WEB APPLICATION FOR CROWD MANAGEMENT SYSTEM

A MINI PROJECT-II REPORT

Submitted by

SABITHA T.	**(1805120)**
SAMAYANJALI M.	**(1805121)**
YOGAPRIYA P.	**(1805161)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

SRI RAMAKRISHNA ENGINEERING COLLEGE

[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Accredited by NAAC with 'A' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,

COIMBATORE – 641 022
ANNA UNIVERSITY : CHENNAI 600 025

JULY 2021

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

16IT259 – MINI PROJECT - II

Certified that this Mini Project - II Report **“Web Application for Crowd Management System”** is the bonafide work of **“Sabitha.T(1805120), Samayanjali.M(1805121), Yogapriya.P(1805161)”** who carried out the project under my supervision.

SIGNATURE
Dr.M.Senthamil Selvi
HEAD OF THE DEPARTMENT
Professor,
Information Technology,
Sri Ramakrishna Engineering College,
Coimbatore-641022.

SIGNATURE
Dr.J.Anitha
SUPERVISOR
Associate Professor,
Information Technology,
Sri Ramakrishna Engineering College,
Coimbatore-641022.

Submitted for the Mini Project Viva-Voce Presentation held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank the Almighty for his blessings on us to complete this project work successfully.

With profound sense of gratitude, we sincerely thank the Managing Trustees**Thiru D.Lakshmi Narayanaswamy** and Joint Managing Trustee **R.Sundar,** for having provided us the necessary infrastructure required for the completion of our project.

With profound sense of gratitude, we sincerely thank the **Head of the Institution, Dr.N.R.Alamelu,** for her kind patronage, which helped in pursuing the project successfully.

With immense pleasure, we express our hearty thanks to the **Head of the Department, Dr.M.SenthamilSelvi,** Department of Information Technology for her encouragement towards the completion of this project.

With profound sense of gratitude, we sincerely thank the Academic Coordinator, **Dr.K.Deepa, Professor**, Department of Information Technology for her kind patronage and encouragement towards the completion of the project.

With immense pleasure, we thank our Project Coordinator, **Dr.J.Anitha, Associate Professor**, Department of Information Technology for their encouragement, valuable guidance and keen interest towards the completion of the project

We convey our thanks to all the teaching and non-teaching staff members of the IT Department of our college who rendered their cooperation by all means for completion of this project.

ABSTRACT

The steady increase in population and overcrowding has become an unavoidable factor in any public gatherings or on the street during any festive occasions. The intelligent monitoring technology has been developing in recent years and human tracking has made a lot of progress. In this project, a method to manage the crowd by keeping in track the count of the people in the scene is proposed. The interactive web application for the system that will provide the count of the human in the region using OpenCV Python. HOG feature is trained for human detection. Human tracking is achieved by indicating the direction of movement of the person. Currently, Computer Vision (CV) is one of the most popular research topics in the world. This is because it can support the human daily life. Moreover, CV can also apply to various theories. Human Detection is one of the most popular research topics in Computer Vision. In this project, we present a study of technique for human detection from video, which is the Histograms of Oriented Gradients or HOG by developing a piece of application to import and detect the human from the video. The HOG Algorithm is used to analyze every frame from the video to find and count people. The results of the analysis will be helpful in managing the crowd in any area with high density of crowds.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Modern technology provides solutions which make people’s life not only easier but also safer. Being in a highly crowded place affects not only the human level of comfort but mainly the human level of safety. In this time when COVID-19 is spreading rapidly, it is essential to maintain social distance and avoid large public gatherings at one place to break the chain of the disease spread. Many people, knowingly or unknowingly, gather and roam on the streets. Keeping an eye on all these activities is not an easy job. The authorities need reliable technology that can survey such places to prevent any unnecessary movement. Human detection is the technology that people usually use to detect objects in images or videos. It is also a crucial step in the video-based surveillance systems. The aim is to identify and monitor humans for security purposes in the crowded environment such as airports, bus terminals or train stations. One example is the video which captured by the CCTV. It has been processed to detect and track the movements of human, both whole and partial of the body. magnitude of flow. HOG has three steps, including preprocessing, HOG feature descriptor used for human detection is calculated on a 64128 patch of an image. Then the magnitude of gradient is calculated. Lastly is to calculate Histogram of Gradients in 88 cells

CHAPTER 2

LITERATURE SURVEY

Various researches have been done for Web Application using crowd management system. This research is done prior to taking up the project and understanding the various methods that were used previously.

