

procesamiento-de-imagenes

October 7, 2024

Jacob Valdenegro Monzón A01640992

```
[2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
import os

image_folder = "/content/Reto"

# Función para mostrar una imagen
def show_image(img, title="Image", cmap_type='gray'):
    plt.figure(figsize=(6, 6))
    plt.imshow(img, cmap=cmap_type)
    plt.title(title)
    plt.axis('off')
    plt.show()

# Cargar todas las imágenes desde la carpeta
def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder, filename), cv2.IMREAD_GRAYSCALE)
        if img is not None:
            images.append(img)
    return images

# Aplicar binarización por threshold
def apply_threshold(image, threshold_value=127):
    _, binary_img = cv2.threshold(image, threshold_value, 255, cv2.
    ↪THRESH_BINARY)
    return binary_img

# Aplicar erosión y dilatación
def apply_erosion(image, kernel_size=(3, 3), iterations=1):
    kernel = np.ones(kernel_size, np.uint8)
    eroded_img = cv2.erode(image, kernel, iterations=iterations)
    return eroded_img
```

```

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

# Cargar imágenes
images = load_images_from_folder(image_folder)

# Procesar cada imagen
for idx, img in enumerate(images):
    print(f"Procesando imagen {idx+1}")

    # Mostrar imagen original
    show_image(img, title=f"Imagen Original {idx+1}")

    # Aplicar binarización
    binary_img = apply_threshold(img, threshold_value=127)
    show_image(binary_img, title=f"Imagen Binarizada {idx+1}")

    # Aplicar erosión
    eroded_img = apply_erosion(binary_img, kernel_size=(3, 3), iterations=1)
    show_image(eroded_img, title=f"Imagen Erosionada {idx+1}")

    # Aplicar dilatación
    dilated_img = apply_dilation(binary_img, kernel_size=(3, 3), iterations=1)
    show_image(dilated_img, title=f"Imagen Dilatada {idx+1}")

