Project Progress Report on

A Framework for Fall Detection Using Audio and Video Features

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING
Submitted by:

Medha Bisht	**2017341**
Yogesh Thakur	**2017397**
Aryamann Singh	**2017313**
Divyam Kholia	**2017324**

Under the Guidance of
Dr. Durgaprasad Gangodkar
Professor, Department of Computer Science & Engineering

Project Team ID: MP23CE004
Project Progress Report No: 1

Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
2023-24

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project progress report entitled **"A Framework for Fall Detection Using Audio and Video Features"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Dr. Durgaprasad Gangodkar, Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Medha Bisht	University Roll no: 2017341	Signature:
Yogesh Thakur	University Roll no: 2017397	Signature:
Aryamann Singh	University Roll no: 2017313	Signature:
Divyam Kholia	University Roll no: 2017324	Signature:

The above mentioned students shall be working under the supervision of the undersigned on the **"A Framework for Fall Detection Using Audio and Video Features"**

Supervisor **Head of the Department**

Examination

Name of the Examiners: **Signature with Date**

1. (Prof) Dr. Santosh Kumar
2. Dr. Satya Prakash Mourya

Table of Contents

Chapter 1

Introduction and Problem Statement

1.1 Basic Overview:

According to WHO data an estimated 684 000 fatal falls occur each year, for seniors aged 79 and above, falls are the primary reason for injury-related deaths, making it the second leading cause of unintentional death, after road injuries. The financial cost of fall related injuries and long-term care are substantial all over the world, usually leading to some sort of disability.

The current methods commonly used to detect falls are portable sensors which need to be worn or embedded on various parts of the human body, in order for them to be able to detect falling events experienced by the person. Our framework for fall detection is based on vision, that can automatically monitor and detect falls, recognize distress calls from the injured and send out help messages to local emergency numbers for timely medical care. When the system identifies a potential fall and determines distress through audio cues, it automatically triggers a call for help.

1.1.1 Vision:

We recognize the serious impact of falls, causing injuries to fatalities, especially with a growing elderly population. In response, there's a strong need for improved surveillance, particularly advanced fall detection systems. Our vision is a system using vision-based technology to automatically monitor and detect falls. It goes beyond detection, identifying distress calls from the injured.

1.1.2 Why a fall detection framework using audio and video features?

The main issue with existing fall detection systems is to differentiate any fall from daily life activities like crouching, sitting down etc. So, the event of fall can be divided into three parts one is the pre-fall phase represents the daily life activities. Secondly, the critical phase which represents the movement of body towards the ground or the shock of the body’s impact with the ground. Thirdly, the post fall phase representing the motionlessness of the person after falling on the ground.

1.2 Problem Statement:

Falls can lead to severe injuries, diminished quality of life, and, in unfortunate cases, even fatal consequences. As the elderly population grows worldwide, there is a demand for better surveillance systems, specifically fall detection systems to tackle this issue. A fall detection system based on vision, that can automatically monitor and detect falls, recognize distress calls from the injured and send out help messages to local emergency numbers for timely medical care. When the system identifies a potential fall and determines distress through audio cues, it automatically triggers a call for help.

Chapter 2

Objectives

The objectives of the proposed work are as follows:

1. **Detect fall in complex situation:** To accurately detect falls on the basis of the video and live time camera feed, even in challenging environments.
2. **Distress Call:** To accurately detect falls on the basis of the distress call made by the person during or after the fall for help, even in noisy environment.
3. **Fast Response:** To minimize the delay between the occurrence of a fall and the system's detection for emergency situations.
4. **Removal of False alarm:** To minimize the number of false alarms, which can be caused by other activities, such as sitting down or lying down.
5. **Real-Time Assistance:** To provide alerts quickly so that help can be provided to the person who has fallen as soon as possible.
6. **User Assistance:** To Provide additional information about the fall, such as potential injuries, to assist caregivers and medical professionals.

Chapter 3

Project Work Carried Out

3.1. Research Done so Far:

A fall detection system implemented using Media Pipe and OpenCV works in the following steps:

1. Initialization:

- Capture video stream: OpenCV captures video from a camera.
- Media Pipe setup: Media Pipe's Holistic model is loaded for pose estimation.