2.1.1 Anuj Mohan, Constantine Papageorgiou and Tomaso Poggio "Component based people detection"

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.

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 component based 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.

2.1.2 Tao Zhao and Bo Wu “ DDMCMC approach”

A model based approach to interpret the image observations by multiple partially occluded human hypotheses in a Bayesian framework. This approach to segmenting and tracking multiple humans emphasizes the use of shape models.
The optimal solution is obtained by using an efficient sampling method, data-driven Markov chain Monte Carlo (DDMCMC), which uses image observations for proposal probabilities. Knowledge of various aspects, including human shape, camera model, and image cues, are integrated in one theoretically sound framework. To improve the computational efficiency, we use direct image features from a bottom-up image analysis as importance proposal probabilities to guide the moves of the markov chain. This method is able to successfully detect and track humans in the scenes of complexity with high detection and low false alarm rates. Here a more accurate 3-D model composed of three ellipsoids was used. To deal with the occlusion problem, a joint probability for multiple humans has been considered. Finally, the human detection and tracking problem was formulated as a Maximum A Posteriori (MAP) problem simultaneously. A sophisticated sampling algorithm, Data Driven Markov Chain Monte Carlo, is used to find the best configuration for the MAP problem. Some positive results for a crowd of a dozen people were obtained. To reduce the dependence on an accurate foreground contour, this may be easily corrupted by noise. This is a time consuming task.

2.1.3 Antoni B.Chan, Zhang-Sheng John Liang and Nuno Vasconcelos "Counting people without people models"

A privacy-preserving system for estimating the size of inhomogeneous crowds, composed of pedestrians that travel in different directions, without using explicit object segmentation or tracking or models.

Video locations are then scanned sequentially; a patch is extracted at each location, and assigned to the mixture component of largest posterior probability. The location is declared to belong to the segmentation region associated with that component. Before extracting features from the video segments, it is important to consider the effects of perspective. Because objects closer to the camera appear larger, any feature extracted from a foreground object will account for a smaller portion of the object than one extracted from an object farther away. This makes it important to normalize the features for perspective. Features such as segmentation area or number of edges should vary linearly with the number of people in the scene. These features capture segment shape and size. In this approach textures inside the foreground are used to estimate the crowd density or the number of people. Gaussian Process Regression was adopted to ascertain the relationship between 28 different features and the number of people. To get more accurate results, the crowd was segmented into two components based on their moving directions before estimation

2.1.4 Ya-Li Hou and Grantham K.H. Pang "Neural Network and EM based people counting and individual detection"

Neural network based people counting and EM based individual detection in a low resolution image with complicated scenes. To detect the number of people from the image we take the difference between the current frame with the previous frame in order to get the foreground pixels based on some predefined threshold. In our method we are setting the threshold in such a way that people moving little bit show some scattered pixels. To minimize the difference between the moving and static people we convert all the pixels whose intensity value is greater or equal to threshold as white, whereas remaining pixel intensity whose value is below threshold are converted into black. Once the foreground pixels are extracted, one can observe that some people show incomplete body shape, while others show some disperse pixels in the foreground image. Additionally, the incomplete body shape is usually from the moving people while the disperse pixels are mainly from the stationary people.

2.2 PROPOSED SYSTEM

The crowd is a gathering of people at some place. In this pandemic time, it is considered to be very dangerous being in a crowd as some deadly diseases like COVID-19 and flu can easily spread. Also, large crowds result in accidents if preventive measures are taken with proper planning. It is very much evident that large crowding is the part of the normal life and this exposes a new problem crowd management. Excessive crowding and poor crowd management results in loss of precious life very easily. It is not feasible to count or monitor all the people at various places like malls, shops, schools, colleges, railway stations, airports etc., at looking at them manually. The complexity of monitoring, tracking and counting increases as the size of the crowd increases. Hence the need for the development of an automated crowd management system will be highly appreciated by the society.

