Chelomo Lubliner

Code support for Master Thesis

October 2023

1. **Initialization**

Relevant Files : globals.py, create\_files\_shoes.py

**globals.py**: We have to choose if we are currently working with the Old or the new Dataset, we just have to change the first line OLD\_DATASET = True/False.

**create\_files\_shoes**: We are uploading the contacts\_data files and transform the list of 0/1 to an array of matrixes that correspond to the shape of Old/New dataset, and convert then to images via PIL.Image.fromarray(). The folder shoes will be saved in contacts\_data() function.

We will also get a superposition of all shoes, the superposed\_pixels that means images that contains only pixels that occurs at least i times. Superposed\_pixels\_reversed is the images that contains pixels that occurs max i times.

1. **Remove Noise**

Relevant Files : remove\_noise.py, contour.py

We define a prototype image that is superposed image of pixels that occurs at least 18 times.

For removing the noise, we do a bitwise AND between every image and the prototype image, the save the new cleaned image in Cleaned\_Shoes folder. Superpose\_image\_prototype() function is for visualization of which pixels of the image we keep and which pixels will be deleted.

These functions will be called in the contour.py file.

1. **Get Contour**

Relevant Files : extreme\_values\_x\_y.py, contour.py

We iterate over every X and get min and max Y. Then, iterate over every Y and get min and max X. The union of the two results give us the extreme values. dicts\_points() return a dictionary of key=X (or Y) and all the Ys (respectively Xs) that corresponds to this X (or Y). Then we select the min\_max.

contour.py first do the remove noise, and then calculate the extreme\_values and save them into folders Extreme\_Values.

1. **Convex Hull**

Relevant Files : convex\_hull.py

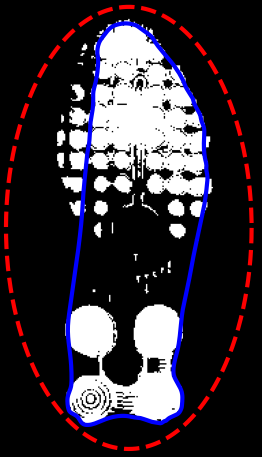
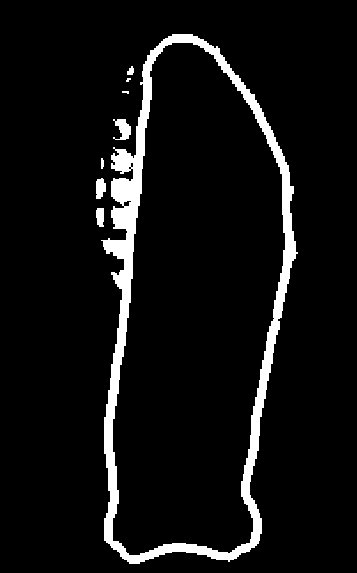
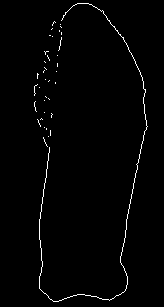
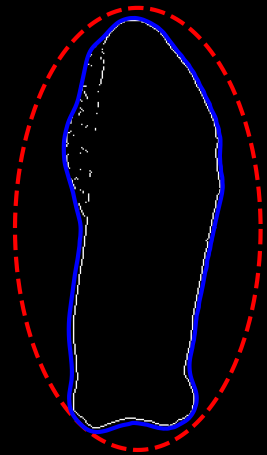
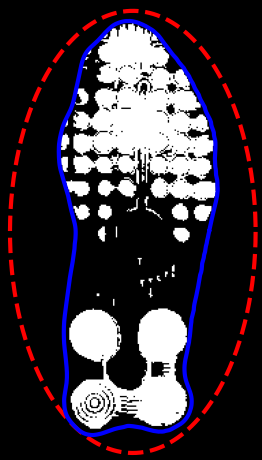
Only one convex from scipy.spatial, on the entire shoe. Get the points of the convex, and add them to the shoe pixels of the shoe. Save the convex in Convex\_Hull\_html folder. We note that convex hull encompasses all points, so the result will not be different if we run it on all the shoe or only on the extreme values. Since it’s computationally better to run on less points, we will run this algorithm on the extreme’s values.

1. **Active-Contour Snake**

Relevant Files : active\_contour\_snake.py

This is the algorithm that give us the best contour of a shoe. There are several steps:

* Get the image from cleaned\_shoes -> get\_image()
* First implementation of active\_contour
  + init\_snake() gives the initial shape that will fit to the shoe.
  + first\_snake = active\_contour()
* combined\_snake\_arr()
  + get image from the snake -> get\_snake\_image()
  + get extreme\_values of the snake (because it’s too thick)
  + get the part of the shoe that is out of the first snake (out\_first\_snake\_arr)
  + combine the out\_first\_snake\_arr and the snake\_arr. Then calculate the extreme values a second times.
  + This image comb\_extremes\_images.
* Second\_snake
  + Run a second time the active-contour algorithm but on comb\_extremes\_images and not on the original image. With differents parameters than the first time (Beta = 0.10)
* Save the original image with the second snake.

1. **Distance of RAC**

Relevant Files : extreme\_values.py

We have now two methods to get the contour of a shoe : Convex-Hull and Active-Contour. The locations\_data file contains the RACs, for each shoe, and their coordinates x,y. First, we need to convert the x/y coordinates to pixels that corresponds to the rows and columns. Then We will calculate the difference between each RAC and the extremities. For each method (Convex/ Snake) we will calculate both the horizontal distance (the closest point (Xi,Yi) where Yi = YRAC ) and the shortest distance. aspix\_x() and aspix\_y transforms a x/y coordinate to a row or column number. init\_locations\_new() upload the original locations\_data file and add new columns, convert x/y to row/columns. Since the locations\_data file start at shoe N.1 I removed one for each number shoe because I’m starting with shoe N.0. this function will also divide by two the coordinates x/y for the new dataset because the correct interval is: x\_coordinates ϵ [-0.25,0.25], y\_coordinates ϵ [-0.5,0.5]. For the horizonal distance, we need only the two points that are at the extremities, but sometimes there are several points one after the other, so we will take only the points that don’t follow each other. To check if a RAC is inside the shoe or not, we can multiplicate the distance of the two points and we are supposed to get a negative result, because there are on an opposite side. The 4 distances are: HORIZ\_DIST\_SNAKE, DIST\_SNAKE, HORIZ\_DIST\_CONV, DIST\_CONV.

The conclusion of this part is in Images/[New\_Shoes|Old\_Shoes]/Saved/locations\_new.csv