TẠP CHÍ KHOA HỌC ĐẠI HỌC HẠ LONG
Scientific Journal of Ha Long University
http://uhl.edu.vn/

ỨNG DỤNG KỸ THUẬT PHÂN NGƯỠNG TRONG XỬ LÝ ẢNH VÀO KHOANH VÙNG CHỮ SỐ ẢNH THẺ ĐIỆN THOẠI

Lê Thị Bích Phượng[1*], Lê Sỹ Phương[1]

[1]Trường Đại học Công đoàn

**Email: phuonglb@dhcd.edu.vn*

Ngày nhận bài: 06/05/2022 *Ngày nhận bài sửa sau phản biện: 21/09/2022* *Ngày chấp nhận đăng: 26/09/20222*

TÓM TẮT

Bài báo này trình bày kết quả nghiên cứu về các phương pháp phát hiện vùng chữ số, đi sâu phân tích kỹ thuật chọn ngưỡng sử dụng trong việc tách các chữ số trong dãy chữ số ở thẻ cào điện thoại. Trên cơ sở lý thuyết về xử lý ảnh, nhóm tác giả đề xuất phương pháp chọn ngưỡng không cố định sử dụng vòng lặp và đưa vào thực nghiệm qua ngôn ngữ Python, sử dụng thư viện OpenCV. Các kết quả thu được nhằm mục đích so sánh, đánh giá hiệu quả của 03 phương pháp: chọn ngưỡng cố định, chọn ngưỡng không cố định và chọn ngưỡng không cố định theo vòng lặp. Từ đó, đưa ra kết luận về tính tối ưu của cải tiến chọn ngưỡng không cố định (theo vòng lặp).

Từ khóa: *chọn ngưỡng, ngưỡng cố định, OpenCV, phát hiện vùng chữ số, xử lý ảnh*

APPLICATION OF THRESHOLDING IN IMAGE PROCESSING INTO FINDING COUNTER OF PHONE CARD

ABSTRACT

This article presents results of a study on methods of detecting digit areas, going into the technical analysis to select thresholds used in separating digits in the number of digits in the phone scratch card. Based on the theory of image processing, the authors propose the method of choosing the non-fixed threshold using the loop and putting into practice through the Python language, using the OpenCV library. The results are for the purpose of comparing and evaluating the effectiveness of 03 methods: selecting the fixed threshold, selecting the non-fixed threshold, and selecting the non-fixed threshold according to the loop. From there, the author drew conclusions about the optimization of the non-fixed threshold (by loop).

Keywords: *choose a fixed threshold, detection of digits, image processing, OpenCV, threshold*

1. ĐẶT VẤN ĐỀ

Những năm gần đây, nhận dạng chữ số trong ảnh đã có những bước tiến vượt bậc, với nhiều ứng dụng được áp dụng như: nhận diện biển số xe xe máy, ôtô trong hình ảnh tĩnh nhận được từ camera chụp hay trong video; xác định số điện qua hình ảnh công tơ điện, đọc số trong thẻ cào điện thoại hay nhận diện số báo danh trong ảnh quét của phiếu trả lời trắc nghiệm. Để phát hiện chữ số trong ảnh, hệ thống nhận diện thường được xây dựng theo quy trình 05 bước: thu thập file ảnh (1), tiền xử lý ảnh (2), xác định vị trí của vùng chữ số (3), khoanh vùng từng chữ số (4), nhận diện và hiển thị chữ số (5) (Viola & Jones, 2020).

Ở đây, trong khuôn khổ nghiên cứu phương pháp nhận diện chữ số trong ảnh thẻ cào điện thoại, nhóm tác giả sẽ đề cập chi tiết hơn đến bước 3 (xác định vị trí của vùng chữ số) và bước 4 (khoanh vùng từng chữ số). Để xác định được vị trí của vùng chứa dãy chữ số trong thẻ cào điện thoại, ta xây dựng bộ trọng số và thử nhiều lần để tìm ra được bộ trọng số thích hợp nhất.

