Computer Vision-Based Self-Inflicted Violence Detection in High-Rise Environments using Deep Learning

Mallikharjuna Rao K[1, *], Deepesh Agrawal[2], Shikhar Reyya[3], and Priykrit Varma[4]

[1, *]Assistant Professor, Data Science and Artificial Intelligence
rao.mkrao@gmail.com, mallikharjuna@iiitnr.edu.in
https://orcid.org/my-orcid?orcid=0000-0002-6096-3963
[2,3,4]Computer Science and Engineering
[1,2,3,4]International Institute of Information Technology Naya Raipur, India
[2]deepesh21100@iiitnr.edu.in, [3]shikhar21100@iiitnr.edu.in
[4]priykrit21100@iiitnr.edu.in

***Corresponding Author**

Abstract - This article presents a system that utilizes computer vision techniques to identify suspicious behavior in high-risk areas. The technology employs cameras to detect individuals engaged in any suspicious activity or situated in hazardous locations. The system then analyzes the video using deep learning and computer vision techniques to identify those at risk. Upon detecting suspicious behavior or a self-inflicted violence attempt, the system promptly notifies the appropriate authorities, such as emergency medical services or law enforcement, enabling them to take preventive action. Videos are sequences of frames. We use a deep sort algorithm to extract the appearance and motion and connect different frames for real-time tracking with excellent performance and accuracy, even in crowded environments the information collected through video analysis can contribute to a better understanding of the underlying factors leading to such incidents.

Keywords: self-inflicted violence, Computer vision, deep learning, Real-time camera surveillance, object detection, object tracking

1. Introduction

Globally, suicide is a significant issue, causing over 700,000 deaths each year [1]. One common method of suicide is jumping from high places like bridges or tall buildings. It not only puts the person attempting it at risk but also causes mental distress for those who witness it and the emergency responders. Traditional methods like physical barriers and increased monitoring by security guards may or may not effectively prevent self-inflicted violence in these locations. To resolve this issue, we need an innovative and effective system to quickly identify individuals who might be showing signs of suicidal behavior and alert the nearby authorities for help. We can potentially spot this behavior by analyzing real-time video using the camera. Our goal in this study is to use these methods to create a system that detects signs of suspicious human behavior near bridges and other elevated areas.

One way of committing suicide is by jumping from a height, which is a problem for global public health. Victims first suffer from mental distress due to some social life issues like being bullied by friends or having low self-esteem, etc. After this, the victim decides to commit suicide. Figure 1 below shows the accident statistics in highly populated areas.

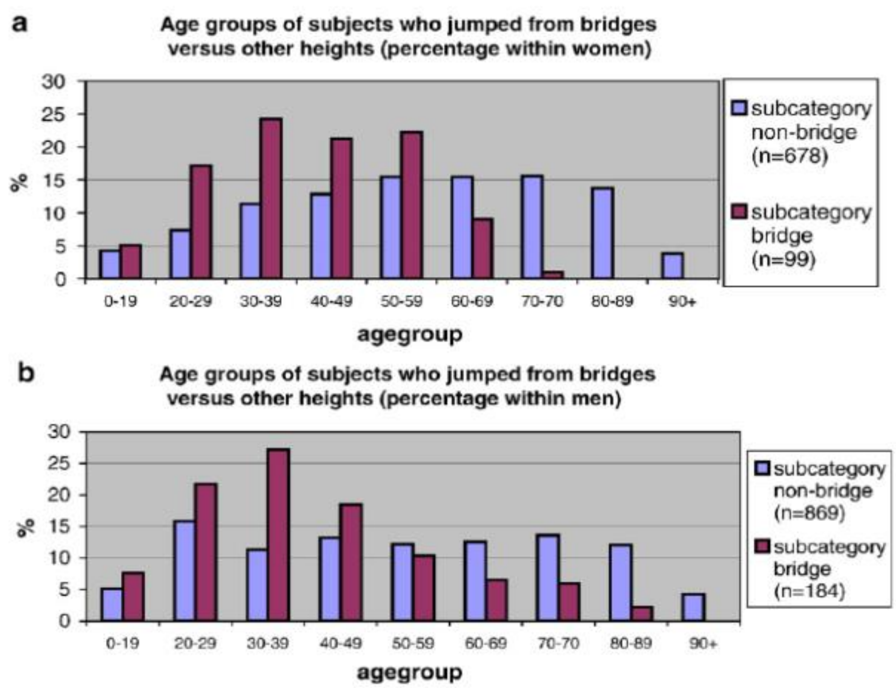

Fig. 1. Age groups (%) of women (panel a) and men (panel b) who jumped from bridges versus other heights.[1]

Creating physical barriers and increasing surveillance are commonly used methods to prevent self-inflicted violence. However, they are not foolproof and may not always effectively stop such attempts. Therefore, there is a need for innovative and effective systems to detect self-inflicted violence in real-time, identify individuals showing suicidal tendencies, and alert the appropriate authorities to intervene promptly. In this paper, we aim to create a computer vision-based system with the help of deep learning with the following objectives:

- To create a system for real-time video surveillance that can identify Self-Inflicted Violence behavior near high-rise places.
- To analyze people's activities and movements to spot those at risk.
- To provide a method for alerting and contacting authorities nearby, like emergency medical services or cops, to stop probable suicide attempts.
- To make suggestions for further study and create self-inflicted violence detection systems in high-risk regions, the primary goal of the suggested system is to save lives.