```

Procesando imagen 1

Imagen Original 1

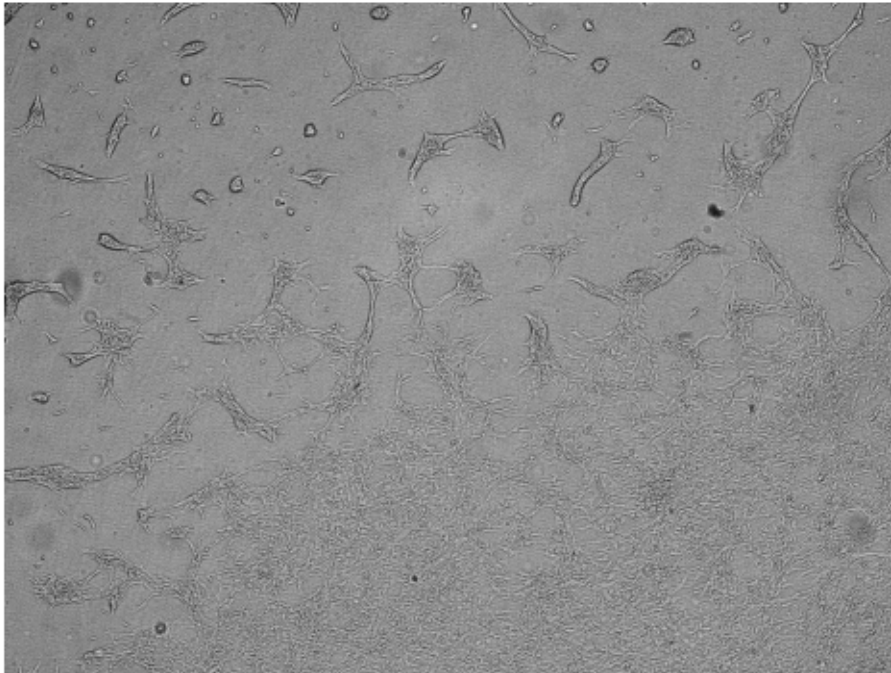


Imagen Binarizada 1

