rectangle_allignment.py

1) Importing libraries. } already explained in rectangle-numbering pdf.
2) def stackImages() }

3) getContours()

→ extra information from that of previous pdf.

↳ within getContours() function:

Code just after:

```
rectangleContours.append(rect_dict)
if contourType == "line":
```

↳ This block will find similar lines within line contours using center, angle & area.

- If difference of center of two line contours are very small, it checks for angle.
- If difference of angle of two line contours are very small, it checks for area
- If difference of area of two line contours are very small, two lines are similar;

So, we append them as list in similarline & delete them from lineContours list.

otherwise, single line is appended & deleted

Also, we need to continuously check for IndexOutOfRange error checking whether 'j' is less than length of current updated lineContours. As we are deleting elements, its length goes on decreasing.

This process is done until all lineContours items are deleted.

After that from similarLineList, we compare area of each line & line with small area is appended to lineContours list.

How i compute x_start, y_start, points for
x_end, y-end rectangles. ~~b~~ horiz-
the rotation of
ontally ?

LOGIC

I am alligning rectangle horizontally at the bottom point i.e. box[3]. So,
x_start = box[3][0]
y_start = box[3][1]

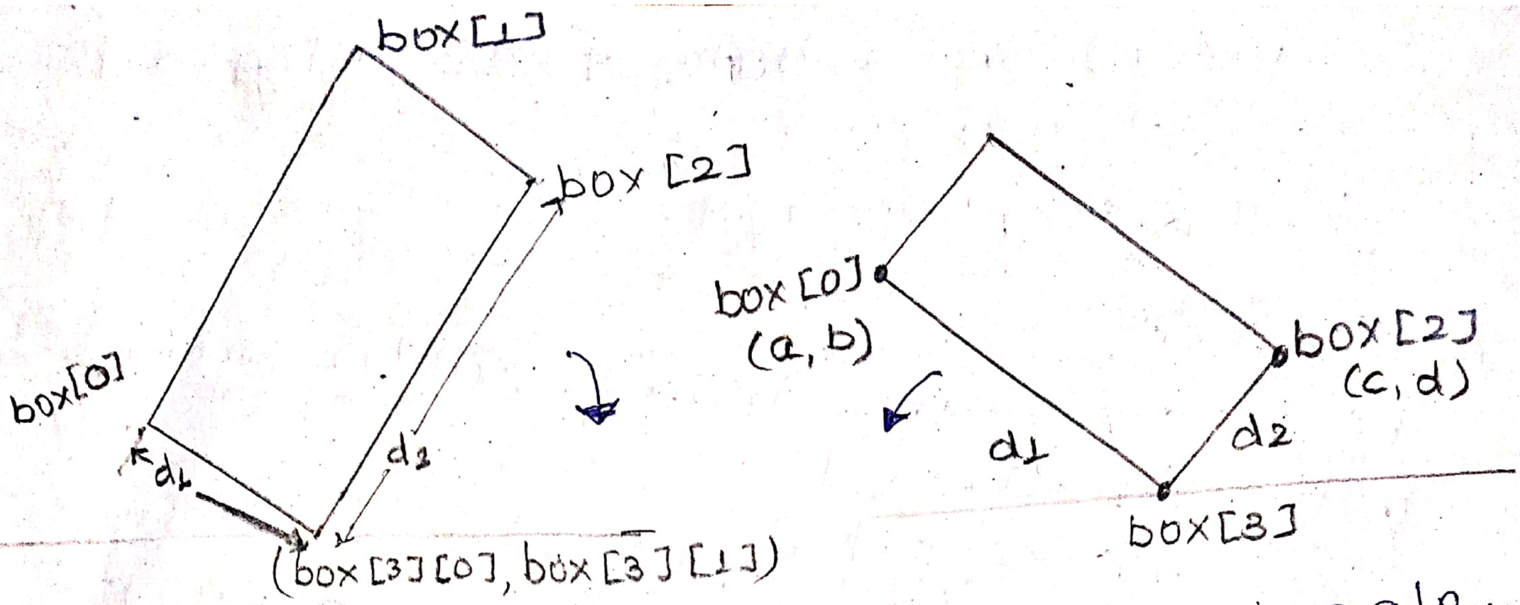


fig. Possible Two orientation of Rectangle.

