Fall And Gesture Detection Using Smart Sentinel

Divya Mahale[1], Sulakshana S. Malwade[2]
[1]*Student, Dept. Of Polytechnic, Dr. Vishwanath Karad's MIT World Peace University*
[2]*Assistant Professor, Dept. Of Polytechnic, Dr. Vishwanath Karad's MIT World Peace University*

***Abstract*—Smart Sentinel is an AI-powered activity detection system based on computer vision technology to monitor homes, workplaces, and community areas. By processing the video feeds, the system identifies incidents such as health emergencies requiring immediate attention. It mainly relies on fall detection and gesture recognition for assessing incident severity.**

Once a fall is detected, the system waits for some time to ensure the fall needs attention before it sends out notifications to authorities. If the subject is okay, then they can gesture thumbs up to let Smart Sentinel know all is okay and no help is needed. This combination of fall detection and gesture detections can dramatically reduce false positives and make the system more dependable.

With enhancements in computer vision technology, it is now feasible to build fall detection and gesture detection models with adequate accuracy. The system is useful in addressing needs in security, health monitoring, and community assistance domains.

***Index Terms*—Artificial Intelligence, Machine Learning, Activity Detection, Fall Detection and Gesture Detection**

I. INTRODUCTION

Technology is the all-time enabler to many exciting solutions in various disciplines. Computer vision technology has been able to successfully mimic human vision and with right post processing algorithms the system can provide dependable solutions that are available 24 x 7.

With the help of AI based computer vision technologies, this system will continuously process video feeds from CCTV cameras, webcams and IoT devices. Key feature of the system is detection of health-related incidents (such as seizures [1], [8] or falls [2], [4], [9], [10]) and sending alerts [14], [15] to appropriate entities.

Smart Sentinel cleverly uses gesture recognition [11], [12], [13], which allows easy assessment of the severity of an incident. As the system detects an incidence, it can play a message to talk to the subject. It will ask the subject if they are alright and if they need any help. It can further request the subject to respond with a "thumbs up" to confirm all is okay or to respond with a "thumbs down" to raise alert. This approach helps minimize false alarms and improves user experience.

A. Smart Sentinel Objectives

New-age AI based Computer Vision based Smart Sentinel solution can monitor the video feed, detect incidents, report, and take necessary actions [3] 24 x 7. Once the system detects an incident, it sends alerts to authorized personnel, and health organizations and make announcements to get the attention of people in the area.

- Smart Sentinel benefits include:
  - Easy implementation
  - Can be used by non-technical users
  - Reliable and dependable solution
  - Works 24 x 7
  - Can be extended to use night vision and/or thermal cameras (with required model training)
- Smart Sentinel can be adopted in areas like:
  - Old age living
  - Daycares
  - Community centers
  - Sensitive office spaces

B. Smart Sentinel Features

- Reliable AI Monitoring: Ensures safety and well-being by monitoring health incidents in autonomous mode. No user intervention or action required.
- Gesture Detection: Detects gestures like a thumbs up (to confirm safety) or thumbs down (to request help) at any times and also after a fall to confirm the incident.
- Real-Time actions: Sends alerts to authorized personnel and makes announcements for guidance.
- 24/7 Availability: Works continuously, ensuring dependable monitoring round the clock.

- Continuous Improvement: Improves with times through continuous training and reinforcement using real data.

C. Existing Solutions

Currently the non-AI solutions depend on staff for detection of health incidents. Human errors due to a brief distraction or a moment of inattention may lead to serious consequences as severe as loss of life if the incident is not detected in time.

Traditionally, CCTV or other video monitoring solutions [5], [6], [7] captured the video feeds. And a dedicated team of security or admin staff were required to manually monitor the video feeds. This approach is obviously very resource intensive and error prone as the personnel may not be able to identify all the incidences. And if CCTVs are not implemented then, the staff needs to patrol the area periodically. The frequency of patrol seriously increases risk of identifying health incidents in timely manner.

