

FEMB

Take a look at my projects!

People Counter 5 – Morphological Transformations

First of all, sorry for taking so long for this post, I’ve been really busy with work and couldn’t find the time to write this.

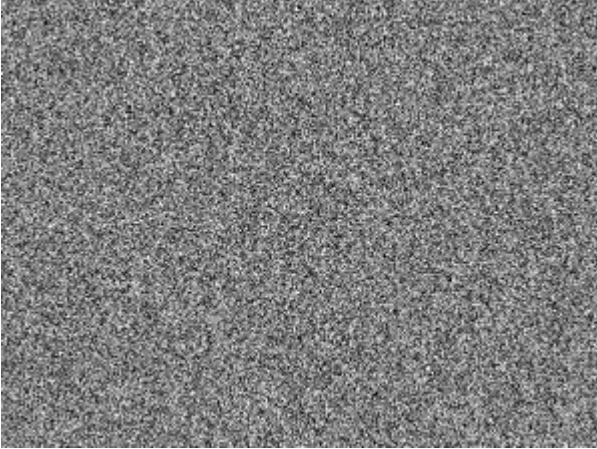
Let’s start! This fifth part of the tutorial will cover basic transformations.

I recommend reading [this](#) and [this](#) first to get an overview of what a transformation is and what it might be used for (You’ll only need to read about Erosion and Dilation , Opening and Closing).

Esentially, we will use erosion and diltation on binarized images (black and white). In a very general manner, erosion expands a black portion of the image into a white portion. Dilation, on the other hand, expands a white portion of the image into a black portion.

To do this operations you also need to specify a kernel or structuring element (strel). This is a matrix that is convoluted on the image of size n*n that defines the area to use when calculating the value of each pixel.

Let try them on this image:



Use this code:

```
import cv2
import numpy as np

img = cv2.imread("noise.png")
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)

kernel = np.ones((3,3),np.uint8)

erosion = cv2.erode(img,kernel,iterations = 1)
dilation = cv2.dilate(img,kernel,iterations = 1)

cv2.imwrite("erode.png",erosion)
cv2.imwrite("dilate.png",dilation)
```

We’re using the **threshold** method to binarize the image. Convert it from color to only black and white, two values (not the same as grayscale). Look at how it works [here](#).

Run it and look at the output images. Also, try changing the kernel’s size (5,5 or 9,9, for example) and see what happens.

You can also combine both operations, running one after the other.

Doing and erosion and then a dilation is called **opening**.

On the other hand, doing a dilation and then an erosion is called **closing**.

Let’s try them on this image:



Use the following code:

```
import cv2
import numpy as np

img = cv2.imread("letters.jpg")
ret,thresh1 = cv2.threshold(img,200,255,cv2.THRESH_BINARY)

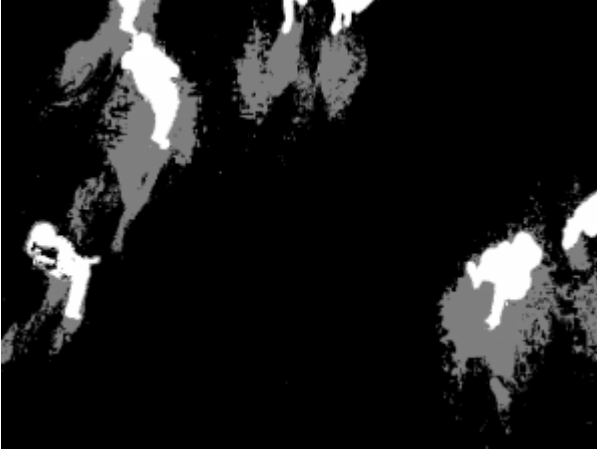
kernel = np.ones((5,5),np.uint8)

opening = cv2.morphologyEx(thresh1, cv2.MORPH_OPEN, kernel)
closing = cv2.morphologyEx(thresh1, cv2.MORPH_CLOSE, kernel)

cv2.imwrite("letters_closing.png",closing)
cv2.imwrite("letters_opening.png",opening)
```

Look at the output and try to understand what happened during closing and opening.

Now try incorporating this code to our people counter, right after background subtraction, to take the shadows (gray color) out and make the video stream clear (take out any noise), to take it from this:



To this:



It’s right to even try combining opening and closing operations, even with different sizes of kernel. Experiment with them!

You can also use this operations on color images to get cool results, although that’s not useful for the counter.

(Click on the images to view them full size)

Original	Opening (9,9)	Closing (9,9)

That’s it for now. If you have any questions leave them in the comments.

Next, we’ll look at finding contours from the clean clean image to later detect them as moving people.

3 thoughts on “People Counter 5 – Morphological Transformations”

- luis.ibarra

September 13, 2016 at 12:27 pm

Hey bro, nice tutorial. I’m also from México and I’m working on a similar project. Look, i have already realized BS, Filtering, contours detection and a basic blob tracking system. However i still have some false–positives which i would like to reduce to o. I’m interested on the “Defining a person” section you mention over your first post. Could you please give me a hand on this? i’d appreciate it.

[Log in to Reply](#)
- Titus

October 13, 2016 at 10:40 am

Hello Fede,

Your guide is so nice, able to make a python script upto filtering, as we know, other steps are very important to do the counting (IN and OUT) in this project, so please help me on this.

Can you please update the step 6,7,8 and 9 ??

If you have very minimal script with simple documentation for rest of the steps, can you please send me an email ?

