

A simple detector for crowd counting using OpenCV and Python

Martí Gelabert Gómez

November 14, 2022

Contents

1	Introduction	1
2	procedure	1
3	algorithms	2
3.1	Background Removal	2
3.1.1	Image Averaging	3
3.1.2	Image Substraction	3
4	Gabor Filter	3
5	What form do the People have?	3

1 Introduction

The assignment consists in **counting** the number of people that appear on the given images. There was not an established way to approach this problem. In this case I have taken the decision of using the computer vision algorithms seen in class and not rely on deep learning or any kind of artificial intelligence algorithm. This is because not by de amount of data needed for training, due to there are really god datasets out there, like MS COCO, but because of the effort of building a good performing detector architecture. Use an already set framework as YOLO or Mask-RCNN would be the way to accomplish the task, but I would be thinking about it as "cheating". Therefore, this way, using the content seen in class, I will be more cautious about my decisions and It will be much easier to justify them.

In the following document we will be focusing on the process taken and the algorithms used, their implementation, their behebeours we obtained and their performance.

2 procedure

The program takes the following steps :

1. Import images as black and white.

2. Apply Histogram equalization to the images.
3. Generate an image from the averaging from all the images we have with the Histogram equalization already applied.
4. Subtract the background to the images using the average image.
5. Apply a thresholding algorithm to binarize the image.
6. Apply a dilation operation into the binarized images to expand the whites.
7. Use a contour algorithm to extract the different regions containing persons.
8. Obtain the bounding boxes from these regions.
9. Count them and compare the number of detections to the real cuantity.

3 algorithms

In this section we will discuss the algorithms selected and the output we obtain from them in the application of our problem.

3.1 Background Removal

Background removal is a technique that allows to remove the background from the image, this way, the output will be only the foreground in the form of a binarized image.

THRESH_OTSU funciona bien salvo en los casos donde empieza a haber sompras en la imagen, a la hora de aplicar el thresholding obtendremos un número muy grande de falsos positivos en las pequeñas montaás que hace la arena, las dimensionas que tienen son similares a las de una cabeza humana por lo que si las eliminasemos estariamos eliminando buenas detecciones.



Figure 1: Binarized image using OTSU and the empty beach image

3.1.1 Image Averaging

3.1.2 Image Substraction

4 Gabor Filter

A band pass filter generated by a function of various parameters.

$$filter(x, y; \sigma, \theta, \lambda, \gamma, \phi) = \exp\left[-\frac{x^2 - \gamma^2 \cdot y^2}{2\sigma^2}\right] \cdot \exp\left[i\left(2\pi\frac{x}{\lambda} + \phi\right)\right] \quad (1)$$

The parameters of `ksize` allows to select the size of our kernel filter, in case we are using a really big shape we will overlook details if the shapes are small. The same reasoning can be applied with a small filter, may overlook shapes too big for it. Therefore, must be tested with different sizes to reach an idoneal spot, if your features are tiny or bigger, you have to take that in count.

If we are looking for **horizontal-like** features, applying an horizontal filter will allow us to maintain those characteristics and block the vertical ones and viceversa in the other cases.

5 What form do the People have?

In general the shapes that a person can describe may suffer alot of distorsion depending of the angle where the frame was took, the person posture, etc. Not always will be a perfect pose to the researchers to easily identify if an object is a person or not. There is no key shape that we could use, but we can try use common sense. The images gathered, are related to people moving standing up, swimming or just taking sunbathing.