

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Subject

Xử lý ảnh

Nhóm: 03

Chủ đề:

Phân tích và nhận dạng biển số xe

Thành viên nhóm:

Họ và tên	Mã SV
Trần Trung Hiếu	B18DCVT162
Trương Quang Khải	B18DCVT229
Quản Trường Huy	B18DCDT098
Nguyễn Nam Anh	B18DCDT009

Mục lục

I. Phát biểu bài toán	4
1. Đề tài : <i>Phân tích và nhận dạng biến số xe</i>	4
2. Giới thiệu bài toán:	4
II. Phát biểu hoạt động ứng dụng	4
1. Vấn đề thực tại:	4
2. Ứng dụng thực tế:	4
3. Giải thích vận hành:	7
III. Thuật toán	7
1. Các phương pháp sử dụng:	7
2. Sơ đồ thuật toán:	10
IV. Cài đặt	10
1. Thư viện:	10
2. Ngôn ngữ lập trình :	11
3. Code triển khai:	11

Bảng phân công công việc

Thành viên nhóm	Công việc
Trương Quang Khải	<ul style="list-style-type: none"> - Code và tìm tài liệu về các hàm sử dụng trong xử lý ảnh - Viết phát biểu bài toán trong báo cáo
Trần Trung Hiếu	<ul style="list-style-type: none"> - Code và tìm tài liệu về các hàm sử dụng - Thuyết trình và chạy code demo - Viết phần cài đặt trong báo cáo
Nguyễn Nam Anh	<ul style="list-style-type: none"> - Tìm hiểu code về phân tích và nhận biết biển số xe - Viết phần phát biểu hoạt động ứng dụng trong báo cáo
Quản Trường Huy	<ul style="list-style-type: none"> - Tìm hiểu code về phân tích và nhận biết biển số xe - Viết phần Thuật toán trong báo cáo

I. Phát biểu bài toán

1. Đề tài : *Phân tích và nhận dạng biển số xe*
2. Giới thiệu bài toán:

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó. Xử lý ảnh có nhiều ứng dụng trong thực tiễn như: hệ thống tin địa lý, quân sự, y học, giao thông,...

Cụ thể, trong đề tài này nhóm em sẽ đi chuyên sâu về ứng dụng của xử lý ảnh trong việc giám sát xe và trông giữ xe.

II. Phát biểu hoạt động ứng dụng

1. Vấn đề thực tại:

Hiện nay, các phương tiện tham gia giao thông đang tăng nhanh từng ngày. Những vấn đề liên quan đến mục đích giám sát cũng để thuận tiện cho việc truy xuất và tìm ra biển số xe nhanh chóng khi có sự cố xảy ra rất cần thiết. Chính vì vậy, bài toán trên đã phát triển ứng dụng nhận diện biển số xe là phần không thể thiếu trong hệ thống giám sát phương tiện giao thông.

2. Ứng dụng thực tế:

