A Mini Project Report on

"AIFDS – AI based Fall Detection System using YOLOv5-s and DeepSparse"

Submitted in partial fulfillment of the requirement for
Degree in Bachelor of Engineering (Information Technology)

By

Joel Dsilva (5019116)
Aryan Koul (5019129)
Madhur Thakkar (5019156)

Guided by:

Prof. Archana Shirke

Department of Information Technology
Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai – 400703

University of Mumbai
2021-2022

CERTIFICATE

This is to certify that the project entitled

"AIFDS – AI based Fall Detection System using YOLOv5-s and DeepSparse"

Submitted By

Joel Dsilva (5019116)
Aryan Koul (5019129)
Madhur Thakkar (5019156)

In partial fulfillment of degree of **T.E** in **Information Technology** for term work of the mini project is approved.

External Examiner

_____

Internal Examiner

_____

Internal Guide

_____

Head of the Department

_____

Principal

_____

Date: -

College Seal

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Joel Dsilva (5019116)

Aryan Koul (5019129)

Madhur Thakkar (5019156)

Date:

Location:

ABSTRACT

A fall in this study is defined as an event in which a person suddenly and inadvertently collapses from an upright position and the person's legs can no longer support oneself. These incidents of falling can have serious impact such as injury or in the worst-case scenario can cause even death if they do not get help immediately. Aid can be given more quickly if the occurrence of falls can be immediately detected. We propose a fall detection algorithm using YOLOv5-s. The people in the image are treated as individual objects by the algorithm and bounding box is drawn around them with high accuracy and speed. The proposed technique measures the length and breadth of the bounding box around the person. Fall is detected when the width of the box exceeds the height by 5%. In order to improve the performance of YOLOv5-s model the system proposes a CPU runtime engine which provides GPU-class performance on CPU using Neuralmagic's DeepSparse.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

INTRODUCTION

As the world is going online so should the printed media which is mostly available only offline. There is a need of a method for storing printed texts in online format and thus preserving it for a longer time and for the future generations. We could store many valuable texts which till now have been bounded to books only. OCR, or optical character recognition, scans document images and turns them into text documents. This intelligent conversion allows documents to be searched with relevant keywords.

1.1 Background

A fall is defined as an event which results in a person coming to rest inadvertently on the ground or floor or other lower level. Worldwide people are facing injuries caused by falling and sometimes even people are dying due to a fall. According to WHO's survey, falls are a leading cause of unintentional injuries in adults older than 65 years old, with 37.3 million falls requiring medical attention and 646,000 resulting in deaths annually. This is especially common among senior citizens living alone. The current birth rate is lower than the death rate which implies that the population is ageing and hence the fatalities caused due to falls are also increasing. In such a dire situation the creation and improvement of fall detection and prevention technologies is the need of the hour. Currently the fall detection technologies can be broadly divided into three classes:

- Wearable devices
- Environmental sensors
- Image detectors

The wearable devices use technologies like accelerometers, gyroscopes, etc to detect falls. A major problem which hinders their use is the changes in temperature, humidity and frequent recharging. Along with these hardware requirements they also cost a lot as they need continuous maintenance and subscription fees. Sometimes the wearer may not be comfortable using them. Environmental sensor technology uses various types of sensors to detect the sounds of falls, sudden changes in pressure, heat signatures and ultrasonic sounds. The major issue with these

technologies are differentiating animals and objects from humans and detection of falls in crowded places.

1.2 Motivation

Fall detection is an important service for the healthcare, especially for the elderly, and a reliable system to detect a fall early is a necessity to reduce the post-effects of falls. As the number of people over 60 years is growing faster than any other age group, a challenge to handle the increasing number of fall accidents emerge. An automatic fall detection system could increase the independent living ability among elderly, and also reduce the manual labor in terms of presence of support staff. Several approaches for fall detection systems have been proposed over the years.

For this project, a vision-based solution is desirable since there are developed special cameras installed in many places. A vision-based system is also interesting when considering recent years of advancement in deep learning, especially image classification. At a wider scope, not just fall detection for indoor usage, cameras are installed at several public places, which would make it possible to use the already installed infrastructure for deployment of a fall detection system over a larger area.

1.3 Problem Definition

The project aims to make a fall detection system which takes camera live feed as an input and continuously monitors people and detects fall in it. So whenever the system is able to detect the fall, it should be able to send alert mails to the registered user along with the image of the fallen person. The system will also be resource optimized i.e. it uses less resources compare to other systems available in the system and to do so deepsparse is used along with yolo. Since we all know YOLO is a state-of-the-art, real-time object detection system. The algorithm applies a single neural network to the full frame. It looks at the whole frame at test time and makes predictions with a single network evaluation which makes it extremely fast. It helps us in detecting whether the object in the frame is a person or anything else.

1.4 Scope/ Assumptions

- Makes use of multiple neural networks to find out the model that is the most efficient for fall detection.
- Able to detect a person from the input camera live feed which can be seen by anyone on any device which has the internet access.
- If the person in the input camera live feed falls down, will be able to quickly detect the fall.
- The snapshot of the person falling will be send to the registered mail id. So that the person could know there is something happened to the person and he/she could take necessary steps. Rather than being unaware of the incident and ended up life risk of the person.
- Fall detection module could be implemented in the health sector where it will detect a person falling and alert the appropriate authorities.
- This module could also be implemented in the public locations as well. So that if there is any hit or run case the appropriate authorities are already aware of it.

1.5 Issues/ Limitations

This project will develop a vision-based fall detection system even though there are other approaches. The main goal will be to develop a model using machine learning that can detect falls using images i.e. a video stream. As of now, the following things are left out:-

- Better algorithm could be used for getting better accuracy of fall detection.
- The emergency button could be linked to multiple users to ensure that help is provided in most situations.
- Reminders for various activities could be set by giving appropriate title, description and alert timings as per user convenience.
- Mails are not sent for multiple fall detection in a single frame. In one frame only one fall is detected.
- The algorithm which is being used for detecting a person is Yolov5-s and it is only available for Ubuntu operating system and Intel processors and does not support any other operating system and processor like Windows, MAC, AMD etc.

CHAPTER 2

LITERATURE SURVEY AND ANALYSIS

LITERATURE SURVEY AND ANALYSIS

The purpose of a literature review is to gain an understanding of the existing research and debates relevant to a particular topic or area of study, and to present that knowledge in the form of a written report.

2.1 Related work

Fall Detection is one of the most useful system for hospitals, family members, parents to keep an eye on their patients, elderly members, new born babies. So that if they fall due to any reason and they end up getting seriously injured or are not able to get up, they could get an alert message that their loved ones are in a problem and they should help them. Rather than being unaware of the situation and ending up either losing or facing serious injuries by their loved ones.

[1] Human fall detection:

In this project they are applying an algorithm for fall detections consistes of video acquisition from an external source such as a surveillance camera, or a saved video or clip. Firstly, a video is acquired as an input. The video is divided into frames, usually being 30 frames per second. There were marks placed on the head of the human in the video. A frame could be regarded as a single image. The human in the image is identified using various methods. Finally the features are extracted and the occurrence of fall is detected on the basis of patterns in these features. The fall is detected on the basis of these features:

- Foreground Extraction
- Centroid Acceleration
- Centroid
- Head Position
- Number of Ones
- Head speed
- Motion Vector
- Centre Speed

- Aspect Ratio
- Fall Angle

[2] Fall Detection Systems for Elderly Care: A Survey

In this paper, they have proposed a fall detection system based on machine learning. Their system detects falls by classifying different activities into fall and non-fall actions and alert the registered user in case of emergency. They have used SisFall dataset with variety of activities of multiple participants calculate features. Machine learning algorithms SVM and decision tree are used to detect the falls on the basis of calculated features. Their system was able to acquire an accuracy up to 96% by using decision tree algorithm.

[3] IoT Based Fall Detection Monitoring and Alarm System For Elderly

This paper presents an IoT based fall detection monitoring and alarm system for the elderly using 3-axis Accelerometer. In the proposed system for detection of falling, the elderly patient's acceleration data are continuously acquired by using a wearable sensor and stored on a cloud server, using an IoT board. To access the stored data, an android application is designed for the medical expert to examine the fall in the patient and provide the needed assistance, if needed. A threshold-based approach for the fall detection has been used to get the sensor data and set the threshold on accelerometer readings. A complete algorithm has been designed for the detection of genuine fall.

[4] Automatic Body Fall Detection System for Elderly People using Accelerometer and Vision Based Technique

In this project generally wearable sensor and vision based technique are used that automatically detecting body fall as early as possible. Accelerometer is used for measuring or maintaining orientation and angular velocity. In vision based procedure first procure casings or video arrangements from the camera is done. The division module separates the body outline from the foundation. For Feature Extraction GLCM method used. SVM method is used for classification.

By using these methods human body fall can surely be detected and preventive measures can be taken.

[5] In Falling Detection System Based on Machine Learning

In this proposed system, a dataset of videos containing falling actions has been utilized via dividing each video into many shots that are consequently being converted into gray- level images. Then, for detecting the moving objects in videos, the foreground is firstly detected, then noise and shadow are deleted to detect the moving object. Finally, a number of features, including aspect ratio and falling angle, are extracted and a number of classifiers are being applied in order to detect the occurrence of falling. Experimental results, using 10-fold cross validation, shown that the proposed falling detection approach based on Linear Discriminant Analysis (LDA) classification algorithm has outperformed both support vector machines (SVMs) and Knearest neighbor (KNN) classification algorithms via achieving falling detection with accuracy of 96.59 %.

Table 2.1 Comparison of Different DeepSparse Yolo based model:

Model Name	Stub	Description	FPS
yolov5l-pruned	zoo:cv/detection/yolov5-l/pytorch/ultralytics/coco/pruned-aggressive_98	Sparse YOLOv5l model trained with full FP32 precision that recovers 98% of its baseline mAP	25 – 30
yolov5l-pruned_quant	zoo:cv/detection/yolov5-l/pytorch/ultralytics/coco/pruned_quant-aggressive_95	Sparse INT8 quantized YOLOv5l model that recovers 95% of its baseline mAP	30 – 35
yolov5s-pruned	zoo:cv/detection/yolov5-s/pytorch/ultralytics/coco/pruned-aggressive_96	Sparse YOLOv5l model trained with full FP32 precision that recovers 96% of its baseline mAP	30 – 35
yolov5s-pruned_quant	zoo:cv/detection/yolov5-s/pytorch/ultralytics/coco/pruned_quant-aggressive_94	Sparse INT8 quantized YOLOv5s model that recovers 94% of its baseline mAP	35- 40
yolov3-pruned	zoo:cv/detection/yolo_v3-spp/pytorch/ultralytics/coco/pruned-aggressive_97	Sparse YOLOv3 model trained with full FP32 precision that recovers 97% of its baseline mAP	15 - 20
yolov3-pruned_quant	zoo:cv/detection/yolo_v3-spp/pytorch/ultralytics/coco/pruned_quant-aggressive_94	Sparse INT8 quantized YOLOv3 model that recovers 94% of its baseline mAP	25 – 30
yolov5l-base	zoo:cv/detection/yolov5-l/pytorch/ultralytics/coco/base-none	Dense full precision YOLOv5l model	20 - 25
yolov5s-base	zoo:cv/detection/yolov5-s/pytorch/ultralytics/coco/base-none	Dense full precision YOLOv5s model	23 - 28
yolov3-base	zoo:cv/detection/yolo_v3-spp/pytorch/ultralytics/coco/base-none	Dense full precision YOLOv3-SPP model	12 - 18

2.2 Existing System

[2.2.1] Bay Alarm Medical

Bay Alarm Medical provides one of the most reliable and affordable life-saving medical alert systems with fall detection. These systems are backed by a reliable, 24-hour medical alarm emergency center that is based in the USA. The Automatic Fall Detection Alarm System from Bay Alarm Medical automatically calls for help when you are unable to do so. It is compatible with both their Cellular and In-Home and medical alert systems.

