How to use openCV's connected components with stats in python?

I am looking for an example of how to use OpenCV's ConnectedComponentsWithStats() function in python, note this is only available with OpenCV 3 or newer. The official documentation only shows the API for C++, even though the function exists when compiled for python. I could not find it anywhere online.

python opencv connected-components

asked Mar 7 '16 at 21:16

Zack Knopp
524 1 4 10

1 Answer

The function works as follows:

```
# Import the cv2 library
import cv2
# Read the image you want connected components of
src = cv2.imread('/directorypath/image.bmp')
# Threshold it so it becomes binary
ret, thresh = cv2.threshold(src,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
# You need to choose 4 or 8 for connectivity type
connectivity = 4
# Perform the operation
output = cv2.connectedComponentsWithStats(thresh, connectivity, cv2.CV_32S)
# Get the results
# The first cell is the number of labels
num_labels = output[0]
# The second cell is the label matrix
labels = output[1]
# The third cell is the stat matrix
stats = output[2]
# The fourth cell is the centroid matrix
centroids = output[3]
```

Labels is a matrix the size of the input image where each element has a value equal to its label.

Stats is a matrix of the stats that the function calculates. It has a length equal to the number of labels and a width equal to the number of stats. It can be used with the OpenCV documentation for it:

> Statistics output for each label, including the background label, see below for available statistics. Statistics are accessed via **stats[label, COLUMN]** where available columns are defined below.
>
> - **cv2.CC_STAT_LEFT** The leftmost (x) coordinate which is the inclusive start of the bounding box in the horizontal direction.
> - **cv2.CC_STAT_TOP** The topmost (y) coordinate which is the inclusive start of the bounding box in the vertical direction.
> - **cv2.CC_STAT_WIDTH** The horizontal size of the bounding box
> - **cv2.CC_STAT_HEIGHT** The vertical size of the bounding box
> - **cv2.CC_STAT_AREA** The total area (in pixels) of the connected component

Centroids is a matrix with the x and y locations of each centroid. The row in this matrix corresponds to the label number.

edited Jul 4 '16 at 17:14

answered Mar 7 '16 at 21:16

Zack Knopp
524 1 4 10

I must say that for some reason, I had to use cv2.THRESH_BINARY instead of cv2.THRESH_BINARY+cv2.THRESH_OTSU, then I had to cast src to integer and thresh to float in order for it to work. I don't know why, but it didn't work otherwise. – Бојан Матовски Jun 27 '16 at 7:54

I don't understand why you create the labels matrix when it is then part of the output anyway? – ypnos Jul 1 '16 at 14:14

1 @ypnos You don't need to for connected components with stats, but do for connected components without stats. I think that part was just left over from me doing it the other way. I fixed it now. Cheers! – Zack Knopp Jul 4 '16 at 17:13

1 can some one explain how to use the labels? How to check if a centroid is what label? – recurf Dec 7 '16

at 22:28

1 Each component in the image gets a number (label). The background is label 0, and the additional objects are numbered from 1 to `num_labels-1`. The centroids are indexed by the same numbers as the labels. `centroids[0]` isn't particularly useful--it's just the background. `centroids[1:num_labels]` is what you want. – krs013 Feb 25 '17 at 21:43