2. Frame Processing:

- Frame reading: Each frame is read from the video stream.
- Pose estimation: Media Pipe analyses the frame and generates 3D key points representing the body's pose.
- Feature extraction: Relevant features for fall detection are calculated from the key points, such as height, orientation, velocity, acceleration, and angles between limbs.

3. Fall Detection Logic:

- Feature analysis: The extracted features are compared to predefined thresholds for fall detection.
- Temporal analysis: Features are analysed over multiple frames to differentiate sudden changes from regular movements.
- State machine: (Optional) A state machine tracks the person's overall motion sequence ("standing," "falling," "recovering") for improved accuracy.
- Fall detection: If the features and sequence satisfy the fall detection criteria, a fall event is confirmed.

Here's a simplified breakdown of the workflow:

1. Video frame captured.
2. Pose estimated with Media Pipe.
3. Key point data extracted.

4. Relevant features calculated.
5. Features compared to fall detection thresholds.
6. Temporal analysis performed.
7. Fall confirmed based on features and sequence.
8. Alert triggered, and appropriate action taken.

Technical details:

- Media Pipe: Provides real-time pose estimation with 3D key points.
- OpenCV: Handles video capture, image processing, and thresholding.
- Feature analysis: Specific features chosen and thresholds set depending on the application and desired sensitivity.
- Fall detection logic: Can be simple rule-based or involve more complex machine learning models.

Benefits of using Media Pipe and OpenCV:

- Open source and free to use.
- Easy to set up and implement.
- Flexibility to customize and optimize for specific needs.
- Active community and resources available for support.

3.2 Research Progress Overview:

To estimate human body, pose in a video/image we used media pipe – pose estimation by google. Its land marker task lets us to detect different points of human body in an image or video as well as in live feed. So, to detect a fall the model measures if the center of gravity of a person in the feed is in between the coordinates of the feet else the system detects a fall in the feed.

- **Step 1: Extracting Landmarks**

  Using the media pipe landmark function, we extracted the landmarks for different body parts of a human in the feed. The coordinates for different joints are calculated in different manner the different manners are displayed in the images below.

```
# finding the position of body part

dot_LEFT_SHOULDER_X= int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_SHOULDER].x * image_width)
dot_LEFT_SHOULDER_Y= int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_SHOULDER].y * image_hight)

dot_LEFT_ELBOW_X= int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ELBOW].x * image_width)
dot_LEFT_ELBOW_Y= int(results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_ELBOW].y * image_hight)


# generating co ordinates for joints

dot_LEFT_ARM_SHOULDER_ELBOW_X = int( (dot_LEFT_SHOULDER_X+dot_LEFT_ELBOW_X) / 2)
dot_LEFT_ARM_SHOULDER_ELBOW_Y = int( (dot_LEFT_SHOULDER_Y+dot_LEFT_ELBOW_Y) / 2)

LEFT_ARM_SHOULDER_ELBOW = [    dot_LEFT_ARM_SHOULDER_ELBOW_X    ,     dot_LEFT_ARM_SHOULDER_ELBOW_Y     ]

# getting co ordinates for angle calculation

elbow_l = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
shoulder_l = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
hip_l = [landmarks[mp_pose.PoseLandmark.LEFT_HIP.value].x,landmarks[mp_pose.PoseLandmark.LEFT_HIP.value].y]
```

Fig – 3.2.2 – Extracting landmarks.

- **Step 2: Calculating Angles**

  Using the landmarks extracted from the last step we calculate the angles between different joints of human body. The method is displayed in images below.

```
# calculating angle

angle_shoulder_l = calculate_angle(elbow_l, shoulder_l, hip_l)
```

Fig – 3.2.3 – Calculating angles using extracted landmarks.

```
def calculate_angle(a,b,c):
    a = np.array(a) # First
    b = np.array(b) # Mid
    c = np.array(c) # End

    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle >180.0:
        angle = 360-angle

    return int(angle)
```

Fig – 3.2.4 – Function of calculating angles.