The following table highlights the advancements and potential benefits of transitioning from existing methods to an AI-driven system for activity detection and incident monitoring.

Table I: Methodology Comparison

Point of Comparison	Existing Methodology	New Methodology
System Type	Manual and camera-based	AI-powered system
Human Involvement	Requires constant human monitoring or patrolling.	Minimal human involvement, as the system autonomously detects and alerts about incidents.
Coverage and Scalability	Limited by number of staff members and size of area to be monitored.	High coverage and scalable for various environments.
Reliability	Prone to errors due to human fatigue, distractions, or missed observations.	More reliable with AI detecting incidents in real-time.
Technology	Basic use of cameras.	Computer Vision, IoT integration, and gesture recognition.
False Alarm Management	No robust mechanism to reduce false alarms.	Gesture recognition (e.g., thumbs up/down) to confirm or dismiss alerts.
Alert Mechanism	Relies on human observers to recognize and act on incidents.	Automated alerts to designated contacts or systems.
Cost	Expensive as need to have multiple skilled people working in shifts to provide 24 x 7 support.	Will become cost effective as number of users increase making solution affordable and economical.

II. SMART SENTINEL ARCHITECTURE

The following diagram depicts Smart Sentinel architecture. All the individual components seamlessly work together to deliver an efficient and scalable solution.

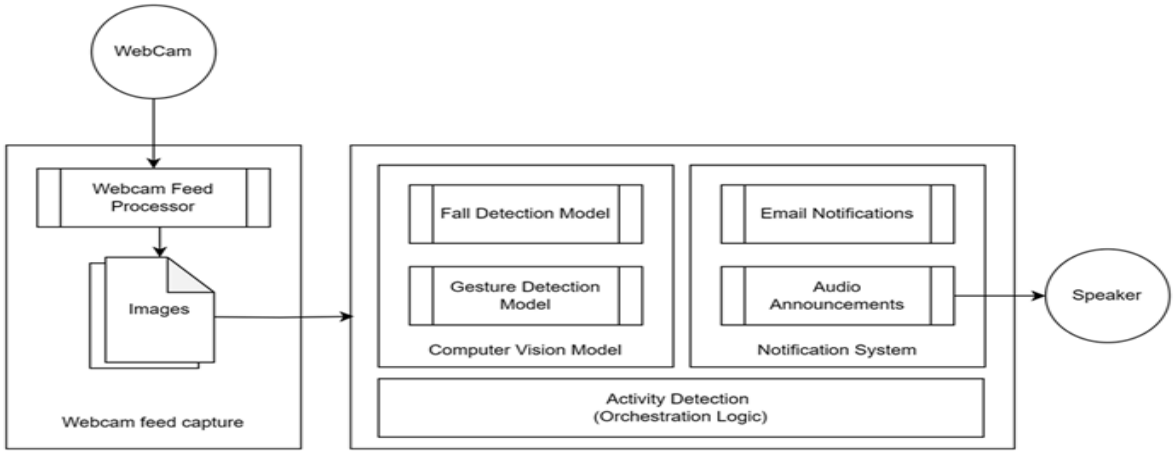

Fig 2. Smart Sentinel architecture

1. Webcam (or IoT or CCTV Camera)

The Webcam is used to capture video to monitor the environment in real-time. It can also be IoT devices or CCTV cameras.

2. Webcam Feed Processor

The Webcam Feed Processor manages the incoming video stream from the webcam. It extracts individual image frames from the video feed. This stage involves pre-processing activities like resizing and filtering to ensure the images are optimized for the next steps.

3. Images

The Images represent the individual frames taken from the webcam's video feed. These images are forwarded to the Computer Vision Model for analysis. Each image captures a moment in the environment using which the models detect falls or gestures.

4. Computer Vision Model

The Computer Vision Model is an integral part of the system, that uses image processing and machine learning to process the images. All images are sent to two different models.

- Fall Detection Model: This model determines if a person has fallen by identifying postures and positions consistent with falls.
- Gesture Detection Model: This model detects thumbs down and thumbs up gestures that allows the system to decide course of action to whether raise alerts or ignore the incidence.

