Trường Đại học Công nghệ - ĐHQGHN
Khoa Công nghệ thông tin

BÁO CÁO

GIỮA KỲ XỬ LÝ ẢNH

Mã lớp học phần: INT3404 1

Giảng viên hướng dẫn: Nguyễn Thị Ngọc Diệp

Các thành viên trong nhóm

1. Phan Đức Trung 19020471
2. Lê Văn An 19020205
3. Nguyễn Gia Cát Thành 19020443
4. Lê Trần Lâm Bình 19020226

Hà Nội, 2021

1. Đề bài: Table cell structure detection

Yêu cầu:

- Áp dụng thuật toán xác định ô trong bảng
- Tính toán tỉ lệ phần chồng chéo (iou)
- Tính độ chính xác trung bình của thuật toán

Dữ liệu: bộ dữ liệu có sẵn để test thuật toán.

2. Giải thích cách làm

Đầu tiên, import tất cả thư viện cần dùng:

```
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [9, 8]
import cv2
import numpy as np
import xml.etree.ElementTree as ET
import glob
import os
```

Viết các hàm cần dùng trong quá trình xác định ô và đánh giá thuật toán.

```
def imshow(image):
    if (len(image.shape)==2):
        plt.imshow(image,cmap="gray")
    else:
        plt.imshow(image[:,:,::-1])
def resize(img, ratio):
    width = int(img.shape[1] * ratio)
    height = int(img.shape[0] * ratio)
    dim = (width, height)
    return cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
```

Hàm getCorrectBound() là kết quả chính xác đã có sẵn, được sử dụng để đánh giá thuật toán. Hàm này sử dụng file xml để trả về một mảng có chứa tọa độ của các đường viền cần xác định.

Hàm iou đánh giá độ chính xác của thuật toán, nhận tham số đầu vào là tọa độ các điểm trên trái và dưới phải của đường viền từ thuật toán và từ kết quả chính xác. Hàm này trả về tỉ lệ phần trăm chính xác của thuật toán so với kết quả mong đợi. Thuật toán có hàm đánh giá iou với giá trị lớn hơn 0.5 được xem là một thuật toán tốt.

```python
# get coordinate of correct bounding box
def getCorrectBound(filename):
    name = filename.rsplit('.', 1)[0]
    tree = ET.parse(name + ".xml")
    root = tree.getroot();
    bounds = root.findall('.//bndbox')

    # variable for store bounding box
    bound_coordinates = []
    for bound in bounds:
        # xmin, ymin, xmax, ymax
        coordinate = []
        for tag in bound.findall('./*'):
            coordinate.append(int(tag.text));
        bound_coordinates.append(coordinate)

    #draw bounding box on picture
    image = cv2.imread(filename)
    copy_image = image.copy()

    for bound in bound_coordinates:
        cv2.rectangle(copy_image,(bound[0],bound[1]),(bound[2],bound[3]),(0,0,255),1)
    return bound_coordinates
```

```python
def iou(boxA, boxB):
    # determine the (x, y)-coordinates of the intersection rectangle
    xTopLeft = max(boxA[0], boxB[0])
    yTopLeft = max(boxA[1], boxB[1])
    xBottomRight = min(boxA[2], boxB[2])
    yBottomRight = min(boxA[3], boxB[3])
    # caculate intersection rectangle's area
    interArea = max(0, xBottomRight - xTopLeft + 1) * max(0, yBottomRight - yTopLeft + 1)
    # caculate boxes's area
    boxAArea = (boxA[2] - boxA[0] + 1) * (boxA[3] - boxA[1] + 1)
    boxBArea = (boxB[2] - boxB[0] + 1) * (boxB[3] - boxB[1] + 1)
    # union area = areas - the interesection area
    iou = interArea / float(boxAArea + boxBArea - interArea)
    # return the intersection over union value
    return iou
```