- **Step 3: Extracting Points for Base of Human Body**

  This part is most important as we are calculating and generating the co ordinates of the base of human body which will be one of the deciding factors of a fall. These points will be extracted using the landmarks extracted in the *Step 1*. These points are denoted as points of action and the way these points of actions are calculated is shown below.

```
# points of actions for both feet

Point_of_action_LEFT_X = int(
    ((dot_LEFT_FOOT_INDEX_X +  dot_LEFT_HEEL_X)/2) )

Point_of_action_LEFT_Y = int(
    ((dot_LEFT_FOOT_INDEX_Y+   dot_LEFT_HEEL_Y)/2) )


Point_of_action_RIGHT_X = int(
    ((dot_RIGHT_FOOT_INDEX_X +  dot_RIGHT_HEEL_X)/2) )

Point_of_action_RIGHT_Y = int(
    ((dot_RIGHT_FOOT_INDEX_Y+   dot_RIGHT_HEEL_Y)/2) )


# co ordinates between Feet

Point_of_action_X = int ( (Point_of_action_LEFT_X +  Point_of_action_RIGHT_X)/2 )

Point_of_action_Y = int ( (Point_of_action_LEFT_Y +  Point_of_action_RIGHT_Y)/2 )


# co ordinates between feet
Point_of_action = [Point_of_action_X , Point_of_action_Y]
```

Fig -3.2.5 – Points of action calculation.

- **Step 4: Detecting Fall**

  To detect the fall, we subtract the value of x coordinate of base (point of action) with the x coordinate of the center of mass/gravity of the body and then if it meets certain conditions then it is considered as a fall else the person is considered as standing. The conditions are in the image below.

```
#fall case
fall = int(Point_of_action_X - dot_BODY_X )


#case falling and standing

falling = abs(fall) > 50
standing = abs(fall) < 50
```

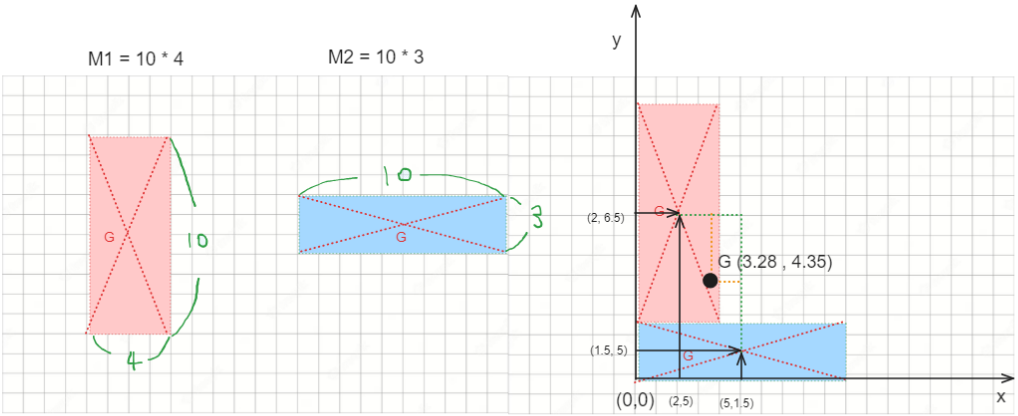
Fig – 3.2.6 – Fall Detection.

3.3 Algorithm Used:

The model uses media pipe by google to detect and find all the landmarks for all the major parts of the body and later using these coordinates finds the base of the human in the feed and the center of mass of that human for upper, lower and full body and then if the center of mass of the body goes out from the base of the human body, then it considers it as a fall.

$$(m1 * x1 + m^2 * x^2)/(m1 + m^2)$$

Fig – 3.3.1 – Formula to calculate rectangle's center of gravity



(M1*X1 + M2*X2) / (M1 + M2) = (40*2 + 30*5) / (40+30) = 3.28

(M1*Y1 + M2*Y2) / (M1 + M2) = (40*6.5 + 30*1.5) / (40+30) = 4.35

Fig – 3.3.2 – Example of center of mass of a rectangle

So, measuring the fall for human body will work like as follow:

Human's Body center of mass * 0.75 > | x-axis center of feet – x-axis center of mass of body |

