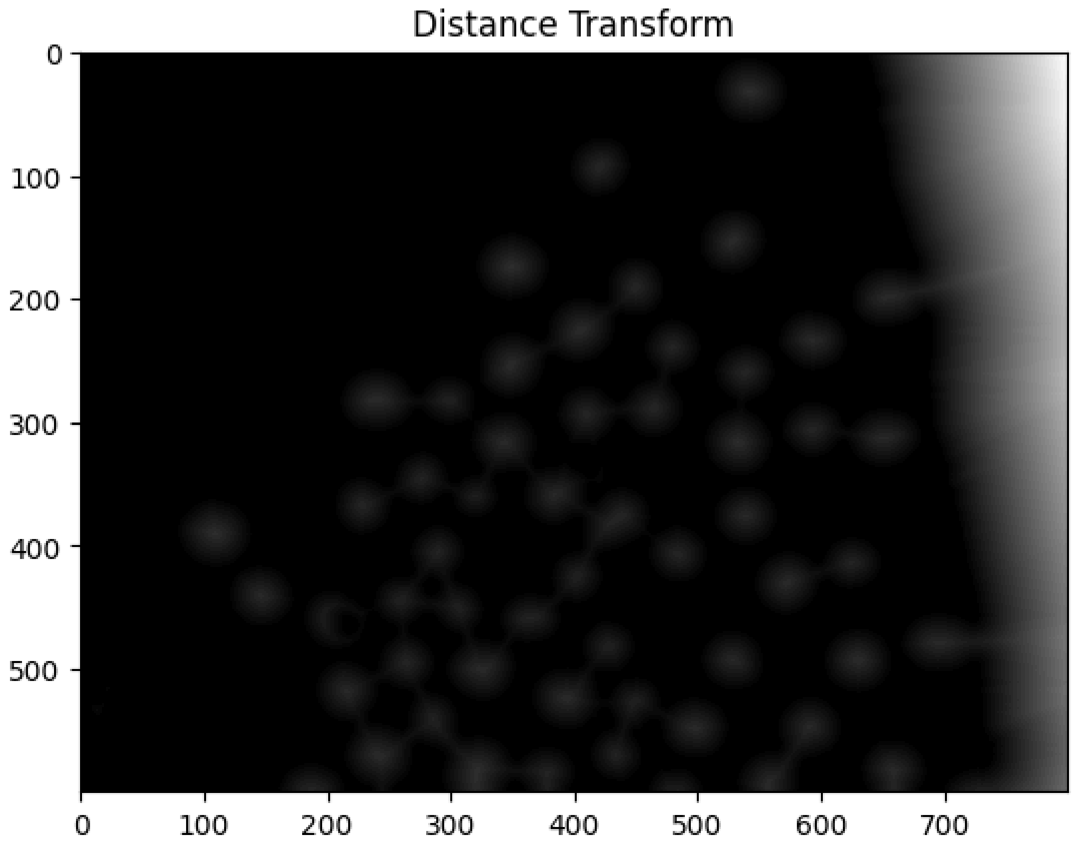
Total Cells Detected: 69

```
Cell 1: Area = 2883, Perimeter = 169, Position = (144, 117), Local Density = 0
Cell 2: Area = 2662, Perimeter = 164, Position = (328, 142), Local Density = 0
Cell 3: Area = 2671, Perimeter = 166, Position = (397, 155), Local Density = 0
Cell 4: Area = 2943, Perimeter = 173, Position = (459, 187), Local Density = 0
Cell 5: Area = 3232, Perimeter = 258, Position = (352, 205), Local Density = 0
Cell 6: Area = 2291, Perimeter = 168, Position = (406, 212), Local Density = 1
Cell 7: Area = 3047, Perimeter = 173, Position = (99, 245), Local Density = 0
Cell 8: Area = 2976, Perimeter = 176, Position = (200, 251), Local Density = 0
Cell 9: Area = 2102, Perimeter = 155, Position = (415, 266), Local Density = 2
Cell 10: Area = 2761, Perimeter = 167, Position = (523, 274), Local Density = 0
Cell 11: Area = 2334, Perimeter = 164, Position = (466, 286), Local Density = 0
Cell 12: Area = 4312, Perimeter = 331, Position = (368, 268), Local Density = 1
Cell 13: Area = 2012, Perimeter = 148, Position = (317, 301), Local Density = 0
Cell 14: Area = 2453, Perimeter = 161, Position = (187, 309), Local Density = 0
Cell 15: Area = 2296, Perimeter = 150, Position = (248, 311), Local Density = 1
Cell 16: Area = 2981, Perimeter = 196, Position = (427, 331), Local Density = 0
Cell 17: Area = 2140, Perimeter = 151, Position = (491, 333), Local Density = 0
Cell 18: Area = 2407, Perimeter = 154, Position = (545, 333), Local Density = 0
Cell 19: Area = 2345, Perimeter = 158, Position = (75, 336), Local Density = 0
Cell 20: Area = 2258, Perimeter = 159, Position = (351, 338), Local Density = 1
Cell 21: Area = 1434, Perimeter = 137, Position = (280, 342), Local Density = 2
Cell 22: Area = 837, Perimeter = 98, Position = (302, 350), Local Density = 1
Cell 23: Area = 2011, Perimeter = 141, Position = (143, 354), Local Density = 0
Cell 24: Area = 1621, Perimeter = 155, Position = (200, 359), Local Density = 2
Cell 25: Area = 528, Perimeter = 71, Position = (200, 381), Local Density = 2
Cell 26: Area = 2981, Perimeter = 259, Position = (462, 374), Local Density = 0
Cell 27: Area = 5015, Perimeter = 569, Position = (357, 359), Local Density = 1
Cell 28: Area = 2579, Perimeter = 162, Position = (406, 385), Local Density = 0
Cell 29: Area = 2528, Perimeter = 162, Position = (95, 390), Local Density = 0
Cell 30: Area = 2102, Perimeter = 145, Position = (263, 394), Local Density = 0
Cell 32: Area = 5317, Perimeter = 305, Position = (547, 414), Local Density = 0
Cell 33: Area = 804, Perimeter = 96, Position = (180, 404), Local Density = 3
Cell 34: Area = 813, Perimeter = 94, Position = (165, 418), Local Density = 1
Cell 35: Area = 2668, Perimeter = 164, Position = (361, 423), Local Density = 1
Cell 36: Area = 2087, Perimeter = 147, Position = (443, 427), Local Density = 0
Cell 37: Area = 1711, Perimeter = 130, Position = (229, 437), Local Density = 1
Cell 38: Area = 2316, Perimeter = 155, Position = (86, 446), Local Density = 1
Cell 39: Area = 308828, Perimeter = 7748, Position = (426, 242), Local Density = 2
Cell 40: Area = 2555, Perimeter = 163, Position = (569, 463), Local Density = 0
Cell 41: Area = 1861, Perimeter = 151, Position = (323, 464), Local Density = 2
Cell 42: Area = 1785, Perimeter = 139, Position = (193, 469), Local Density = 1
Cell 43: Area = 3707, Perimeter = 246, Position = (394, 460), Local Density = 1
Cell 44: Area = 3401, Perimeter = 252, Position = (442, 493), Local Density = 1
Cell 45: Area = 2629, Perimeter = 170, Position = (496, 489), Local Density = 0
Cell 46: Area = 2811, Perimeter = 194, Position = (660, 490), Local Density = 1
Cell 48: Area = 1978, Perimeter = 146, Position = (288, 497), Local Density = 3
Cell 49: Area = 1162, Perimeter = 116, Position = (87, 492), Local Density = 3
Cell 50: Area = 1112, Perimeter = 111, Position = (103, 509), Local Density = 1
Cell 51: Area = 1408, Perimeter = 142, Position = (329, 511), Local Density = 3
Cell 52: Area = 932, Perimeter = 106, Position = (352, 521), Local Density = 1
Cell 53: Area = 908, Perimeter = 106, Position = (233, 509), Local Density = 3
Cell 54: Area = 230, Perimeter = 55, Position = (236, 530), Local Density = 3
Cell 55: Area = 669, Perimeter = 88, Position = (254, 521), Local Density = 4
Cell 56: Area = 2088, Perimeter = 221, Position = (22, 508), Local Density = 1
Cell 57: Area = 842, Perimeter = 100, Position = (51, 523), Local Density = 2
Cell 58: Area = 2959, Perimeter = 171, Position = (553, 521), Local Density = 0
Cell 59: Area = 1411, Perimeter = 140, Position = (181, 518), Local Density = 1
Cell 60: Area = 836, Perimeter = 101, Position = (202, 528), Local Density = 3
Cell 61: Area = 3160, Perimeter = 206, Position = (634, 529), Local Density = 1
Cell 62: Area = 2723, Perimeter = 165, Position = (411, 527), Local Density = 2
Cell 63: Area = 2177, Perimeter = 172, Position = (480, 537), Local Density = 1
Cell 64: Area = 768, Perimeter = 84, Position = (481, 563), Local Density = 2
Cell 65: Area = 2841, Perimeter = 170, Position = (148, 560), Local Density = 0
Cell 66: Area = 3082, Perimeter = 203, Position = (378, 571), Local Density = 0
Cell 67: Area = 11277, Perimeter = 741, Position = (287, 555), Local Density = 1
Cell 68: Area = 1255, Perimeter = 125, Position = (428, 572), Local Density = 2
Cell 69: Area = 588, Perimeter = 80, Position = (454, 580), Local Density = 2
```