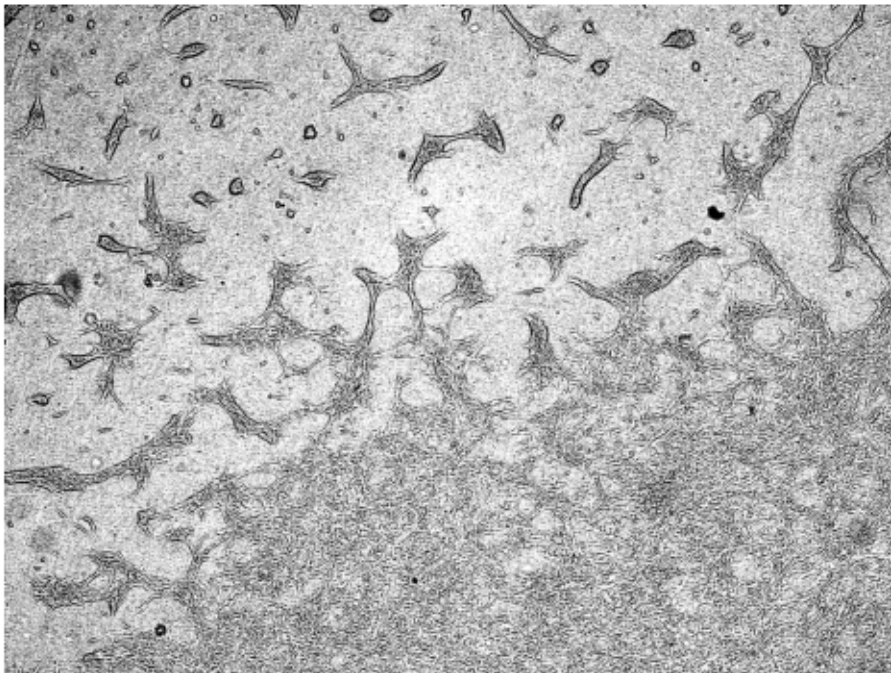


Imagen Erosionada 1

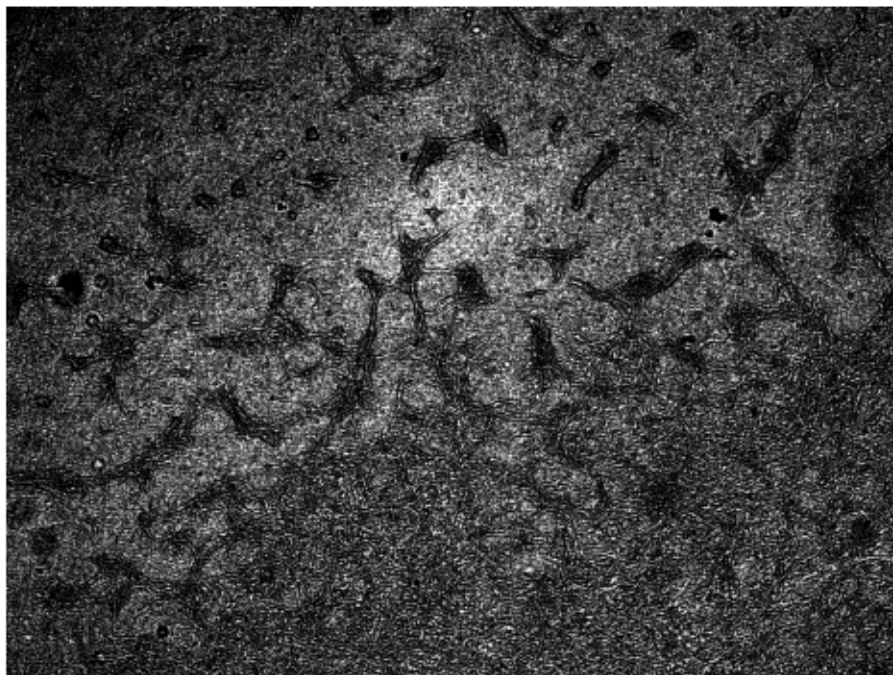
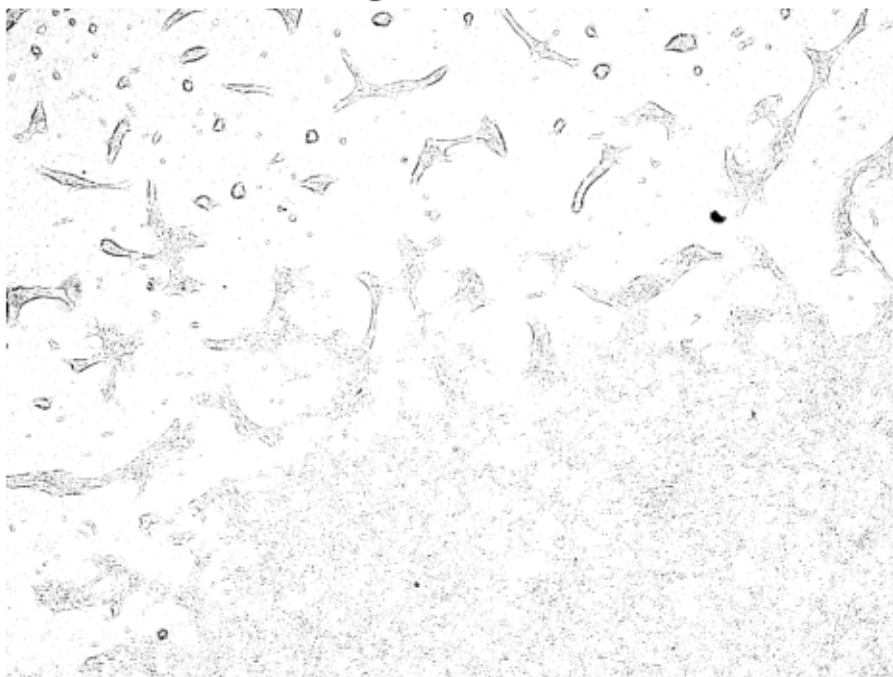