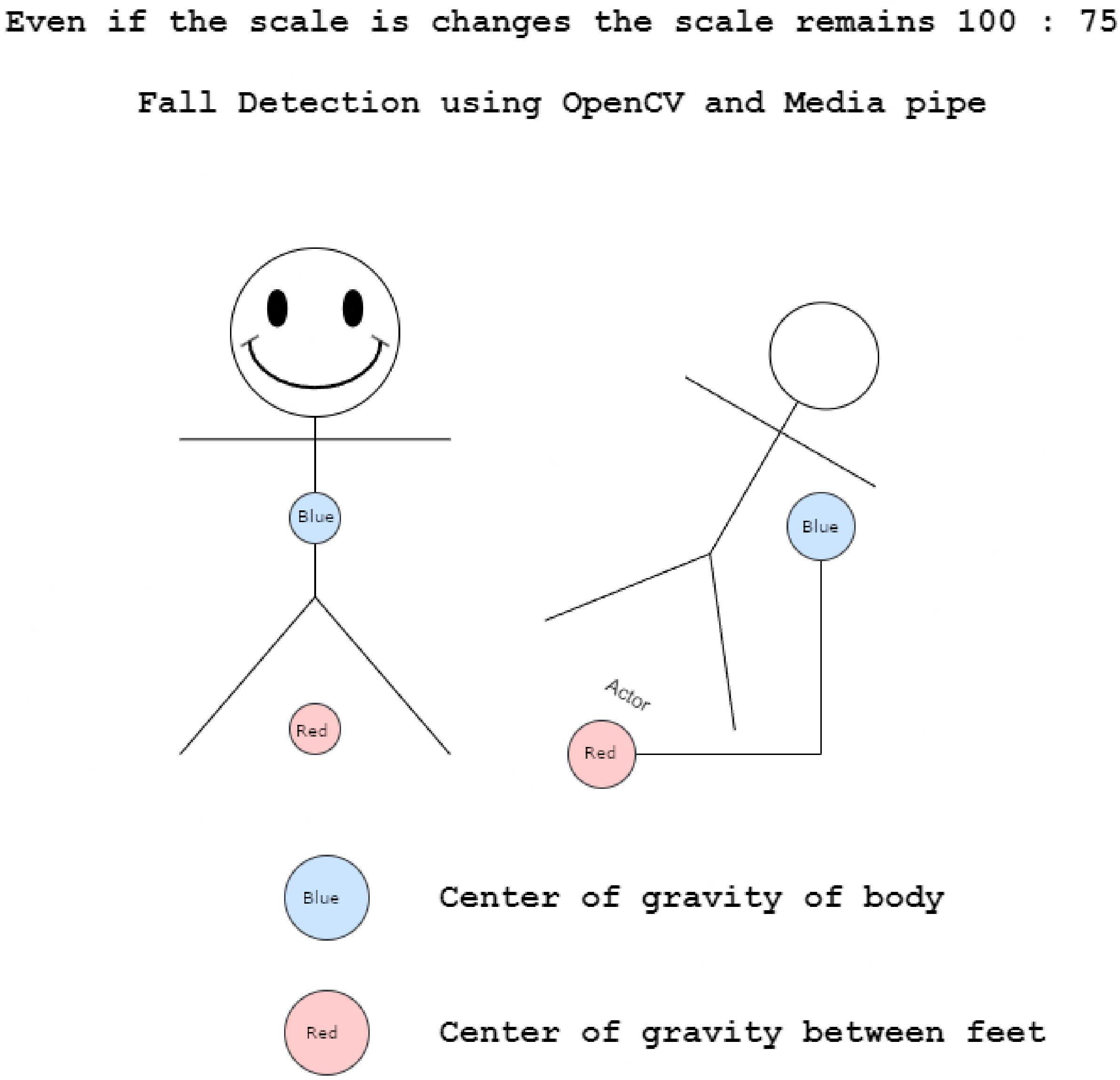


Fig – 3.3.3 – Representation.

3.4 Tools used:

- **Media Pipe** – For pose estimation
- **Open CV** – For using videos/images as input