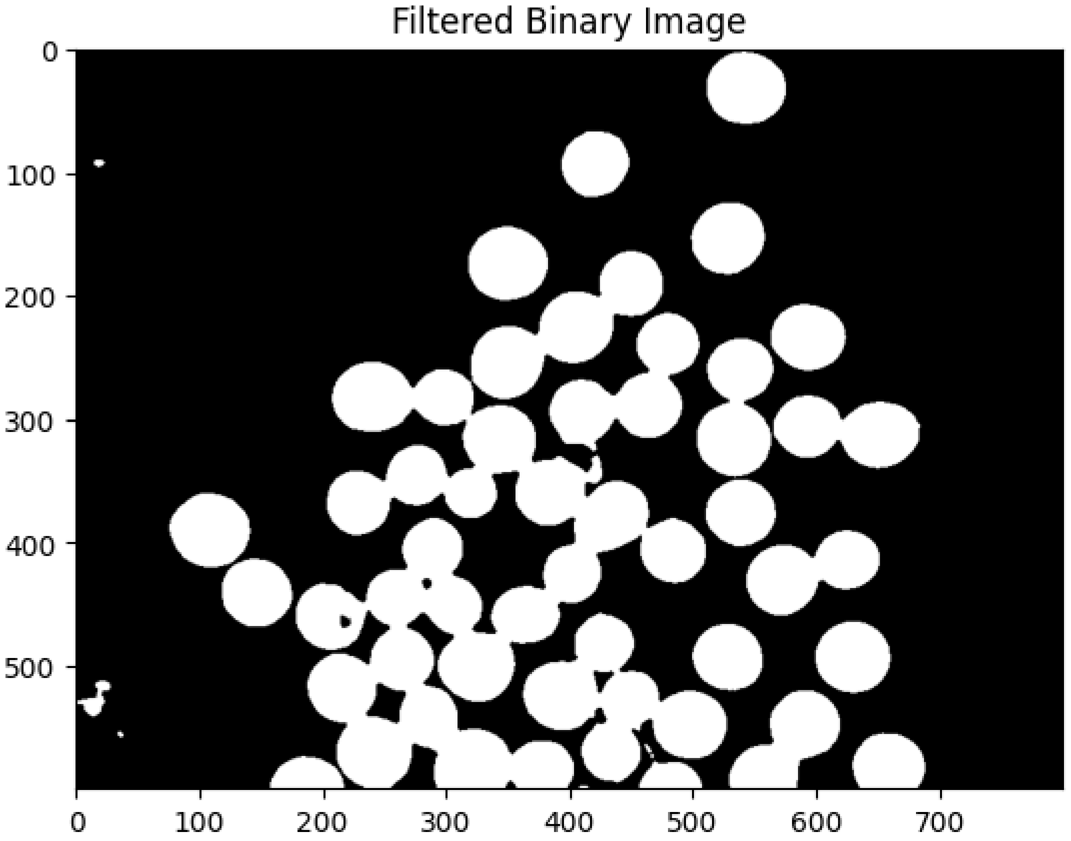

Original Image
0
100
200
300
400
500
0
100
200
300
400
500
600
700

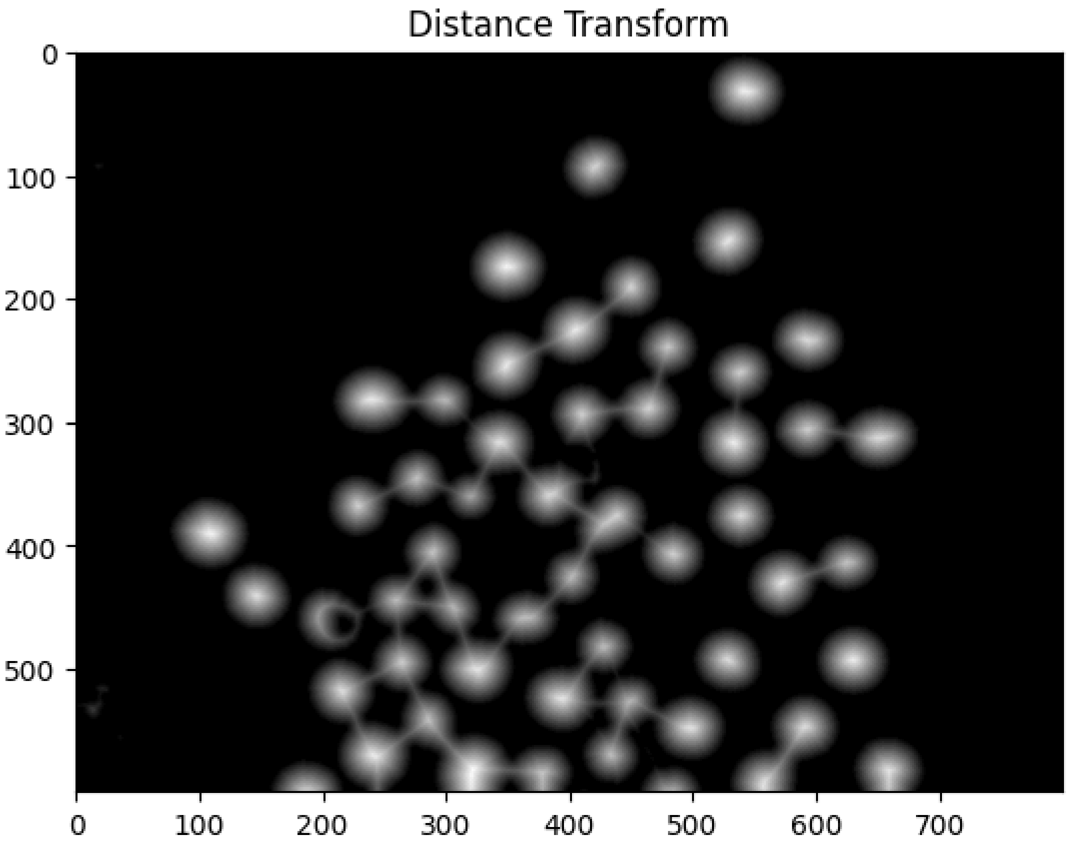

Enhanced Contrast
0
100
200
300
400
500
0
100
200
300
400
500
600
700

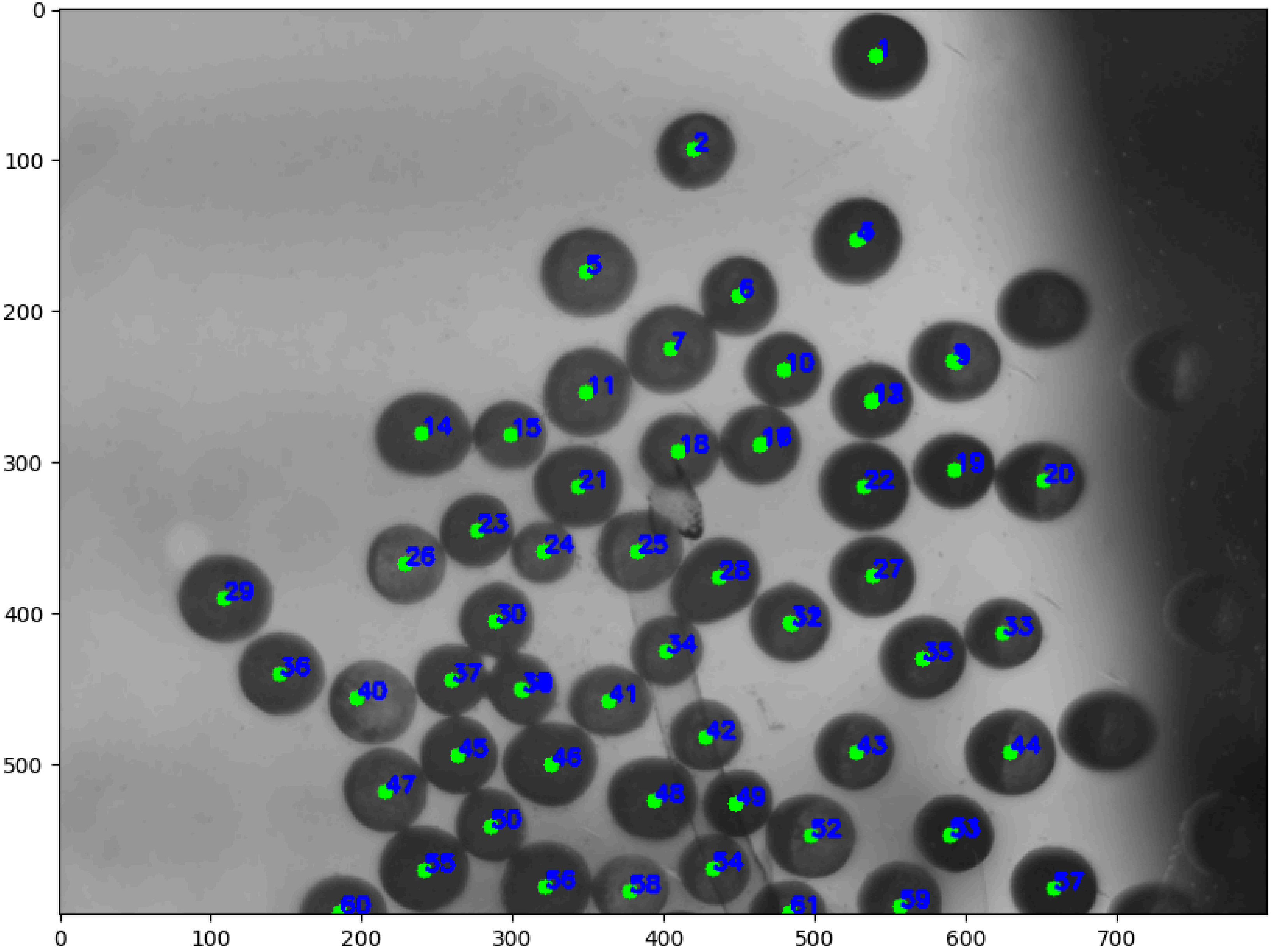

Filtered Binary Image
0
100
200
300
400
500
0
100
200
300
400
500
600
700



```
num_cells < 10: Reapplying area filtering with stricter threshold 50000
```