Imagen Dilatada 1



Procesando imagen 2

Imagen Original 2

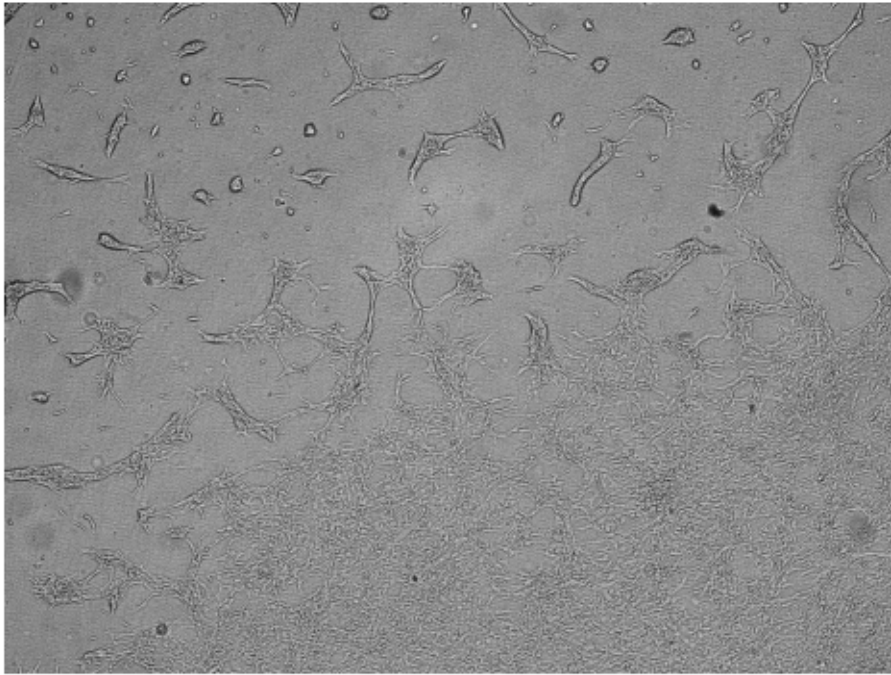


Imagen Binarizada 2

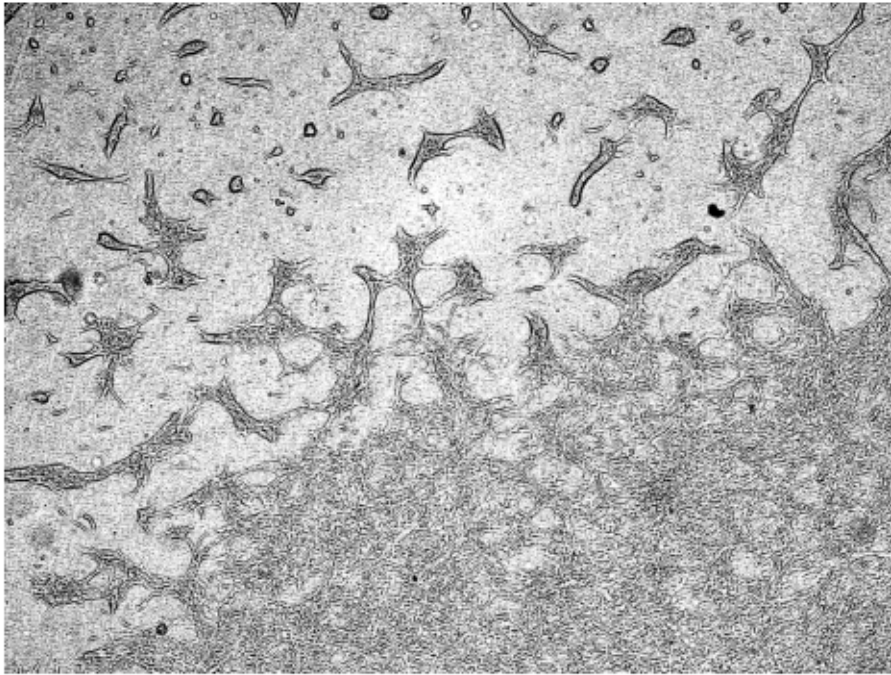


Imagen Erosionada 2

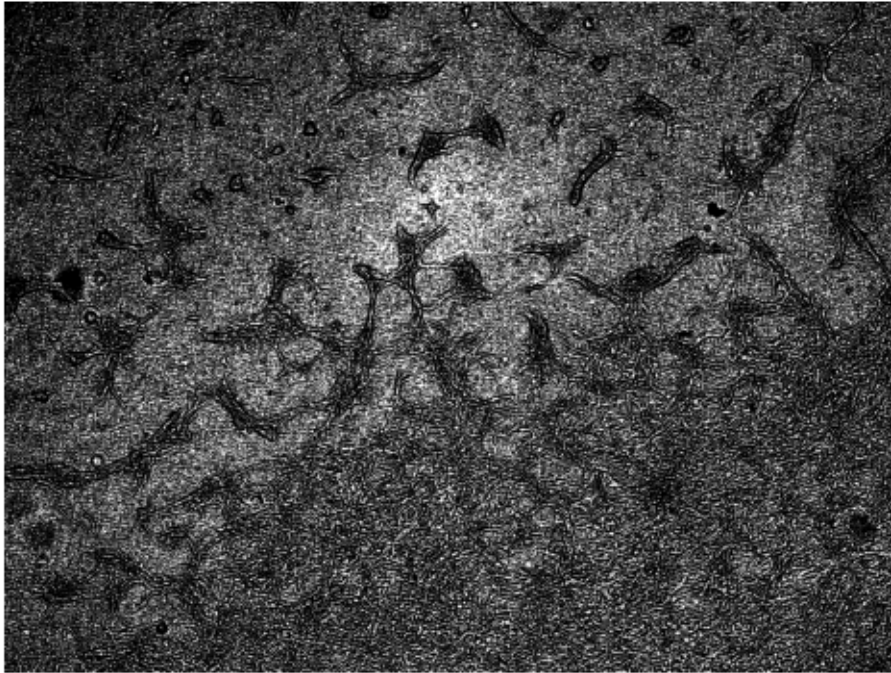
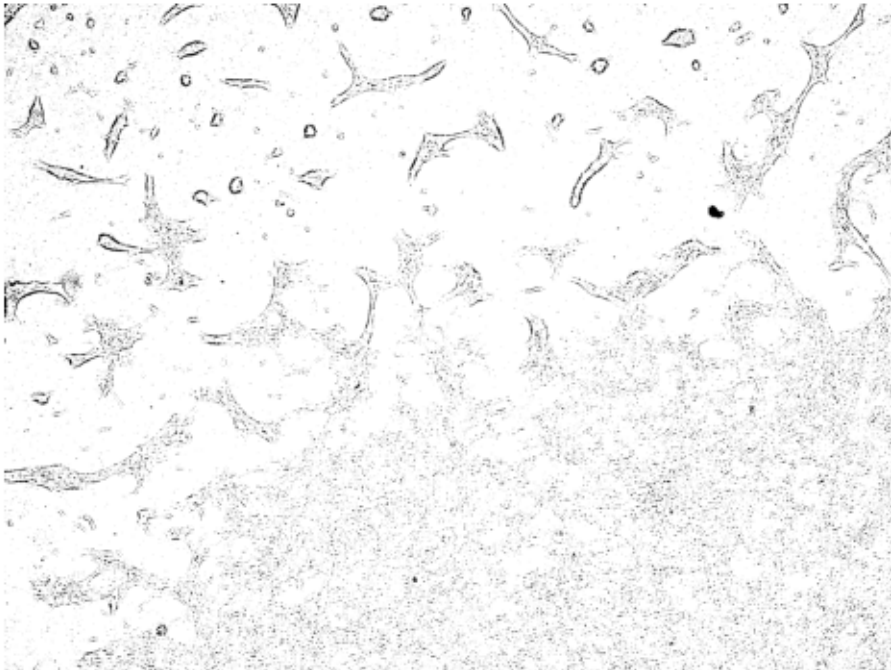