Sau khi định vị được vùng chứa dãy chữ số, chúng ta cần phải phân biệt được các đối tượng cần quan tâm với phần còn lại của ảnh, hay còn gọi là nền ảnh. Những đối tượng này có thể tìm ra được nhờ các kỹ thuật phân đoạn ảnh, tách phần tiền cảnh ra khỏi hậu cảnh trong ảnh. Khái niệm phân đoạn (segmentation) được hiểu là một quá trình chia ảnh ra các vùng con khác nhau mà trong mỗi vùng chứa các thực thể có ý nghĩa cho việc phân lớp – mỗi thực thể được xem là một đối tượng mang những thông tin đặc trưng riêng. Có rất nhiều kỹ thuật phân đoạn ảnh, và trên thực tế, không có một kỹ thuật phân đoạn nào có thể áp dụng cho mọi loại ảnh và cũng không có một kỹ thuật phân đoạn ảnh nào là hoàn hảo. Nhóm tác giả lựa chọn phương pháp phân đoạn dựa vào ngưỡng để định vị các vùng nhỏ chứa chữ số trong thẻ cào điện thoại.

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Phương pháp phân tích tài liệu

Thông qua việc phân tích các tài liệu có sẵn giúp cho đề tài có được những kiến thức cơ bản và nền tảng về thuật toán phân ngưỡng, thư viện xử lý phân ngưỡng trong OpenCV. Nhờ tìm hiểu các công trình nghiên cứu đi trước giúp cho nhóm tác giả nắm được các phương pháp của các nghiên cứu trước đây về phân ngưỡng trong xử lý ảnh, giúp cho bài báo có phương pháp luận và các luận cứ, luận chứng chặt chẽ và xác thực hơn.

2.2. Phương pháp nghiên cứu thực nghiệm

Nhóm nghiên cứu sử dụng ý tưởng chọn ngưỡng nhiều lần để thực hiện bài toán tìm vùng chứa các chữ số đơn trong ảnh và thực nghiệm trên 61 ảnh thẻ với các mệnh giá khác nhau. Bước đầu, ảnh được đưa về kích thước chung, sau đó, định vị vùng chứa dãy chữ số dựa trên bộ giá trị bao gồm: tọa độ điểm bắt đầu vùng, chiều cao và độ rộng của vùng. Và mục tiêu của nghiên cứu là tìm ra phương pháp định vị các vùng nhỏ chứa chữ số.

Thực nghiệm so sánh phương pháp lấy ngưỡng cố định và lấy ngưỡng không cố định (theo vùng), lấy ngưỡng không cố định (theo vòng lặp) để đưa ra kết luận nghiên cứu.

3. NỘI DUNG NGHIÊN CỨU

3.1. Thuật toán phân ngưỡng

3.1.1. Giới thiệu chung

Ngưỡng (Threshold) dùng để chỉ một giá trị mà người ta dựa vào để phân hoạch một tập hợp thành các miền phân biệt (Ngô Diên Tập, 2001).

Trong kỹ thuật này, biên độ của các tính chất vật lý của ảnh (như là độ phản xạ, độ truyền sáng, màu sắc, v.v.) là một đặc tính đơn giản và rất hữu ích. Nếu biên độ đủ lớn đặc trưng cho ảnh thì chúng ta có thể dùng ngưỡng biên độ để phân đoạn ảnh. Thí dụ, biên độ trong bộ cảm biến hồng ngoại có thể phản ánh vùng có nhiệt độ thấp hay vùng có nhiệt độ cao. Đặc biệt, kỹ thuật phân ngưỡng theo biên độ rất có ích đối với ảnh nhị phân như văn bản in, đồ họa, ảnh màu hay ảnh X-quang (Lương Mạnh Bá & Nguyễn Thanh Thủy, 2003).