- Ứng dụng đọc ảnh biển số
- + Đầu tiên ta đưa 1 ảnh biển số xe từ máy tính vào



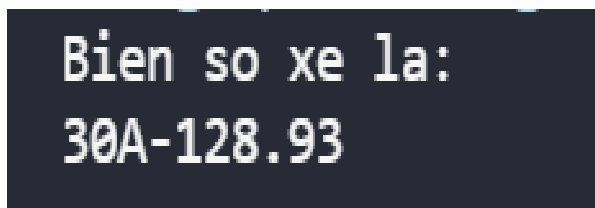
- + Định vị biển số



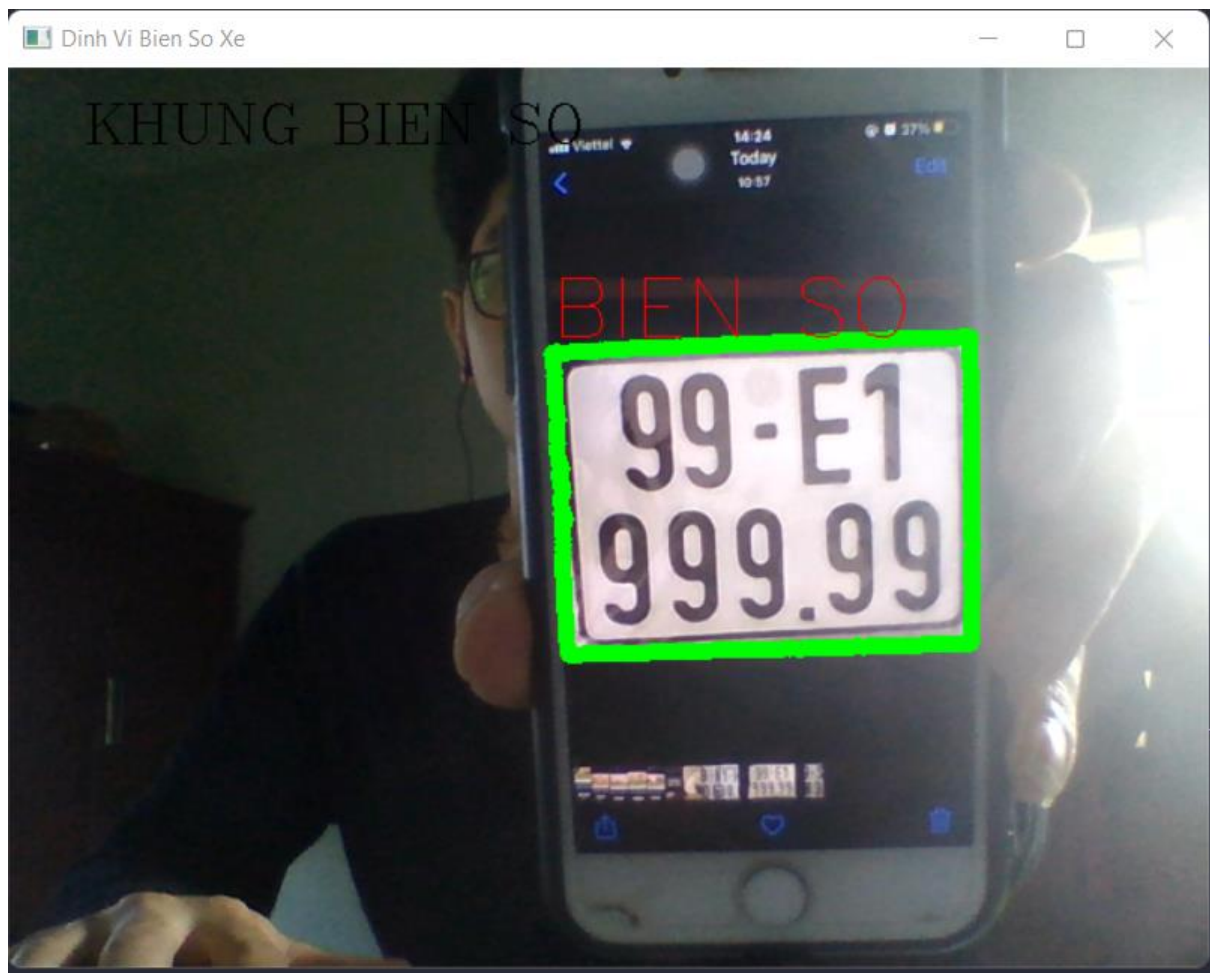
+ Sau khi xử lý ta đc ảnh



+ Hiện thị biển số dạng text



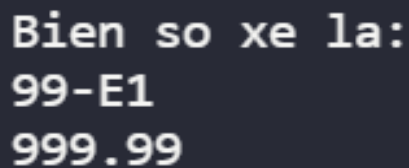
- Ứng dụng đọc ảnh realtime
- + Đầu tiên ta đưa trực tiếp ảnh vào thông qua Cam và định vị biển số



+ Sau khi xử lý ta đc ảnh



+ Hiện thị biển số dạng text



Bien so xe la:
99-E1
999.99

3. Giải thích vận hành:

- Bài toán chạy dựa trên đầu vào sẽ đưa hình ảnh hoặc trực tiếp thông qua camera của máy tính
- Chương trình sẽ sử dụng hàm contours để định vị đúng vị trí của biển số xe để xác định biển số
- Sau đó chúng ta sử dụng các hàm xử lý ảnh và phân tích nhận diện ký tự
- Sau khi xác định được biển số, xử lý và phân tích ảnh biển số xe đó thì chúng ta dùng Tesseract OCR để đọc các ký tự trong ảnh rồi chuyển thành text để giảm công sức đánh máy