Imagen Dilatada 2



Procesando imagen 3

Imagen Original 3

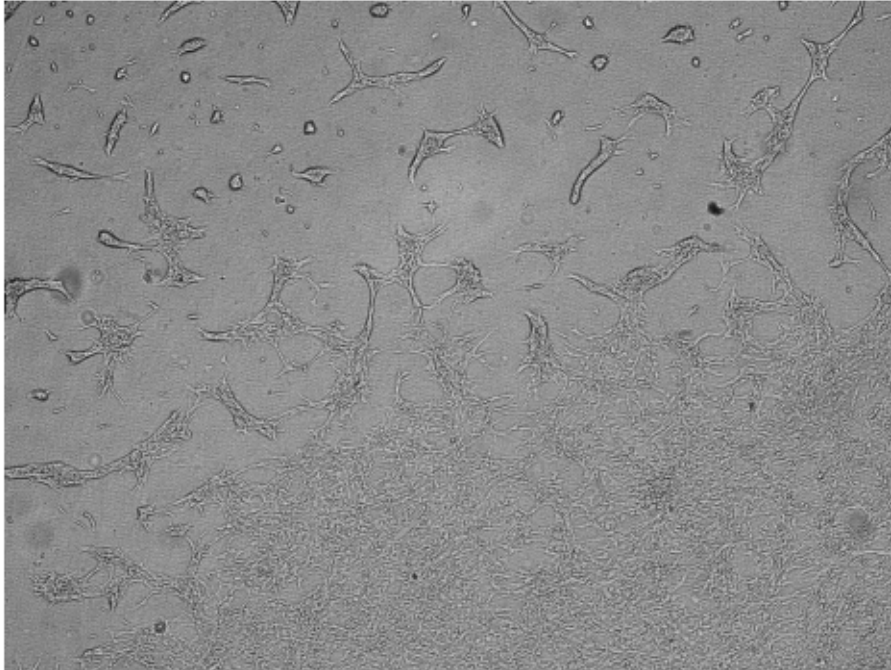


Imagen Binarizada 3

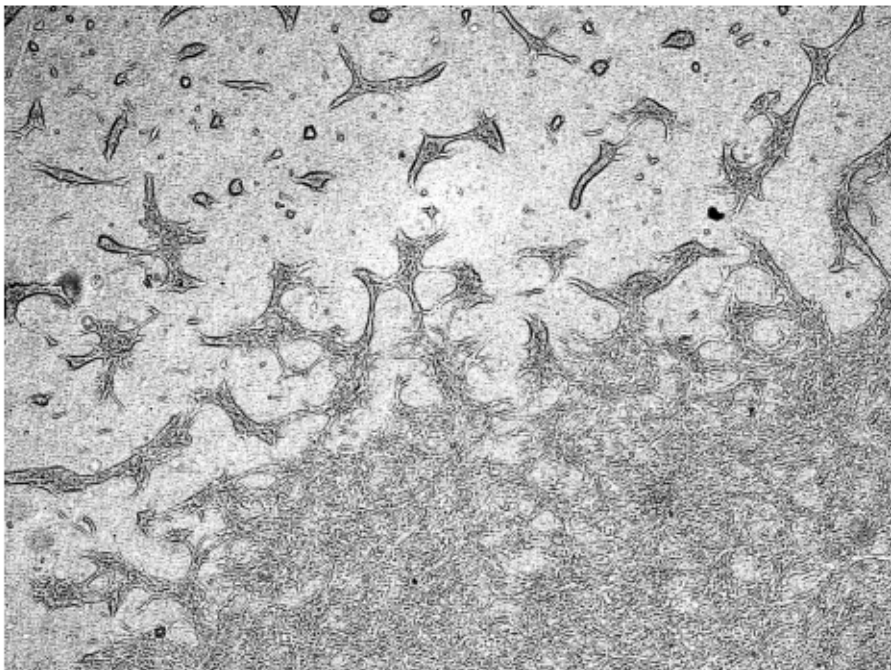


Imagen Erosionada 3

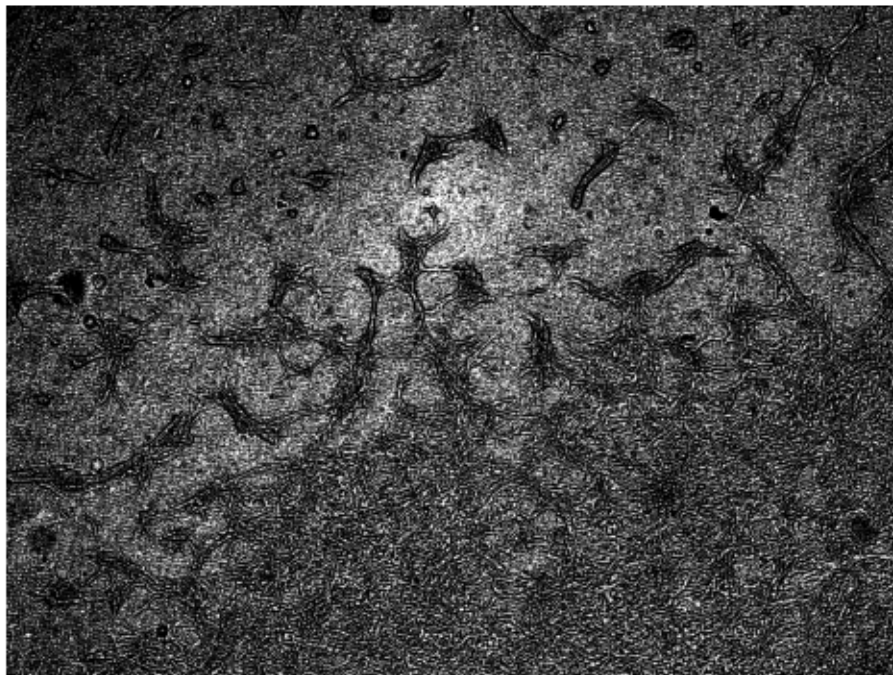
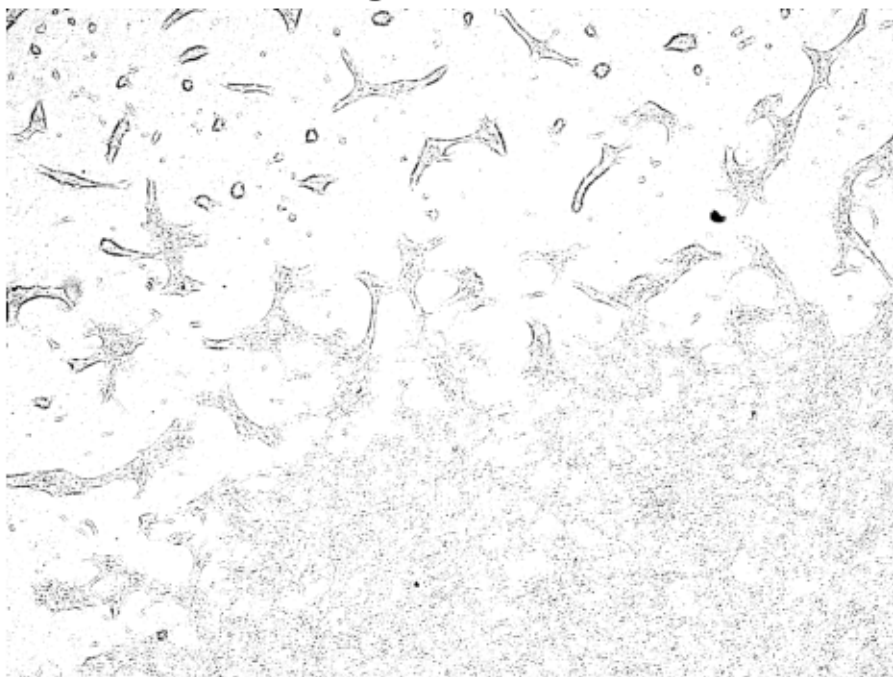