Việc chọn ngưỡng trong kỹ thuật này là một bước vô cùng quan trọng, thông thường người ta tiến hành theo các bước chung như sau:

° Xem xét lược đồ xám của ảnh để xác định đỉnh và khe. Nếu ảnh có nhiều đỉnh và khe thì các khe có thể sử dụng để chọn ngưỡng;

° Chọn ngưỡng T sao cho một phần xác định trước p của toàn bộ số mẫu là thấp hơn T;

° Điều chỉnh ngưỡng dựa trên xét lược đồ xám của các điểm lân cận;

° Chọn ngưỡng bằng cách xem xét lược đồ xám của những điểm thoả tiêu chuẩn đã chọn.

Một thuật toán đơn giản trong kỹ thuật này là: giả sử rằng chúng ta đang quan tâm đến các đối tượng sáng (object) trên nền tối

(background), một tham số T – gọi là ngưỡng độ sáng, sẽ được chọn cho một ảnh $f[x,y]$ theo cách:

If $f[x,y] > T$ $f[x,y]$ = object = 1

Else $f[x,y]$ = Background = 0.

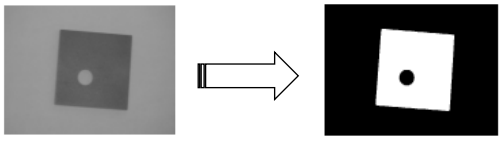
Ngược lại, đối với các đối tượng tối trên nền sáng, chúng ta có thuật toán sau:

If $f[x,y] < T$ $f[x,y]$ = object = 1

Else $f[x,y]$ = Background = 0.

(Lương Mạnh Bá & Nguyễn Thanh Thủy, 2003)

Để dễ hình dung hơn, ta xét một ví dụ bộ lọc ngưỡng (Threshold Filter) đơn giản trong xử lý ảnh. Với mỗi pixel trong hình đa mức xám (grayscale) ở trên giá trị sẽ trong khoảng 0 – 255 vậy pixel nào lớn hơn ngưỡng là 120 ta gán giá trị cho nó thành đen (0), ngược lại gán giá trị trắng (255). Kết quả thu được như Hình 1, hình ảnh được phân chia thành màu đen và trắng:



Hình 1. Hình ảnh trước và sau áp dụng kỹ thuật phân ngưỡng

Vấn đề chính là chúng ta nên chọn ngưỡng T như thế nào để việc phân vùng đạt được kết quả cao nhất. Trong thực tế, giá trị ngưỡng thường được xác định dựa vào những điểm đặc biệt (ví dụ ở trung bình), dựa vào kinh nghiệm khảo sát. Có rất nhiều thuật toán chọn ngưỡng: ngưỡng cố định, ngưỡng không cố định, dựa trên lược đồ, sử dụng Entropy, sử dụng tập mờ, chọn ngưỡng thông qua sự không ổn định của lớp và tính thuần nhất của vùng, v.v.. Trong phần thực nghiệm, nhóm tác giả sẽ phân tích về hai thuật toán chọn ngưỡng đó là chọn ngưỡng cố định và chọn ngưỡng không cố định (Dalal & Triggs, 2005).

3.1.2. Phân ngưỡng trong OpenCV

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở chuyên dùng để xử lý các vấn đề liên quan đến thị giác máy tính. Nhờ một hệ thống các giải thuật chuyên biệt, tối ưu cho việc xử lý thị giác máy tính, vì vậy tính ứng dụng của OpenCV là rất lớn, có thể kể đến như: nhận dạng ảnh (nhận dạng khuôn mặt, các vật thể), xử lý ảnh (khử nhiễu, điều chỉnh độ sáng), nhận dạng cử chỉ và còn rất nhiều ứng dụng khác nữa.

Ta có thể sử dụng nhiều ngôn ngữ lập trình khác nhau để làm việc với OpenCV như: C^{++}, Java, Python, C#, v.v. (Meier, 2015).