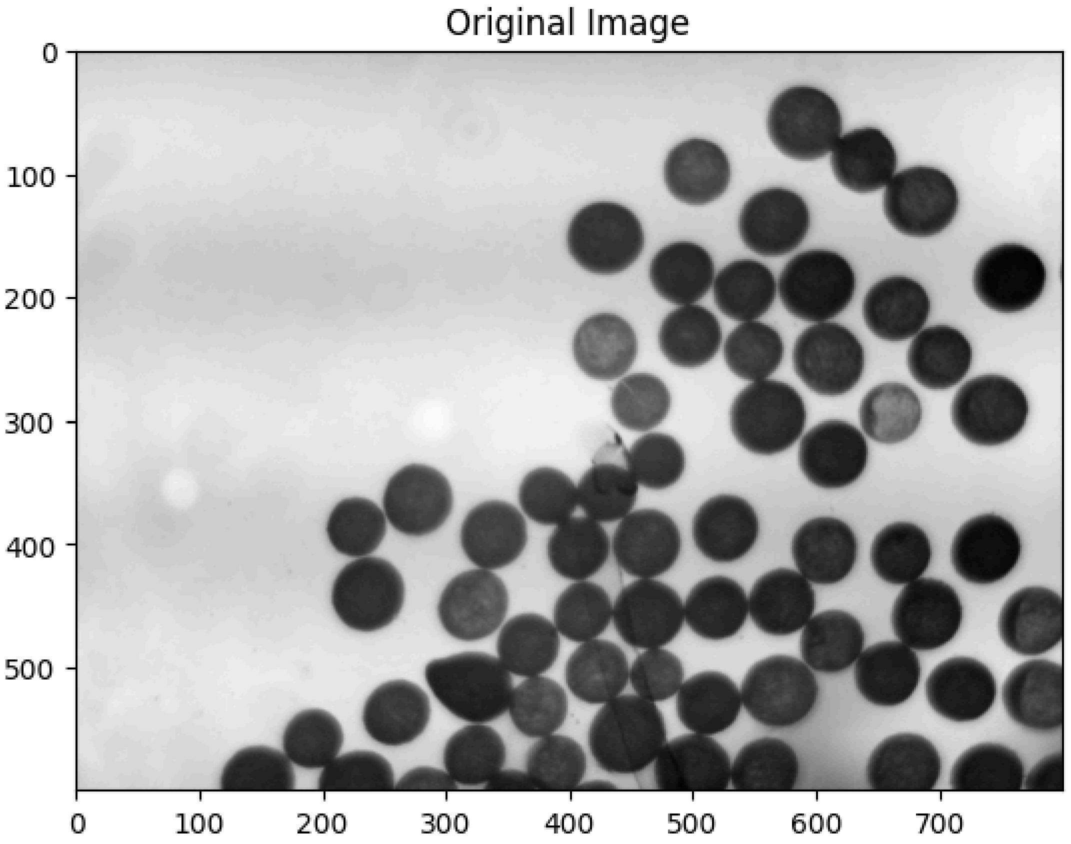

Filtered Binary Image
0
100
200
300
400
500
0
100
200
300
400
500
600
700

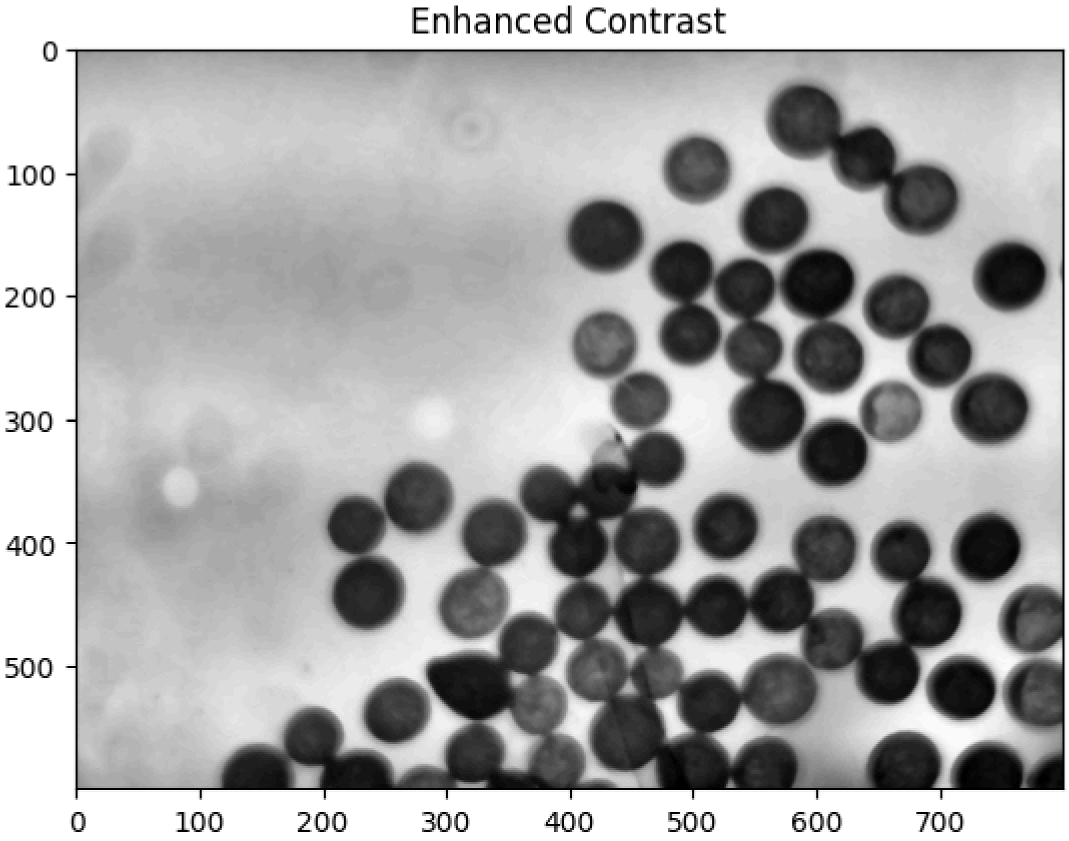

Distance Transform
0
100
200
300
400
500
0
100
200
300
400
500
600
700

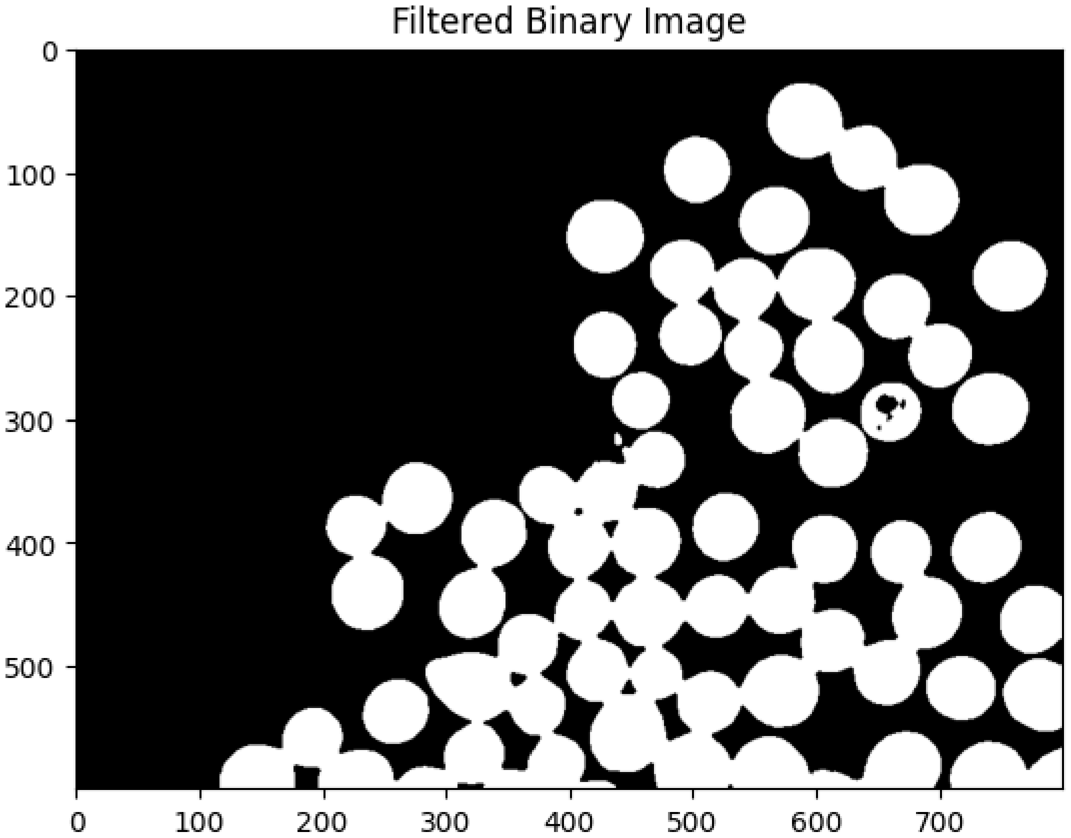


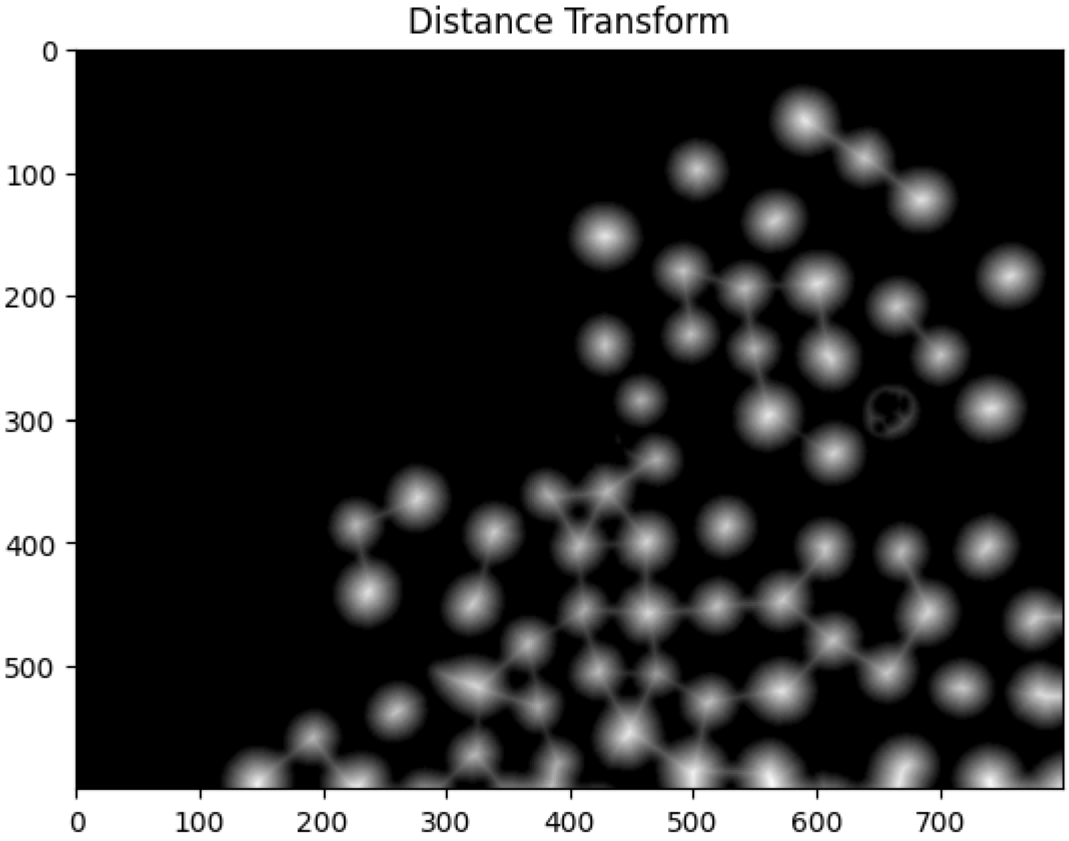
Total Cells Detected: 61

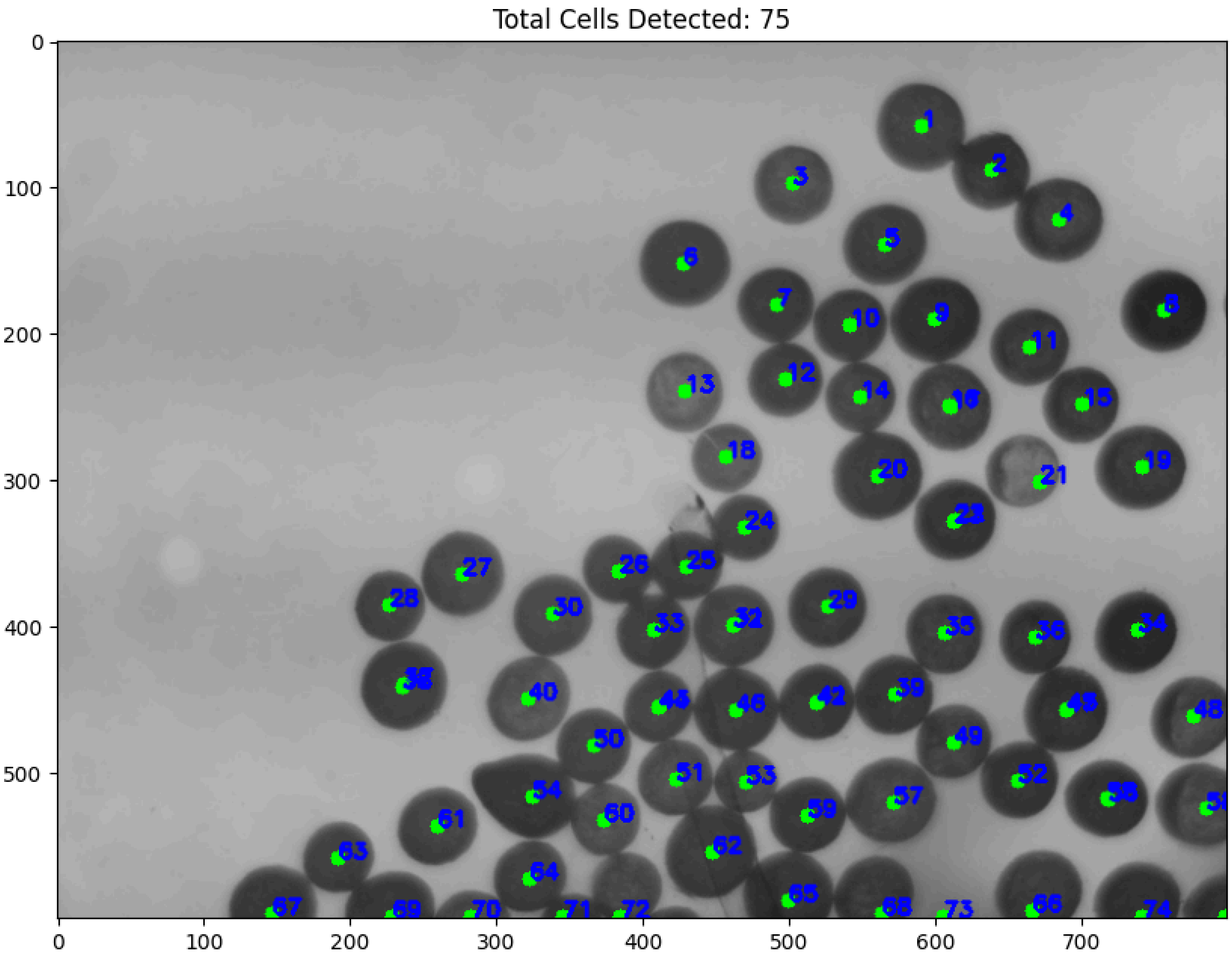
```
Cell 1: Area = 1729, Perimeter = 155, Position = (532, 28), Local Density = 1
Cell 2: Area = 1207, Perimeter = 116, Position = (558, 37), Local Density = 1
Cell 3: Area = 1407, Perimeter = 140, Position = (412, 91), Local Density = 1
Cell 4: Area = 833, Perimeter = 101, Position = (434, 98), Local Density = 1
Cell 5: Area = 1577, Perimeter = 148, Position = (538, 149), Local Density = 2
Cell 6: Area = 1078, Perimeter = 112, Position = (514, 160), Local Density = 2
Cell 7: Area = 1, Perimeter = 1, Position = (529, 154), Local Density = 2
Cell 8: Area = 3011, Perimeter = 174, Position = (350, 174), Local Density = 0
Cell 9: Area = 1618, Perimeter = 151, Position = (449, 185), Local Density = 1
Cell 10: Area = 552, Perimeter = 75, Position = (449, 207), Local Density = 3
Cell 11: Area = 2750, Perimeter = 169, Position = (404, 226), Local Density = 2
Cell 12: Area = 1519, Perimeter = 144, Position = (584, 230), Local Density = 2
Cell 13: Area = 975, Perimeter = 105, Position = (608, 239), Local Density = 1
Cell 14: Area = 1991, Perimeter = 148, Position = (479, 240), Local Density = 2
Cell 15: Area = 2751, Perimeter = 168, Position = (349, 254), Local Density = 1
Cell 16: Area = 1044, Perimeter = 108, Position = (547, 252), Local Density = 2
Cell 17: Area = 1096, Perimeter = 117, Position = (530, 268), Local Density = 2