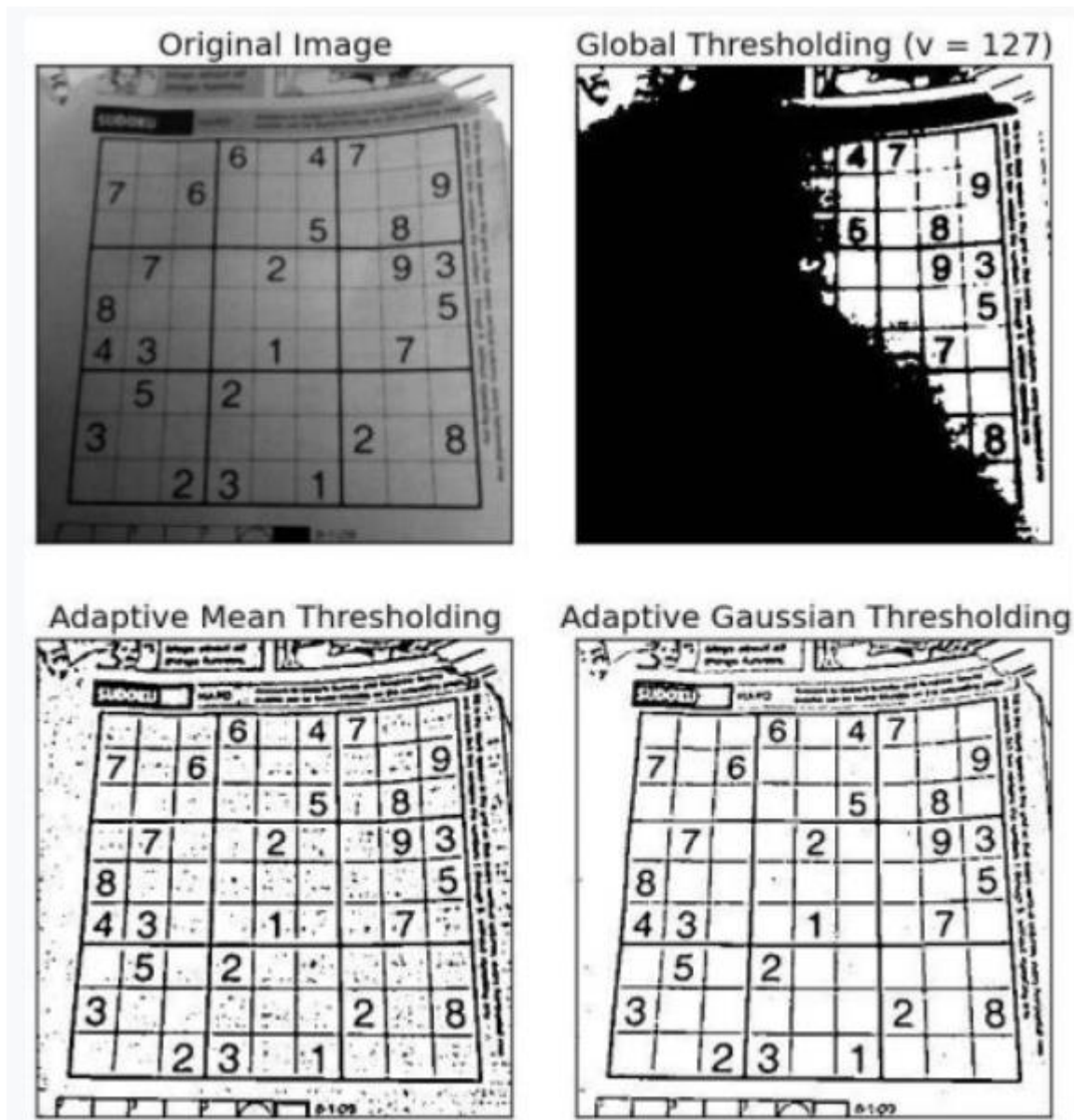
III. Thuật toán

1. Các phương pháp sử dụng:

a. Thuật toán tách ngưỡng (adaptiveThreshold)

Thực tế khi chụp, hình ảnh chúng ta nhận được thường bị ảnh hưởng của nhiều, ví dụ như là bị phơi sáng, bị đèn flash, ...

Một trong những cách được sử dụng để giải quyết vấn đề trên là chia nhỏ bức ảnh thành những vùng nhỏ (region), và đặt giá trị ngưỡng trên những vùng nhỏ đó -> adaptive thresholding ra đời.