Imagen Dilatada 3



Procesando imagen 4

Imagen Original 4

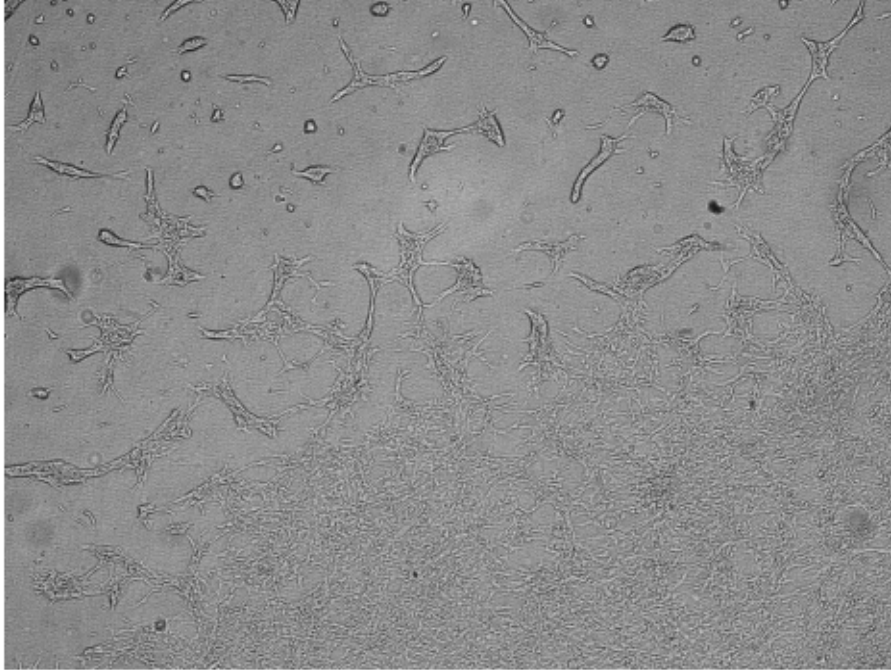


Imagen Binarizada 4