Cell 18: Area = 647, Perimeter = 81, Position = (240, 265), Local Density = 3
Cell 19: Area = 1084, Perimeter = 110, Position = (224, 288), Local Density = 3
Cell 20: Area = 1, Perimeter = 1, Position = (240, 283), Local Density = 3
Cell 21: Area = 1051, Perimeter = 112, Position = (255, 288), Local Density = 4
Cell 22: Area = 1864, Perimeter = 153, Position = (297, 283), Local Density = 1
Cell 23: Area = 1071, Perimeter = 115, Position = (472, 281), Local Density = 2
Cell 24: Area = 1110, Perimeter = 110, Position = (455, 297), Local Density = 3
Cell 25: Area = 1577, Perimeter = 145, Position = (410, 289), Local Density = 3
Cell 26: Area = 598, Perimeter = 80, Position = (409, 311), Local Density = 3
Cell 27: Area = 1029, Perimeter = 107, Position = (584, 299), Local Density = 2
Cell 28: Area = 620, Perimeter = 81, Position = (610, 305), Local Density = 3
Cell 29: Area = 504, Perimeter = 70, Position = (593, 321), Local Density = 2
Cell 30: Area = 2598, Perimeter = 164, Position = (652, 313), Local Density = 1
Cell 31: Area = 2456, Perimeter = 160, Position = (344, 316), Local Density = 1
Cell 32: Area = 2764, Perimeter = 167, Position = (533, 317), Local Density = 1
Cell 33: Area = 1858, Perimeter = 138, Position = (275, 345), Local Density = 1
Cell 34: Area = 1501, Perimeter = 139, Position = (320, 358), Local Density = 2
Cell 35: Area = 2689, Perimeter = 200, Position = (387, 357), Local Density = 0
Cell 36: Area = 2116, Perimeter = 145, Position = (229, 368), Local Density = 0
Cell 37: Area = 1485, Perimeter = 141, Position = (530, 373), Local Density = 1
Cell 38: Area = 859, Perimeter = 101, Position = (553, 381), Local Density = 1
Cell 39: Area = 2486, Perimeter = 154, Position = (435, 378), Local Density = 1