CHAPTER 3

PROJECT DESCRIPTION

3.1 INTRODUCTION

The HOG descriptor is a local descriptor which is appropriate for pedestrian detection. The descriptor is applied to upright human body in flat view, mainly consisting of head, torso and limbs. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. The HOG descriptor captures edge or gradient structure that is characteristic of local shapes. The HOG descriptor significantly out performs others in pedestrian detection. As long as the posture of pedestrian is upright, there will be no effect on the detecting results with limb and body segments to change appearance and even move from side to side. It seems that the HOG descriptor shows better inclusiveness to individual differences, except for the rotation of the main body. The descriptor is a multi-dimensional vector.

3.2 FLOW CHART

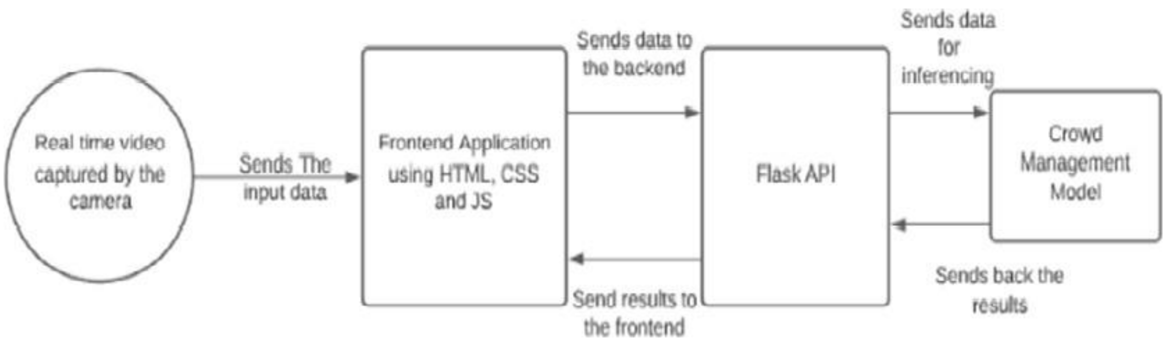

Real time video captured by the camera
Sends The input data
Frontend Application using HTML, CSS and JS
Sends data to the backend
Send results to the frontend
Flask API
Sends data for inferencing
Sends back the results
Crowd Management Model

Fig 3.2.1 Block Diagram

3.3 WORKING PRINCIPLE

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. HOG decomposes an image into small squared cells, computes an histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and return a descriptor for each cell. A feature descriptor is a representation of an image or an image patch that simplifies the image by extracting useful information and throwing away extraneous information. The first step in HOG detection is to divide the source image into blocks. Each block is divided by small regions, called cell. For each pixel within the cell the vertical and horizontal gradients are obtained. A descriptor is assigned to each detector window. This descriptor consists of all the cell histograms for each block in the detector window. The detector window descriptor is used as information for object recognition. The architecture of the model is based on the Faster RCNN algorithm, which is an efficient and popular object detection algorithm which uses deep convolutional networks.

3.4 MERITS AND DEMERITS

3.4.1 MERITS

- It produces good accuracy depending on algorithm and environment.
- High coverage as it is not easily occluded by other objects in the environment and low cost for deployment.
- Count in extremely dense crowd and also detect partially visible humans.
- It will be able to count the both static and moving crowd.

3.4.2 DEMERITS

- Accuracy is low as it unable to distinguish human and other objects and unable to detect people walking side by side while crossing the infrared beam.
- Limited to sparse crowd, poor performance on people with strange clothes.
- Difficult in low resolution, camera position.
- Cannot be used if an accurate counting system is needed.

3. 5. APPLICATIONS

- It is mainly used in real-life for automated public monitoring such as surveillance and traffic control. Different from object detection, Crowd Counting aims at recognizing arbitrarily sized targets in various situations including sparse and cluttering scenes at the same time.
- A lot of businesses or government agencies could use people counter to understand various things like how crowded are public places at a given time or how many people are using a particular street crossing every day etc.
- Another broad area of application can be safety. If you are searching for someone who is lost or if a business would like to ensure an area is human free before starting a big machine. In the image below my model could track a person who is trekking in outback.
- Understanding foot traffic is very important in retail to organise merchandise in aisles, optimise store layout, understand peak times and potentially even protect against theft. You can now put a camera in your store and connect it to Unleashlive AI platform and get real time data that can be ingested and analysed.
- From retractable belt stanchions to rigid rail barriers, these staples of crowd control help establish clear boundaries for everything from waiting lines to walkways. For extra strength, look for mounting solutions that keep stanchions firmly in place either with a strong magnet or with a permanent fixture.