2. Literature Review

2.1 Self-inflicted Violence Prevention Strategies [2]

- Cameras are essential to monitor individuals' suspicious activity and movements at hazardous locations. An automated machine learning system monitors and analyzes live video recordings and alerts if an individual is in danger.
- If suspicious activity is detected, employ a rescue team, including mental health professionals and officers, to handle the situation.
- Deployment of an automated alert-generating self-inflicted violence system. These alerts, containing details about the location and incident where this incident happened, are then transmitted to emergency medical services for swift response and intervention.

2.2. Self-Inflicted Violence Detection Systems [3]

The system aims to detect people's self-harming activities and notify nearby authorities. Various technologies exist to detect self-inflicted violence activities on bridges. The first approach is developing real-time camera surveillance and identifying suspicious behavior. This video's image frame can be fed to a machine-learning algorithm to detect suspicious behavior in real time. After detection, the system can alert the nearby officials' people so they can check in the specified area.

2.3. Computer Vision Techniques

We can locate people in hazardous places using object detection algorithms like YOLO v5 [4-5-16-20]. By detecting the movement and activities of individuals, using this object detection algorithm system can identify various actions and patterns that may lead to suicidal behavior. Additionally, a face recognition algorithm is used to identify the person who is at risk of self-inflicted violence. The pose estimation algorithm [6-7] can identify the position and activities considered one criterion to detect a person's behavior. Machine learning algorithms can be trained on typical behavioral patterns and then used to detect abnormalities that can be signs of an impending Self-Inflicted Violence attempt. These are only a few examples of computer vision methods that might be applied in a project to detect Self-inflicted violence [8-18-19].

2.4 Deep Learning Algorithms

To find patterns and behavior that may be a sign of suicidal behavior using deep learning. These are some of the deep learning algorithms that would be applied in our system:

- Convolutional Neural Networks (CNNs) [9-18] will help detect persons and their position, such as pacing or standing on the brink of a bridge, in the image frame provided by real-time video.
- Deepsort is an object-tracking algorithm that can be used for movement tracking in behavior analysis, ultimately leading to Self-Inflicted Violence detection.
- Long Short-Term Memory Networks (LSTMs) can evaluate data sequences, such as movement patterns or sign measures, to spot alterations that point to suspicious behavior. Non-Maxima Suppression (NMS) can remove minor probability instances of detected objects [9]

The examples provided are merely instances of deep learning algorithms that could be utilized in a system. Deepsort is an algorithm that combines deep learning with classical tracking methods to achieve high accuracy in tracking [9]. The algorithm we choose depends on its nature and complexity, as well as available resources for training.

2.5 Research Gaps

1. Checking the system's effectiveness in different environmental conditions such as varying lighting, weather, and visibility.
2. Assess the real-time response efficiency of the system in notifying authorities. Check the time lag between detection, notification, and the arrival of an emergency message, aiming for the quickest possible response.

2.6 Relevant Case Studies

In 2019, a research project by the Korea Advanced Institute of Science and Technology (KAIST) [10] for preventing self-inflicted violence. They were using security cameras for real-time monitoring. Researchers have developed a deep learning-based method to detect suspicious activities in high-raised areas. This system can identify any suspicious activities, such as someone leaning over a bridge or pacing back and forth, which are suggestive of suicidal behavior.

When this system detects any person in danger, it notifies nearby authorities so that professionals can come to save the person. Also, speakers can be used in hazardous areas so that victims think twice before performing unnecessary activities during research on the South Korean bridge. The KAIST identified those who are at risk with high accuracy. This method sounds helpful to decrease the rate of suicide as compared to the current scenario of 700000 per year [10]. The use of deep learning algorithms and computer vision is highlighted in this research paper.

3. Methodology

3.1 Data used

We first identify the persons who are in danger in high-raised areas and having suicidal thoughts or suspicious behavior through real-time cameras, which will then be the data fed frame by frame to the machine learning algorithm and find a specific suspicious pattern or behavior suggestive of a potentially dangerous situation. We detect if someone is at risk at significant heights and close to harmful places. To determine whether someone is considering self-harm, we employ computer vision algorithms to analyze several components of the video footage, such as people's motions and facial expressions. The architectural view of the proposed model is shown in Figure 2. For instance, we can train computer algorithms to detect abrupt stops, pacing back and forth, or unusual movements that might be indicators of suicidal thoughts. This research focuses on using technology to spot these warning signs.