Cell 40: Area = 1902, Perimeter = 159, Position = (99, 387), Local Density = 1
Cell 41: Area = 1094, Perimeter = 112, Position = (125, 396), Local Density = 2
Cell 42: Area = 1937, Perimeter = 138, Position = (289, 406), Local Density = 2
Cell 43: Area = 1087, Perimeter = 108, Position = (475, 399), Local Density = 2
Cell 44: Area = 1037, Perimeter = 106, Position = (492, 415), Local Density = 1
Cell 45: Area = 441, Perimeter = 68, Position = (624, 399), Local Density = 3
Cell 46: Area = 725, Perimeter = 91, Position = (638, 418), Local Density = 3
Cell 47: Area = 816, Perimeter = 104, Position = (610, 419), Local Density = 4
Cell 48: Area = 1, Perimeter = 1, Position = (625, 415), Local Density = 3
Cell 49: Area = 1971, Perimeter = 155, Position = (402, 423), Local Density = 0
Cell 50: Area = 2522, Perimeter = 160, Position = (571, 431), Local Density = 1
Cell 51: Area = 2485, Perimeter = 157, Position = (146, 441), Local Density = 1
Cell 52: Area = 1839, Perimeter = 149, Position = (258, 445), Local Density = 2
Cell 53: Area = 862, Perimeter = 102, Position = (299, 443), Local Density = 3
Cell 54: Area = 918, Perimeter = 110, Position = (313, 461), Local Density = 1
Cell 55: Area = 351091, Perimeter = 9042, Position = (394, 269), Local Density = 4
Cell 56: Area = 2042, Perimeter = 147, Position = (363, 459), Local Density = 0
Cell 57: Area = 1992, Perimeter = 147, Position = (427, 485), Local Density = 0
Cell 58: Area = 1458, Perimeter = 140, Position = (519, 489), Local Density = 1
Cell 59: Area = 912, Perimeter = 105, Position = (542, 498), Local Density = 1
Cell 60: Area = 2768, Perimeter = 166, Position = (629, 493), Local Density = 0
Cell 61: Area = 2087, Perimeter = 149, Position = (264, 495), Local Density = 0
```