[2.2.2] AutoAlert fall detection

When a fall is detected, AutoAlert connects quickly to our Response Center, even if you're unconscious or unable to move. The two-way voice communication capability enables you to speak directly with one of our highly trained Lifeline Response Associates. They talk with you, assess the situation, and dispatch the help you need. If the Response Associate is unable to make verbal contact with you, they may default to calling emergency services.

[2.2.3] FallSafety App

FallSafety home uses intelligent fall detection and automatic alerts for emergency assistance. Both big and small falls are set to be detected by default. It uses proven technology. Protects your privacy and only shares your location during an emergency. Works with iPhone and Apple Watch Series 3, 4, 5, SE, and 6. Designed for people at risk from falls. Fall Detection + Panic Button + Fast Heart Rate Detection.

[2.2.4] iMedAlert

iMedAlert is a powerful and user-friendly medical alert application for your phone. In case of an emergency, iMedAlert iPhone not only can immediately alert emergency responders by phone or alarm, but also can broadcast rescues message with your current GPS location and complete medical profile (e.g. your personal information, blood type, medical history, allergies, medication) to selected receipts via SMS and email. The accuracy of GPS location is varied with the performance of your phone's OS and GPS device.

2.3 Requirements Analysis

- A User-Friendly and minimal interface is required for easy and smooth handling of the website.
- The application is a combination of various existing systems.

- The use of visual data will increase the accuracy of fall detection.
- There are no intermediate people as the application connects directly to caretakers thereby reducing maintenance costs.

CHAPTER 3

SYSTEM DESIGN

SYSTEM DESIGN

3.1 Architectural Diagram/ block diagram

An architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components. It is a multiple view model that addresses different features and concerns of the system. It standardizes the software design documents and makes the design easy to understand by all stakeholders. It is an architecture verification method for studying and documenting software architecture design and covers all the aspects of software architecture for all stakeholders.

Architecture diagram is a system diagram used to abstract the overall design of the software system and the relationships, constraints and boundaries between components. This is an important tool because it provides an overview of the physical implementation of the software. System and its development plan.

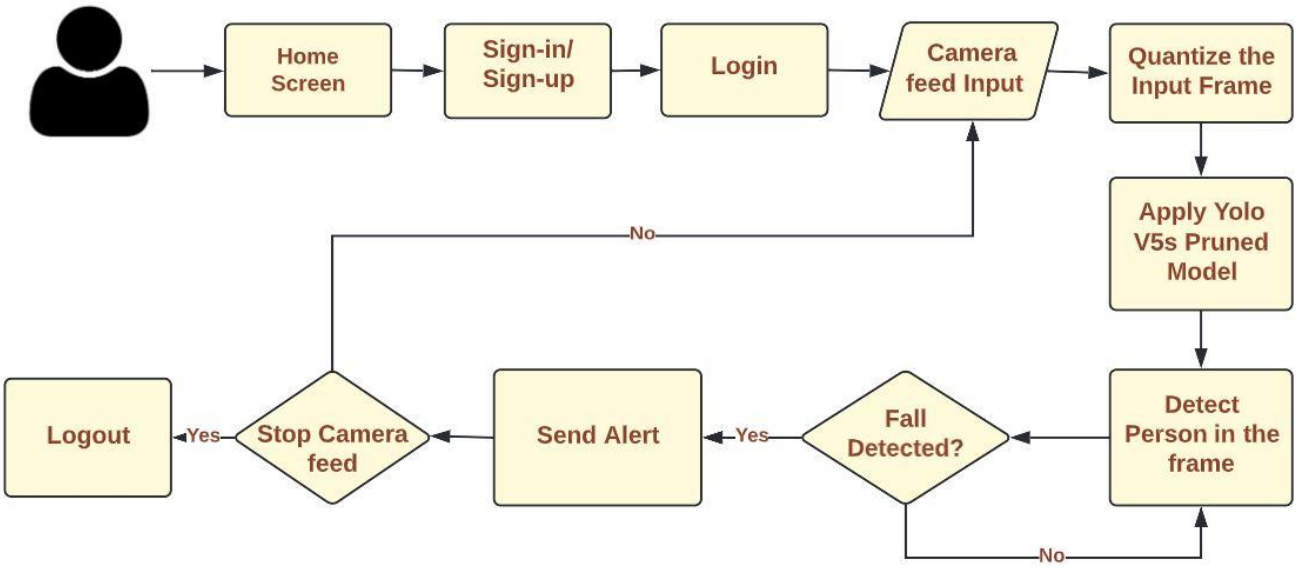

Figure 3.1 Architectural block diagram

In Figure 3.1 we have briefly described the overall architecture of fall detection system . A fall detection system receives an input in the form of video which contains any object. The object can be a person or anything. When our model detects a person using Yolo V5s Pruned Model, it starts

applying the Fall Detection Algorithm on it. If the fall is detected it sends an alert message to the respective authorities via email. And, if the fall is not detected it keeps on taking the camera feed as an input and applying the respective algorithms.

- **Camera Feed Input:** Input video for the system might be acquire by webcam or any other camera. That camera footage is known as Camera Feed Input.
- **Quantize the Input Frame:** It is the process of converting the video frame into contiguous numpy array. It helps in consuming less storage and increasing the resource optimization. It also ends up being more efficient as an input to the Yolo Model.
- **Yolo V5s Pruned Model:** YOLO is a state-of-the-art, real-time object detection system. The algorithm applies a single neural network to the full frame. The main advantage of YOLO is that it looks at the whole frame at test time and makes predictions with a single network evaluation which makes it extremely fast. It helps us in detecting whether the object in the frame is a person or anything else.
- **Fall Detection:** For Fall Detection we are using a purely hit and trial based model. In which we have set the measurements of the bounding box of the person that if width of the bounding box of the person is 5% i.e. 1.05 times greater than the height of the bounding box of the person, then fall is detected or else keep the fall detecting.
- **Alert:** When the fall is detected send an alert mail to the logged in mail id along with the frames of the fallen person.
- **Stop Camera Feed:** If the user wants to stop giving the input and stop the detection. He can simply do it by just clicking on the button or else he can simply keep it going and going.
- **Logout:** When the client is done using our application he/she can logout from our system and stop the camera live feed.

3.2 Flow chart

Flow Diagrams help us to understand a complex system in sequence of actions or movements taking place in it and various paths of objects.

The Figure 3.2 shows the overall flow of the proposed system. From user entering the system and seeing the home page to the fall detection and alert message send to the registered mail id.

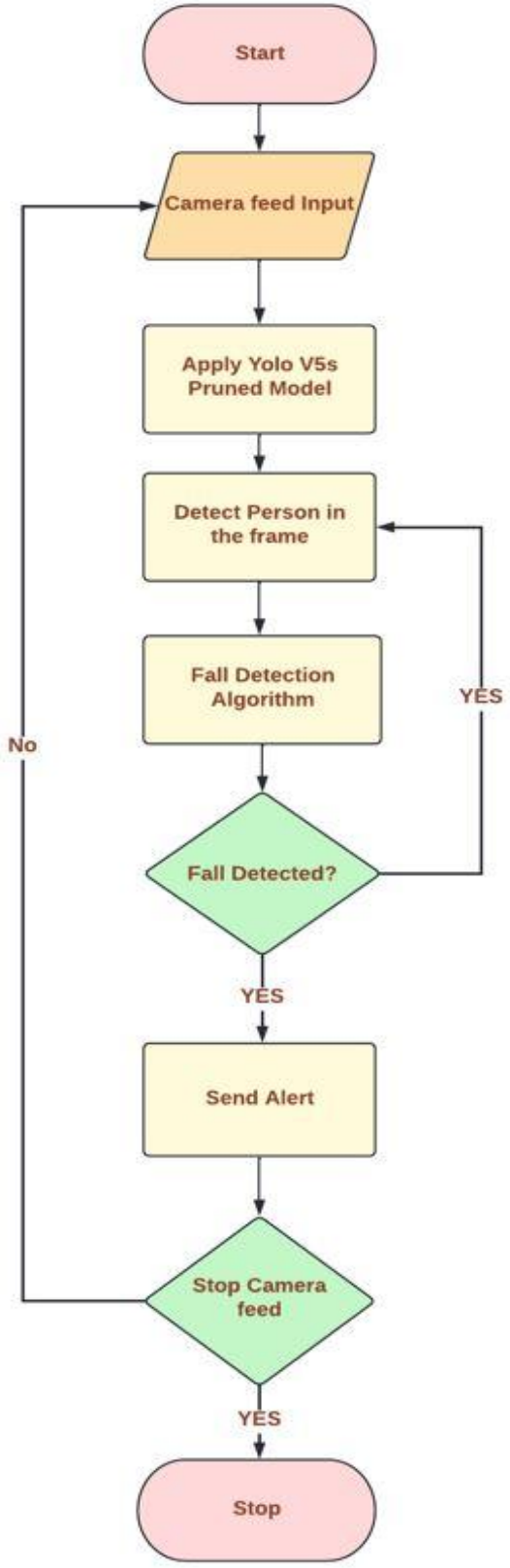

Figure 3.2 Flowchart

- **Start:** On viewing the website the homepage will be visible where the option to sign in or sign up will be there. After successfully logging in to the system, the user will be able to go to the fall detection.

- **Camera feed Input:** The user will then be prompted to feed the input from his/her camera of choice.

- **Apply Yolo V5s Pruned Model:** The frames from the camera feed which are in continuous form will be separated into array batches using numpy for better and easy understanding of image by the model. Then those matrices will be given as an input toh the YOLO model for detecting a person. The YOLO is a state-of-the-art, real-time object detection system. The algorithm applies a single neural network to the full frame. The main advantage of YOLO is that it looks at the whole frame at test time and makes predictions with a single network evaluation which makes it extremely fast. It helps us in detecting whether the object in the frame is a person or anything else.

- **Detect Person in the frame:** When the person is successfully been detected by the YOLO model a bounding box will be drawn around the person in the frame.

- **Fall Detected:** If the model detects that the width of the bounding box exceeds the height with 5% or more then fall will be detected.