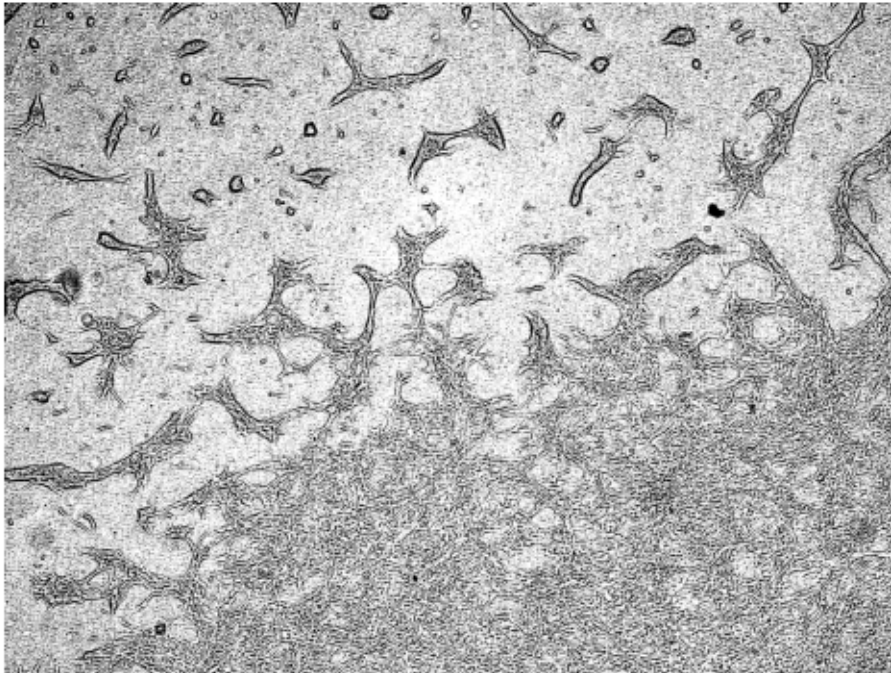


Imagen Erosionada 4

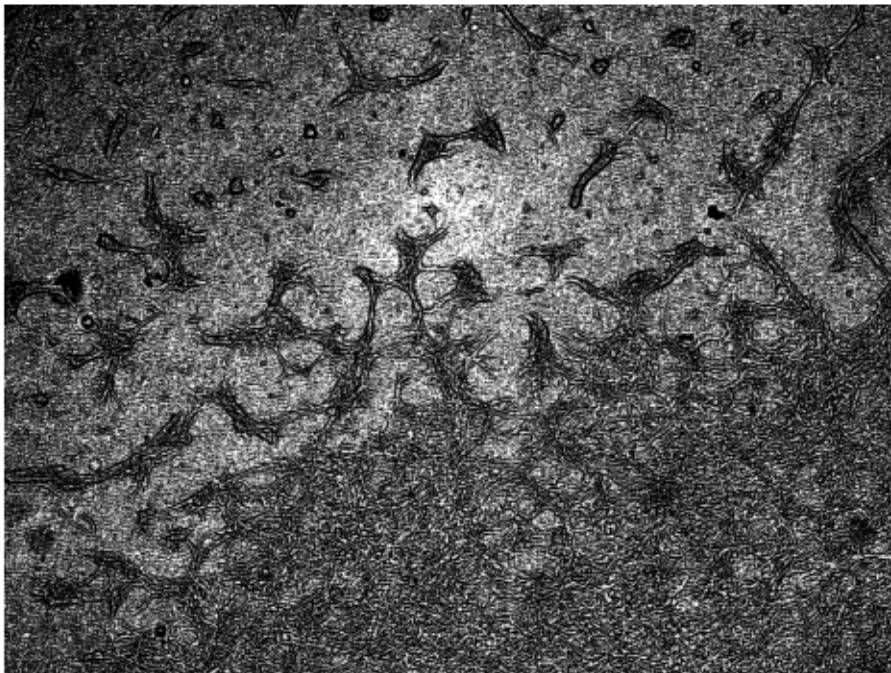
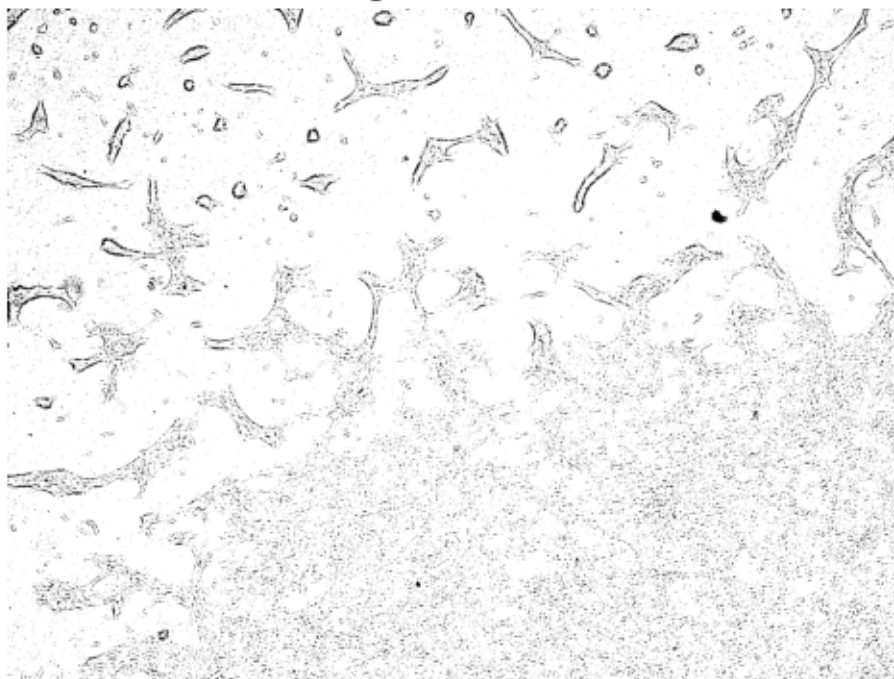