Các kỹ thuật phân ngưỡng trong OpenCV:

Sử dụng hàm Threshold

Trong thư viện OpenCV có cung cấp các hàm phục vụ cho việc phân ngưỡng ảnh, bao gồm:

```
ret,thresh1 =
    cv.threshold(img,127,255,cv.THRESH_BINARY)
ret,thresh2 =
 cv.threshold(img,127,255,cv.THRESH_BINARY_INV)
ret,thresh3 =
     cv.threshold(img,127,255,cv.THRESH_TRUNC)
ret,thresh4 =
    cv.threshold(img,127,255,cv.THRESH_TOZERO)
ret,thresh5 =
 cv.threshold(img,127,255,cv.THRESH_TOZERO_INV)
```

Mục đích: dùng để biến đổi ảnh xám src về thành ảnh đen trắng bằng cách sử dụng một ngưỡng cố định.

Cấu trúc:

```
threshold(src, img, thres, max_value, type)
```

Ngưỡng để xác định đen và trắng được truyền qua tham số thres. Ngoài ra, bạn có thể định nghĩa lại màu sáng nhất thông qua max_value (ngưỡng trắng). Hàm threshold hộ trợ một số loại biến đổi như sau:

° type = 0 Threshold Binary: Biến đổi đen trắng, nếu giá trị trên ảnh xám lớn hơn ngưỡng threshold, đấy là điểm ảnh trắng và ngược lại;

° type = 1 Threshold Binary, Inverted: Biến đổi đen trắng ngược với Threshold Binary;

° type = 2 Truncate: Nếu giá trị điểm ảnh lớn hơn thres, giá trị sẽ được đặt lại bằng thres (chặn mức sáng nhất);

° type = 3 Threshold to Zero: Nếu giá trị điểm ảnh nhỏ hơn thres, điểm ảnh đó sẽ trở thành màu đen;

° type = 4 Threshold to Zero, Inverted: Ngược lại với type = 3.

Tùy vào ngưỡng threshold, chúng ta sẽ có kết quả hình khác nhau.

Sử dụng hàm adaptiveThreshold

Bên cạnh việc hỗ trợ các kỹ thuật phân ngưỡng cố định, trong OpenCV cũng hỗ trợ kỹ thuật phân ngưỡng thích nghi (adaptiveThreshold). Để sử dụng, ta gọi hàm:

```
ret,th1 =
cv.threshold(img,127,255,cv.THRESH_BINARY)

th2=cv.adaptiveThreshold(img,255,cv.ADAPTI
VE_THRESH_MEAN_C,cv.THRESH_BINARY,11,2)

th3=cv.adaptiveThreshold(img,255,cv.ADAPTIV
E_THRESH_GAUSSIAN_C,cv.THRESH_BINARY,11,2)
```

Mục đích: Như ở ví dụ trên, chúng ta có thể thấy kết quả phụ thuộc rất nhiều vào ngưỡng threshold. Trong thực tế, đối với các chương trình xử lý ảnh real-time, việc sử dụng một threshold cố định là điều không khả thi do cường độ sáng thay đổi liên tục. Để khắc phục tình trạng này, chúng ta có thể sử dụng hàm adaptiveThreshold do OpenCV hỗ trợ giúp biến đổi ảnh xám src về ảnh đen trắng với ngưỡng sáng tối đa maxValue.

Cấu trúc:

```
adaptiveThreshold (src, dst, maxValue,
adaptiveMethod, thresholdType, blockSize, C)
```

AdaptiveThreshold hỗ trợ 2 phương pháp tự động chọn ngưỡng threshold:

° ADAPTIVE_THRESH_MEAN_C: giá trị ngưỡng là trung bình của vùng lân cận;

° ADAPTIVE_THRESH_GAUSSIAN_C: giá trị ngưỡng là tổng trọng số của các giá trị vùng lân cận với trọng số theo cửa sổ gaussian;

° Cùng với 2 chế độ thresholdType là THRESH_BINARY và THRESH_BINARY_INV. Các tham số blockSize và C dùng để chọn số lượng điểm lân cận để tính ngưỡng threshold tốt nhất.