CHAPTER 4

RESULTS

Therefore, the web application on crowd management system has been developed and deployed successfully using CNN and HOG. The flask application is made accessible to the authorities or admin of any organization. By doing so, one can easily detect the crowd and can be aware of overcrowding difficulties and can reduce or eliminate them to a greater extent.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

The scope of the project is to develop the model into the user interactive web application so that it will be easier to use. The system can be implemented in the public places where the real time visuals will be captured by the camera and alert the authorities accordingly. In terms of research recommendations for future study, we strongly recommend that students and researchers who are interested in the Human Detection using HOG techniques to carry on further study by mainly focus on the quality assessment by using number of videos. Moreover, we also recommend that the further research should have high performance devices because the more the performance devices is related to the more ability to compute and detect the human in the high resolution videos and provide a very good detection results. After all, this program should be developed in parallel with statistical program for more convincible results.

APPENDIX

APPENDIX 1

SOFTWARE AND HARDWARE DESCRIPTION

- VISUAL STUDIO CODE
- LIBRARIES
    1. OpenCV
    2. imutils
    3. numpy
    4. argparse

VISUAL STUDIO CODE

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. It is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services). Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings.

LIBRARIES

1. OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.. The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings in Python, Java and MATLAB/OCTAVE.If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10.The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

2. imutils

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3. Translation is the shifting of an image in either the x or y direction. To translate an image in OpenCV you would need to supply the (x, y)-shift, denoted as (tx, ty) to construct the translation matrix M: Rotating an image in OpenCV is accomplished by making a call to cv2.getRotationMatrix2D and cv2.warpAffine. Further care has to be taken to supply the (x, y)-coordinate of the point the image is to be rotated about. These calculation calls can quickly add up and make your code bulky and less readable. The resize function of imutils maintains the aspect ratio and provides the keyword arguments width and height so the image can be resized to the intended width/height while (1) maintaining aspect ratio and (2) ensuring the dimensions of the image do not have to be explicitly computed by the developer. Skeletonization is the process of constructing the "topological skeleton" of an object in an image, where the object is presumed to be white on a black background. OpenCV does not provide a function to explicitly construct the skeleton, but does provide the morphological and binary functions to do so.

3. Numpy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. The ndarray (NumPy Array) is a multidimensional array used to store values of same datatype. These arrays are indexed just like Sequences, starts with zero.

4. Argparse

The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sys.argv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments. The argparse is a standard module; we do not need to install it.A parser is created with ArgumentParser and a new parameter is added with add_argument(). Arguments can be optional, required, or positional.

APPENDIX 2

SOURCE CODE

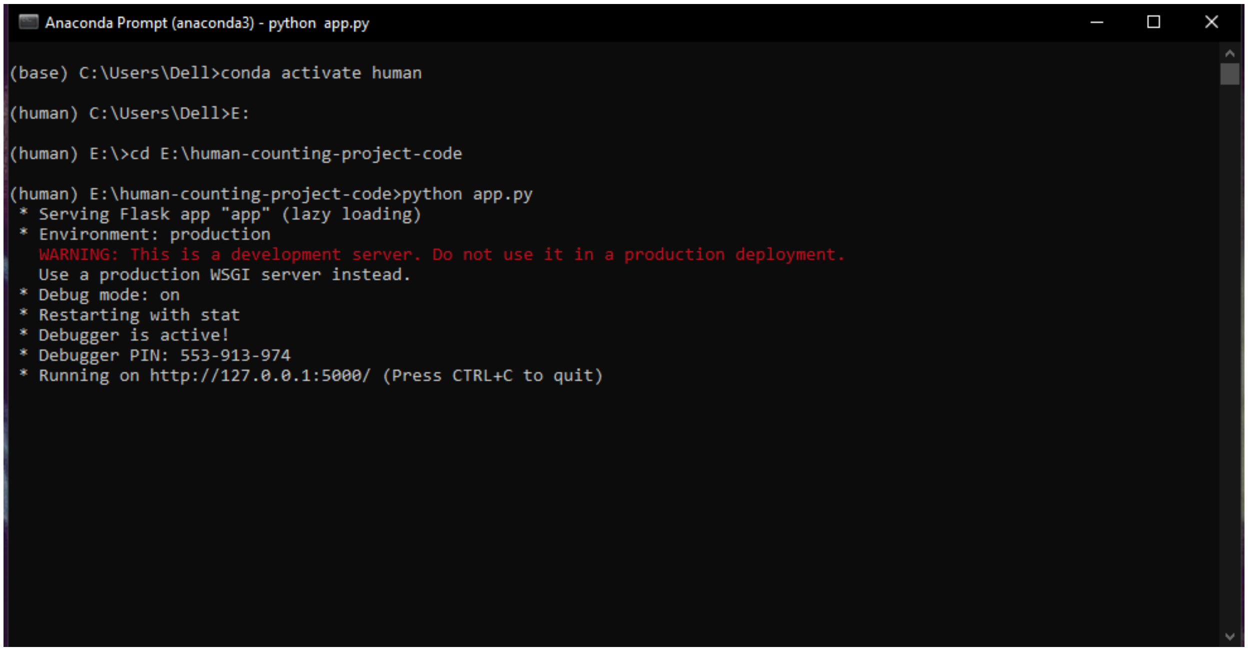
```
import cv2
from cv2 import cv2
import imutils
import numpy as np
import argparse

def detect(frame):
    bounding_box_cordinates, weights =  HOGCV.detectMultiScale(frame,
winStride = (4, 4), padding = (8, 8), scale = 1.03)

    person = 1
    for x,y,w,h in bounding_box_cordinates:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 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 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 detectByCamera( writer):
    #url='http://192.168.80.102:8080'
    #video = cv2.VideoCapture(url)
    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 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 humanDetector(args):
    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(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 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()
    # print("Variable loaded")
    HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
```

```
print("Accessed cam")

args = argsParser()
humanDetector(args)
print("Passed args")
print("Running")
```