Imagen Dilatada 4



Procesando imagen 5

Imagen Original 5

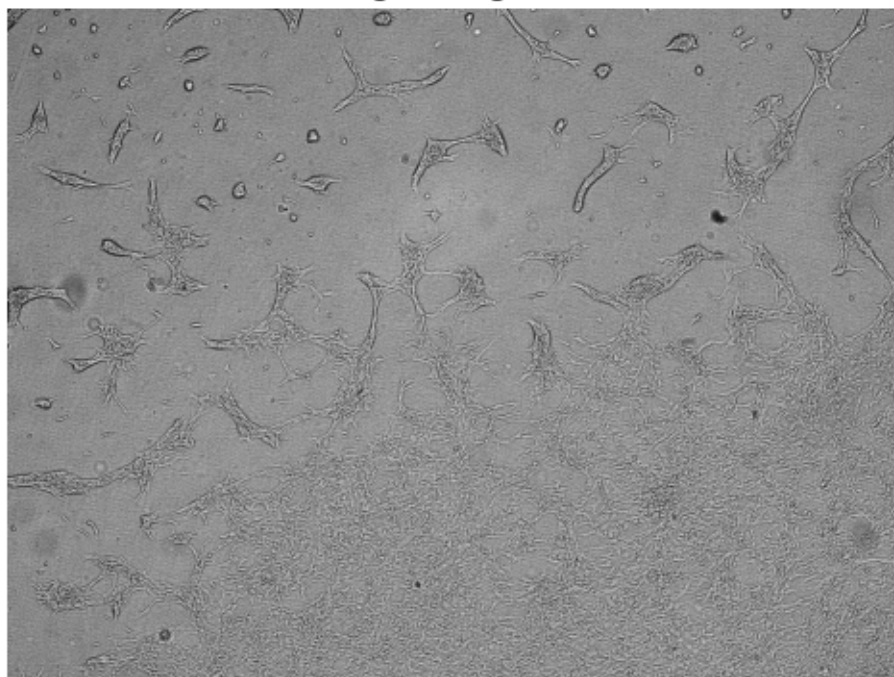


Imagen Binarizada 5

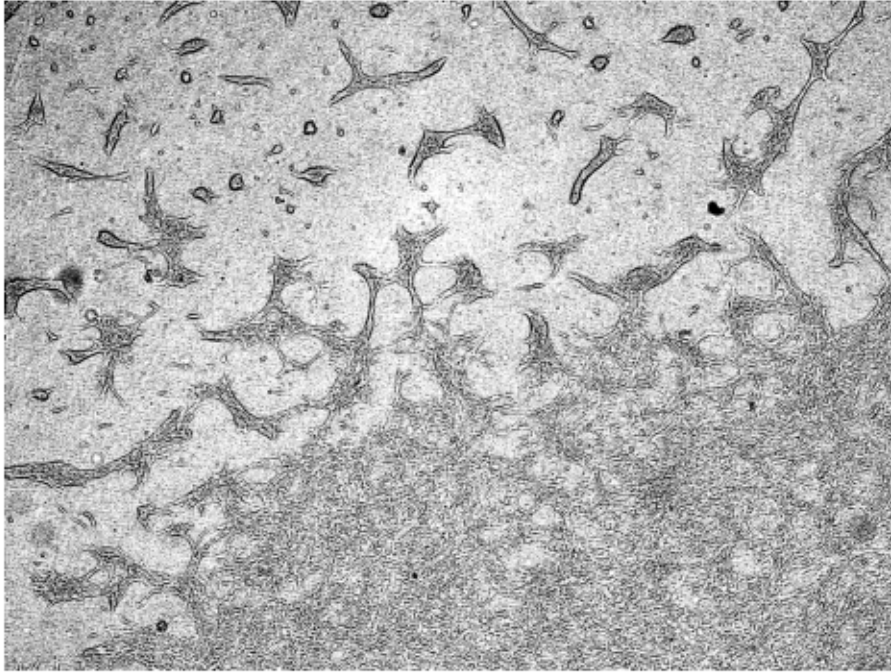


Imagen Erosionada 5

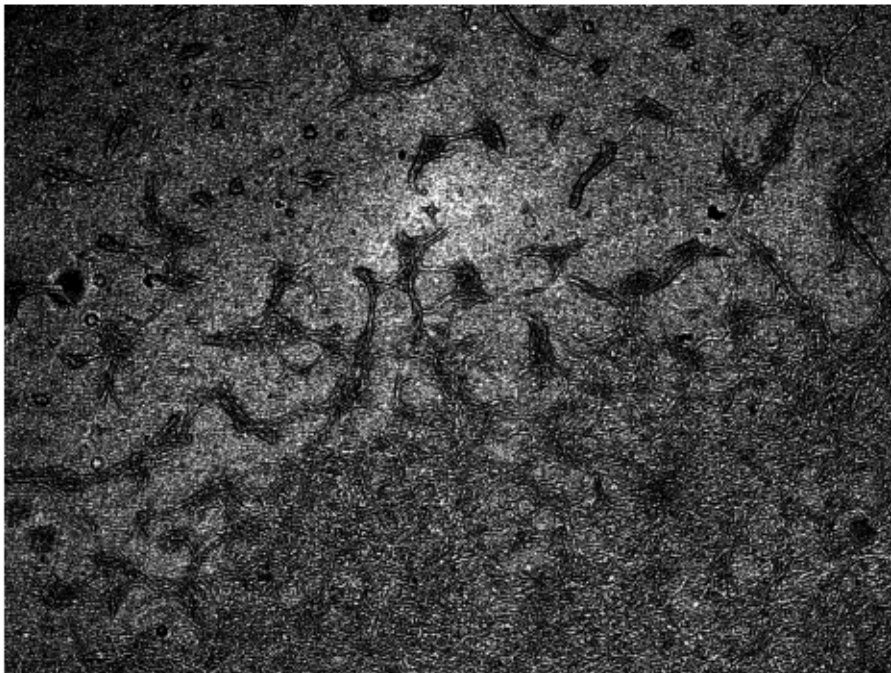


Imagen Dilatada 5

