

Môn học: Nhập môn tính toán đa
phương tiện

BÁO CÁO CUỐI KỲ

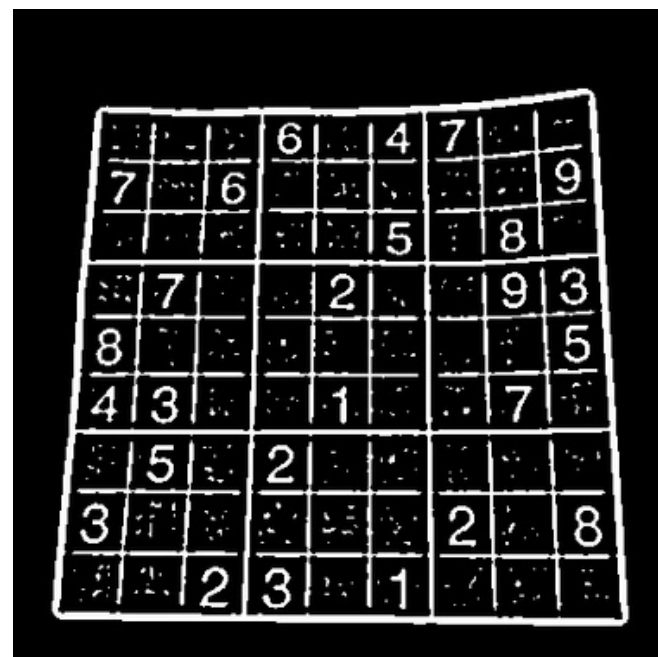
Giảng viên: Phạm Văn Huy

Thành viên:

- 51900340 – Nguyễn Lê Sông Hương
- 51900846 – Trần Đức Văn



YÊU CẦU ĐỀ BÀI



Input:

- Hình ảnh chứa bàn cờ sudoku

Quá trình:

- Convert về Gray scale
- Tạo mask
- Khử nhiễu, tạo res
- Chuyển Input về nhị phân
- Contours
- Loại bỏ chi tiết thừa

Output:

- Hình nhị phân chứa hình bàn cờ sudoku bao gồm những con số

Bước 1: Import các thư viện

```
import cv2 as cv
import numpy as np
import os
```



Bước 2:

- **imread()**: đọc input
- **GaussianBlur()**: làm mờ
- **COLOR_BGR2GRAY**: chuyển input về grayscale
- **zeros(gray.shape)**: tạo mask cùng kích thước input

```
path = 'input.jpg'
if os.path.isfile(path) == False:
    path = 'input.jpg'
image = cv.imread(path)
image = cv.GaussianBlur(image, (5,5), 0)
gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
mask = np.zeros((gray.shape), np.uint8)
```

Bước 3: Khử nhiễu

- **getStructuringElement()**: Tạo kernel dạng Ellipse
- **morphologyEx()**: khử nhiễu bằng closing
- **normalize()**: tạo res để chuẩn hóa dữ liệu thương của gray và closing

```
kernel1 = cv.getStructuringElement(cv.MORPH_ELLIPSE, (11,11))  
closing = cv.morphologyEx(gray, cv.MORPH_CLOSE, kernel1)  
div = np.float32(gray)/(closing)  
res = np.uint8(cv.normalize(div,div,0,255,cv.NORM_MINMAX))
```



Input sau khi khử nhiễu

Bước 4:

- **adaptiveThreshold()**: chuyển về ảnh nhị phân

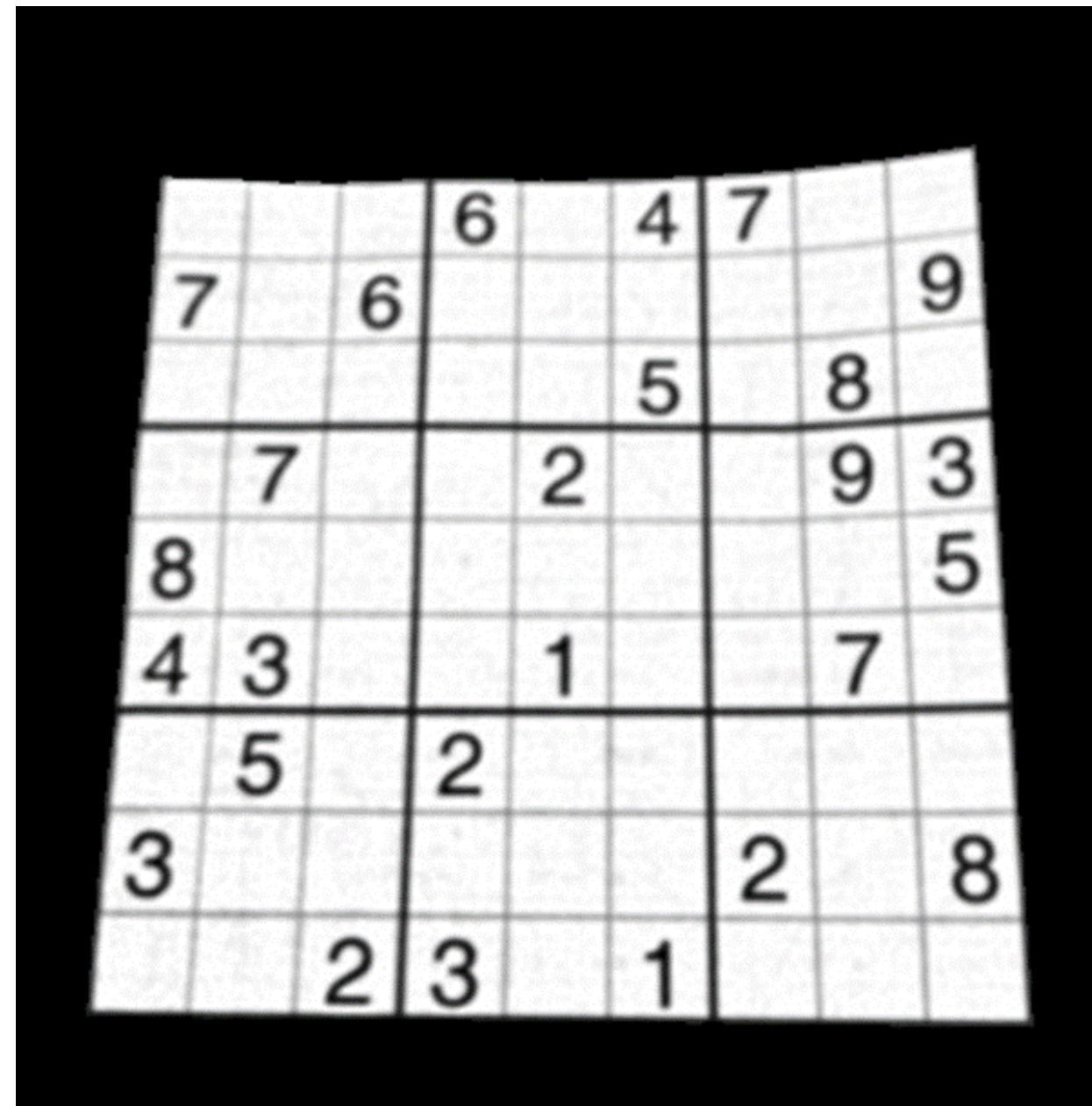
```
thresh = cv.adaptiveThreshold(res, 255, cv.THRESH_BINARY_INV, 1, 19, 2)
```



Bước 5:

- **findContours()**: tìm các đường nét khép kín
- **contourArea()**: chạy vòng lặp duyệt từng đường nét đã tìm trên
- **drawContours()**: vẽ những dữ liệu có trong contours đã tìm
- **bitwise_and()**: loại bỏ các chi tiết thừa ở ngoài contours

```
contours, hierarchy= cv.findContours(thresh, cv.RETR_TREE,  
cv.CHAIN_APPROX_SIMPLE)  
max_area = 0  
best_cnt = None  
for i in contours:  
    area = cv.contourArea(i)  
    if area > 1000:  
        if area > max_area:  
            max_area = area  
            best_cnt = i  
  
cv.drawContours(mask,[best_cnt],0,255,-1)  
cv.drawContours(mask,[best_cnt],0,0,2)  
  
res = cv.bitwise_and(res,mask)
```



Input được loại bỏ các chi tiết thừa

Bước 6: Đưa về nhị phân và
khử nhiễu bằng erode và
closing

```
blur = cv.GaussianBlur(res, (5,5), 0)
thresh = cv.adaptiveThreshold(blur, 255, 1, 1, 11, 2)
kernel = np.ones((2,2),np.uint8)
erosion = cv.erode(thresh,kernel,iterations = 1)
closing = cv.morphologyEx(erosion, cv.MORPH_CLOSE, kernel)
```

			6		4	7		
7		6						9
					5		8	
	7			2			9	3
8								5
4	3			1			7	
	5		2					
3						2		8
	2	2	3		1			

Output

Thank You!