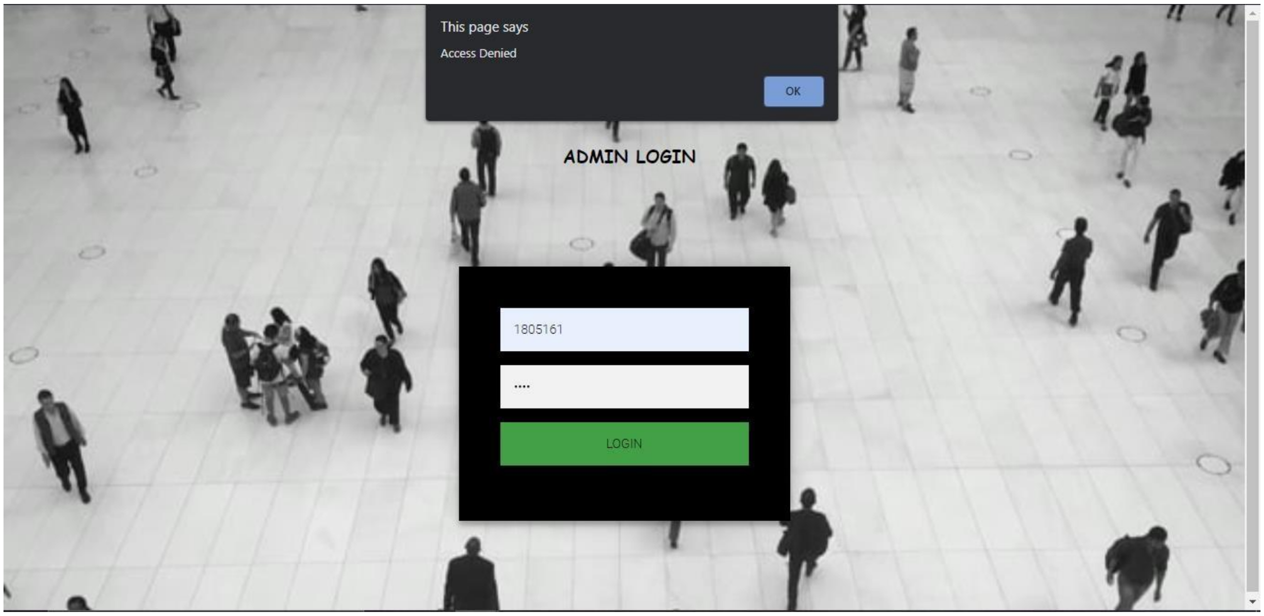
APPENDIX 3

SCREEN SHOTS



```
(base) C:\Users\Dell>conda activate human

(human) C:\Users\Dell>E:

(human) E:\>cd E:\human-counting-project-code

(human) E:\human-counting-project-code>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 553-913-974
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig A3.1 Command Prompt



This page says

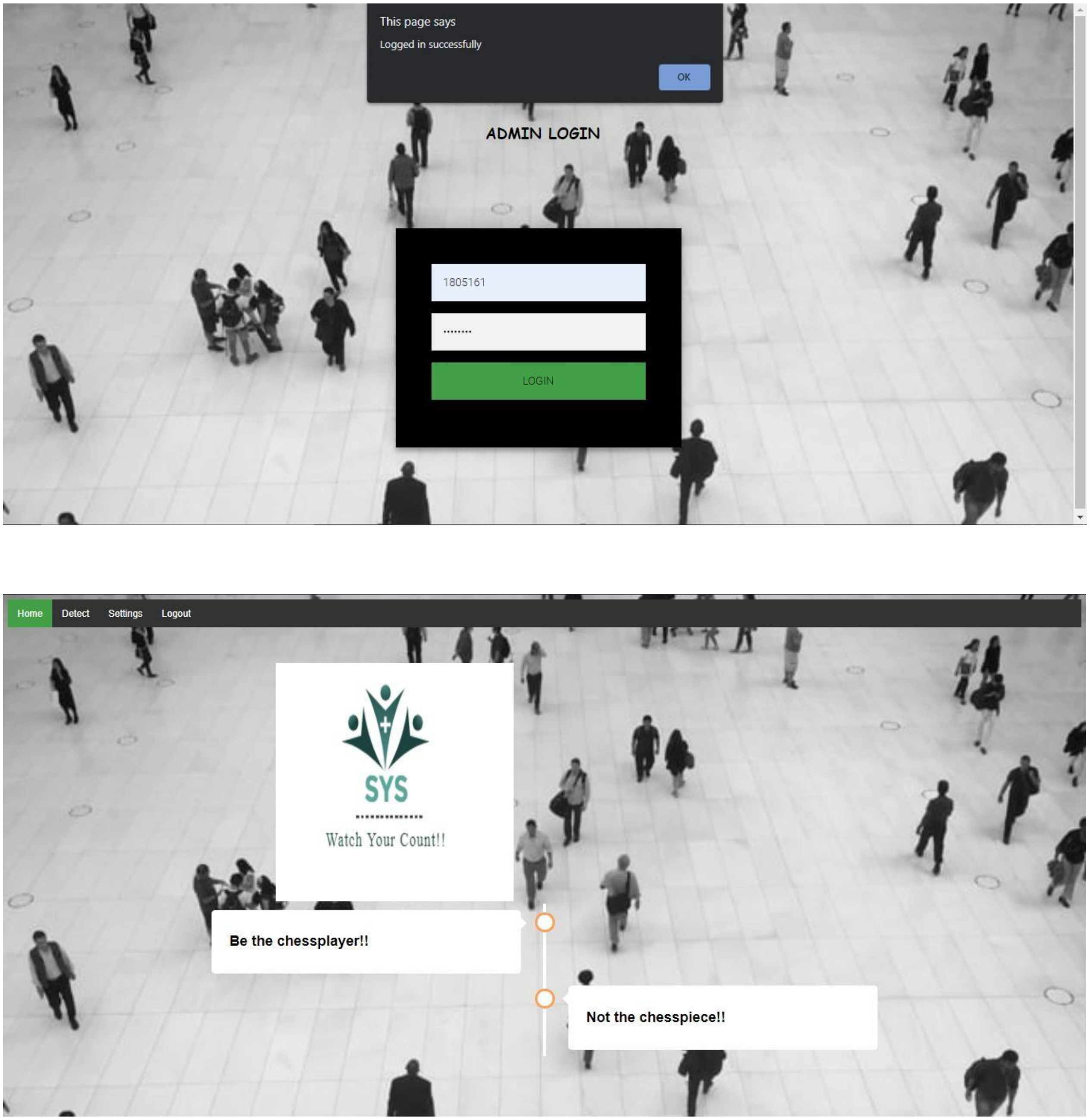
Access Denied

ADMIN LOGIN

1805161

····

Fig A3.2 Login Page



This page says

Logged in successfully

ADMIN LOGIN

1805161

••••••••

Watch Your Count!!

Be the chessplayer!!

Not the chesspiece!!

Fig A3.3 Home Page

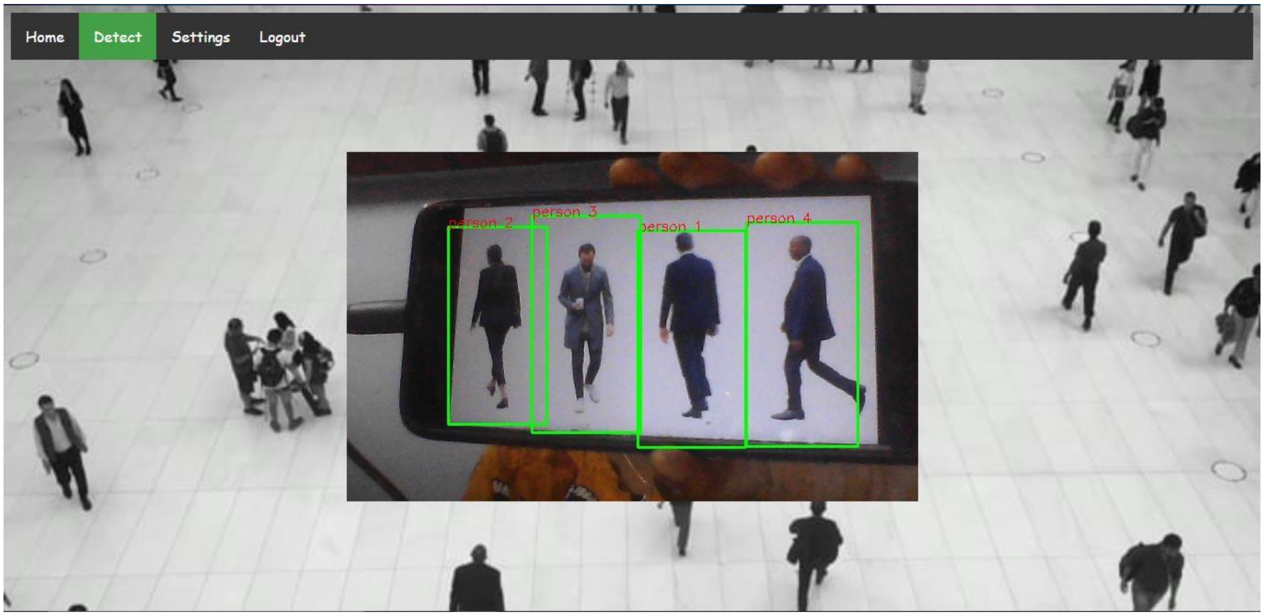
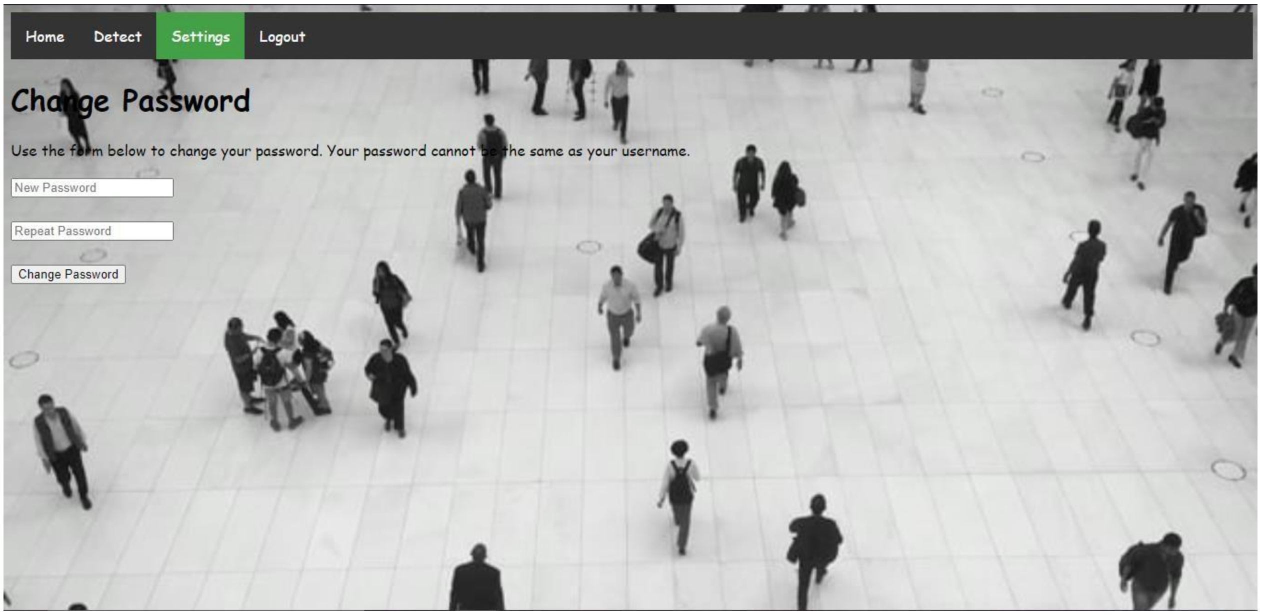


Fig A3.4 Detect Page



Change Password

Use the form below to change your password. Your password cannot be the same as your username.

Fig A3.5 Settings Page

REFERENCES