- **Send Alert:** An alert mail will be sent to the registered user along with the image of the fall detection.

- **Stop Camera Feed:** When the user is done using our system, he/she can then choose whether they want to continue monitoring or stop the process.

- **Stop:** The user have successfully stopped the input camera feed and have logged out from the system.

CHAPTER 4

IMPLEMENTATION DETAILS

IMPLEMENTATION DETAILS

This section includes all the details of how the proposed system is implemented and minimum system requirements for the system to run smoothly.

4.1 Software requirements (Software/ Hardware)

4.1.1 Hardware requirements:

- PC, Mac or laptop with x86-64 (64-Bit) compatible processors
- Processor only Intel
- Operating System only Ubuntu
- 512 MB of free RAM
- Web camera for video footage

4.1.2 Software requirements:

The following describes the software needed in-order to develop the Image2Text application:

- **Frontend:**

  - Hyper Text Markup Language (HTML): The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. HTML is the code that is used to structure a web page and its content.

  - Cascading Style Sheets (CSS): Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is a stylesheet language used to describe the presentation of a document written in HTML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

  - JavaScript (JS): JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. JS is a programming language that

conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype- based object-orientation and first-class functions. It allows us to add dynamic behaviour to the webpage and add special effects to the webpage. On websites, it is mainly used for validation purposes. JavaScript helps us to execute complex actions and also enables the interaction of websites with visitors.

- **Database:**

  - MySQL: MySQL is an open-source relational database management system. MySQL is a database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

- **Backend:**

  - Python Flask: Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. Flask aims to keep its core functionality small yet typically extensible to cover an array of small and large applications. Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

  - YOLOv5s: YOLO is a state-of-the-art, real-time object detection system. The algorithm applies a single neural network to the full image. The main advantage of YOLO is that it looks at the whole image at test time and makes predictions with a single network evaluation which makes it extremely fast.

  - OpenCV: OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.

  - Neuralmagic DeepSparse: The DeepSparse Engine is a CPU runtime that delivers GPU-class performance by taking advantage of sparsity within neural networks to reduce compute required as well as accelerate memory bound workloads.

Software will be using:

- OS : Ubuntu 20.04
- Language : Python
- Tool Used : VS Code

4.2 Solution Approach/ Methodology

The methodology of the application involved reviews of numerous papers, case studies, articles, blogs and documentation.

In figure 4.2 we have given an overview of how the website is created and what are the different methods included in the website. A website is created which is simple to use and user friendly. Flask is used to connect the frontend with the backend. The backend model used for fall detection is made on VS Code using python. The system is a purely mathematical formula based model integrated with Deep sparse. The entire code has been written in python which is easy to understand with sufficient comments. The threshold values are determined through trial and error i.e. if the width of the rounding box around the person is 5% i.e. 1.05 times greater than the height of the rounding box then the fall will be detected.

Figure 4.2 Methodology

The user will start from the home page where he/she need to signup and create an account on the website. Then using that account credentials they need to login on the website and then they can start with the fall detection.

They need to give the camera access to the website for the detection of the person and alerting the user when there is fall. The camera feed is usually in the form of a video. The first step is to quantize the input frame i.e. converting those video frames into contiguous numpy array and it could provide those arrays to the YOLO V5s model for person detection. It helps in consuming less storage and increasing the resource optimization. It also ends up being more efficient as an input to the Yolo Model.

In YOLO V5s Pruned Model where from each frame the YOLO model will be scanning the objects and detecting whether there is a person in the frame or not. If the person is detected it will create a bounding box around the person and show the accuracy percentage i.e. how much percent the model is sure that the detected object is a person or not. For time being the accuracy percent is set to 60% i.e. if the model is 60% or more than 60% sure that the detected object is a person then only the model is allowed to categories the object as a person.

Fall in fall detection is an event in which a person suddenly and inadvertently collapses from an upright position and the person's legs can no longer support oneself. So to detect whether the person in the input camera feed is fallen or not we are going to use a mathematical formula i.e. if the width of the rounding box around the person is 5% i.e. 1.05 times greater than the height of the rounding box then there is fall.

These incidents of falling can have serious impact such as injury or in the worst-case scenario can cause even death if they do not get help immediately. Aid can be given more quickly if the occurrence of falls can be immediately detected. Therefore, alert message is send to the registered mail id along with a snapshot of the fall whenever the fall is detected and the person is lying for about 100 frames which are around 40 - 50 seconds. If the person gets in a not fall position from fall position within these 40 – 50 seconds, no mail is send to the registered mail id.

The user can stop the camera feed at any instance he/she wants. He/she can also logout from their respective accounts and end the camera feed.

CHAPTER 5

EXPERIMENTAL RESULTS

EXPERIMENTAL RESULTS

The result or output of each activity of the system is displayed in the form of screenshots including short explanation of the features present on the respective pages.

5.1 GUI (Graphical User Interface)

The user will first see the home page of the system as shown in the figure 5.1.1. The user will be provided with the Login option where he/she can create their account and use the fall detection. Without logging in the system, every option will direct the user to Login Page, except the Contact Us page.

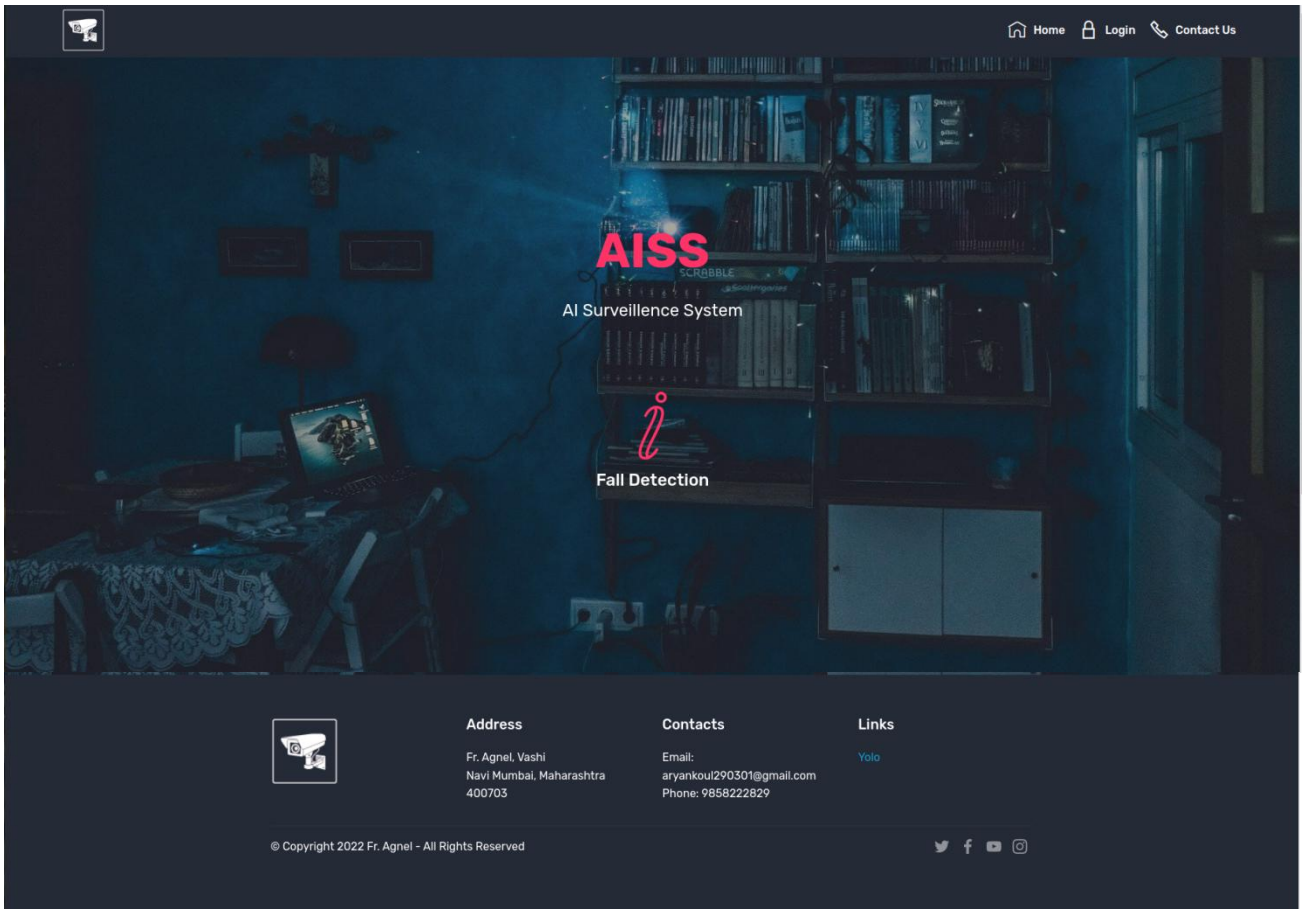

Figure 5.1.1 Home Page

When the user clicks on the Login button, he/she will be directed to the sign-up page as shown in the figure 5.1.2. There user is allowed to create an account and enter his/her credentials.

