PEDESTRIAN DETECTION AND COUNT FOR THE NUMBER OF FRAMES IN AN IMAGE

INTRODUCTION:

* In this era of technology, we are facing many problems arising due to the population spurt with overcrowding being one of them. The lack of crowd control leads to disasters and ends up causing a lot of casualties. Coming up with solutions to keep track of the number of people is not only a requirement but a necessity .With the help of computer vision we’re able to prevent such dire circumstances. Data manipulation and analysis is a major step towards coming up with a good solution against this problem. The program we have come up with will help with counting the number of relevant objects in each dataset. Here we specifically target human beings as objects. There are many applications for such a software. Detection is the first step in prevention. From a small robbery to saving a person’s life, it can have applications in all areas of life. It can be used for military, security, health care, education and so many more purposes. It can be used in security systems in households specially with elderly people who live alone.Family members can track their acitivity in case something bad happens or in case of a robbery they can be alerted immediately. It can be used in banks to track the number of people in specfic areas of a bank.It can be used in military bases where number of people allowed to be there is limited. With covid-19 a surge of crowds has been seen in hospitals this can be managed better using this software.Head count systems can be improved using this software which is a boon for the education department. There are many other applications like counting the number of students traveling in a bus on a trip. Here counting will help to keep the track of students when they get off a bus. Pedestrian detection can be used as an application in large universities that provides information as to how they can divide students in the food courts, assembly sessions, etc. These applications provide an easy manner to maintain the universities.

A crowd of people in a city

Description automatically generated

CASE STUDY:

In this section of the paper two recent tragedies have been discussed – The Morbi bridge tragedy and the Seoul Halloween stampede.

* The Morbi bridge tragedy

On 30 October 2022, five days after reopening, the Morbi bridge collapsed at 6:40 p.m. More than five hundred people were on the bridge at the time of the collapse, far exceeding the official capacity of 125. Security footage of the bridge showed the structure shaking violently and people holding onto cables and fencing on either side of the bridge before the walkway gave way. Images of rescue and recovery operations showed the walkway had divided at its midpoint, with some pieces still hanging from snapped cables. A survivor said that so many people on the bridge could barely move and that pieces of the bridge crushed some victims.

So, the aim of this software would be to not only prevent this sort of overcrowding but also help with the headcount of the number of victims in case something like this does end up happening.



* The Seoul Halloween stampede

On the night of 29 October 2022 a crowd rush occurred during Halloween festivities in the Itaewon neighbourhood of Seoul, South Korea. At least 159 people were killed, and 196 others were injured. The victims were mostly young adults. The pedestrian count software could’ve easily collected the overcrowding data and that could’ve helped the police in preventing this tragedy with better crowd control.



ABSTRACT:

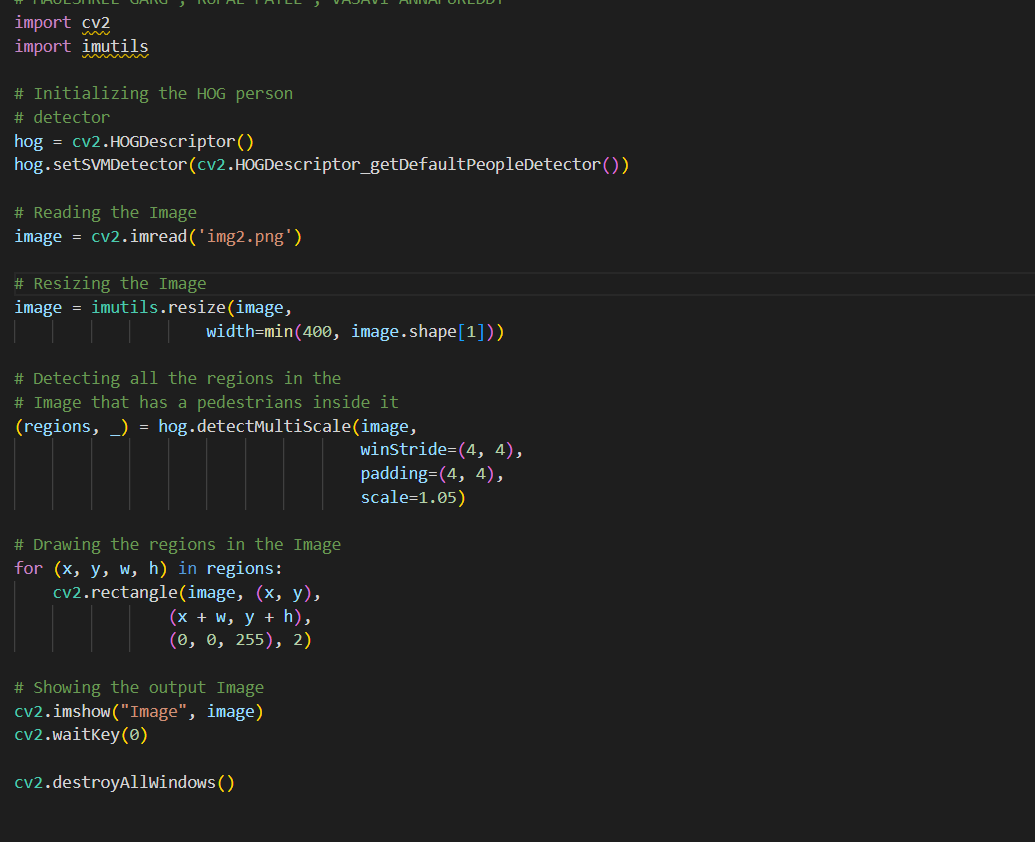
This paper describes a vision based pedestrian detection and tracking system which is able to count people in very crowded situations like escalator entrances in underground stations or on streets. The proposed system uses certain specific geometric features to compute regions of interest and prediction of objects, extracts shape information from the image frames to detect individuals and applies texture features to recognize people. A search strategy creates trajectories and new pedestrian hypotheses and then filters and combines those into accurate counting events. We show that counting accuracies up to 79 % can be achieved.