Enhanced Contrast
0
100
200
300
400
500
0
100
200
300
400
500
600
700


Filtered Binary Image
0
100
200
300
400
500
0
100
200
300
400
500
600
700


Distance Transform
0
100
200
300
400
500
0
100
200
300
400
500
600
700


Total Cells Detected: 75
0
100
200
300
400
500
0
100
200
300
400
500
600
700

```
Cell 1: Area = 2882, Perimeter = 169, Position = (590, 58), Local Density = 0
Cell 2: Area = 2296, Perimeter = 156, Position = (638, 89), Local Density = 0
Cell 3: Area = 2272, Perimeter = 151, Position = (503, 98), Local Density = 0
Cell 4: Area = 2768, Perimeter = 168, Position = (685, 123), Local Density = 0
Cell 5: Area = 2473, Perimeter = 157, Position = (566, 139), Local Density = 0
Cell 6: Area = 1785, Perimeter = 156, Position = (419, 149), Local Density = 1
Cell 7: Area = 1082, Perimeter = 113, Position = (444, 158), Local Density = 1
Cell 8: Area = 2108, Perimeter = 147, Position = (491, 180), Local Density = 0
Cell 9: Area = 2688, Perimeter = 163, Position = (757, 184), Local Density = 0
Cell 10: Area = 2851, Perimeter = 174, Position = (600, 191), Local Density = 1
Cell 11: Area = 2401, Perimeter = 178, Position = (539, 198), Local Density = 1
Cell 12: Area = 2256, Perimeter = 150, Position = (665, 209), Local Density = 0
Cell 13: Area = 2000, Perimeter = 140, Position = (498, 232), Local Density = 1
Cell 14: Area = 1552, Perimeter = 144, Position = (428, 234), Local Density = 2
Cell 15: Area = 564, Perimeter = 72, Position = (429, 256), Local Density = 3
Cell 16: Area = 1160, Perimeter = 146, Position = (542, 242), Local Density = 3
Cell 17: Area = 684, Perimeter = 96, Position = (560, 249), Local Density = 3
Cell 18: Area = 2122, Perimeter = 145, Position = (700, 249), Local Density = 0
Cell 19: Area = 1310, Perimeter = 123, Position = (601, 240), Local Density = 3
Cell 20: Area = 1284, Perimeter = 117, Position = (619, 259), Local Density = 1
Cell 21: Area = 784, Perimeter = 96, Position = (450, 278), Local Density = 5
Cell 22: Area = 437, Perimeter = 65, Position = (471, 285), Local Density = 4
Cell 23: Area = 414, Perimeter = 64, Position = (457, 299), Local Density = 5
Cell 24: Area = 1, Perimeter = 1, Position = (458, 286), Local Density = 5
Cell 25: Area = 2827, Perimeter = 167, Position = (741, 292), Local Density = 0
Cell 26: Area = 2819, Perimeter = 170, Position = (561, 298), Local Density = 1
Cell 27: Area = 321280, Perimeter = 8555, Position = (346, 256), Local Density = 0
Cell 28: Area = 601, Perimeter = 78, Position = (614, 311), Local Density = 2
Cell 29: Area = 904, Perimeter = 106, Position = (627, 333), Local Density = 2
Cell 30: Area = 912, Perimeter = 104, Position = (599, 333), Local Density = 2
Cell 31: Area = 1126, Perimeter = 126, Position = (461, 331), Local Density = 5
Cell 32: Area = 607, Perimeter = 85, Position = (482, 337), Local Density = 2
Cell 33: Area = 1983, Perimeter = 144, Position = (430, 361), Local Density = 3
Cell 34: Area = 1776, Perimeter = 138, Position = (382, 362), Local Density = 2
Cell 35: Area = 2551, Perimeter = 161, Position = (277, 364), Local Density = 0
Cell 36: Area = 1405, Perimeter = 145, Position = (228, 381), Local Density = 1
Cell 37: Area = 545, Perimeter = 79, Position = (226, 402), Local Density = 3
Cell 38: Area = 2313, Perimeter = 151, Position = (526, 387), Local Density = 0
Cell 39: Area = 2344, Perimeter = 160, Position = (338, 393), Local Density = 0
Cell 40: Area = 1158, Perimeter = 111, Position = (471, 391), Local Density = 1
Cell 41: Area = 1237, Perimeter = 116, Position = (454, 409), Local Density = 4
Cell 42: Area = 2007, Perimeter = 140, Position = (407, 403), Local Density = 4
Cell 43: Area = 1176, Perimeter = 115, Position = (747, 395), Local Density = 3
Cell 44: Area = 628, Perimeter = 80, Position = (721, 404), Local Density = 4
Cell 45: Area = 1, Perimeter = 1, Position = (739, 404), Local Density = 3
Cell 46: Area = 2303, Perimeter = 150, Position = (607, 405), Local Density = 0
Cell 47: Area = 656, Perimeter = 79, Position = (737, 421), Local Density = 4
Cell 48: Area = 2026, Perimeter = 150, Position = (669, 408), Local Density = 0
Cell 49: Area = 1638, Perimeter = 153, Position = (241, 432), Local Density = 3
Cell 50: Area = 1, Perimeter = 1, Position = (237, 441), Local Density = 3
Cell 51: Area = 1052, Perimeter = 111, Position = (230, 455), Local Density = 2
Cell 52: Area = 2267, Perimeter = 150, Position = (572, 447), Local Density = 1
Cell 53: Area = 2397, Perimeter = 157, Position = (321, 450), Local Density = 0
Cell 54: Area = 1045, Perimeter = 110, Position = (528, 444), Local Density = 2
Cell 55: Area = 1044, Perimeter = 110, Position = (510, 459), Local Density = 3
Cell 56: Area = 1561, Perimeter = 145, Position = (425, 440), Local Density = 4
Cell 57: Area = 925, Perimeter = 103, Position = (404, 464), Local Density = 3
Cell 58: Area = 1227, Perimeter = 113, Position = (699, 448), Local Density = 3
Cell 59: Area = 2376, Perimeter = 159, Position = (464, 457), Local Density = 3
Cell 60: Area = 1293, Perimeter = 118, Position = (681, 467), Local Density = 2
Cell 61: Area = 2252, Perimeter = 154, Position = (775, 462), Local Density = 0
Cell 62: Area = 2165, Perimeter = 150, Position = (612, 480), Local Density = 0
Cell 63: Area = 2039, Perimeter = 148, Position = (365, 484), Local Density = 1
Cell 64: Area = 2429, Perimeter = 181, Position = (426, 501), Local Density = 1
Cell 65: Area = 2258, Perimeter = 150, Position = (657, 505), Local Density = 1
Cell 66: Area = 5275, Perimeter = 400, Position = (505, 499), Local Density = 2
Cell 67: Area = 2868, Perimeter = 174, Position = (319, 515), Local Density = 0
Cell 68: Area = 1158, Perimeter = 112, Position = (708, 511), Local Density = 1
Cell 69: Area = 1101, Perimeter = 111, Position = (727, 526), Local Density = 1
Cell 70: Area = 2803, Perimeter = 166, Position = (571, 520), Local Density = 0
Cell 71: Area = 2270, Perimeter = 164, Position = (777, 523), Local Density = 0
Cell 72: Area = 2191, Perimeter = 170, Position = (514, 531), Local Density = 1
Cell 73: Area = 6878, Perimeter = 452, Position = (389, 553), Local Density = 0
Cell 74: Area = 2195, Perimeter = 147, Position = (259, 537), Local Density = 0
Cell 75: Area = 2886, Perimeter = 175, Position = (447, 557), Local Density = 0
```

