HUMAN DETECTION USING PYTHON

A PROJECT REPORT

Submitted by

YOKESH.R 200701303
HARISHKUMAR.S 200701515
VIGNESHWARAN.R 200701513

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

NOVEMBER 2022

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this thesis titled "**HUMAN DETECTION USING PYTHON** " is the bonafide work of "YOKESH.R(200701303), HARISHKUMAR.S(200701515),VIGNESHWARAN.R(200701513)**"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. M.Jaeyalakshmi M.E,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Chennai - 602 105

Submitted to Project Viva-Voice Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

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.In our project we are going to detect humans by using videos by using some of the machine learning and open cv algorithms we will identify the human and assign a private id and therefore the count it accordingly.The research for biometric authentication of a person has reached far but the real-time tracking of human beings has not gained much importance .for better advancements we can implement this in future with advanced technologies.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and ViceChairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr.P.Kumar, Head of the Department of Computer Science and Engineering,** and our project guide **Ms. M. Jaeyalakshmi M.E,** for her encouragement and guiding us throughout the project. We would like to thank our project coordinator, for her encouragement and guidance towards the successful completion of this project and to our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

HARISH KUMAR.S.

YOKESH.R.

VIGNESHWARAN.R

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

This chapter resembles the brief introduction about the most widely used field of study “Computer Vision”. Here talked about the various aspects and uses of computer vision, basic meaning and keywords like detection, enumeration needed for our project.

1.1) Computer Vision

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. It is most widely used field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs.different types of computer vision include image segmentation, object detection, facial recognition, edge detection, pattern detection, image classification, and feature matching.Computer Vision itself is a big domain and is divided into various subdomains like scene reconstruction, object detection, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual serving, 3D scene modeling, and image restoration.

1.2) Application of Computer Vision

It has various different application[1] that too in various fields. Some of them are listed below:

- Object Detection
- Screen Reader
- Intruder Detection
- Code and Character Reader
- Robotics
- Motion Analysis
- Image Restoration

There are many left to list as it is very wide topic and here in this project we have used one of the application i.e. Object Detection.

1.3) Detection and Enumeration in Computer Vision

Detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.

For Detection process in computer vision, there are various methods and each one have different level of accuracy according to their advancement level, like is some methods that is invented in very early stage, they give more cases of false detection as compared to the advanced methods that had been discovered after that.

And here we have used the human as an entity which we are detecting our project and along with that, we are also counting humans through image, video and camera.

1.4 OpenCV

The OpenCV is an open source library of computer vision and image processing etc. It plays an important role in real time image operation, and it is an important part of today computer systems. By using this software the user can process image, detect objects and this library of OpenCV is gradually evolving because of its ability to perform more complex tasks in processing images etc in a consistent manner. This library has been applied extensively in companies, public bodies (like Government bodies), well established software companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda make extensive use of OpenCV. OpenCV is highly rated as it involves state of Art Computer Vision and Machine Learning algorithms, Deep Learning etc. Deep Learning helps in self driving cars and those cars will be using OpenCV to gather colors either be it on road traffic lights or other cars light, it may can passing light, stop light or indicator lights.

1.5 Objectives

- The ultimate aim of the project is to detect the human by using the concept of machine learning
- To Detect the human in a video.
- The aim is to detect with an accuracy level.

CHAPTER 2

LITERATURE REVIEW

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, 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.

2.1 Existing System

In the existing system they have gone through with the opencv but while detection of the humans they got the wrong outputs.There is no exact human detection with accuracy.

2.1.1 Advantages of the existing system

1) Less processing time

2.1.2 Drawbacks of the existing system

1) The output may be fault.

2) Unable to detect humans

2.2 Proposed System

In the proposed system we are detecting humans using open cv and by hog descriptor algorithm so the accuracy level of detecting humans is so high

.

2.2.1 Advantages of proposed system:

1) makes implementation more easy.

2)provides continuous survilleance

3)It is a very efficient way of delivering alerts to admin

4)As you have access to the net 24*7,you can train system anytime and from anywhere

2.2.2 IMPACT OF HUMAN DETECTION:

During this COVID-19 situation, we should always maintain equal social distance and to possess a limited number of individuals in each place. to take care of these within the public places like shopping malls, theatre, complexes and lotsof other crowded places where entrance and exit should get on an equivalent side. The Prediction border will count the entire number of in and total number of individuals out by assigning the individual identity number. If the people inside the place is exceeded the limit given within the admin system.

CHAPTER 3

SYSTEM DESIGN

3.1 Development Environment

3.1.1 Hardware Requirements

The hardware requirements may serve as the basis for a contract for the training of the models used and should therefore be a complete and consistent specification of the whole system. They are used by machine learning engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

COMPONENT	SPECIFICATION
PROCESSOR	Intel Core i5.
RAM	8 GB DDR4 RAM
GPU	NVIDIA GEFORCE 960
MONITOR	15” COLOR
HARD DISK	32 GB
PROCESSOR SPEED	MINIMUM 500MHZ
RASPBERRY PI	4B
WEBCAM OR PI CAMERA	ANY VERSION

3.1.2 Software Requirements

- Python 3.7
- Pycharm
- Visual studio code
- Open cv package
- Numpy package

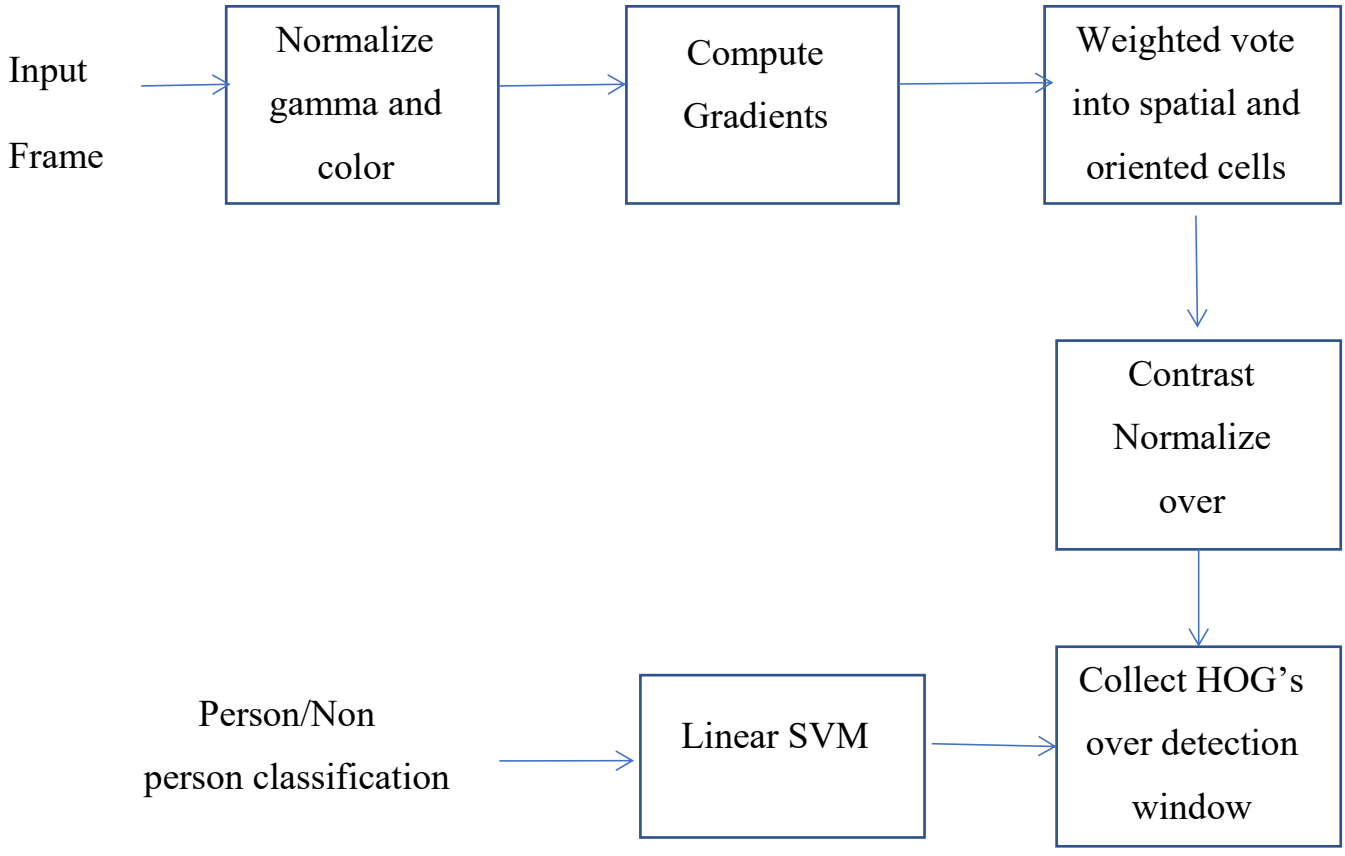
CHAPTER 4
PROJECT DESCRIPTION

4.1 HISTOGRAM OF ORIENTED GRADIENTS

It is well known that Histogram of Oriented Gradient (HOG) is quite a popular method of detecting people in static image. HOG is developed from the SIFT feature and is used to describe the body's shape information by gradient direction histogram of small pieces. The image is first divided into blocks while these blocks overlap with each other. Each block contains four cells. For each pixel I(x,y), the orientation θ(x,y) and the magnitude m(x,y) of the gradient are calculated by

$$dx = I(x+1, y) - (x-1, y)$$
$$dy = I(x, y+1) - (x, y-1)$$
$$\theta(x, y) = \tan^{-1}\left(\frac{dy}{dx}\right)$$
$$m(x, y) = \sqrt{dx^2 + dy^2}$$

A histogram is calculated for each cell, and the volume of each bin is the sum of magnitude of the pixels whose orientations are in the corresponding angle interval. In our implementation, each block contains 2×2 cells, so a block can be represented by a 36-dimensional vector and the resized window is 64×128.