5. Activity Detection (Orchestration Logic)

This is the orchestration logic that manages the whole solution. It takes the images and feeds to computer vision models and then processes their outputs trigger appropriate actions. It processes predictions from both the fall and gesture detection models to decide if a detected activity should prompt an alert.

6. Notification System

The Notification System delivers alerts when fall or gestures are identified. It includes two main functions:

- Email Notifications: Sends email alerts to predefined recipients in the event of fall or thumbs down gesture.
- Audio Announcements: Real-time audio feedback or alerts through a connected speaker to communicate with the subjects or other personnel.

7. Speaker

The Speaker plays the audio announcements generated by the Notification System. It facilitates real-time communication with those in the monitored area.

A. Data Flow Diagram

The following Data Flow Diagrams show how the data shall flow in Smart Sentinel platform.

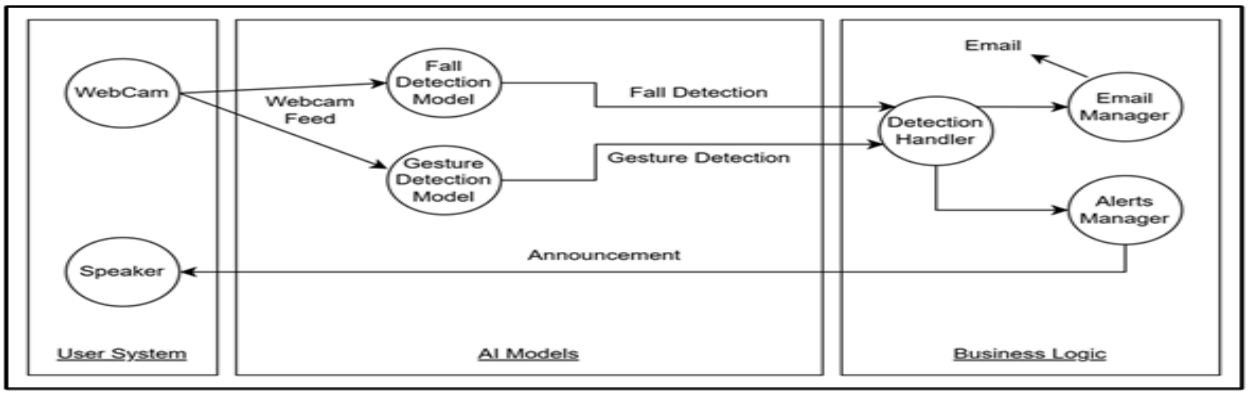

Fig 3: Level 2 DFD

B. Activity Diagram

The following Activity Diagrams show how the various components shall interact in Smart Sentinel platform.