Test and store all data; Organized data into well-structured tabular format:

In [ ]:

```
# Test
image_dir = "/content/images/"
image_paths = [os.path.join(image_dir, f) for f in os.listdir(image_dir) if f.endswith(".png")]

for img in image_paths:
    count_cells_from_distance_transform(img)


# Convert lists to DataFrames
df_cells = pd.DataFrame(cell_data_list)
df_images = pd.DataFrame(image_data_list)

# Save DataFrames to CSV
df_cells.to_csv("cell_data.csv", index=False)
```

```
df_images.to_csv("image_summary.csv", index=False)

# Display DataFrames
print("\n Processing Completed. Saved as 'cell_data.csv' and 'image_summary.csv'.")
print("\n Cell Data Preview:")
print(df_cells.head())

print("\n Image Summary Preview:")
print(df_images.head())
```

Support natural language or SQL-like queries:

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np



def load_data(choice):
    if choice == "cell":
        return pd.read_csv(cell_data_path)
    elif choice == "summary":
        return pd.read_csv(image_summary_path)
    else:
        print("N/A")
        return None

def run_query(query_string, df):
    try:
        result = df.query(query_string)
        print(f"Query executed: {query_string}")
        display(result.head())
        return result
    except Exception as e:
        print(f"Query error: {e}")

def plot_histogram(column_name, df, data_type="cell"):
    print("\n")
    plt.figure(figsize=(8, 6))
    plt.hist(df[column_name], bins=20, color="blue", alpha=0.7, edgecolor="black")
    plt.xlabel(column_name)
    plt.ylabel("Frequency")
    plt.title(f"Histogram of {column_name} ({data_type.capitalize()} Level Data)")
    plt.grid(True)
    plt.show()

def plot_scatter(x_col, y_col, df, data_type="cell"):
    print("\n")
    plt.figure(figsize=(8, 6))
    plt.scatter(df[x_col], df[y_col], color="green", alpha=0.6)
    plt.xlabel(x_col)
    plt.ylabel(y_col)
    plt.title(f"Scatter Plot of {x_col} vs {y_col} ({data_type.capitalize()} Level Data)")
    plt.grid(True)
    plt.show()

def plot_heatmap(column_name, df, data_type="cell"):
    print("\n")
    plt.figure(figsize=(8, 6))

    heatmap, xedges, yedges = np.histogram2d(df["X Position"], df["Y Position"], bins=[50, 50], weights=df[column_name])

    plt.imshow(heatmap.T, origin="lower", cmap="hot", aspect="auto")
    plt.colorbar(label=column_name)
    plt.xlabel("X Position")
    plt.ylabel("Y Position")
    plt.title(f"Heatmap of {column_name} ({data_type.capitalize()} Level Data)")
    plt.show()
```

Output tabular-format examples and show distributions:

In [4]:

```
cell_data_path = "cell_data.csv"
image_summary_path = "image_summary.csv"

data_choice = "cell"
df = load_data(data_choice)

# Example query: Find cells with Area > 1000 and Local Density > 2
query_result1 = run_query("Area > 1000 and `Local density` > 2", df)

# Example query: Find Cells in a Specific Image
query_result2 = run_query("`Image Name` == '26_01_2024_11.png'", df)

# Example query: Find Cells in the Left Half of an Image (X Position < 250)
query_result3 = run_query("`X Position` < 250", df)

# Example plot: Area Distribution
plot_histogram("Area", df, "cell")

df_sum = load_data("summary")
plot_histogram("Total Cells", df_sum, "summary")
```