[1] T. Xu, X. Chen, G. Wei and W. Wang, *"Crowd counting using accumulated HOG,"* 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 1877-1881, doi: 10.1109/FSKD.2016.7603465.

[2] D. B. Sam, S. Surya and R. V. Babu, *"Switching Convolutional Neural Network for Crowd Counting,"* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 4031-4039, doi: 10.1109/CVPR.2017.429.

[3] C. Shang, H. Ai and B. Bai, *"End-to-end crowd counting via joint learning local and global count,"* 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 1215-1219, doi: 10.1109/ICIP.2016.7532551.

[4] S. Huang et al., *"Body Structure Aware Deep Crowd Counting,"* in IEEE Transactions on Image Processing, vol. 27, no. 3, pp. 1049-1059, March 2018, doi: 10.1109/TIP.2017.2740160.

[5] Tianjun. Ma, Qingge. Ji, Ning. Li *"Scene invariant crowd counting using multi-scales head detection in video surveillance"* IET Digital Library Volume 12, Issue 12, December 2018, p. 2258 – 2263.

[6] Viresh Ranjan, Hieu Le, Minh Hoai *"Iterative Crowd Counting"* Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 270-285.

[7] L. Zhang, M. Shi and Q. Chen, *"Crowd Counting via Scale-Adaptive Convolutional Neural Network,"* 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 2018, pp. 1113-1121, doi: 10.1109/WACV.2018.00127.

[8] Vishwanath A. Sindagi, Vishal M. Patel, *“A survey of recent advances in CNN-based single image crowd counting and density estimation”,* Pattern Recognition Letters, Volume 107, 2018.

[9] Lamba S., Nain N. (2017) , *“Crowd Monitoring and Classification: A Survey”*. In: Bhatia S., Mishra K., Tiwari S., Singh V. (eds) Advances in Computer and Computational Sciences. Advances in Intelligent Systems and Computing, vol 553. Springer, Singapore.

[10] Miaojing Shi, Zhaohui Yang, Chao Xu, Qijun Chen, *“Revisiting Perspective Information for Efficient Crowd Counting”,* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7279-7288.

[11] S. Gong, E. -B. Bourennane and X. Yang, *"MFNet: Multi-feature convolutional neural network for high-density crowd counting,"* 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2020, pp. 0384-0390, doi: 10.1109/IEMCON51383.2020.9284903.

[12] Y. Tian, Y. Lei, J. Zhang and J. Z. Wang, *"PaDNet: Pan-Density Crowd Counting,"* in IEEE Transactions on Image Processing, vol. 29, pp. 2714-2727, 2020, doi: 10.1109/TIP.2019.2952083.

[13] Misbah Ahmad, Imran Ahmed, Kaleem Ullah, Maaz Ahmad, *"A Deep Neural Network Approach for Top View People Detection and Counting",* Ubiquitous Computing Electronics & Mobile Communication Conference (UEMCON) 2019 IEEE 10th Annual, pp. 1082-1088, 2019.

[14] X. Pan, H. Mo, Z. Zhou and W. Wu, *"Attention Guided Region Division for Crowd Counting”,* ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 2568-2572, doi: 10.1109/ICASSP40776.2020.9053761.

[15] M. B. Shami, S. Maqbool, H. Sajid, Y. Ayaz and S. S. Cheung, *"People Counting in Dense Crowd Images Using Sparse Head Detections,"* in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2627-2636, Sept. 2019, doi: 10.1109/TCSVT.2018.2803115.