Every human being has certain features in common and the software can be trained using those common features as a reference point. This project aims at developing and evaluating the performance of a novel counting system developed for environments with high pedestrian flows .Using an image acquisitions device we get our input data and we give an apt output based on algorithm.

ALGORITHM:

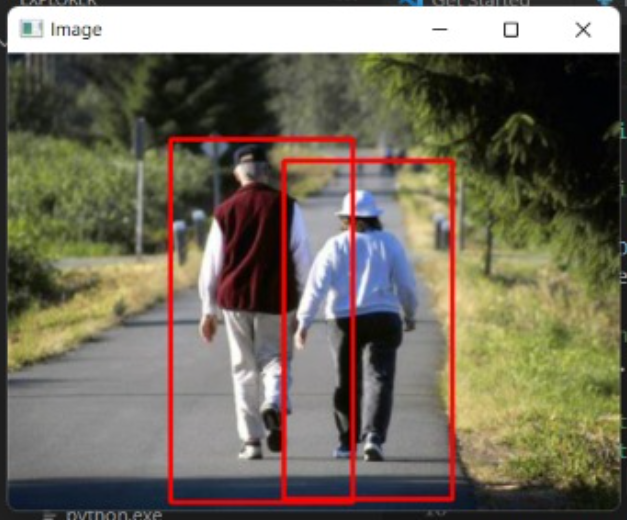
* Step-1: Start
* Step-2: Import cv2 and imutils packages to python.
* Step-3 : initialise the hog detector that is imported through the imutils package and use the getdefaultpeopledetector() function to detect the shape of a human being in the image().
* Step-4: Read the image by using imread function and storing it in the variable called image.
* Step-5: resize the image to the size required for processing.
* Step-6: Use a method to detect all the regions in an image by using region splitting techniques using multiscale. This way the software will detect the number of objects.
* Step-7: CV2 import will be used to draw a red rectangular box around the object that has been identified.
* Step-8: the output image is displayed.
* Step-9: Stop

PROGRAM:



RESULT:





CONCLUSION:

The result proves that our algorithm is able to perform the counting application with a high accuracy rate. This can very well be used to replace traditional tracking methods. Detecting the people will help keep the crowded areas safe. The properties should be implemented to detect in a very fast, robust manner.

REFERENCES:

1. <https://ieeexplore.ieee.org/abstract/document/4020729>
2. <https://en.wikipedia.org/wiki/2022_Morbi_bridge_collapse#:~:text=On%2030%20October%202022%20a%20pedestrian%20suspension%20bridge,New%20Year%2C%20following%20a%20lengthy%20closure%20for%20repairs>.
3. <https://en.wikipedia.org/wiki/Seoul_Halloween_crowd_crush>
4. <https://www.geeksforgeeks.org/pedestrian-detection-using-opencv-python/>