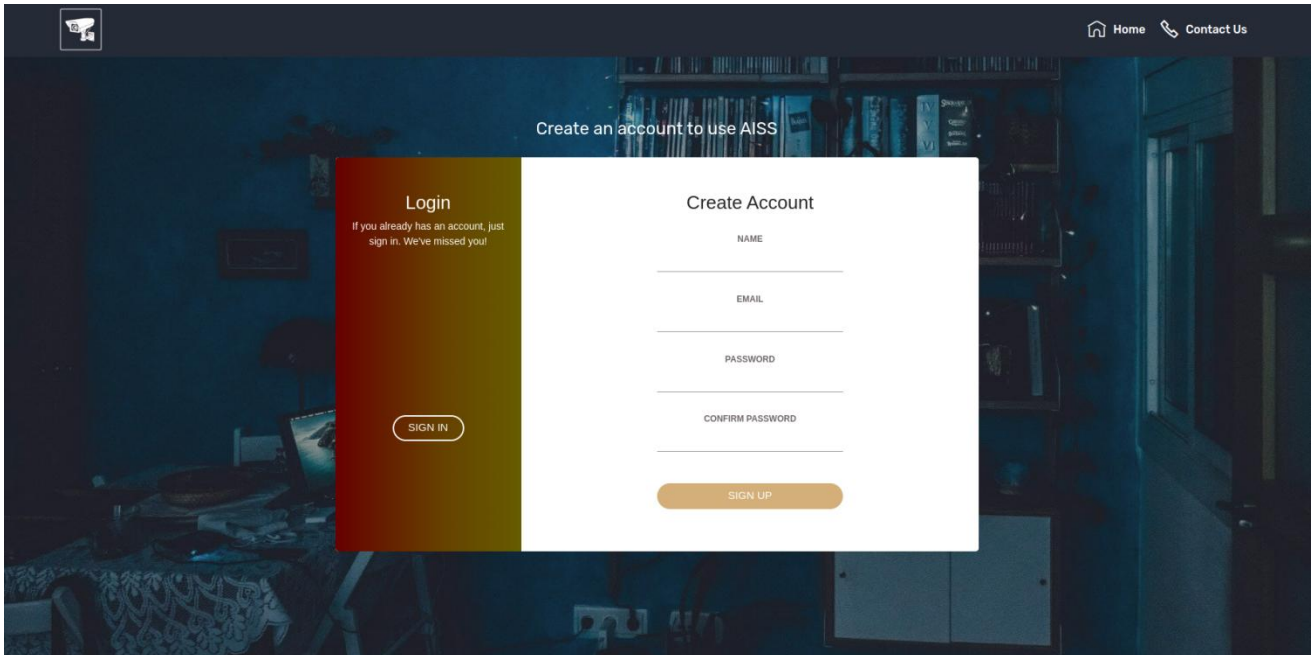

Figure 5.1.2 Creating Account

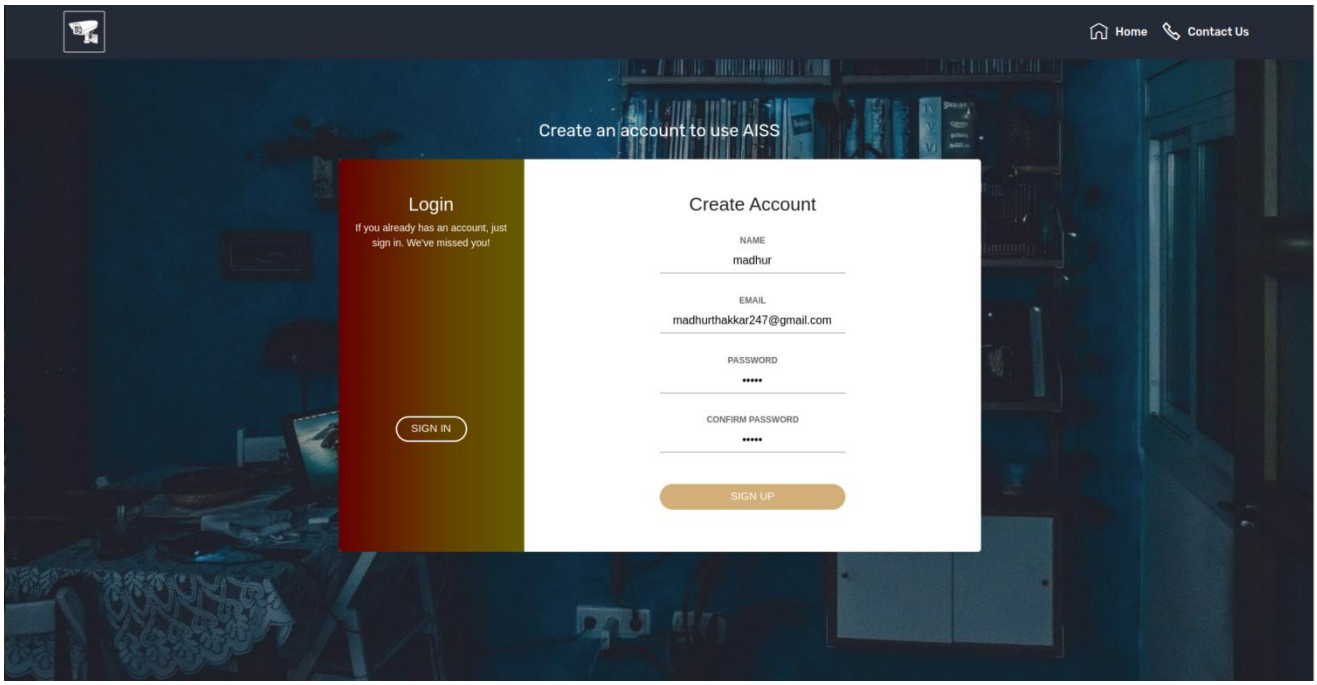

Figure 5.1.3 Sign Up

When the user enters their respective details and click on the sign-up button as shown in figure 5.1.3 an OTP will be sent to the entered mail. The OTP will be consisting of an one time password as shown in figure 5.1.4.

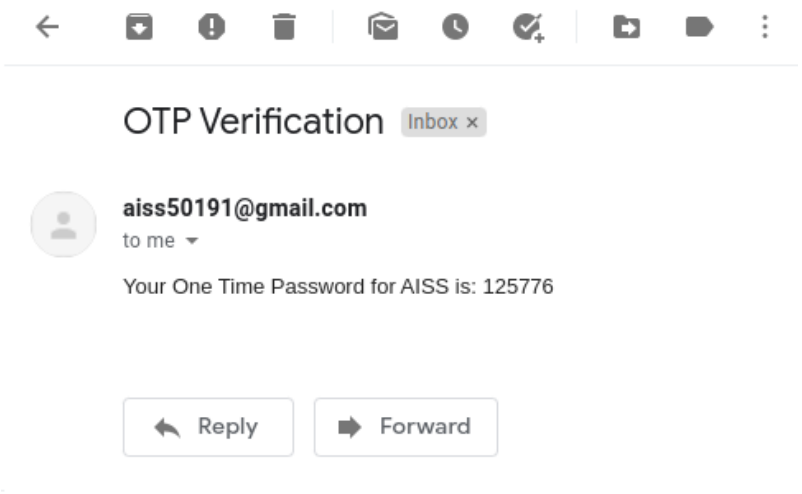

Figure 5.1.4 OTP Verification Mail

The user can enter that numeric password in the boxes as shown in figure 5.1.5 where after clicking on the sign-up button user is guided. If the user enters the wrong OTP he/she will get an alert message and will be requested to enter the right OTP.

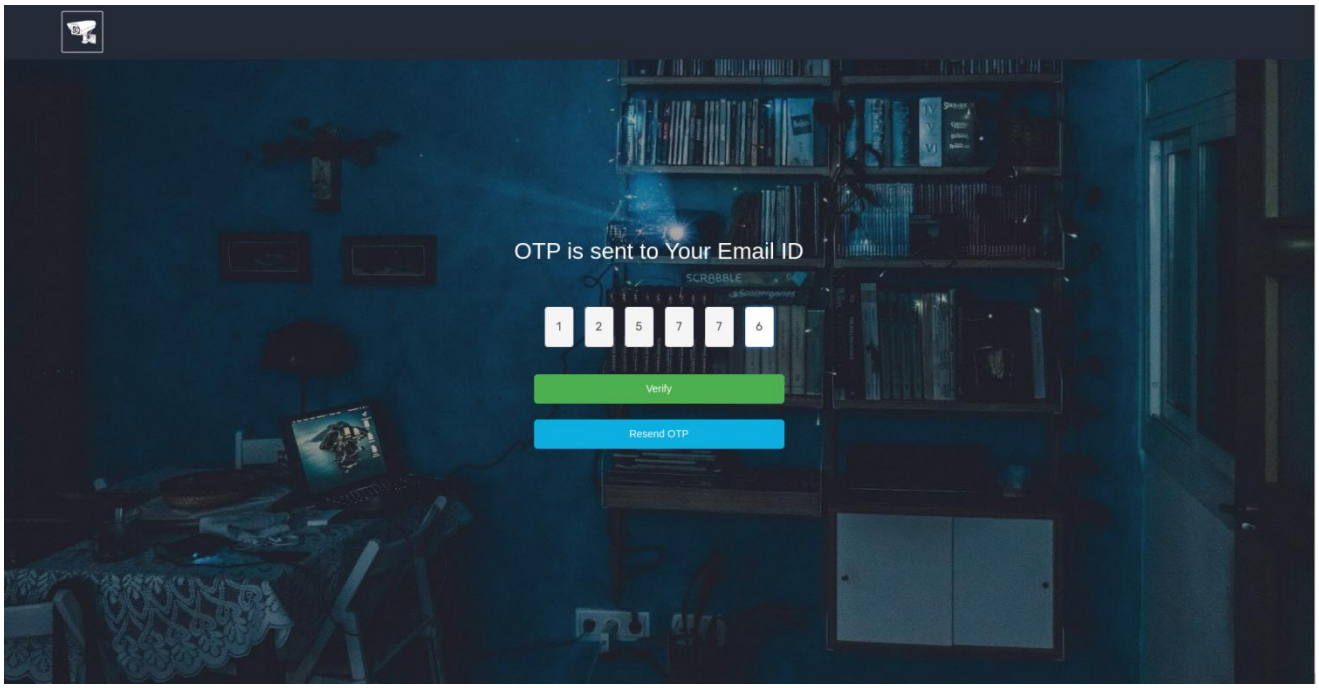

Figure 5.1.5 OTP Page

When the user enters the right OTP his/her account will be created successfully and his/her account credential will be saved in our database as shown in figure 5.1.6. For the security purposes we have also done password hashing for more users to trust our website.

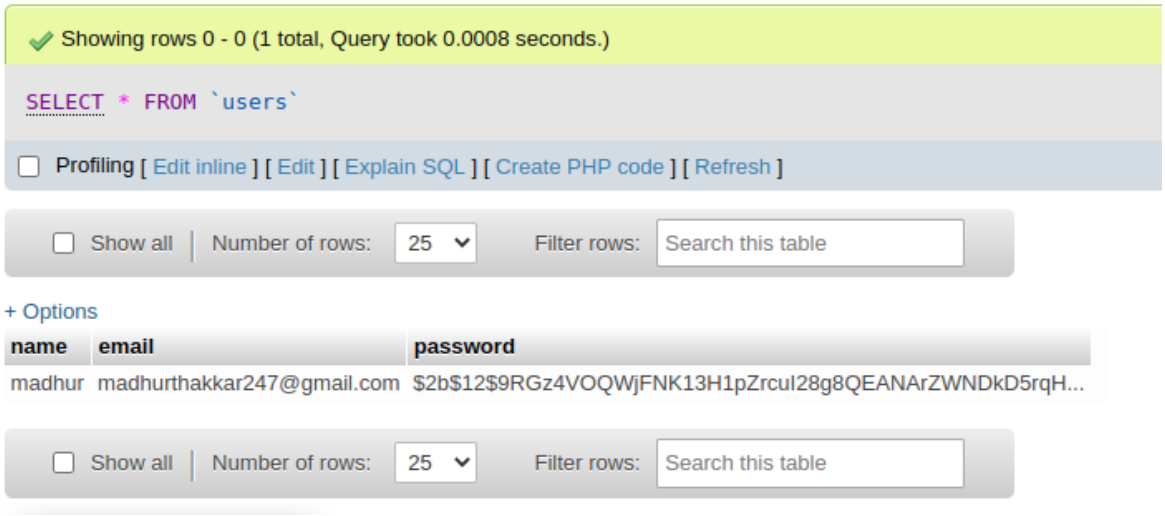

Figure 5.1.6 Password Hashing

Once the sign-up is done successfully, the user can go to the login page and enter their credentials to login into their system as shown in figure 5.1.7.