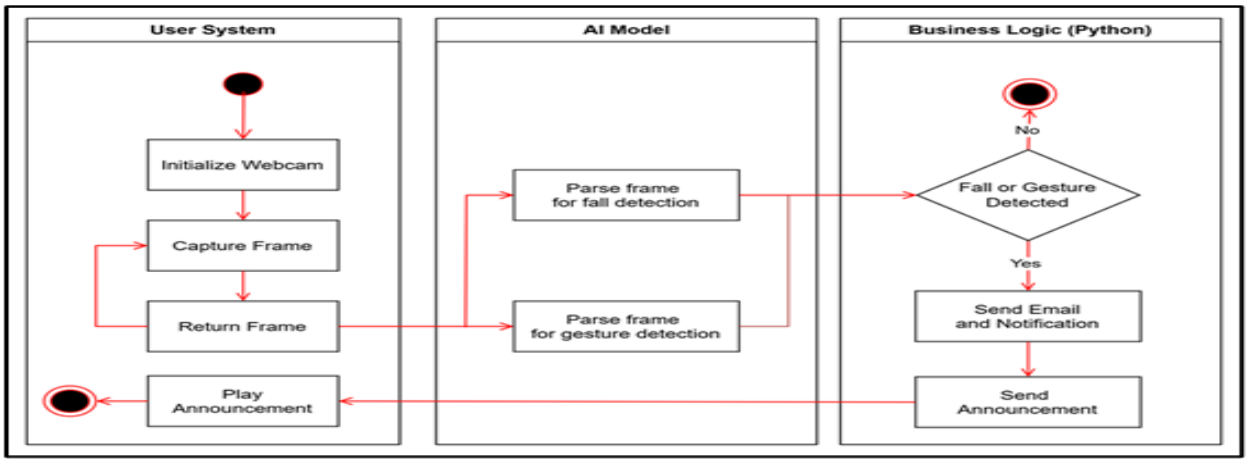

Fig 4: Activity Diagram

C. Algorithm

The following Algorithm outlines the orchestration logic of Smart Sentinel platform.

Table II: Smart Sentinel Algorithm

Step	Module	Action	Remark
1	Main	Initialise flgFall = false flgThumbsUp = false, flgThumbsDown = false	
2	Main	Initialise array elements arrFall[] = 0 and arrThumbsUp[]=0, arrThumbsDown[]=0, i=0, j=0	Array [60] for 2 seconds
3	Main	Initilise fallTime = NULL;	
4	Main	While True	Endless Loop
5	Main	Capture camera feed	

Step	Module	Action	Remark
6	Main	If flgFall == False	Check fall only when fall is not detected skip otherwise
7	Main	Pass camera feed to Fall Detection Model	Currently assume 30 fps
8	Fall Detection	Predict Fall from the image	Returns 0: Nothing or 1: Fall detected
9	Main	If i< 60 then i=i++ else i=0	set array pointer. If at end set to start
10	Main	If fall detected then arrFall[i] = 1 else arrFall[i]=0	Record fall
11	Main	countFall = 0	Reset count
12	Main	For p=0 to 59 countFall+=arrFall[p]	Add array elements to get total true values of fall
13	Main	If countFall >= 48 then flgFall = True	Check if 80% of array elements are 1
14	Main	If flgFall then playsound(), for q=0 to 59 arrThumbsUp[q]=0, arrThumbsDown[q]=0, fallTime = now()	Play recorded message "Are you okay, show thumbs up" And reset gesture flags to track gesture post fall
15	Main	End If	
16	Main	Pass camera feed to Gesture Detection Model	Check for gesture always. With or without fall
17	Gesture Detection	Predict Gesture from the image	Returns 0: Nothing/Other, 1: Thumbs up, 2: Thumbs Down
18	Main	If j< 60 then j=j++ else j=0	set array pointer. If at end set to start
19	Main	If gesture = 1 detected then arrThumbsUp[j] = 1 else arrThumbsUp[j]=0	
20	Main	If gesture = 2 detected then arrThumbsDown[j] = 1 else arrThumbsDown[j]=0	
21	Main	countThumbsUp = 0; countThumbsDown = 0	
22	Main	For q=0 to 59 countThumbsUp+=arrThumbsUp[q]	Add array elements to get total true values of gesture
23	Main	If countThumbsUp >= 48 then flgThumbsUp = True, flgFall = false	Check if 80% of array elements are 1
24	Main	For q=0 to 59 countThumbsDown+=arrThumbsDown[q]	Add array elements to get total true values of gesture
25	Main	If countThumbsDown > = 48 then flgThumbsDown = True, flgFall = True	Check if 80% of array elements are 1
26	Main	If flgThumbsDown OR (fallTime is Not NULL and now() - fallTIme > 2 seconds) then playsound(), sendEmail(), flgThumbsUp = 0, flgThumbsDown = 1	Play recorded message "Help is on the way. Hold on" send email to predefined email address. Play message when thumbs down or no response for 2 sec
27	Main	If flgThumbsUp then playsound()	Play recorded message "All okay then. Take care".

Step	Module	Action	Remark
28	Main	If flgThumbsUp == True OR flgThumbsDown == True	Check if relevant gestures detected
29	Main	flgFall = false; fallTime = NULL; flgThumbsUp = False; flgThumbsDown = False Wait (15 minutes)	Wait once gesture (with or without fall detected. Cool off period
30	Main	End If	
31	Main	End While	Endless Loop

D. AI Models

Fall Detection Model

The following table outlines the salient points of Fall Detection AI Model of Smart Sentinel platform.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14,714,688
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 256)	2,097,408
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 3)	771

Total params: 16,812,867 (64.14 MB)
Trainable params: 2,098,179 (8.00 MB)
Non-trainable params: 14,714,688 (56.13 MB)

Fig 5: Fall Detection Model

Parameter Summary

- Total Params: 16,812,867 (64.14 MB) The total number of parameters in the model.
- Trainable Params: 2,098,179 (8.00 MB) Parameters that the model will update during training (e.g., dense layers and possibly some of VGG16 if fine-tuned).
- Non-Trainable Params: 14,714,688 (56.13 MB) Parameters that remain frozen, likely those from the pre-trained VGG16 model.

Functionality

- Input Features: Processes input data, likely from images or sensor data, with the VGG16 model extracting key features.
- VGG16: Pre-trained model acts as a feature extractor to capture complex patterns related to falls or incidents.
- Dense Layers:
  - The first dense layer learns from the flattened VGG16 output, reducing it to 256 features.
  - Dropout helps regularize the model to prevent overfitting.
  - The final dense layer outputs 3 categories (e.g., fall, standing or walking, sitting) based on learned features.
- Output: Classifies the input data into one of three categories (likely "fall", "standing or walking", and "sitting" scenarios) using the final dense layer.

Gesture Detection Model

The following table outlines the salient points of Gesture Detection Model of Smart Sentinel platform.

Model: "sequential"

Layer (type)	Output Shape	Param #
dropout (Dropout)	(None, 42)	0
dense (Dense)	(None, 20)	860
dropout_1 (Dropout)	(None, 20)	0
dense_1 (Dense)	(None, 10)	210
dense_2 (Dense)	(None, 3)	33

Total params: 1,103 (4.31 KB)
Trainable params: 1,103 (4.31 KB)
Non-trainable params: 0 (0.00 B)

Fig 6: Gesture Detection Model

Parameter Summary

- Total Params: 1,103 (4.31 KB).
- Trainable Params: 1,103 (4.31 KB).
- Non-Trainable Params: 0 (0.00 KB).

Functionality

- Input Features: Accepts input data with 42 features.
- Dropout: Two dropout layers prevent overfitting by reducing reliance on specific neurons during training.
- Dense Layers:
  - Extract key features using two hidden dense layers with 20 and 10 neurons, respectively.
  - The final dense layer predicts the gesture class with 3 output neurons.

- Output: Classifies input data into one of three gesture categories using the final dense layer.

III. SMART SENTINEL RESULTS

1. Smart Sentinel can successfully detect fall and/or gestures as trained.
2. The logic of seeking confirmation post fall resulted in reducing false positive significantly as it allowed person to tell Smart Sentinel to ignore causal falls.
3. Smart Sentinel was able to interact with people by playing prerecorded audio. It allowed Smart Sentinel to check on people or inform them that help is coming.
4. Smart Sentinel was able to successfully seek help by informing authorities by sending SOS email.

The following figure shows how different incidences are recognised by Smart Sentinel platform.

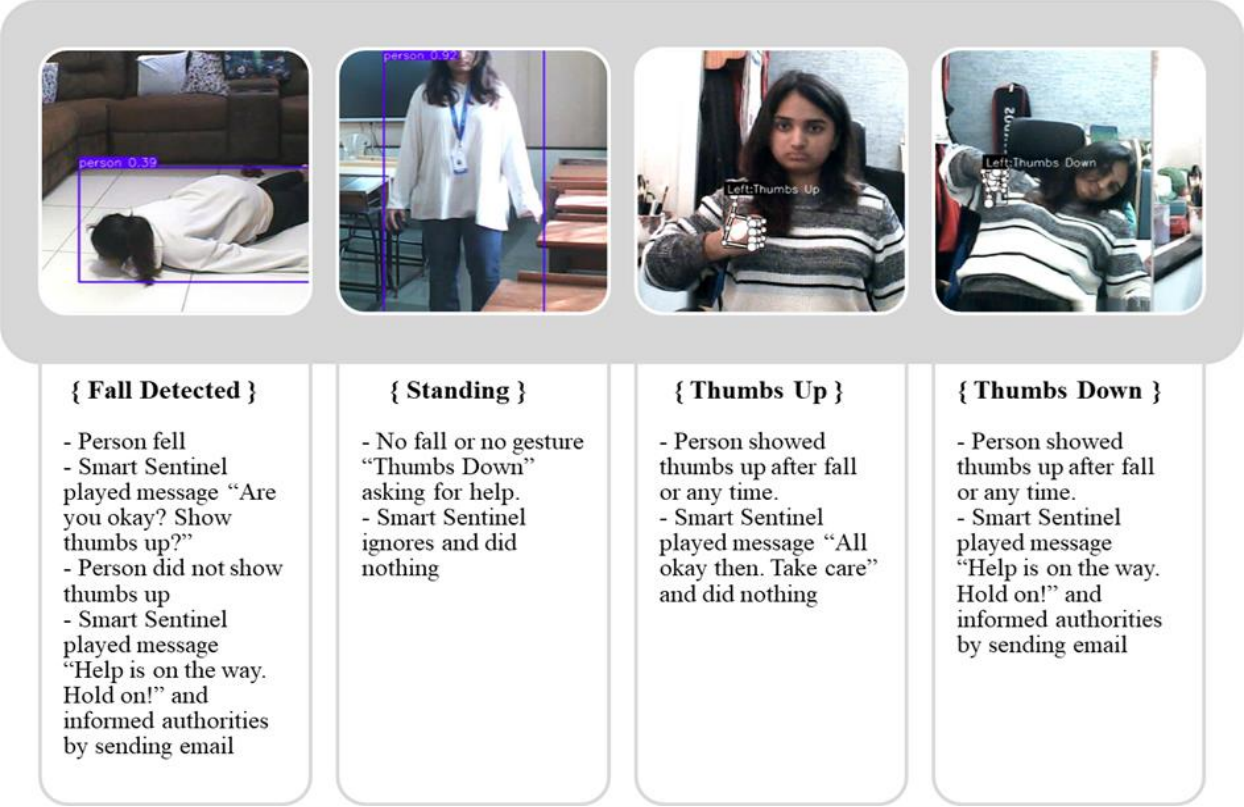

Fig 7: Smart Sentinel Performance

IV. CONCLUSION

The Smart Sentinel platform is a significant advancement in ensuring the safety and well-being of people using its cutting-edge Activity Detection AI models. This system monitors incidents in real time and immediately issues alerts. It allows users to interact with the system by means of intuitive and simple hand gestures. Smart Sentinel greatly minimizes the chances of oversight and human error by providing a reliable technology solution. Its simple implementation and accessibility to individuals without technical background makes it a versatile and dependable system. It can be easily used in a range of settings, from elderly and childcare to community areas and sensitive work environments. This solution shows how new age AI can genuinely enhance our lives without compromising on safety and well-being.

V. FUTURE SCOPE

The Smart Sentinel platform lays a solid groundwork for incorporating advanced technologies that enhance

safety, well-being, and proactive management of incidents across various settings.

A. *Improving AI Models for Activity Detection*: Future updates to the Smart Sentinel could integrate more advanced AI and machine learning models, enabling the identification of a wider spectrum of incidents.

B. *IoT Device Integration*: Incorporating IoT-enabled sensors - such as wearables, motion detectors, and environmental monitors. This would enable the system to monitor heart rate, temperature, and air quality.

C. *Multi-Lingual User Interfaces*: Boost accessibility, the platform could adopt multilingual interfaces, allowing users from various regions and cultural backgrounds to interact seamlessly with the system.

D. *Predictive and Preventive Analytics*: Utilizing AI to forecast potential incidents based on behavioral trends and environmental conditions could pave the way for preventive strategies.

E. *Adoption in Diverse Industries and Scenarios*: Broadening the use of this platform in manufacturing, healthcare, and public infrastructure, can help tackle wider safety issues.

VI. ACKNOWLEDGMENT

Our sincere thanks to the Department of Polytechnic, Dr. Vishwanath Karad's MIT World Peace University, for their support and encouragement without which this paper would not have been possible.

REFERENCES

[1] Jing Tian, Weiyu Yu, Jinquan Chen, Junke Lin, Mingfeng Wen, Yingxin Li, Jianxin Zhong, Keqiang Chen, Xuchu Feng (2020), Literature Survey. In Automated Analysis of Seizure Behavior in Video: Methods and Challenges (pp. 1-3), China.

[2] Hiroaki Kingetsu, Takeshi Konno, Shuji Awai, D. Fukuda, T. Sonoda (2019), Introduction. In Video-based Fall Risk Detection System for the Elderly (pp. 1-3), Japan.

[3] Sanjay Adhikesaven (2022), Problem and Motivation. In An IndustrialWorkplace Alerting and Monitoring Platform to Prevent Workplace Injury and Accidents (pp. 1-3), California.

[4] Alex D. Edgcomb, F. Vahid (2012), Introduction and Fall Detection on Raw Video. In Automated fall detection on privacy-enhanced video (pp. 1-3), California.

[5] K. Ullah, Imran Ahmed, Misbah Ahmad, I. Khan (2019), Related work, dataset and results. In Comparison of Person Tracking Algorithms Using Overhead View Implemented in OpenCV (pp. 1-4), Pakistan.

[6] Ibrar Ahmed, Bharat Bhushan Naib (2023), Related Work, Proposed Work and Results & Discussion. In Object Motion Tracking and Detection in Surveillance Videos using Resnet Architecture (pp. 1-4), India.

[7] S. Aarthi, S. Juliet (2021), Literature Review. In A Comprehensive Study on Human Activity Recognition (pp. 2-4), India

[8] Rizzah Grace Llanes, Rosula S. J. Reyes (2022), Methodology. In Stress Detection in Video Feed: Utilizing Facial Action Units as Indicators in Various Machine Learning Algorithms (pp. 1-3), Philippines.

[9] Srijita Ghatak, Rishita Mitra, Washef Ahmed, Kunal Chanda (2023), Contributions and Results. In A Simple Fall Detection Scheme for Early Detection of Falls in Elderly People (pp. 1-5), Kolkata

[10] Dara Ros, Rui Dai (2023), Introduction and Performance Evaluation. In A Flexible Fall Detection Framework Based on Object Detection and Motion Analysis (pp. 1-6), Ohio.

[11] Priyal Raj, Sakshi Pandey, Yuvraj Singh, Monu Singh (2023), Introduction. In Hand Sign & Gesture Recognition System (pp. 1-3), India

[12] Narjes Heidari, Javid Norouzi, M. Helfroush, H. Danyali (2024), Introduction. In Dynamic Hand Gesture Recognition with 2DCNN-LSTM and Improved Keyframe Extraction (pp. 1-3), Iran

[13] Ediyanus Pabendon, H. Nugroho, A. Suheryadi, Prasti Eko Yunanto (2017), Introduction. In Hand Gesture Recognition System Under Complex Background Using Spatio Temporal Analysis(pp. 1-2), Indonesia

[14] Trevor Lanigan, Stephanie Anderson, Alexus Cottonham, Gregory Marzolf (2024), Introduction and Background and Motivation. In Expanding the Emergency Alert System Enterprise with Space-Based Emergency Alerts(pp. 1-4), USA

[15] Hans Ziegler, Michael Klafft (2013), Introduction. In Privacy-aware and distributed system design for personalised alerting (pp. 1-3), Mexico