3.2. Thực nghiệm

Thực nghiệm sử dụng ý tưởng chọn ngưỡng nhiều lần để thực hiện bài toán tìm contour (vùng chứa các chữ số đơn trong ảnh). Ngoài kết quả thực nghiệm của phương pháp này, còn có sự so sánh với các phương pháp chọn ngưỡng khác bao gồm: chọn ngưỡng cố định (Binary Threshold), chọn ngưỡng thích nghi (Adaptive Threshold).

3.2.1. Thu thập dữ liệu

Sau khi chụp để lấy các ảnh thẻ cào điện thoại, ta cắt để sao cho ảnh chỉ còn lại vùng thẻ, không có nền khác. Ảnh chia nhóm theo giá của thẻ. Số lượng cụ thể các loại thẻ thu được để tiến hành thử nghiệm như sau:

Bảng 1. Số lượng mẫu trong thực nghiệm

Nhóm (mệnh giá)	Số lượng thẻ
20.000	21
10.000	14
50.000	14
100.000	8
200.000	4
Tổng số	**61**

3.2.2. Đưa ảnh về kích thước chung

Các ảnh thẻ có mệnh giá khác nhau, sau khi cắt ảnh, ta thấy mỗi ảnh có kích thước khác nhau. Để định vị được vị trí của dãy chữ số, ta đưa các ảnh về chung độ rộng là 600 pixel.

```
image=imutils.resize(image,height=600)
```

3.2.3. Định vị vùng chứa dãy chữ số

Với mỗi loại thẻ khác nhau, ta có bộ trọng số dựa vào chiều cao, độ rộng của ảnh thẻ để lấy ra vùng chứa dãy chữ số. Đầu ra sẽ là bộ giá trị bao gồm: tọa độ điểm bắt đầu vùng, chiều cao và độ rộng của vùng.

3.2.4. Định vị các vùng nhỏ chứa chữ số

Để định vị các vùng nhỏ chứa chữ số, nghiên cứu sử dụng kỹ thuật lấy biên bằng phân ngưỡng với giá trị ngưỡng thay đổi theo các vòng lặp. Với mỗi ngưỡng khác nhau sẽ lấy ra được một số contour. Cho đến khi số contour chưa đủ 13 chữ số (số chữ số trong 1 thẻ cào điện thoại), sẽ lặp lại quy trình: lấy ngưỡng → định vị vùng chứa chữ số.

Ngoài phương pháp trên, để đánh giá được hiệu quả của phương pháp trên, nghiên cứu so sánh với hai phương pháp lấy ngưỡng phổ biến: lấy ngưỡng cố định và ngưỡng không cố định.

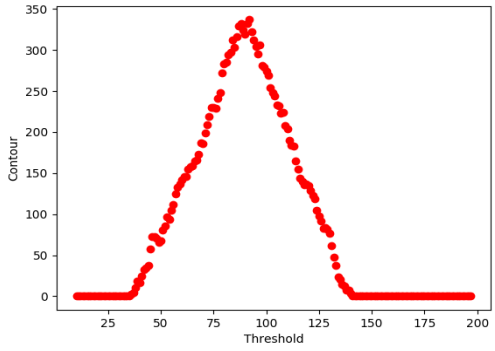
Ngưỡng cố định

Đây là một phương pháp chọn ngưỡng độc lập với dữ liệu ảnh. Tuy nhiên, sau khi đã thử tất cả các ngưỡng có thể, qua kết quả thu được

khi sử dụng các ngưỡng cố định khác nhau, ta thấy ở ngưỡng 92 thu được nhiều contour nhất (số coutour này là tổng của tất cả các contour trong từng thẻ điện thoại có trong thực nghiệm).