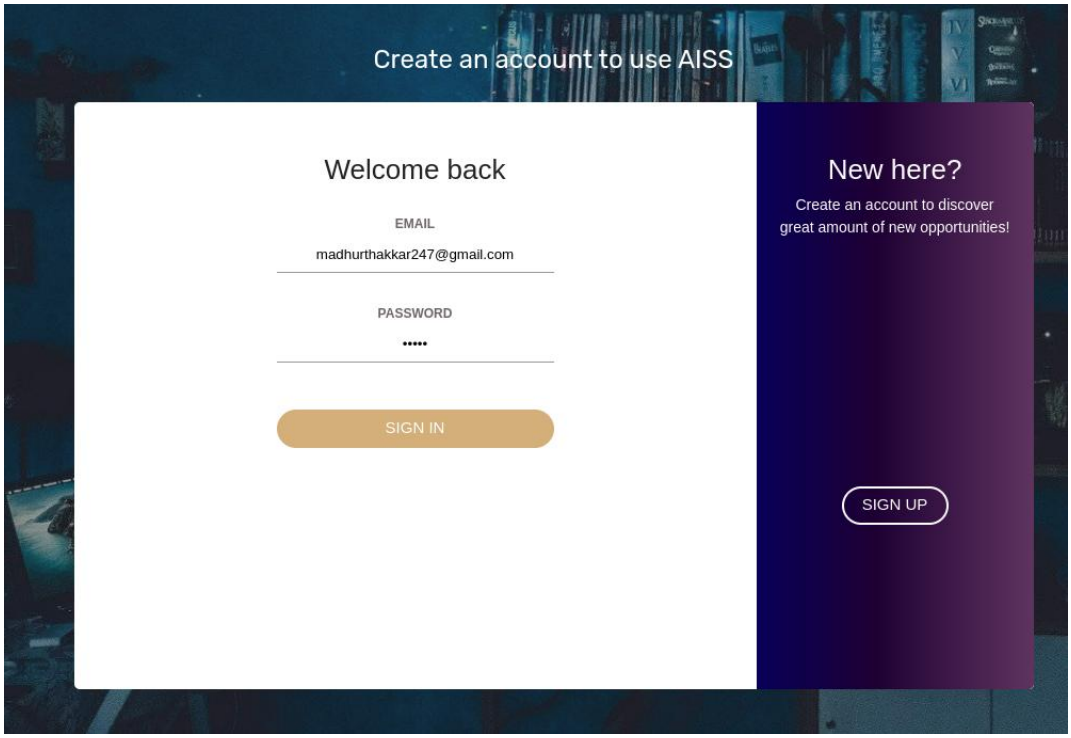

Figure 5.1.7 Login Page

In figure 5.1.8 snapshot of the home page is pasted after the login is done.

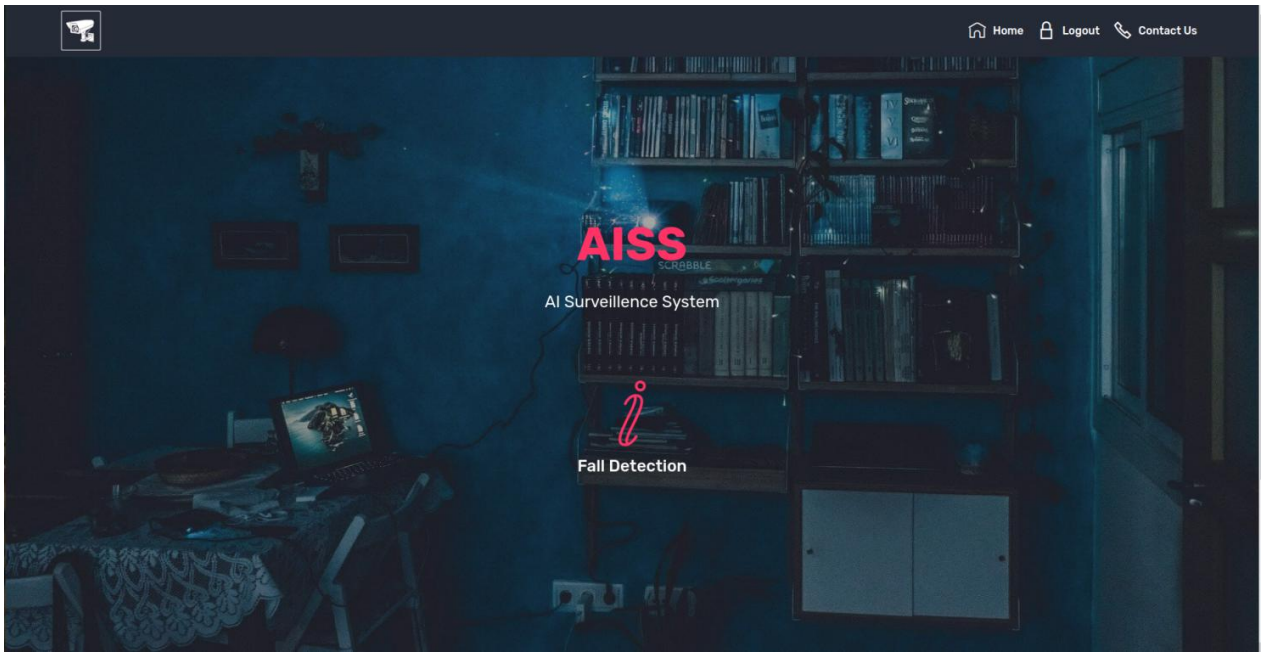

Figure 5.1.8 Home Page After Login

Once the user have logged in he/she is allowed to use our fall detection feature. When the user clicks on the "i" symbol he/she will be directed to the fall detection page as shown in figure 5.1.9**.**

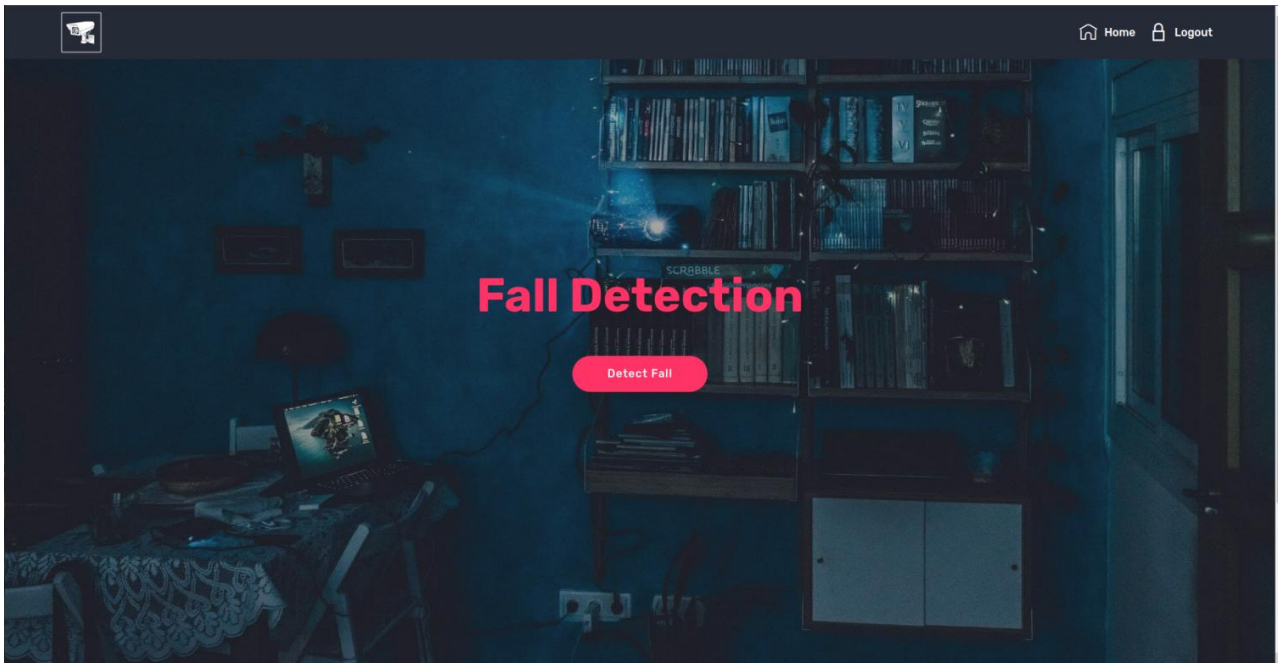

Figure 5.1.9 Fall Detection Page

After clicking on the Detect Fall button user will be providing the live camera feed to our system and our model will start detecting the persons as shown in figure 5.1.10 where a bounding box will be created around the person. The person detected in the snapshot is detected with an accuracy of 90% and the fps i.e. frames per second provided by our system is around 30-50 fps.