Hình ảnh ví dụ về adaptive Gaussian

b. Hàm tìm contour (findContour)

Hàm tìm contour-là “tập các điểm-liên-tục tạo thành một đường cong (curve) (boundary), và không có khoảng hở trong đường cong đó, đặc điểm chung trong một contour là các các điểm có cùng /gần xấp xỉ một giá trị màu, hoặc cùng mật độ. Dùng để phân tích hình dạng đối tượng, phát hiện đối tượng và nhận dạng đối tượng

c. Hàm ApproxPolyDP()

Một thuật toán được áp dụng nhằm giảm số lượng điểm có trong Contour (đơn giản hóa Contour)

d. Hàm `contourArea()`

Hàm tính toán một vùng đường bao. Được tính bằng công thức Green.

e. hàm `boundingRect()`

Hàm tính toán và trả về hình chữ nhật giới hạn phía trên bên phải nhỏ nhất cho tập hợp điểm được chỉ định hoặc các pixel khác không của hình ảnh tỷ lệ xám

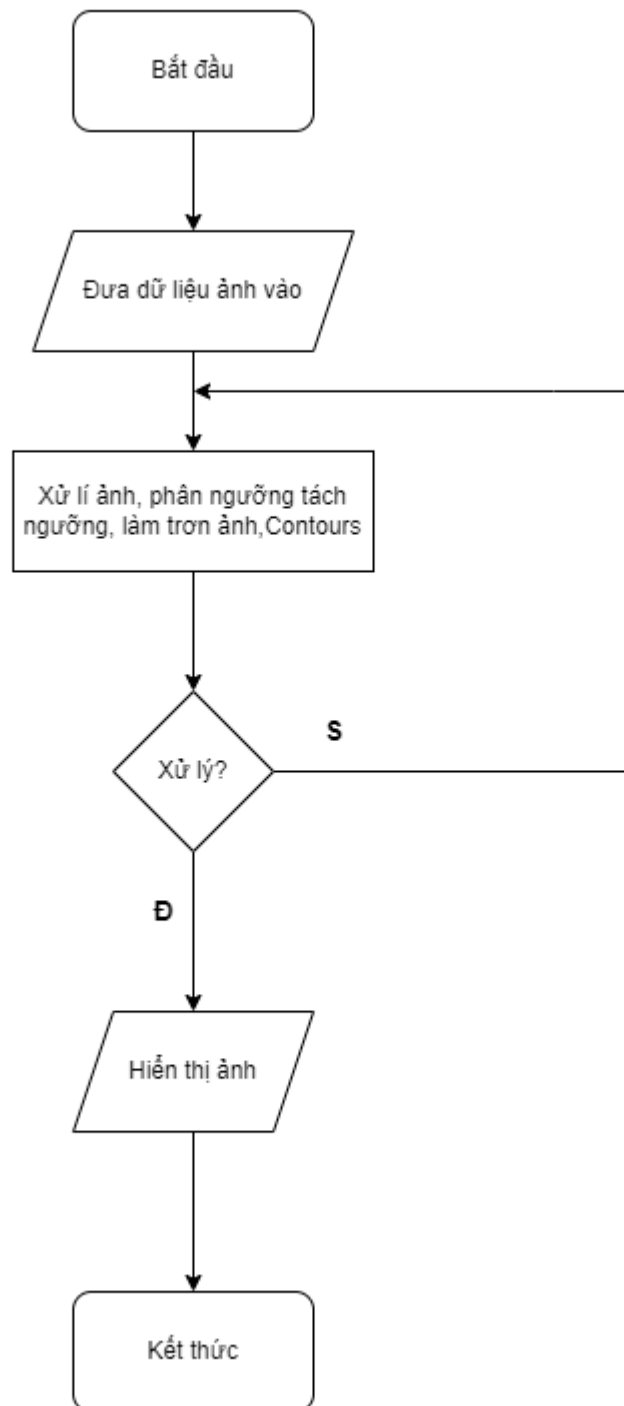
f. Hàm `morphologyEx()`

Phép biến đổi hình thái nâng cao bằng cách sử dụng sự co và giãn nở làm các phép toán cơ bản. Bất kỳ hoạt động nào có thể được thực hiện tại chỗ. Trong trường hợp hình ảnh đa kênh, mỗi kênh được xử lý độc lập

g. Hàm `Gaussian`.