Lấy giá trị ngưỡng 92 và sử dụng phương pháp lấy ngưỡng cố định, trong OpenCV ta gọi lệnh:

```
thresh = cv2.threshold(gray, 92, 255, cv2.THRESH_BINARY_INV)
```



Biểu đồ 1. Kết quả thu được khi sử dụng các ngưỡng cố định khác nhau

Ngưỡng không cố định (adaptive threshold)

Với ngưỡng cố định, có thể kết quả thu được không tốt trong tất cả các trường hợp, đặc biệt là khi mà hình ảnh có ánh sáng khác nhau ở các khu vực khác nhau. Trong trường hợp đó, một kỹ thuật được đề xuất, đó là adaptive threshold hay còn gọi là lấy ngưỡng thích nghi. Trong thuật toán này, ngưỡng được tính cho một vùng nhỏ của hình ảnh. Vì vậy, trên một ảnh, các ngưỡng khác nhau cho các khu vực khác nhau. Phương pháp này cho chúng ta kết quả tốt hơn ngay cả khi hình ảnh với ánh sáng khác nhau.

Trong thư viện OpenCV, cung cấp hai thuật toán cho adaptive threshold:

° cv.ADAPTIVE_THRESH_MEAN_C: giá trị ngưỡng bằng giá trị trung bình của các điểm thuộc vùng;

° cv.ADAPTIVE_THRESH_GAUSSIAN_C: giá trị ngưỡng bằng tổng trọng số của các giá trị điểm lân cận được tính qua cửa sổ Gaussian.

Với bài toán này, xét về tổng số lượng contour thu được, Adaptive Thresh Gaussian thu được kết quả khả quan hơn so với Adaptive Thresh Mean (Bảng 2).

Bảng 2. Tổng số contour khi sử dụng Adaptive Thresh Mean và Adaptive Thresh Gaussian

Phương pháp sử dụng	Tổng số contour
Adaptive Thresh Mean	412
Adaptive Thresh Gaussian	506

Ngưỡng không cố định (sử dụng vòng lặp)

Ý tưởng thuật toán: khi sử dụng ngưỡng cố định thì việc xác định giá trị ngưỡng phù hợp là rất khó khăn. Hơn nữa, giá trị ngưỡng phù hợp sẽ khác nhau cho các ảnh khác nhau. Ngay cả khi sử dụng ngưỡng thích nghi, kết quả cũng có thể không được tốt do ảnh quá tối hoặc quá sáng. Trong phương pháp này ngưỡng sẽ được thay đổi qua các vòng lặp. Ban đầu, ngưỡng m sẽ bằng trung bình các pixel trong ảnh cộng thêm 10, giá trị 10 chỉ đơn giản làm ảnh không quá tối hay bị phân đoạn ở những nét chữ mảnh. Sau đó, ngưỡng này sẽ được giảm xuống. Với mỗi ngưỡng khác nhau, ta thu được một số các contour nhất định hoặc không thu được contour nào. Cho đến khi chưa thu được 13 contour thì vòng lặp vẫn được tiếp tục. Vòng lặp dừng lại khi gặp một trong hai trường hợp: định vị được 13 contour hoặc giá trị của $m = 10$

° **Bước 1:** gán m = mean của điểm ảnh + 10

° **Bước 2:** THRESH_BINARY_INV

- $m = m - 1$
- Threshold (với ngưỡng m)
- Lưu các contour vào mảng
- Kiểm tra các contour
  - ✓ Kiểm tra diện tích contour
  - ✓ Kiểm tra hình dáng contour
  - ✓ Kiểm tra chiều cao contour
  - ✓ Kiểm tra vị trí contour

° **Bước 3:** lưu các contour thỏa mãn, cho đến khi nào đủ 13 contour hoặc $m = 10$ thì dừng, nếu không sẽ quay lại bước 2.