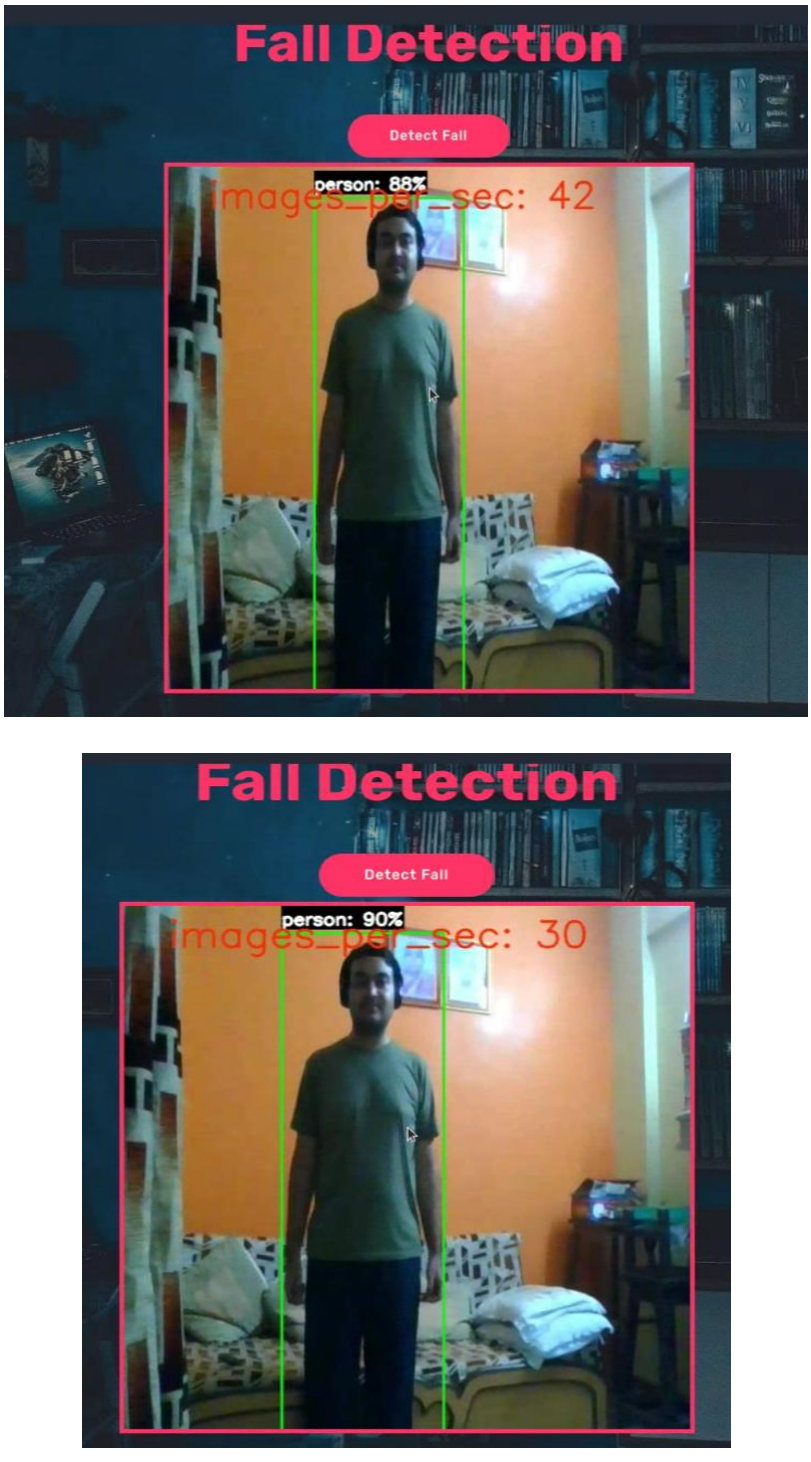

Figure 5.1.10 Person Detection

Once the person is detected by the model, the system will start detecting the fall and once the fall is detected as shown in figure 5.1.11 it will keep on detecting the person in the fall condition for around 100 fps and once the person is fallen for around 40 - 50 seconds, the system will send an alert message to logged in mail id along with the snapshot of the fallen person as shown in figure 5.1.12.

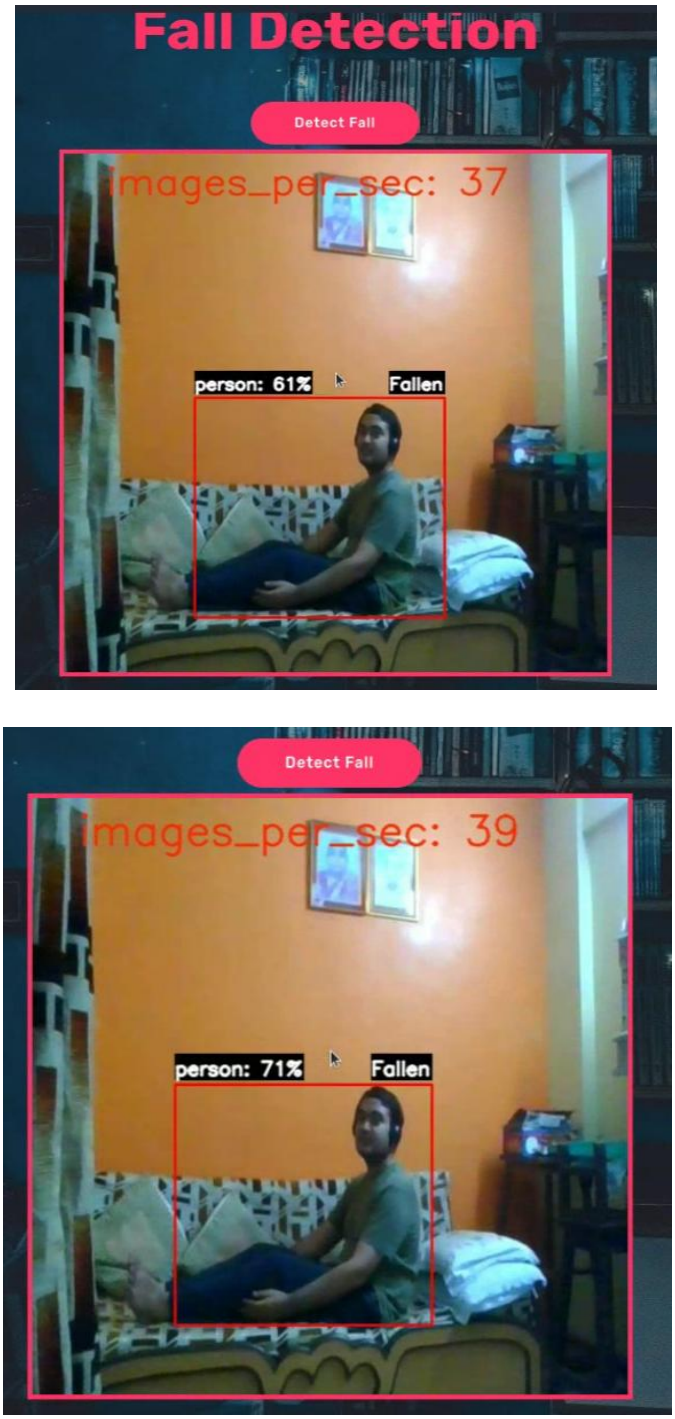

Figure 5.1.11 Fall Detection

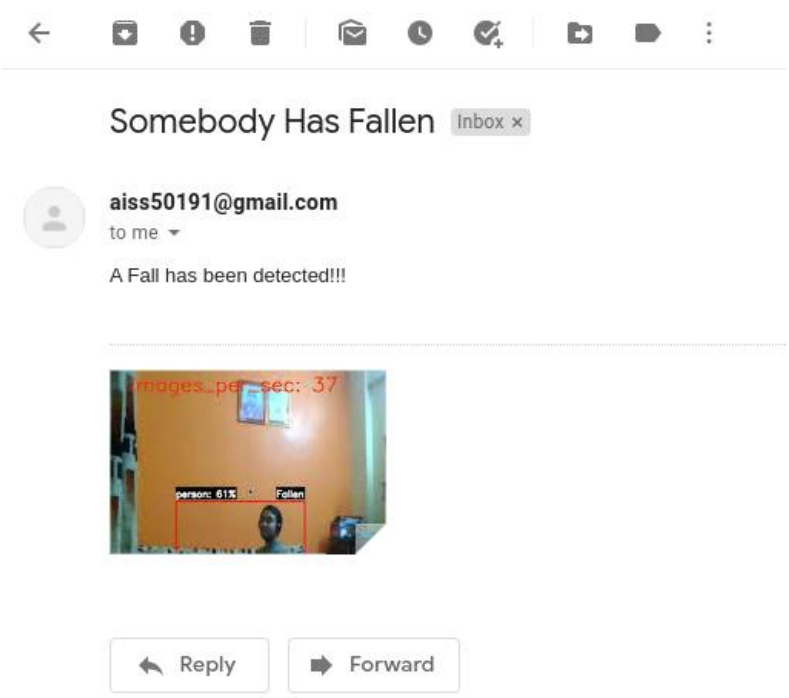

Figure 5.1.12 Alert Mail

Contact us page is also available for the user as shown in figure 5.1.13 where the user can give their valuable feedback on the system.

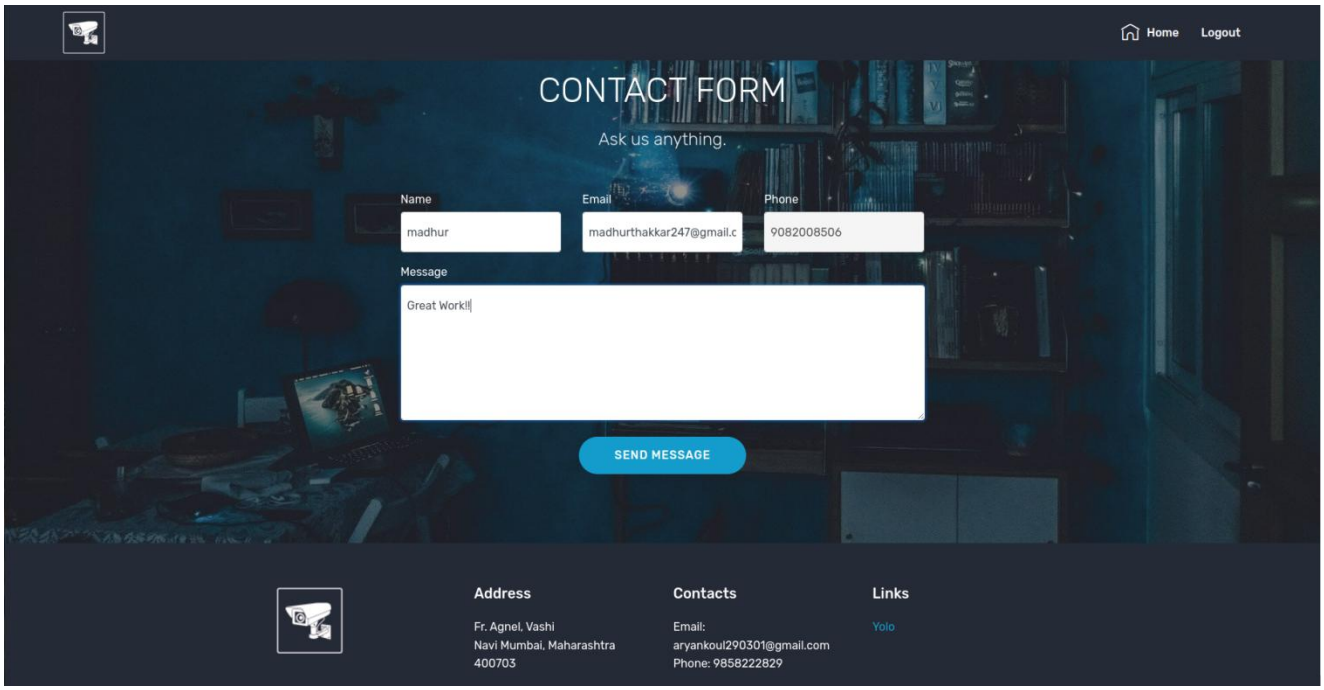

Figure 5.1.13 Feedback Page

And once the feedback is given by the user he/she will be directed to the thank you page where the social media websites of the system are available as we can see in the figure 5.1.14.

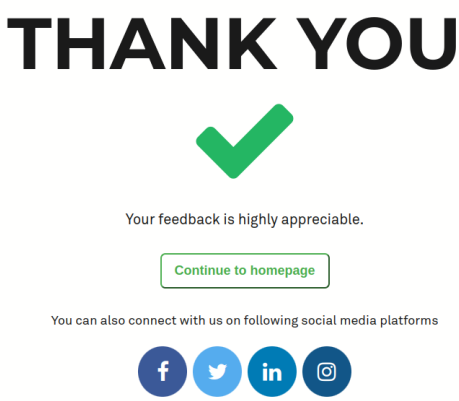

Figure 5.1.14 Social Media Page

Finally the Fig 5.1.15 shows the feedback database where the user's name, phone number and the feedback they have given is stored so that the system could be updated for better user experience.