Sử dụng làm trơn ảnh để loại bỏ nhiễu đầu vào.

2. Sơ đồ thuật toán:



IV. Cài đặt

1. Thư viện:

- **OpenCV**

- **Pytesseract OCR** : Nhận diện văn bản tiếng Anh là OCR (Optical Character Recognition). Chuyên dùng để đọc các ký tự trong ảnh rồi chuyển thành text để giảm công sức đánh máy. Trong đó phổ biến nhất là nhận diện văn bản bằng Tesseract.
2. Ngôn ngữ lập trình :
- **Python**
3. Code triển khai:
- a. Ứng dụng đọc ảnh biển số:
- **Đọc và xử lý ảnh**

```
# LOAD THU VIEN VA MODUL CAN THIET
import cv2
import pytesseract

#DOC HINH ANH - TACH HINH ANH NHAN DIEN
img = cv2.imread('5.jpg')
cv2.imshow('Image_Ori', img)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
thresh = cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,11,2)
contours,h = cv2.findContours(thresh,1,2)
largest_rectangle = [0,0]
for cnt in contours:
    lenght = 0.01 * cv2.arcLength(cnt, True)
    approx = cv2.approxPolyDP(cnt, lenght, True)
    if len(approx)==4:
        area = cv2.contourArea(cnt)
        if area > largest_rectangle[0]:
            largest_rectangle = [cv2.contourArea(cnt), cnt, approx]
x,y,w,h = cv2.boundingRect(largest_rectangle[1])

image=img[y:y+h, x:x+w]
cv2.drawContours(img,[largest_rectangle[1]],0,(0,255,0),8)

cropped = image[y:y+h, x:x+w]
cv2.imshow('Xac dinh bien so xe', image)

cv2.drawContours(img,[largest_rectangle[1]],0,(255,255,255),18)
```

- **Chuyển thành file text**

```
#DOC HINH ANH CHUYEN THANH FILE TEXT
pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3,3), 0)
thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
cv2.imshow('Crop', thresh)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=1)
invert = 255 - opening
data = pytesseract.image_to_string(invert, Lang='eng', config='--psm 6')
print("Bien so xe la:")
print(data)
cv2.waitKey()
```

b. Ứng dụng đọc ảnh realtime:

- Đọc và xử lý ảnh

```
# LOAD THU VIEN VA MODUL CAN THIET
import cv2
import pytesseract
#DOC HINH ANH - TACH HINH ANH NHAN DIEN
cap = cv2.VideoCapture(0)
# Bắt đầu một vòng lặp
while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, 2)
    cv2.putText(frame, "KHUNG BIEN SO", (40, 40), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0))
    # Tạo đường viền để theo dõi biển số
    contours, h = cv2.findContours(thresh, 1, 2)
    largest_rectangle = [0, 0]
    for cnt in contours:
        lenght = 0.01 * cv2.arcLength(cnt, True)
        approx = cv2.approxPolyDP(cnt, lenght, True)
        if len(approx) == 4:
            area = cv2.contourArea(cnt)
            if area > largest_rectangle[0]:
                largest_rectangle = [cv2.contourArea(cnt), cnt, approx]
    x, y, w, h = cv2.boundingRect(largest_rectangle[1])

    image = frame[y:y + h, x:x + w]
    cv2.drawContours(frame, [largest_rectangle[1]], 0, (0, 255, 0), 8)
    cropped = frame[y:y + h, x:x + w]
    cv2.putText(frame, "BIEN SO", (x, y),
                cv2.FONT_HERSHEY_SIMPLEX, 1.5,
                (0, 0, 255))
    cv2.imshow('Dinh Vi Bien So Xe', frame)
    cv2.drawContours(frame, [largest_rectangle[1]], 0, (255, 255, 255), 18)
```

- Chuyển thành file text

```
# DOC HINH ANH CHUYEN THANH FILE TEXT
pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3, 3), 0)
thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
cv2.imshow('Bien So La', thresh)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=1)
invert = 255 - opening
data = pytesseract.image_to_string(invert, Lang='eng', config='--psm 6')
print("Bien so xe la:")
print(data)
key = cv2.waitKey(1)
if key == 27:
    break
cap.release()
cv2.destroyAllWindows()
```