Để kết quả chính xác hơn, các contour này được đưa vào các hàm kiểm tra, nếu không thỏa mãn sẽ bị loại bỏ khỏi mảng chứa contour. Các hàm kiểm tra bao gồm:

– Kiểm tra về diện tích: đặc trưng của bài toán nhận diện vùng bao chữ số trong điện thoại là các contour có diện tích gần như nhau. Điều kiện thỏa mãn:

50 < diện tích vùng < 1500

– Kiểm tra về hình dáng: vì tỷ lệ giữa chiều cao và độ rộng của các countour chứa chữ số là gần như nhau và cố định. Điều kiện thỏa mãn:

1,2*độ rộng < chiều cao < 4,5*độ rộng

– Kiểm tra về chiều cao: Nếu có ít nhất 3 vùng con có sự tương đồng về chiều cao với vùng này thì vùng đó thỏa mãn:

° Bước 1:

Cho e = 5 là sai số cho phép với các contour có chiều cao khác nhau;

k = 0 là số contour có chiều cao gần giống contour đang xét.

° Bước 2: với mỗi contour có trong mảng chứa contour lấy ra các đối số x, y, w, h tương đương với tọa độ điểm gốc contour, độ rộng, chiều cao của contour. So sánh với các contour x_1, y_1, w_1, h_1. Điều kiện thỏa mãn:

$$|y - y_1| < e < |h - h_1|$$

Tăng $k = k+1$

° Bước 3: nếu $k > 3$, countour đang xét thỏa mãn.

– Kiểm tra về vị trí: Trong dãy số có 13 chữ số nằm liên tiếp cạnh nhau, vì thế vùng countour thỏa mãn khi nó có ít nhất 01 countour nằm gần.

° Bước 1: cho $t = 6$

° Bước 2: với mỗi contour chứa trong mảng, khởi tạo $k = 0$ và lấy ra mảng rec$[x - t, y, w + 2*t, h]$ so sánh với các contour khác có rec1$[x_1 - t, y_1, w_1 + 2*t, h_1]$, nếu thỏa mãn overlapping_area(rec, rec1)>0: $k=k+1$

° Bước 3: nếu $k > 1$, countour đang xét thỏa mãn.

3.3. Đánh giá kết quả

3.3.1. So sánh tổng số contour thu được trên 03 phương pháp

Khi thực nghiệm trên 61 thẻ điện thoại, tổng số chữ số chính xác hay số contour thu được trong trường hợp tốt nhất sẽ là 61×13=793. Bảng 3 cho thấy trong 03 phương pháp chọn ngưỡng được tiến hành, phương pháp ngưỡng không cố định (theo vòng lặp) tìm ra được nhiều contour nhất 689 contour. Điều này là dễ hiểu khi phương pháp này lấy ngưỡng nhiều lần, mỗi lần có thể không lấy được contour nào nhưng cũng có thể lấy được nhiều contour. Tiếp theo là phương pháp lấy ngưỡng không cố định (theo vùng) thu được 506 contour. Và cuối cùng, phương pháp lấy ngưỡng cố định, dù đã chọn ngưỡng lấy được nhiều contour nhất (ngưỡng 92) nhưng số contour thu được cũng chỉ dừng lại ở con số 337 chỉ bằng khoảng 1/2 khi lấy ngưỡng không cố định (theo vòng lặp). Khi độ tương phản của các chữ số và nền không cao, kỹ thuật sử dụng lấy ngưỡng cố định không thu được kết quả như mong muốn. Qua đây, ta có thể thấy, với bài toán phát hiện chữ số trong thẻ điện thoại, xét về số lượng contour thu được, phương pháp ngưỡng không cố định (theo vòng lặp) là tối ưu nhất, sau đó đến lấy ngưỡng không cố định (theo vùng) và sử dụng ngưỡng cố định là phương pháp thu được số contour nhỏ nhất.

Bảng 3. Đánh giá kết quả

Phương pháp	Số Contour	Tỉ lệ Contour chính xác	
		So với số Contour phát hiện được	So với tổng số Contour
Ngưỡng không cố định theo vòng lặp	689/793	97% (667/689)	84,1% (667/793)
Ngưỡng không cố định theo vùng	506/793	79% (400/506)	50,4% (400/793)
Ngưỡng cố định	337/793	96% (324/337)	40,7% (324/793)

Trong 03 phương pháp lấy ngưỡng, sử dụng ngưỡng không cố định (theo vòng lặp) có tỷ lệ số countour chính xác cao nhất (97%). Đáng ngạc nhiên, phương pháp có số contour chính xác tiếp theo lại là phương pháp lấy ngưỡng cố định (96%) và lấy ngưỡng không cố định (theo vùng) chỉ cố 79% số contour thu được là chính xác.