3.5 Results

- **Standing:**

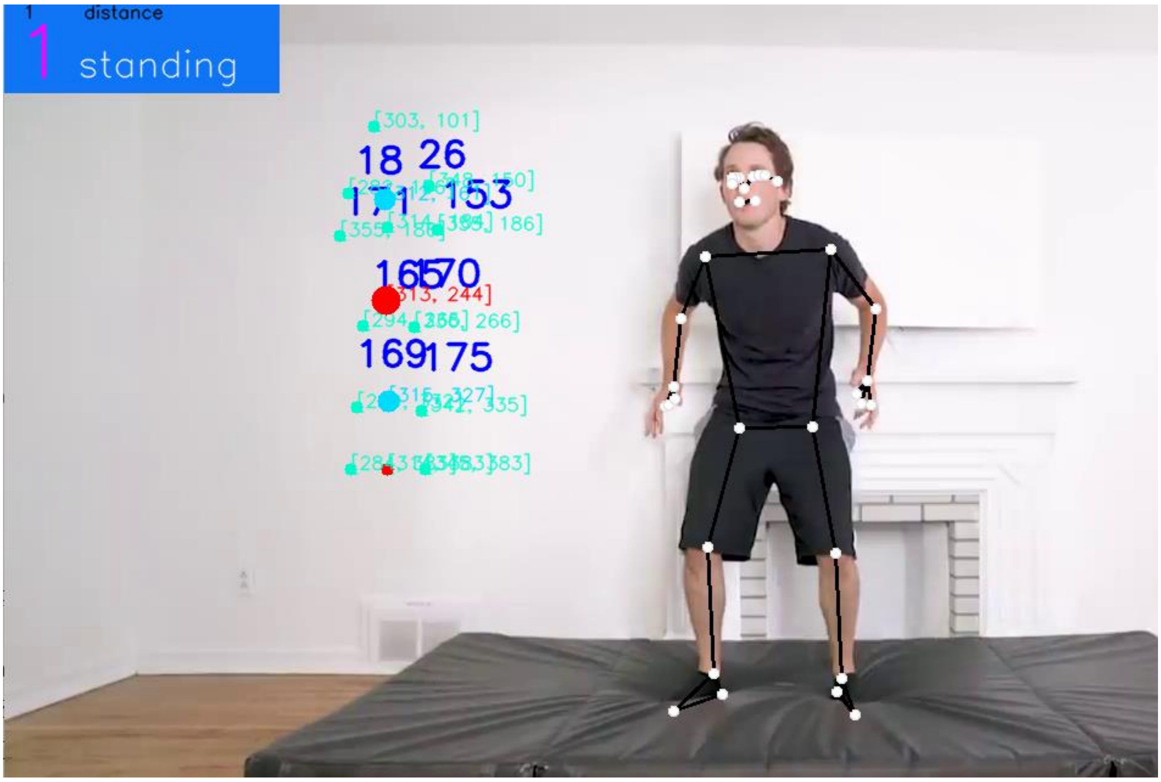


Fig – 3.5.1 – Standing.

Fig – 3.5.2 – Standing as on knees.

