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

1. **International Research Journal of Engineering and Technology (IRJET)**

   (Mrs.R.Preethi, Gollapudi Venkata Praneeth, Evani Sai Sarvan Kumar Sarma)

Counting people in visual surveillance is hard and challenging problem. Automatic counting surveillance of individuals publicly areas is vital for safety control. Previously many techniques and methods are proposed.

These methods/techniques aren't producing accurate and high performance for difficult situations. Now Foreground Extraction and Expectation Maximization (EM) based methods are proposed, which provides a far better accurate solution for counting people and locating a private. This work presents the security precaution of covid-19 for maintaining social distancing. Single shot detector algorithm (SSD) takes the live stream from camera and convolutional neural network (CNN) will identify the human and assign a private id and therefore the count it accordingly. Keywords: Single shot detector (SSD) Algorithm, Deep Learning, Machine Learning, Convolutional neural network (CNN) Algorithm, Live stream

Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio [4] proposed a general example-based framework for detecting objects in static images by components. The technique is demonstrated by developing a system that locates people in cluttered scenes. Especially, the system detects the components of a person’s body in a picture, i.e., the head, the left and right arms, and therefore the legs, rather than the complete body by using four distinct example based detectors. The system then checks to make sure that the detected components are within the proper geometric configuration.

2. **IOSR Journal of Engineering (IOSRJEN)**

   (G.Thomas Prathiba , Y.R.Packia Dhas)

Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio [4] proposed a general example-based framework for detecting objects in static images by components. The technique is demonstrated by developing a system that locates people in cluttered scenes. In particular, the system detects the components of a person’s body in an image, i.e., the head, the left and right arms, and the legs, instead of the full body by using four distinct example based detectors. The system then checks to ensure that the detected components are in the proper geometric configuration.

We calculated the geometric constraints for each component from a sample of the training images, tabulated in Table 1, by taking means of the centroid and top and bottom boundary edges of each component over positive detections in the training set. There are two sets of constraints for the arms, one intended for extended arms and the other for bent arms. Haar wavelet functions are used to represent the components in the images and Support Vector Machines (SVM) to classify the patterns. Four component-based detectors are combined at the next level by another SVM. The results of the component detectors are used to classify a pattern as either a “person” or a “nonperson”. For this purpose uses one classifier, named as Adaptive Combination of Classifiers (ACC) that improves accuracy of people detection. This system performs significantly better than a similar full-body person detector. This suggests that the improvement in performance is due to the componentbased approach and the ACC data classification architecture. While this paper establishes that, this system can detect people who are slightly rotated in depth, it does not determine, quantitatively, the extent of this capability. This is the main drawback of the method and also more time consuming task.

4. GE Global Research Niskayuna, NY 12309, USA

(J. Rittscher A. G. A. Perera N. Krahnstoever)

Various techniques [11, 3, 4] have been applied to construct fast and reliable person detectors for surveillance applications. Classification techniques can for example be applied to decide if a given image region contains a person. Amongst others Nakajimia et al. [11] use Support Vector Machines to approach this problem. Gravrila [3] uses a tree based classifier to represent possible shapes of pedestrians. Griebel et al. [4] use dynamic point distribution models. An alternative to modeling the appearance of an entire person is to design detectors for specific body parts and combine the result of those. The idea of learning part detectors using Ada-Boost and a set of weak classifiers is presented in [8]. A learning approach is then being used to combine the set of weak classifiers with body part detectors, which are further combined using a probabilistic person model. All these approaches require a fair amount of training data to learn the

Various techniques [11, 3, 4] have been applied to construct fast and reliable person detectors for surveillance applications. Classification techniques can for example be applied to decide if a given image region contains a person. Amongst others Nakajimia et al. [11] use Support Vector Machines to approach this problem. Gravrila [3] uses a tree based classifier to represent possible shapes of pedestrians. Griebel et al. [4] use dynamic point distribution models. An alternative to modeling the appearance of an entire person is to design detectors for specific body parts and combine the result of those. The idea of learning part detectors using Ada-Boost and a set of weak classifiers is presented in [8]. A learning approach is then being used to combine the set of weak classifiers with body part detectors, which are further combined using a probabilistic person model. All these approaches require a fair amount of training data to learn the

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

Conclusion

The aim of this project 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)

- Real Time Human Detection and Counting with Python (ris-ai.com)

- **Google Colab :-**

  https://colab.research.google.com/

- **HOG Descriptor :-**

  Training custom SVM to use with HOGDescriptor in OpenCV - Stack Overflow

  OpenCV HOG Hyperparameter Tuning for Accurate and Fast Person Detection (debuggercafe.com)

- **Research Papers :-**

  https://www.cse.msu.edu/~liuxm/publication/Liu_Tu_Rittscher_Perera_Krahnstoever_AVSS05.pdf

  https://www.irjet.net/archives/V8/i4/IRJET-V8I4158.pdf

  https://www.iosrjen.org/Papers/vol3_issue1%20(part-1)/B03110510.pdf