Về bản chất, lấy ngưỡng không cố định (theo vòng lặp) cũng chính là phương pháp lấy ngưỡng cố định, tính cải tiến ở đây thể hiện ở việc lấy ngưỡng nhiều lần. Vì vậy, tỷ lệ số contour chính xác của hai phương pháp này là tương đương nhau vượt trội hơn hẳn so với lấy ngưỡng không cố định (theo từng vùng). Nguyên nhân có thể đến từ việc phân vùng cố định chưa thực sự hợp lý khiến các con số ở biên giữa các vùng bị chia nhỏ ra, chính vì vậy mặc dù số contour nhận nhiều nhưng chỉ là những mảnh của các chữ số, dẫn đến kết quả không chính xác.

3.3.2. So sánh hiệu quả đạt được khi sử dụng 03 phương pháp

Quan sát bảng 3, so với phương pháp lấy ngưỡng cố định và lấy ngưỡng không cố định (theo vùng), lấy ngưỡng không cố định (theo vòng lặp) thu được nhiều contour chính xác nhất – 667 contour (84,1% số contour chính xác trong 61 thẻ). Điều này đến từ việc lấy ngưỡng nhiều lần, mỗi ngưỡng khác nhau lại lấy được một số contour nhất định và những contour này được lưu lại. Vòng lặp kết thúc khi ngưỡng $m = 10$ hoặc số contour mỗi thẻ cào là 13. Kết quả cuối cùng thu được gộp từ mỗi bước lấy ngưỡng khác nhau.

Phương pháp lấy ngưỡng không cố định (theo vùng) thu được số contour chính xác là 50,4 % (tương đương với 400 contour). Điều này cho thấy đối với bài toán tách chữ số trong thẻ cào điện thoại, phương pháp này chưa thực sự là giải pháp tối ưu.

Mặc dù đã chọn ngưỡng thu được nhiều contour nhất nhưng sử dụng ngưỡng cố định cũng chỉ có 40,7% contour chính xác (tương đương với 323 contour), là phương pháp kém hiệu quả nhất trong thực nghiệm này.

4. KẾT LUẬN

Trong bài báo, nhóm nghiên cứu đã nghiên cứu về các thuật toán phân ngưỡng được sử dụng trong bài toán khoanh vùng các chữ số riêng biệt trong ảnh có chứa dãy số. Sử dụng thư viện OpenCV trong Python để so sánh một số phương pháp chọn ngưỡng trên 61 ảnh thẻ cào điện thoại các loại, cho thấy hiệu quả của phương pháp ngưỡng không cố định (theo vòng lặp) so với 02 phương pháp truyền thống là ngưỡng cố định và ngưỡng không cố định theo vùng.

Trong thời gian tới, nhóm tác giả mong muốn được nghiên cứu phát triển thêm một số chức năng:

° Nâng cao hiệu quả chương trình cách ly các vùng kí tự;

° Nhận dạng chữ số trong các trường hợp thẻ điện thoại bị mờ, ánh sáng môi trường yếu;

° Áp dụng các thuật toán trong học máy để nhận diện chữ số thu được từ các contour.

TÀI LIỆU THAM KHẢO

Dalal, N. & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA.

Lương Mạnh Bá & Nguyễn Thanh Thủy. (2003). *Nhập môn xử lý ảnh số*. Nxb Khoa học và Kỹ thuật Hà Nội.

Meier, B. A. (2015). *Python GUI Programming Cookbook*. Packt Publishing Ltd.

Ngô Diên Tập. (2001). *Xử lý ảnh bằng máy tính*. Nxb Khoa học và Kỹ thuật Hà Nội.

Viola, P. & Jones, M. (2020). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA.