ID: 2023-1-60-075

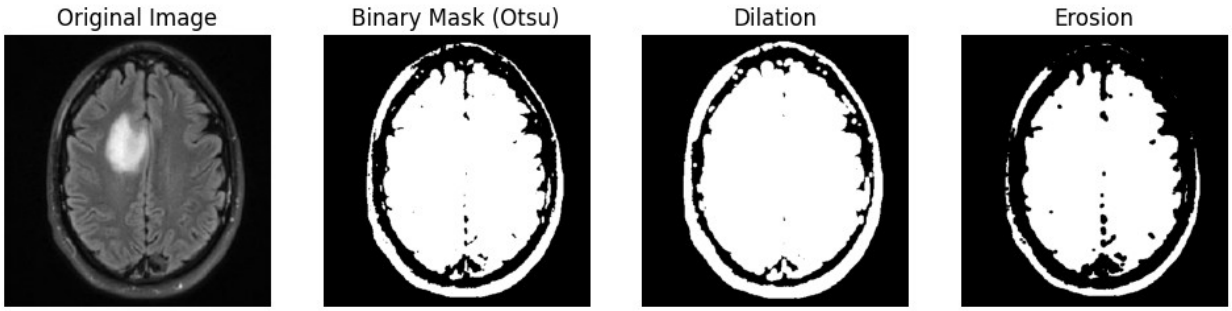
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread("/content/picture1.png", cv2.IMREAD_GRAYSCALE)
_, mask = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5,5))
dilated = cv2.dilate(mask, kernel, iterations=1)
eroded = cv2.erode(mask, kernel, iterations=1)

titles = ["Original Image", "Binary Mask (Otsu)", "Dilation",
"Erosion"]
images = [img, mask, dilated, eroded]

plt.figure(figsize=(12,6))
for i in range(4):
    plt.subplot(1,4,i+1)
    plt.imshow(images[i], cmap='gray')
    plt.title(titles[i])
    plt.axis("off")
plt.show()
```




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

img = cv2.imread("/content/picture2.png")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

edges = cv2.Canny(gray, 50, 150, apertureSize=3)

lines = cv2.HoughLines(edges, 1, np.pi/180, 150)

if lines is not None:
    for rho, theta in lines[:,0]:
```

```
        a = np.cos(theta)
        b = np.sin(theta)
        x0 = a * rho
        y0 = b * rho
        x1 = int(x0 + 1000 * (-b))
        y1 = int(y0 + 1000 * (a))
        x2 = int(x0 - 1000 * (-b))
        y2 = int(y0 - 1000 * (a))
        cv2.line(img, (x1, y1), (x2, y2), (0,0,255), 2)

plt.figure(figsize=(10,6))
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title("Detected Lines with Hough Transform")
plt.axis("off")
plt.show()
```