International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 2 pp. 9-11 April 2020 - Sept 2020 www.csjournals.com

Social Distance Scoring of an Image

Uma S.U[1]

Data Scientist[4], Freelancer, Bangalore, India

Abstract: Based on the recent pandemic of coronavirus, social distancing is a major part of containment measures to stop the spread. A question was raised to see if we can score an image containing multiple people based on how good they are following social distancing. This algorithm is developed to do the same.

Keywords: image distance, social distancing, scoring.

1. Theory

Finding the distance between any two objects in a flat plane like an image where no depth data is present is pretty difficult. We need to find an object in the image whose dimension are known in real life and then approximate every other distance based on it.

Similar to above case, calculating social distancing score requires you to find the distance between people in animage.To achieve this, we took some approximations interms of human body dimension and posture,then we calculated the ratio and used it as base reference for scoring.

Distances can be categorized into 2 types. Distance between the people in parallel plane to the camera sensor and distance between people perpendicular to camera sensor (depth). Distance in plane parallel to camera sensor is referred as horizontal distance , distance perpendicular to camera sensor is referred to as verticle distance.

To calculate horizontal distance, we can use the ratio of human bodies to approximate. In case of verticle distance we used the ratio between face areas to approximate the distance and score them

2. Method

Python language is used to write this program and github[2] link can be accessed here

(https://github.com/frier-sam/Social_Distance_Scoring)

Face-recognition[3] package is used to find the co-ordinated of faces in a image

2.1 Calculations

Based on global averages, human face is 7.4 inches wide and human body is 14.4 inches wide from shoulder to shoulder. To follow social distancing , there needs to be atleast 6ft or 72 inches of distance between the people

If we approximate body width, human body is 2.1 times larger than face width. So if we know the width of the face , we can calculate width of body from shoulder to shoulder (with assumption that they are standing still)

Since 2 people need to maintain a distance of 72inches , that can be calculated interms of ratio to average face width adding shoulder buffer, Calculating this will give us a result of **9** (Approx) .So if 2 people are at a distance of 9 times the averagewidth of their faces, we can assume they are following social distancing. Since everything is relative here, we can directly calculate the pixel lengths and calculate the ratio to compare and score. Scoring is done form 0-10 , 0 represents ratio more than or equal to 9 while 10 represent ratio less than 2

For verticle scoring, Globally adult humans have avariation of 20% interms of their face area, so this is used as base to see if there is a verticle distance that is worth calculating. Getting the percentage of one face w.r.t other (large /small) we can see that larger this ratio, the closer are they are (1 signifies they are on same plane) . These percentages are bucketed to give a score from 0 - 10.

2.2Explanation

On input of a image, it runs through face-detection module

In face-recognition package. This gives us (top,left) ,(bottom,right) coordinates of all the faces as an array.

Faces are sorted based on the left coordinate to get an array of face coordinates from left to right in order. Pick any 2 adjacent faces in the array to calculate score. First thing to check is if the faces overlap from left to right (i.e right coordinate first one face < left coordinate of next face). If they do overlap, then we need to calculate verticle scoring , nor we can go for horizontal scoring . There's a check inside horizontal scoring method to see if the face area ration>20 to decide if the scoring needs to be done in verticle method.

Calculations are done based on the above fashion and a score is calculated for these two faces. After iterating through the whole array , taking 2 adjacent faces at a time, all the scores are averaged to get final score of the image.

This whole application is wrapped as flask application for easier use with a user interface for easier testing

2.3 Logic.

Image input → Find all the faces and their four coordinates → Sort faces from left to right(array k)

If length k > 100

Score 10

Else →For every 2 faces from left to right in k

Calculate Average face length,

Calculate distance between faces,

Calculate area of faces

If different in face areas > 20%

Vertical score the faces

Else

Horizontal score the faces

Return average of all the scores.

2.4 Code

This logic is extended as a python flask application and available on github at this location

https://github.com/frier-sam/Social_Distance_Scoring

Follow the instructions on github to run the application

2.5 caveats and Improvements

Faces needs to be clearly visible in the image to be recognised. This can be mitigated by using packages like Open Cv[5] to abstract the face detection and find the faces even when turned all the way around

Acknowledgements

Contributors for open source package face-recognition that is used in this project

References

[1] U.Uma, Data Scientist

[2] Github , Open Source git repository [source]

[3] Face – Recognition , Open Source package [source]

[4] Data Scientiest, Job Description [source]

[5] Open Cv , Open Source package [source]