```
Query executed: Area > 1000 and `Local density` > 2
```

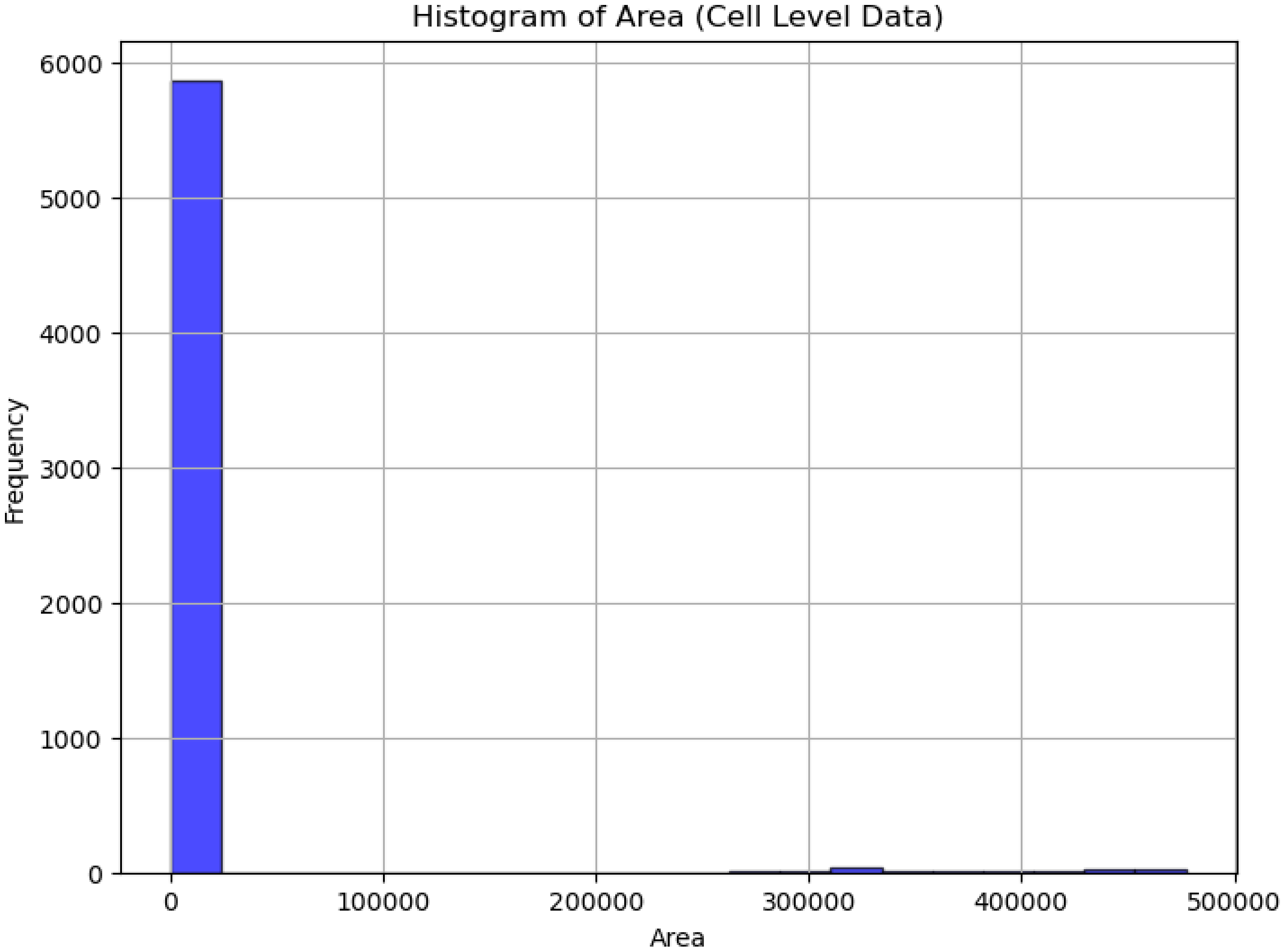
	Image Name	Cell ID	X Position	Y Position	Area	Perimeter	Local density
28	28_01_2024_08.png	8	662	107	1356	124	3
31	28_01_2024_08.png	11	636	123	1064	111	4
44	28_01_2024_08.png	24	618	200	1242	120	3
54	28_01_2024_08.png	34	469	289	1841	156	7
61	28_01_2024_08.png	41	508	289	1367	143	8

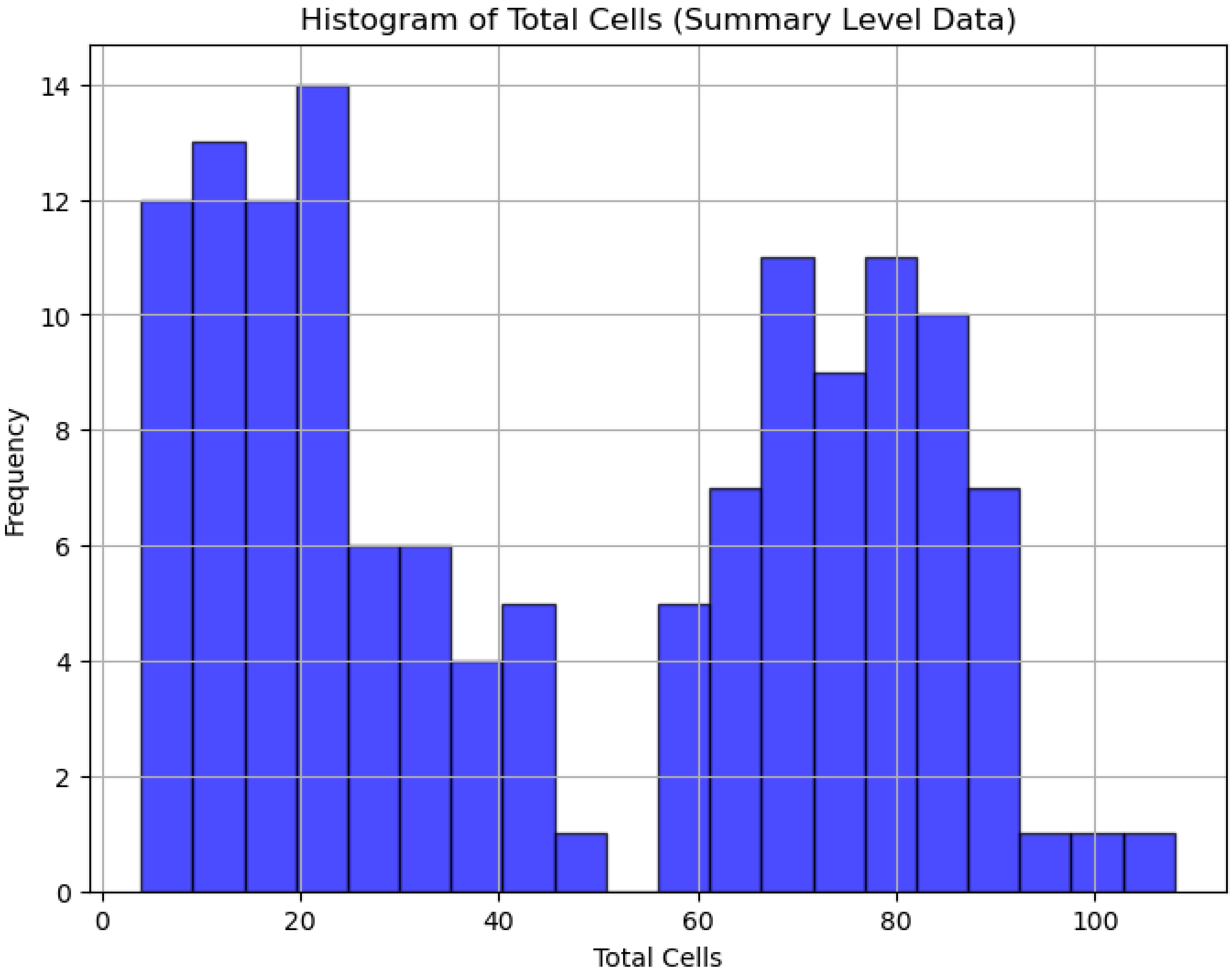
```
Query executed: `Image Name` == '26_01_2024_11.png'
```

	Image Name	Cell ID	X Position	Y Position	Area	Perimeter	Local density
3980	26_01_2024_11.png	6	29	161	618	86	1
3981	26_01_2024_11.png	7	21	167	1	1	1
3982	26_01_2024_11.png	8	422	288	434605	4794	0
3983	26_01_2024_11.png	9	247	229	1596	147	1
3984	26_01_2024_11.png	10	248	251	566	75	2

```
Query executed: `X Position` < 250
```

	Image Name	Cell ID	X Position	Y Position	Area	Perimeter	Local density
0	30_01_2024_18.png	15	48	491	3145	178	0
2	26_01_2024_05.png	1	25	77	751	100	1
3	26_01_2024_05.png	2	13	97	612	108	1
5	26_01_2024_05.png	5	108	302	373	67	0
6	26_01_2024_05.png	6	39	306	3089	292	0


Histogram of Area (Cell Level Data)
Frequency
6000
5000
4000
3000
2000
1000
0
0
100000
200000
300000
400000
500000
Area


Histogram of Total Cells (Summary Level Data)
Frequency
Total Cells
0
2
4
6
8
10
12
14
0
20
40
60
80
100