The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances.

3.4 HIEARCHICAL SEARCH ALGORITHM

After we finish the motion region extraction stage, we set out to utilize HOG feature to reach the detection phase. The detection process is similar to reach all the candidate windows step by step. In this proposed method a hierarchical search algorithm is proposed by using an adaptive threshold. In HOG model training part, the training samples have a fixed size (such as 64x128), while in the blobs to be detected, the expected target might appear at any position and the size would also keep changing (take a man appear from distance location and come towards the camera for example). So we try to use the hierarchical search algorithm to solve these problems. We set a window stride value, and then gradually down scale the target image according to the value until the scale of the size zoom to a predefined size or smaller than the target model (the window size). Target detection is applied to each scale level respectively and a final integration is adapted to all of the results.

CHAPTER 5
RESULTS AND CONCLUSION

5.1 Conclusion:

In this project, we defined to get the required color field from an RGB image. In this various steps are implemented using openCV platform. The main positive point of this method is its color differentiation of a mono color. In the future scope, the detection of the edge detection techniques has different other applications like facial detection color conversion for grey scale image etc. That can also be implemented and human tracking system can also be implemented by using fuzzy logic.

SOURCE CODE:

```
import cv2
import imutils
import numpy as np
import argparse
HOGCV = cv2.HOGDescriptor()
HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())


def detect(frame):
    bounding_box_cordinates, weights = HOGCV.detectMultiScale(frame,
winStride=(4,4), padding=(8,8), scale=1.1)


    person = 1
    for x, y, w, h in bounding_box_cordinates:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 500, 0), 2)
        cv2.putText(frame, f'person {person}', (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1)
        person += 1
```