- **Fallen:**

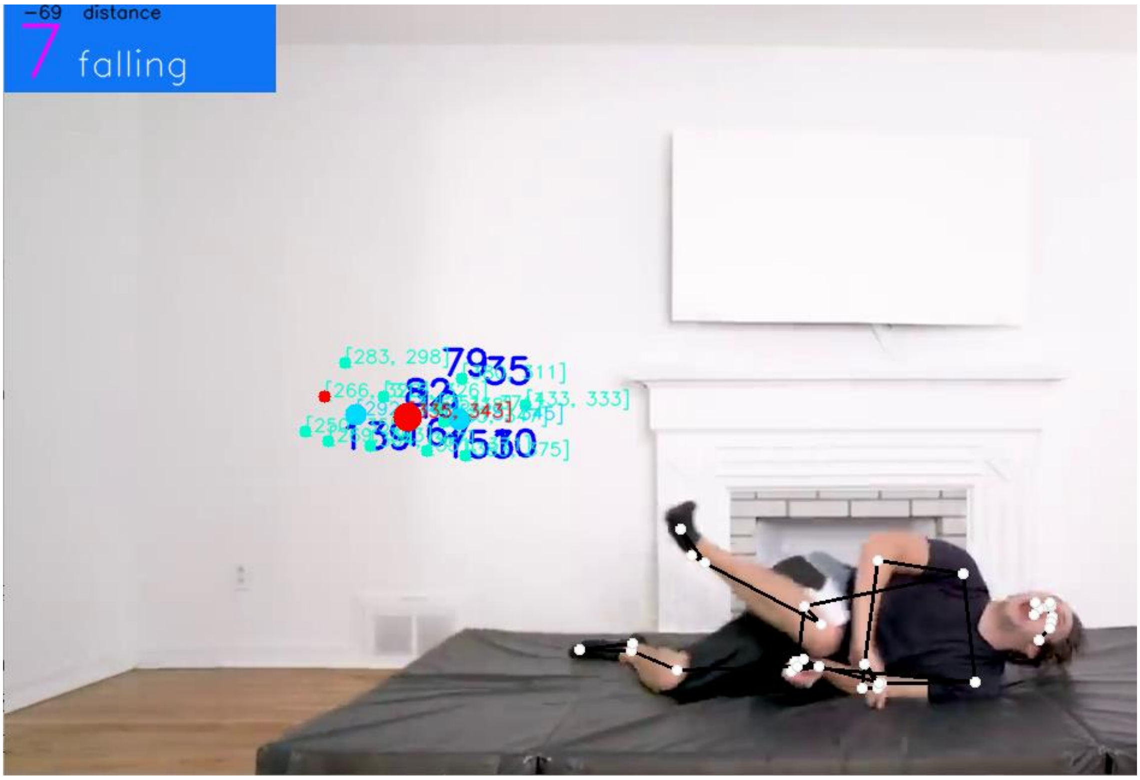


Fig – 3.5.3 – Fallen.

Fig – 3.5.4 – Fallen.

Chapter 4

Future Work Plan

The future work plan of our project are as follows:

Sl. No.	Work Description	Duration in Days
1	We plan to generate alerts to send a notification or any other type of alert whenever a fall is detected. It can be done by using telepot library of python.	**7 – 10 days**
2	We plan to integrate audio features into our fall detection framework. The addition of audio-based analysis aims to provide complementary information to enhance the overall effectiveness of fall detection.	**10 – 14 days**
3	We plan to reduce the false calls as much as possible for better use and making this model as convenient as possible	**5 – 7 days**

Chapter 5

Weekly Task

The report of project work allocated by the supervisor is as follows:

Week No.	Date: From-To	Work Allocated	Work Completed (Yes/No)	Remarks	Guide Signature
1	18th Sept- 24th Sept	Project Work Discussion	Yes		
2	28th Oct- 3rd Oct	Understanding Problem Statement	Yes		
3	10th Oct- 17th Oct	Learning OpenCV	Yes		
4	24th Oct- 30th Oct	Learning and Understanding Mediapipe	Yes		
5	4th Nov- 11th Nov	Implementing Mediapipe to trach human body	Yes		
6	18th Nov- 25th Nov	Implementing Mediapipe to generate coordinates for human body	Yes		
7	22nd Nov- 29th Nov	Calculating angles and detecting falls using algorithm	Yes		
8	30th Nov- 6th Dec	Gather feedback and make adjustments to the work done so far	Yes		

References

[1] V. Viet and D.-J. Choi, "Fall Detection with Smartphone Sensor," in Proc. 3rd Int. Conf. Internet (ICONI), Chonnam National University, 2011, pp. 2011-2011.

[2] V. Vishwakarma, C. Mandal, and S. Sural, "Automatic Detection of Human Fall in Video," School of Information Technology, Indian Institute of Technology, Kharagpur, India.

[3] R. Shu1 and J. Shu2, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," www.nature.com/articles.

[4] L. Hazelhoff, J. Han, and P.H.N. de With, "Video-Based Fall Detection in the Home Using Principal Component Analysis," in Advanced Concepts for Intelligent Vision Systems, J. Blanc-Talon et al. (eds.), Lecture Notes in Computer Science, vol. 5259, Springer, Berlin, Heidelberg, 2008, pp. 27-36.

[5] L. Alhimale, H. Zedan, and A. Al-Bayatti, "The implementation of an intelligent and video-based fall detection system using a neural network," Applied Soft Computing, vol. 18, pp. 59-69, 2014.

[6] Miao Yu, Liyun Gong, and Stefanos Kollias. 2017. Computer vision based fall detection by a convolutional neural network. In Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI '17). Association for Computing Machinery, New York, NY, USA, 416–420.

[7] K. Gunale and P. Mukherji, "Indoor human fall detection system based on automatic vision using computer vision," Journal of Engineering Science and Technology, vol. 13, no. 8, pp. 2587-2605, Aug. 2018.

[8] S. Khawandi, B. Daya, and P. Chauvet, "Implementation of a monitoring system for fall detection in elderly healthcare," Procedia Computer Science, vol. 3, pp. 216-220, 2011.

[9] T.H. Noor, "Human Action Recognition-Based IoT Services for Emergency Response Management," Mach. Learn. Knowl. Extr., vol. 5, pp. 330-345, 2023.

[10] S. Usmani, A. Saboor, M. Haris, M.A. Khan, and H. Park, "Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review," Sensors, vol. 21, pp. 5134, 2021.