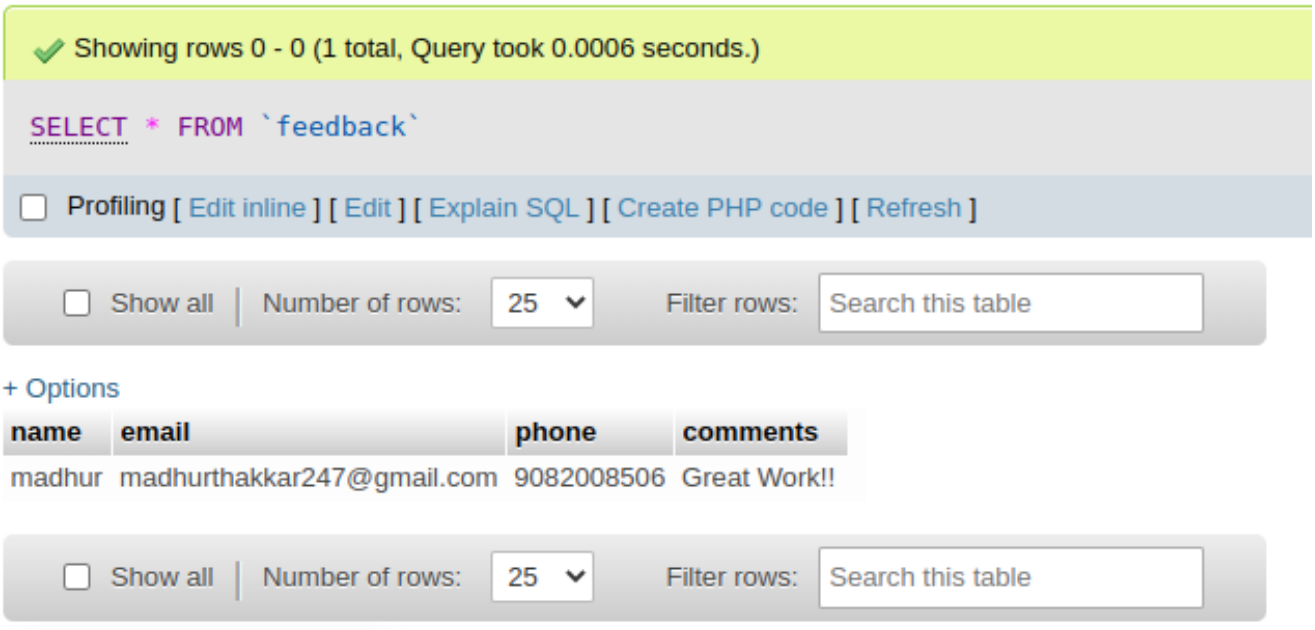

Figure 5.1.15 Feedback Database

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION AND FUTURE WORK

The conclusion of the proposed system is mentioned in this section. The section also highlights the possible future development which can be done in this system.

6.1 Conclusion

As the people are getting busy in their life, they have less time to take care of their loved ones. So, they are relying on the technology. Efficient methods to decrease the risks of people falling are needed to be devised. In the past several attempts have been made to carry out this task using conventional wearable technology or highly demanding computer vision methods. In order to gain high performance without compromising much on accuracy, an approach with resource optimization in mind was explored by AIFDS. The YOLOv5-s model based on Neuralmagic's Deepsparse proposed in this paper has successfully gained much higher performance on CPU with minimal effect on accuracy in the task of Fall Detection as compared to base YOLOv5 models. The system is able to achieve an accuracy of about 80%. The application detects the falls and notify the concerned people via email for immediate help. Hence, the dangers of such falls can be reduced as necessary help can be provided on time.

6.2 Future Work

To improvise further the application can be further enchanted in future by adding additional features that are an essential part of elder's lives. Some of these features are as below:

- Training a custom model based on neuralmagic's DeepSparse Engine for precise detection.
- Providing a common environment to store all their medical files and records.
- Implementing posture recognition for better accuracy of fall detection.
- The emergency button can be linked to multiple users to ensure that help is provided in most situations.
- Reminders for various activities can be set by giving appropriate title, description and alert timings as per user convenience.
- Implementing more AI surveillance system based models like: social distancing, vehicle crashing, object detection etc.
- Mails could be sent for multiple fall detection in a single frame. In one frame only one fall is detected, right now.

REFERENCES

[1]	S. F. Ali, M. Muaz, A. Fatima, F. Idrees and N. Nazar, "Human fall detection," INMIC, 2013, pp. 101-105, doi: 10.1109/INMIC.2013.6731332.
[2]	S. Badgujar and A. S. Pillai, "Fall Detection for Elderly People using Machine Learning," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-4, doi: 10.1109/ICCCNT49239.2020.9225494.
[3]	A. Gupta, R. Srivastava, H. Gupta and B. Kumar, "IoT Based Fall Detection Monitoring and Alarm System For Elderly," 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2020, pp. 1-5, doi: 10.1109/UPCON50219.2020.9376569.
[4]	S. M. Turkane, Swapnil J. Vikhe, C. B. Kadu, P. S. Vikhe, ``Automatic Body Fall Detection System for Elderly People using Accelerometer and Vision Based Technique",vol 8 issue 4 ,2019.
[5]	M. Nadi, N. El-Bendary, A. E. Hassanien and T. -h. Kim, "Falling Detection System Based on Machine Learning," 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), 2015, pp. 71-75, doi: 10.1109/AITS.2015.27. [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

APPENDIX A: CODE SAMPLE

app.py

```
from sys import stdout
import logging
from flask import Flask, render_template, session, request, flash
from flask_socketio import SocketIO, emit
from flask_session import Session
from flask_mysqldb import MySQL
import random
import bcrypt
# from camera import Camera
from utils import base64_to_pil_image
import cv2
from email.mime.text import MIMEText
import secrets
import numpy as np
import base64
from annotate import (_get_save_dir, annotate, _load_model)
import argparse
from Google import Create_Service
from deepsparse_utils import (
    YoloPostprocessor,
)
app = Flask(__name__)
app.logger.addHandler(logging.StreamHandler(stdout))
app.config['MYSQL_HOST'] = '127.0.0.1'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'aissdb'
mysql = MySQL(app)
app.secret_key = secrets.token_bytes(16)
app.config['DEBUG'] = True
app.config['SESSION_PERMANENT'] = False
app.config['SESSION_TYPE'] = "filesystem"
Session(app)
CLIENT_SECRET_FILE = "client_secret.json"
API_NAME = "gmail"
API_VERSION = "v1"
SCOPES = ['https://mail.google.com/']
socketio = SocketIO(app)
args = argparse.Namespace(device=None, engine='deepsparse', fp16=False, image_shape=[
                    416, 416], model_config=None, model_filepath='zoo:cv/detection/yolov5-
s/pytorch/ultralytics/coco/pruned_quant-aggressive_94', name=None, no_save=True, num_cores=None,
quantized_inputs=True, save_dir='annotation_results', source='0', target_fps=None)
model, has_postprocessing = _load_model(args)
postprocessor = (
    YoloPostprocessor(args.image_shape, args.model_config)
    if not has_postprocessing
    else None
)
save_dir = _get_save_dir(args)
```

```
@socketio.on('input image', namespace='/test')
def test_message(input):
    input = input.split(",")[1]
    image_data = np.asarray(base64_to_pil_image(input))
    cv2_img = annotate(args, postprocessor, image_data, model, save_dir,email=session.get('email'))
    cv2_img = cv2.cvtColor(cv2_img, cv2.COLOR_BGR2RGB)
    retval, buffer = cv2.imencode('.jpeg', cv2_img)
    b = base64.b64encode(buffer)
    b = b.decode()
    image_data = "" + b
    emit('out-image-event', {'image_data': image_data}, namespace='/test')
@socketio.on('connect', namespace='/test')
def test_connect():
    app.logger.info("client connected")
@app.route('/')
def index():
    session['login_status'] = False
    return render_template('index.html',status=session.get('login_status'))
@app.route('/login_index')
def login_index():
    return render_template("index.html", status=session.get('login_status'))
@app.route('/verify', methods=['GET','POST'])
def verify():
    if request.method == "POST":
        if request.form['name']!="" and request.form['email']!="" and request.form['password']!="" and
request.form['con_password']!="":
            if request.form['password']==request.form['con_password']:
                ver_email = request.form['email']
                session['email'] = ver_email
                reg = None
                try:
                    cur = mysql.connection.cursor()
                    cur.execute("SELECT * FROM users WHERE email = "'+session.get('email')+'"")
                    reg = cur.fetchone()
                    mysql.connection.commit()
                    cur.close()
                except:
                    print(reg,session.get('email'))
                if reg is None:
                    otp_gen = random.randint(100000,999999)
                    session['otp'] = otp_gen
                    print("verify")
                    print(session.get('otp'))
                    msg="Your One Time Password for AISS is: "+str(session.get('otp'))
                    service = Create_Service(CLIENT_SECRET_FILE, API_NAME, API_VERSION, SCOPES)
                    mimeMessage = MIMEText(msg, 'plain')
                    mimeMessage['to'] = session.get('email')
                    mimeMessage['subject'] = 'OTP Verification'
                    mimeMessage['from'] = 'aiss50191@gmail.com'
                    raw_string = base64.urlsafe_b64encode(mimeMessage.as_bytes()).decode()
                    message = service.users().messages().send(userId='me', body={'raw': raw_string}).execute()
                    print(message['id'])
                    session['name'] = request.form.get('name')
                    session['hashed'] = bcrypt.hashpw(request.form['password'].encode('utf-8'), bcrypt.gensalt())
                    print("hash",session.get('hashed'))
                    return render_template("otp.html")
```

```
            else:
                flash('This Email is already registered')
                return render_template("login.html")
        else:
            flash('Password and Confirm Password Do not Match!')
            return render_template("login.html")
    else:
        flash('Please enter all the details properly!')
        return render_template("login.html")
@app.route('/resend_verify', methods=['GET','POST'])
def resend_verify():
    if request.method == "POST":
        #email = request.form['email']
        otp_gen = random.randint(100000,999999)
        session['otp'] = otp_gen
        print("resend")
        msg="Your One Time Password for AISS is: "+str(session.get('otp'))
        service = Create_Service(CLIENT_SECRET_FILE, API_NAME, API_VERSION, SCOPES)
        mimeMessage = MIMEText(msg, 'plain')
        mimeMessage['to'] = session.get('email')
        mimeMessage['subject'] = 'OTP Verification'
        mimeMessage['from'] = 'aiss50191@gmail.com'
        raw_string = base64.urlsafe_b64encode(mimeMessage.as_bytes()).decode()
        message = service.users().messages().send(userId='me', body={'raw': raw_string}).execute()
        flash(u"New OTP is sent to {0}".format(session.get('email')))
        return render_template("otp.html")
@app.route('/userlogin',methods=['GET','POST'])
def userlogin():
    if request.method=='POST':
        if request.form['loginemail'] != "" and request.form['loginpass'] != "":
            email = request.form['loginemail']
            password = request.form['loginpass']
            reg = None
            try:
                cur = mysql.connection.cursor()
                cur.execute("SELECT * FROM users WHERE email = '"+email+"'")
                reg = cur.fetchone()
                session['hashed'] = reg[2].encode("utf-8")
                session['email'] = reg[1]
                session['name'] = reg[0]
                mysql.connection.commit()
                cur.close()
            except:
                pass
            if reg is None:
                flash('This Email is not registered')
                return render_template("login.html")
            else:
                if bcrypt.checkpw(password.encode('utf-8'), session.get('hashed')):
                    session['login_status'] = True
                    if session.get('login_dest')=="index":
                        return render_template("index.html",status=session.get('login_status'))
                    elif session.get('login_dest')=="falldetection":
                        return render_template("FallDetection.html")
                else:
                    flash('Incorrect Password!')
```

```
                return render_template("login.html")
        else:
            flash('Please enter all the details properly!')
            return render_template("login.html")
@app.route('/authenticate', methods=['GET','POST'])
def authenticate():
    if request.method == "POST":
        otp_get = request.form.getlist('otp')
        otp_enter = ""
        for ele in otp_get:
            otp_enter+=str(ele)
        if len(otp_enter)<1:
            otp_enter = -1
        if int(otp_enter)==session.get('otp'):
            cur = mysql.connection.cursor()
            cur.execute("INSERT INTO users
VALUES('"+session.get('name')+"','"+session.get('email')+"','"+str(session.get('hashed'))[2:-1]+"')")
            mysql.connection.commit()
            cur.close()
            session['login_status'] = True
            if session.get('login_dest')=="index":
                return render_template("index.html",status=session.get('login_status'))
            elif session.get('login_dest')=="falldetection":
                return render_template("FallDetection.html")
        else:
            flash("Incorrect OTP\nTry Again!")
            return render_template("otp.html")
@app.route('/falldetection')
def falldetection():
    return render_template('FallDetection.html')
@app.route('/login/<dest>')
def login(dest):
    session['login_dest'] = dest
    return render_template("login.html")
@app.route('/contactus')
def contactus():
    return render_template("ContactUs.html",status=session.get('login_status'))
@app.route('/feedback',methods=['POST','GET'])
def feedback():
    if request.method=="POST":
        name = request.form['feed_name']
        email = request.form['feed_email']
        phone = request.form['feed_phone']
        comment = request.form['comments']
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO feedback VALUES('"+name+"','"+email+"','"+phone+"','"+comment+"')")
        mysql.connection.commit()
        cur.close()
        return render_template("feedback.html")
if __name__ == '__main__':
    socketio.run(app)
```

