

INTRODUCTION

In this python project, we are going to build the Human Detection and Counting System through Webcam or you can give your own video or images.

Real-Time Human Detection & Counting is a system which will help in many places to optimize and control security. This system is mainly used to detect humans and count them. This system can be used in Malls, Temples, Railway Stations, Airports and in security systems. By using this system, we can find out as well as stop the suspicious activities or abnormal events.

Over the recent years, detecting human beings in a video scene of a surveillance system is attracting more attention due to its wide range of applications in abnormal event detection, human gait characterization, person counting in a dense crowd, person identification, gender classification, fall detection for elderly people, etc.

The detection process generally occurs in two steps: object detection and object classification. Object detection could be performed by background subtraction, optical flow and spatio-temporal filtering. Background subtraction is a popular method for object detection where it attempts to detect moving objects from the difference between the current frame and a background frame in a pixel-by-pixel or block-by-block fashion.

Literature Review/Related work

For this project we need some resources/files to train the model and it also help in minimize the work and time. for all the necessary files we are going to save in project named folder. We are using the google collab platform that enables us to escape the hardware problem but still we have to upload and download the input and output videos. Also, we are using the command prompt to run our project and see the output.

1. International Research Journal of Engineering and Technology (IRJET)
(Mrs.R.Preethi, Gollapudi Venkata Praneeth, Evani Sai Sarvan Kumar Sarma)
2. IOSR Journal of Engineering (IOSRJEN)
(G.Thomas Prathiba , Y.R.Packia Dhas)
3. IEEJ detection counting and Systems 122(12):2011-2019
(Hiroshi Matsuo - Nagoya Institute of Technology)
4. GE Global Research Niskayuna, NY 12309, USA
(J. Rittscher A. G. A. Perera N. Krahnstoever)

Proposed Work and Objectives

In this project first we have Download some sample video for the road detection. Make sure that video must be in mp4 format because it easy to preprocess. After that we need to upload the that video on the google colab. The next step is the to create a model for the road detection

Now we need to preprocess the data for the model, like the resize the frames for the proper execution of our model. In this project we will use Histogram of Oriented Gradient Descriptor (HOG). HOG is a feature descriptor used in computer vision and image processing for the purpose of object detection. This is one of the most popular techniques for object detection. We will use HOGDescriptor with SVM already implemented in OpenCV. this will reduce our time to creating or trainig the model.

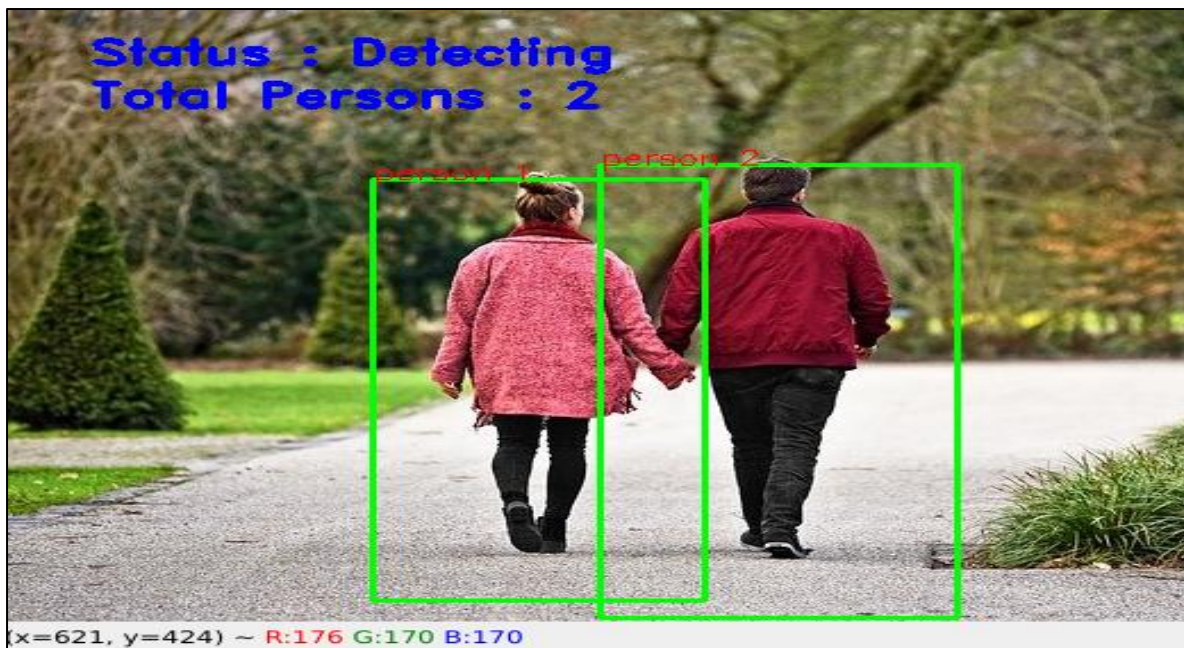
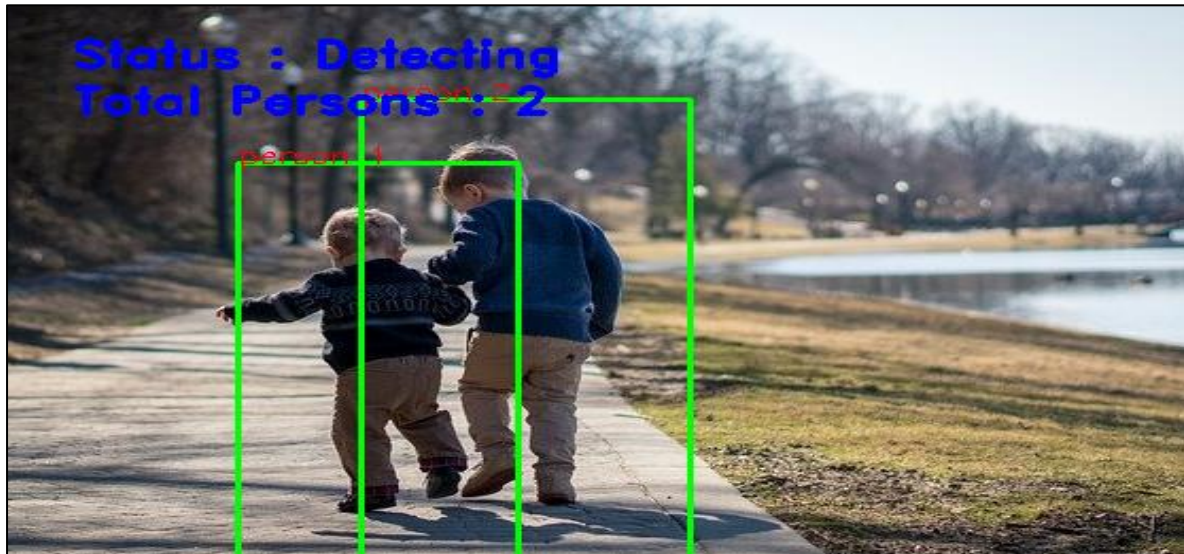
Objectives:

Objectives of this project is to learn some model handing and a create a mini project for the collage Curriculum. And it also gaining some Experience for project handling And the Proper execution of the code and getting the correct result for the Project

Desired Implications

- 1) The first step is to import the google collab files module this module will allow us to upload files from the pc to the google caller project
- 2) we will now import the HOGDescriptor with SVM already implemented in OpenCV. Below code will do this work:
HOGCV = cv2.HOGDescriptor()
HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
(cv2.HOGDescriptor_getDefaultPeopleDetector() calls the pre-trained model for Human detection of OpenCV and then we will feed our support vector machine with it.)
- 3) The Detect() method:
 - Video - A video combines a sequence of images to form a moving picture. We call these images as Frame. So in general we will detect the person in the frame. And show it one after another that it looks like a video.
 - That is exactly what our Detect() method will do. It will take a frame to detect a person in it. Make a box around a person and show the frame and return the frame with person bounded by a green box.
- 4) Everything will be done by detectMultiScale(). It returns 2-tuple:
 - List containing Coordinates of bounding Box of person.
Coordinates are in form X, Y, W, H.
Where x,y are starting coordinates of box and w, h are width and height of box respectively.
 - Confidence Value that it is a person
- 5) Now we create out detector:
 - By using path of the image/video - In this deep learning project, we can take images also. So our method will check if a path is given then search for the video or image in the given path and operate.
- 6) Now we create argparse() methode:
 - The function argparse() simply parses and returns as a dictionary the arguments passed through your terminal to our script. There will be Three arguments within the Parser.
 - **Image:** The path to the image file inside your system
 - **Video:** The path to the Video file inside your system

Output



Conclusion

The aim of this paper is to show that human detection and counting is most useful in this pandemic COVID-19 situation to maintain social distance in the public places by setting the people limit. So that we can easily monitor the people limit in this COVID situation.

We effectively understood that how to create and train model in python. We understood the basic concepts of python. We understood the different libraries of python. Now we are able to create and train machine learning model.

We successfully implemented and run the “Real-Time Human Detection and Counter” in python.

References

- [Python Project - Real-time Human Detection & Counting - DataFlair \(data-flair.training\)](https://data-flair.training/tutorials/real-time-human-detection-counting-python-opencv/)
- [Real Time Human Detection and Counting with Python \(ris-ai.com\)](https://ris-ai.com/real-time-human-detection-counting-python/)
- **Google Colab :-**
<https://colab.research.google.com/>
- **HOG Descriptor :-**
[Training custom SVM to use with HOGDescriptor in OpenCV - Stack Overflow](https://stackoverflow.com/questions/22952275/training-custom-svm-to-use-with-hogdescriptor-in-opencv)
[OpenCV HOG Hyperparameter Tuning for Accurate and Fast Person Detection \(debuggercafe.com\)](https://debuggercafe.com/opencv-hog-hyperparameter-tuning-for-accurate-and-fast-person-detection/)