Tiếp theo là thuật toán xác định ô. Trước tiên là khai báo các biến cho quá trình đánh giá thuật toán. Sử dụng vòng for để duyệt qua tất cả các ảnh có trong thư mục train và lấy ra mảng giá trị tọa độ đường viền chính xác.

```
path = 'train/'
iou_arr = []
total_correct_cells = 0
precision_arr = []

for filename in glob.glob(os.path.join(path, '*.png')):
    image = cv2.imread(filename)
    result = image.copy()
    result = resize(result, 4)

    correctCellsImage = 0
    cellsImage = 0

    # correct bound
    bound_coordinates = getCorrectBound(filename)
```

Chuyển ảnh màu về ảnh xám và thực hiện xác định cạnh bằng Canny. Sử dụng opening để nối các cạnh liền kề lại với nhau, sau đó lần lượt chuyển các cạnh thành màu đen theo chiều dọc và chiều ngang.

```
# detect edge
gray = cv2.cvtColor(result,cv2.COLOR_BGR2GRAY)
canny = cv2.Canny(gray, 10, 50)
cannyres = canny.copy()

#remove_horizontal
horizontal_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (40,1))
remove_horizontal = cv2.morphologyEx(cannyres, cv2.MORPH_OPEN, horizontal_kernel, iterations=1)
cnts = cv2.findContours(remove_horizontal, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
    cv2.drawContours(canny, [c], -1, (0,0,0), 4)

#remove_vertical
vertical_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1,28))
remove_vertical = cv2.morphologyEx(cannyres, cv2.MORPH_OPEN, vertical_kernel, iterations=2)
cnts = cv2.findContours(remove_vertical, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
for c in cnts:
    cv2.drawContours(canny, [c], -1, (0,0,0), 4)
```

Sử dụng dilation để nối các vùng liên quan lại với nhau sau đó erosion để đưa các vùng này về kích thước ban đầu.

```
kernel2 = cv2.getStructuringElement(cv2.MORPH_RECT, (12, 6))
connected = cv2.dilate(canny, kernel2, iterations=2)
connected = cv2.erode(connected, kernel2, iterations=2)
```

Sau khi resize kích thước to hơn để dễ chọn kernel cho quá trình xử lý, ta đưa ảnh về kích thước ban đầu. Sau đó, tìm đường viền từ ảnh chứa các vùng của ô rồi vẽ một đường viền lên ảnh ban đầu.

```
n_connected = resize(connected, 1/4)
contours, hierarchy = cv2.findContours(n_connected.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

dt_bounds = []
image1 = image.copy()
for idx in range(len(contours)):
    x, y, w, h = cv2.boundingRect(contours[idx])
    if w > 4 and h > 4:
        cv2.rectangle(image1, (x, y), (x+w-1, y+h-1), (255, 0, 0), 1)
        dt_bound = [x,y,x+w-1,y+h-1]
        dt_bounds.append(dt_bound)
```

Cuối cùng, sử dụng hai mảng tọa độ từ dữ liệu sẵn và từ thuật toán để tính ra iou trung bình, độ chính xác trung bình và số ô xác định đúng (có iou > 0.5).

```
    for bound in bound_coordinates:
        max_iou = 0
        for dt_bound in dt_bounds:
            iou_value = iou(dt_bound,bound)
            max_iou = max(max_iou,iou_value)
        if (max_iou > 0.5):
            total_correct_cells += 1
            correctCellsImage += 1
        cellsImage += 1
        iou_arr.append(max_iou)
    precisionImage = (filename,round(correctCellsImage/cellsImage,2))
    precision_arr.append(precisionImage)
average_iou = sum(iou_arr)/len(iou_arr)
print("Average of iou is %s" %average_iou)
print("Total of correct cells is %s" %total_correct_cells)
sum_pre = 0
for pre in precision_arr:
    sum_pre += pre[1]
average_pre = sum_pre/len(precision_arr)
print("Average precision is %s" %average_pre)
for precision in precision_arr:
    print("%s %s" %(precision[0],precision[1]))
```

3. Một số kết quả

Hầu hết các ảnh đều xác định chính xác ô trong các trường hợp, ví dụ như:

- Trường hợp có đầy đủ đường phân cách:

Tall i mill. kr	2014	2015	2016	2017
Sum inntekter	-2 064,1	-2 073,5	-2 075,0	-2 076,4
Sum utgifter	1 998,0	2 000,8	2 005,8	2 011,1
Brutto driftsresultat	**-170,1**	**-179,0**	**-181,2**	**-181,9**
Netto finans	90,3	92,5	98,2	102,8
Netto driftsresultat	**-79,9**	**-86,5**	**-83,0**	**-79,1**
Finansiering av investering	71,5	74,0	76,0	78,0
Årets resultat	**-8,4**	**-12,5**	**-7,0**	**-1,1**
Avsatt fond	8,4	12,5	7,0	1,1

- Trường hợp không có đường:

	Papirutskrift		Digital innlevering
6 poeng	Wenche Baardsen	6	Einar Bjaanes
5 poeng	Torbjørn Strømberg	5	Einar Bjaanes
4 poeng	Dag Eidet	4	Dag Eidet
3 poeng	Dag Eidet	3	Torbjørg Bergerud
2 poeng	Wenche Baardsen	2	Ole M Holmen
1 poeng	Terje Håheim	1	Wenche Baardsen

- Trường hợp ảnh màu:

	Vehicle Occupant	
Star Ratings	Length (km)	Percent
5 Stars	0,0	0%
4 Stars	10,1	4%
3 Stars	205,5	79%
2 Stars	39,8	15%
1 Star	5,1	2%
Not applicable*	0,0	0%
Totals	260,5	100%

- Trường hợp hàng chia thành nhiều cột:

Account/Transaction Description	Chart of Accounts Codes	
	Tuition	Boarding
Interest – Bank overdraft	2910	2910
Interest – Recurrent loans	2930	2930

- Trường hợp khoảng cách chữ gần nhau:

KIC ID	Obs. Date	Kp	F555W	F775W
2853029	2013-08-12	15.679	16.017	15.006
4139816	2013-04-12	15.954	16.604	15.141
4813563	2012-11-12	14.254	14.602	13.510
5358241	2013-02-04	15.386	15.656	14.902
5942949	2012-10-29	15.699	16.154	14.990
6026438	2013-05-22	15.549	16.075	14.827
6149553	2013-06-12	15.886	17.004	14.812
6263593	2013-02-14	15.037	15.524	14.275
6435936	2013-08-18	15.849	16.846	14.796
7455287	2013-10-04	15.847	16.720	14.837
8150320	2013-09-02	15.791	16.303	14.985
8890150	2013-08-16	15.987	16.853	14.969
8973129	2013-07-07	15.056	15.329	14.455
9838468	2012-10-28	13.852	14.108	13.324
10004738	2014-01-07	14.279	14.563	13.704
10118816	2012-10-27	15.233	16.000	14.226
10600955	2013-02-10	14.872	15.135	14.253
11305996	2013-03-31	14.807	15.519	13.850
11497958	2013-04-06	15.921	16.807	14.805
11768142	2013-07-31	15.931	17.056	14.895
12256520	2013-07-28	14.477	14.805	13.957
12470844	2013-03-19	15.339	15.636	14.695
12557548	2013-02-06	15.692	16.349	14.936

Tuy nhiên vẫn còn một số trường hợp mà thuật toán chưa thể xác định đúng vị trí ô. Vì thuật toán được áp dụng trong tập dữ liệu lớn nên một số trường hợp như khoảng cách giữa các ô là quá nhỏ gây ra việc các ô bị nhóm lại thành một hoặc khoảng cách các chữ trong một ô quá xa cũng sẽ gây ra những sai sót.

Tổng kết lại, trong tập dữ liệu được cung cấp, thuật toán đã xác định đúng vị trí của 3922 ô, tỉ lệ iou là 0.77, tỉ lệ xác định đúng là 0.86, có thể nhận thấy thuật toán đã đạt được một số mục tiêu nhất định và được đánh giá là một thuật toán tốt với iou lớn hơn 0.5.