index.html

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="generator" content="Mobirise v4.9.7, mobirise.com">
   <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1">
   <link rel="shortcut icon" href="../static/assets/images/cctv_white.png" type="image/x-icon">
   <meta name="description" content="">
   <title>AISS</title>
   <link rel="stylesheet" href="../static/assets/web/assets/mobirise-icons/mobirise-icons.css">
   <link rel="stylesheet" href="../static/assets/web/assets/mobirise-icons-bold/mobirise-icons-bold.css">
   <link rel="stylesheet" href="../static/assets/web/assets/mobirise-icons2/mobirise2.css">
   <link rel="stylesheet" href="../static/assets/tether/tether.min.css">
   <link rel="stylesheet" href="../static/assets/bootstrap/css/bootstrap.min.css">
   <link rel="stylesheet" href="../static/assets/bootstrap/css/bootstrap-grid.min.css">
   <link rel="stylesheet" href="../static/assets/bootstrap/css/bootstrap-reboot.min.css">
   <link rel="stylesheet" href="../static/assets/socicon/css/styles.css">
   <link rel="stylesheet" href="../static/assets/dropdown/css/style.css">
   <link rel="stylesheet" href="../static/assets/theme/css/style.css">
   <link rel="stylesheet" href="../static/assets/mobirise/css/mbr-additional.css" type="text/css">
</head>
<body>
   <section class="menu cid-s3MrIgmP7I" once="menu" id="menu1-5">
      <nav
         class="navbar navbar-expand beta-menu navbar-dropdown align-items-center navbar-fixed-top navbar-
toggleable-sm">
         <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
            data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
            aria-label="Toggle navigation">
            <div class="hamburger">
               <span></span>
               <span></span>
               <span></span>
               <span></span>
            </div>
         </button>
         <div class="menu-logo">
            <div class="navbar-brand">
               <span class="navbar-logo">
                  <a href="http://localhost:5000/login_index">
                     <img src="../static/assets/images/cctv.png" alt="AISS" title="" style="height: 3.8rem;">
                  </a>
               </span>
            </div>
         </div>
         <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav nav-dropdown nav-right" data-app-modern-menu="true">
               <li class="nav-item">
                  <a class="nav-link link text-white display-4" href="http://localhost:5000/login_index"><span
                        class="mbrib-home mbr-iconfont mbr-iconfont-btn"></span>
                     Home</a>
               </li>
```

```
                {% if status==False %}
                <li class="nav-item"><a class="nav-link link text-white display-4"
                        href="http://localhost:5000/login/index"><span
                            class="mobi-mbri mbri-login mbr-iconfont mbr-iconfont-btn"></span>
                        Login</a>
                </li>
                {% else %}
                <li class="nav-item"><a class="nav-link link text-white display-4"
                    href="http://localhost:5000/"><span
                            class="mobi-mbri mbri-login mbr-iconfont mbr-iconfont-btn"></span>
                    Logout</a>
            </li>
            {% endif %}
                <li class="nav-item"><a class="nav-link link text-white display-4"
                        href="http://localhost:5000/contactus"><span
                            class="mobi-mbri mobi-mbri-phone mbr-iconfont mbr-iconfont-btn"></span>
                        Contact Us</a>
                </li>
            </ul>
        </div>
    </nav>
</section>
<section class="header12 cid-s3Mrg3sUiE mbr-fullscreen mbr-parallax-background" id="header12-1">
    <div class="mbr-overlay" style="opacity: 0.4; background-color: rgb(35, 35, 35);">
    </div>
    <div class="container  ">
        <div class="media-container">
            <div class="col-md-12 align-center">
                <h1 class="mbr-section-title pb-3 mbr-white mbr-bold mbr-fonts-style display-1">
                    <br>AISS
                </h1>
                <p class="mbr-text pb-3 mbr-white mbr-fonts-style display-5">
                    AI Surveillence System</p>
                <div class="icons-media-container mbr-white">
                    <div class="card col-12 col-md-6 col-lg-3">
                        <div class="icon-block">
                            {% if status==True %}
                            <a href="http://localhost:5000/falldetection">
                                <span class="mbr-iconfont mbri-italic"
                                    style="color: rgb(255, 51, 102); fill: rgb(255, 51, 102);"></span>
                            </a>
                            {% else %}
                            <a href="http://localhost:5000/login/falldetection">
                                <span class="mbr-iconfont mbri-italic"
                                    style="color: rgb(255, 51, 102); fill: rgb(255, 51, 102);"></span>
                            </a>
                            {% endif %}
                        </div>
                        <h5 class="mbr-fonts-style display-5">
                            Fall Detection</h5>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
```

```
<section class="cid-s3MrrUyL0E" id="footer1-3">
    <div class="container">
        <div class="media-container-row content text-white">
            <div class="col-12 col-md-3">
                <div class="media-wrap">
                    <a href="http://localhost:5000/login_index">
                        <img src="../static/assets/images/cctv.png" alt="AISS" title="">
                    </a>
                </div>
            </div>
            <div class="col-12 col-md-3 mbr-fonts-style display-7">
                <h5 class="pb-3">
                    Address
                </h5>
                <p class="mbr-text">Fr. Agnel, Vashi
                    <br>Navi Mumbai, Maharashtra
                    <br>400703
                </p>
            </div>
            <div class="col-12 col-md-3 mbr-fonts-style display-7">
                <h5 class="pb-3">
                    Contacts
                </h5>
                <p class="mbr-text">
                    Email: aryankoul290301@gmail.com
                    <br>Phone: 9858222829
                </p>
            </div>
            <div class="col-12 col-md-3 mbr-fonts-style display-4">
                <h5 class="pb-3">
                    Links
                </h5>
                <p class="mbr-text">
                    <a class="text-primary" href="https://pjreddie.com/darknet/yolo/">Yolo</a>
                    <br>
                </p>
            </div>
        </div>
        <div class="footer-lower">
            <div class="media-container-row">
                <div class="col-sm-12">
                    <hr>
                </div>
            </div>
            <div class="media-container-row mbr-white">
                <div class="col-sm-6 copyright">
                    <p class="mbr-text mbr-fonts-style display-7">
                        © Copyright 2022 Fr. Agnel - All Rights Reserved
                    </p>
                </div>
                <div class="col-md-6">
                    <div class="social-list align-right">
                        <div class="soc-item">
                            <a href="https://twitter.com" target="_blank">
                                <span class="mbr-iconfont mbr-iconfont-social socicon-twitter socicon"></span>
                            </a>
```

```
                    </div>
                    <div class="soc-item">
                        <a href="https://www.facebook.com" target="_blank">
                            <span class="mbr-iconfont mbr-iconfont-social socicon-facebook socicon"></span>
                        </a>
                    </div>
                    <div class="soc-item">
                        <a href="https://www.youtube.com" target="_blank">
                            <span class="mbr-iconfont mbr-iconfont-social socicon-youtube socicon"></span>
                        </a>
                    </div>
                    <div class="soc-item">
                        <a href="https://www.instagram.com" target="_blank">
                            <span class="mbr-iconfont mbr-iconfont-social socicon-instagram socicon"></span>
                        </a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
<script src="../static/assets/web/assets/jquery/jquery.min.js"></script>
<script src="../static/assets/popper/popper.min.js"></script>
<script src="../static/assets/tether/tether.min.js"></script>
<script src="../static/assets/bootstrap/js/bootstrap.min.js"></script>
<script src="../static/assets/vimeoplayer/jquery.mb.vimeo_player.js"></script>
<script src="../static/assets/parallax/jarallax.min.js"></script>
<script src="../static/assets/smoothscroll/smooth-scroll.js"></script>
<script src="../static/assets/dropdown/js/script.min.js"></script>
<script src="../static/assets/touchswipe/jquery.touch-swipe.min.js"></script>
<script src="../static/assets/theme/js/script.js"></script>
</body>
</html>
```

ACKNOWLEDGEMENT

The making of the project "AI based Fall Detection System using YOLOv5-s and DeepSparse" involves contribution of many people. We owe a great thanks to many people who have helped and supported us during the course of our project. We express our gratitude to Dr. S. M. Khot, Principal of Fr. C. Rodrigues Institute of Technology, H.O.D. of IT department Dr. Vaishali Bodade, the Project Coordinator of IT department Prof. Sharlene Rebeiro for extending their inevitable and valuable support to us.

We would take this opportunity to thank our guide Prof. Archana Shirke for guiding and correcting various documents of ours with attention and care. This synopsis could not have been completed without their knowledge and expertise.
We would also thank our Institution and our faculty members without whom this project would have been a distant reality. We also extend our heartfelt thanks to our families and well-wishers.

Yours sincerely,

Joel Dsilva

Aryan Koul

Madhur Thakkar