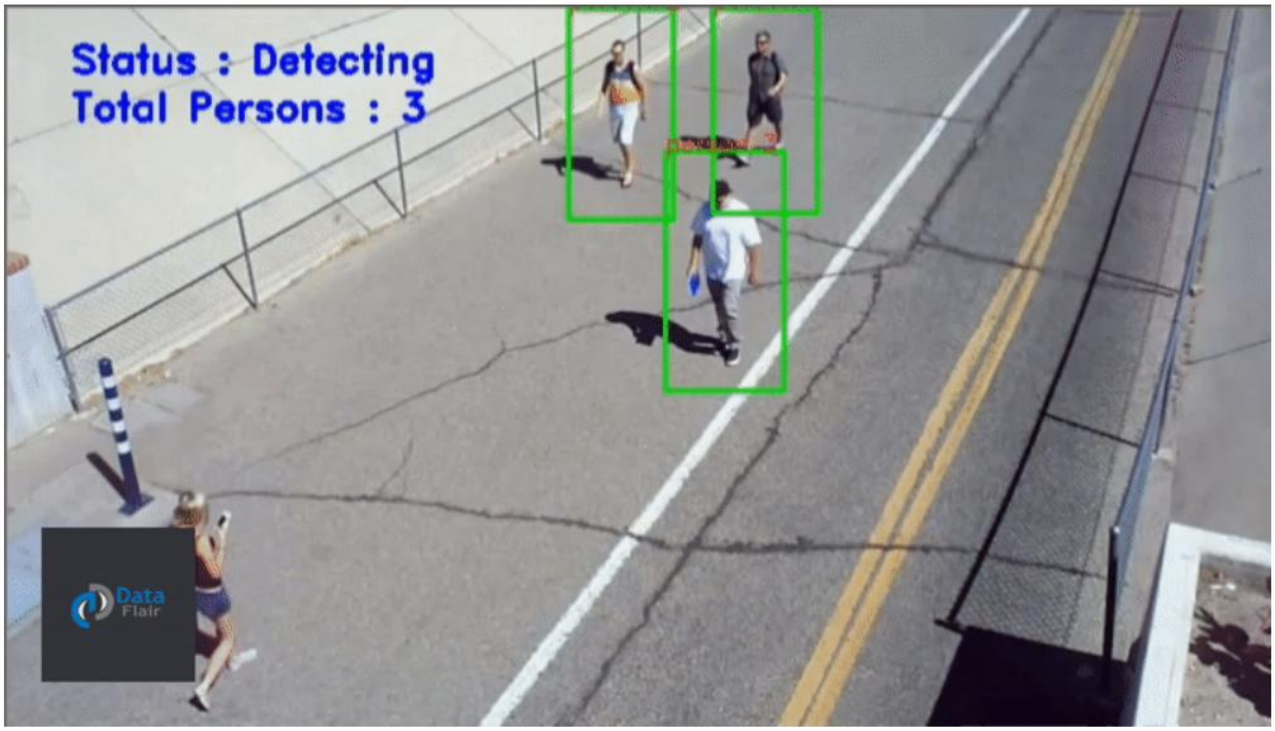
```
    cv2.putText(frame, 'Status : Detecting ', (40, 40),
cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 0, 0), 2)
    cv2.putText(frame, f'Total Persons : {person - 1}', (40, 70),
cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 0, 0), 2)
    cv2.imshow('output', frame)
    return frame
def humanDetector(args, ouput_path=None):
    image_path = args["image"]
    video_path = args['video']
    if str(args["camera"]) == 'true' : camera = True
    else : camera = False
    writer = None
    if args['output'] is not None and image_path is None:
        writer = cv2.VideoWriter(args['output'],cv2.VideoWriter_fourcc(*'MJPG'), 10,
(600,600))
    if camera:
        print('[INFO] Opening Web Cam.')
        detectByCamera(ouput_path,writer)
    elif video_path is not None:
        print('[INFO] Opening Video from path.')
        detectByPathVideo(video_path, writer)
    elif image_path is not None:
        print('[INFO] Opening Image from path.')
        detectByPathImage(image_path, args['output'])
def detectByCamera(writer):
    video = cv2.VideoCapture(0)
    print('Detecting people...')
```

```
    while True:
        check, frame = video.read()


        frame = detect(frame)
        if writer is not None:
            writer.write(frame)


        key = cv2.waitKey(1)
        if key == ord('q'):
            break


    video.release()
    cv2.destroyAllWindows()




def detectByPathVideo(path, writer):
    video = cv2.VideoCapture(path)
    check, frame = video.read()
    if check == False:
        print('Video Not Found. Please Enter a Valid Path (Full path of Video Should
be Provided).')
        return
    print('Detecting people...')
    while video.isOpened():
        # check is True if reading was successful
```

```
        check, frame = video.read()
        if check:
            frame = imutils.resize(frame, width=min(800, frame.shape[1]))
            frame = detect(frame)


            if writer is not None:
                writer.write(frame)


            key = cv2.waitKey(1)
            if key == ord('q'):
                break
        else:
            break
    video.release()
    cv2.destroyAllWindows()




def detectByPathImage(path, output_path):
    image = cv2.imread(path)
    image = imutils.resize(image, width = min(800, image.shape[1]))
    result_image = detect(image)
    if output_path is not None:
        cv2.imwrite(output_path, result_image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
def argsParser():
```

```
    arg_parse = argparse.ArgumentParser()
    arg_parse.add_argument("-v", "--video", default=None, help="path to Video File ")
    arg_parse.add_argument("-i", "--image", default=None, help="path to Image File ")
    arg_parse.add_argument("-c", "--camera", default=False, help="Set true if you
want to use the camera.")
    arg_parse.add_argument("-o", "--output", type=str, help="path to optional output
video file")
    args = vars(arg_parse.parse_args())
    return args
if __name__ == "__main__":
    HOGCV = cv2.HOGDescriptor()
    HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
    args = argsParser()
    humanDetector(args)
```

OUTPUT:



5.2 Future Work

It is particularly very important to collect and train as much real-world data as possible. Model can be optimized for faster inference. Thus, the model's overall performance and accuracy measures can be further enhanced by increasing the dataset used for training with real-world data and we can also track human by using fuzzy logic.

CHAPTER -6
REFERENCES

1. 1. Weiming Hu, Xue Zhou, "Active Counter Based Visual Tracking by Integrating Colors, Shapes and Motions", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 5, MAY 2013

2. G.M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32, NO. 9, SEPTEMBER 2010

3. Claudia Nieuwenhuis, "Spatially Varying Color Distributions for Interactive Multi Label Segmentation", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL 35, NO. 5, MAY 2013

4. HUMAN Identification in Images—Machine Learning Application. Medium. (2020). Retrieved 23 June 2020, from https://towardsdatascience.com/color-identification-inimages-machine-learning-application-b26e770c4c71.

5. Xiong, N., Shen, Y., Yang, K., Lee, C., & Wu, C. (2020). Color sensors and their applications based on real-time color image segmentation for cyber physical systems. Retrieved 23 June 2020, from.