If $d_1 \ngeq d_2$:

Rotating rectangle in clockwise direction.

x_end = x_start + length
↓
size [0]

y_end = y_start - height
↓
size [1]

If $d_1 \geq d_2$:

Rotating rectangle in anti-clockwise direction.

x_end = x_start - length

y_end = y_start - height

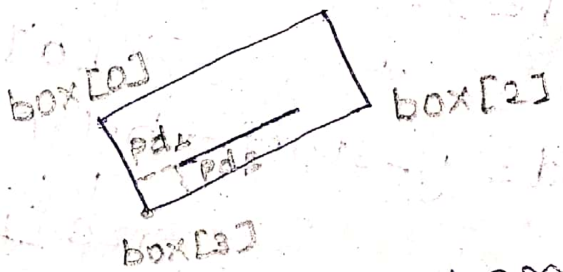
→ Returns list of dictionary or rectangle contours.
line contours

In order to show output of Alligned Rectangles, we use "blank.jpg" image.

After that,
rectangles are drawn with previously calculated
x-start, x-end }
y-start, y-end } points.
so that they are alligned horizontally.

Now, For Rotating Lines lying inside of rectangle, I used following Logic:

- I am going to first compute perpendicular distance from line point that lies at bottom to two adjacent edges connected to vertex box[3]



Since, box[3] is kept fixed & rectangle is rotated. Position of box[3] will remains same in rotated Image too.

Calculation of perpendicular Distance from a point to a line with ~~endpoints~~ equation

$$Ax + By + C = 0$$

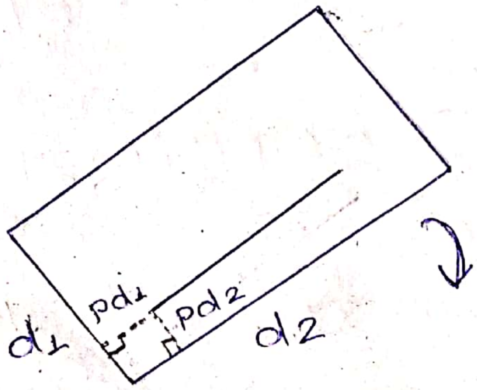
$A = y_1 - y_2$

$B = x_2 - x_1$

$C = x_1 * y_2 - x_2 * y_1$

Perpendicular distance from a point (a,b) ~~(x,y)~~ to line $Ax + By + C = 0$ is:

$$pd = \left| \frac{a \cdot A + b \cdot B + C}{\sqrt{A^2 + B^2}} \right|$$



if d_1 ~~[illegible]~~ $< d_2$:

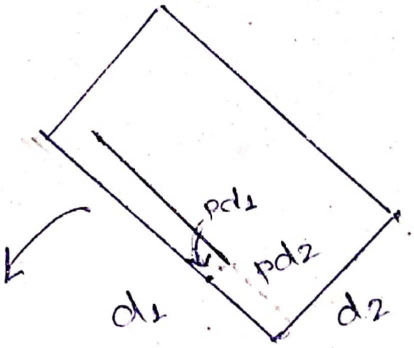
x_start = x_start + Pd_1

y_start = y_start − Pd_2

x_end = x_start + length

y_end = y_start

(Not giving width to line otherwise after ploting it will have greater thickness)



if $d_1 \geq d_2$:

x_start = x_start − Pd_2

y_start = y_start − Pd_1

x_end = x_start − length

y_end = y_start

These were the main logics for this task.
After that I plotted images & list of images in various windows.

THANK YOU !!!