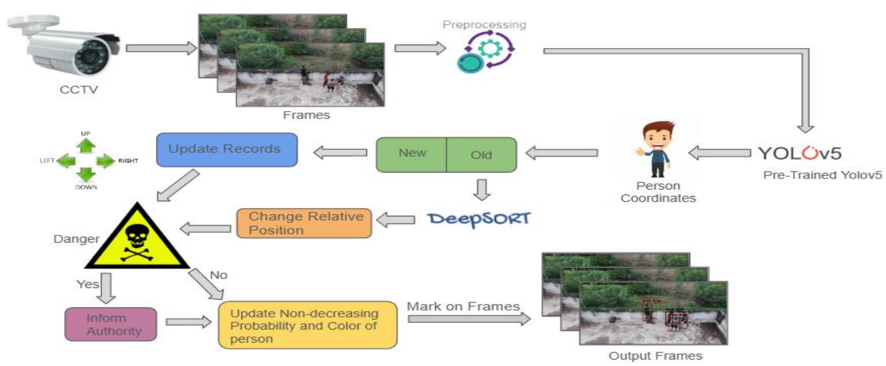

Fig. 2 The architectural view of the proposed model.

3.2 Flow of frame feeds

The feed will be taken from the camera and then preprocessed (resize and convert BGR image to RGB) [11]. This feed will then be forwarded to the pre-trained YOLO v5 model, which will return the number and coordinates of persons. The path indicating back to capture feed will be executed if no person is detected on the feed. If a person is detected, then the algorithm will identify whether the person exists in records or if a new person has entered the scene. DeepSORT will give it a new identifier and start to monitor movements. Then, judgment will be passed on whether a person is in danger. If a person happens to be in a dangerous location, then those concerned with information to authorities will be executed. Suppose a person happens to be in a dangerous location. In that case, direct probabilities will be updated, and frames will be marked with information related to the person, and the algorithm will return these frames and continue with the next frame. The pictorial representation and corresponding working procedure are shown in Figure 3, and the algorithm is below.

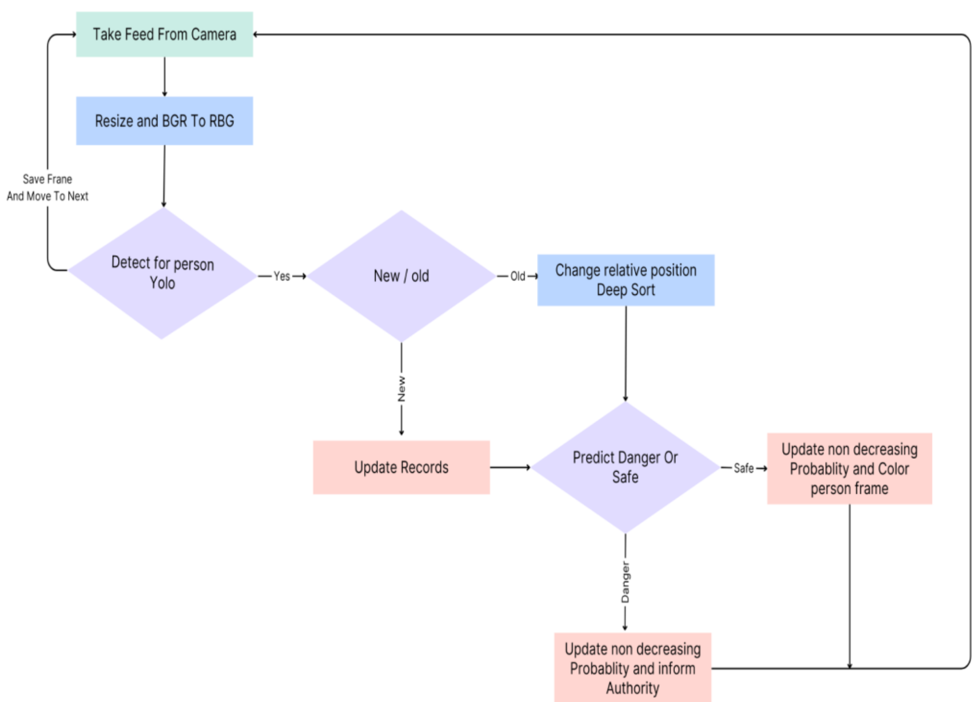

Fig. 3 Flow of frame feeds

Algorithm: Detecting self-inflicted violence at a high-raised area

Input: image frames $f_1, f_2 \ldots\ldots f_i$

Load YOLO v5 model $\Rightarrow Y_{model}$ and DeepSORT object tracker $\Rightarrow D_{model}$
Define hazardous locations (boundary regions) as coordinates (x1, y1, x2, y2)
Image frames $\Rightarrow$ {f1,f2,f3,f4…fn}
Continuously capture frames {f1,f2,f3,f4} from the camera feed in a loop
$RGB_{x,y,z} \Leftarrow$ RGB format frame
while(true)
{
  if f_i != $RGB_{x,y,z}$
    $f_i \Rightarrow RGB_{x,y,z}$
  if $Y_{model} \Leftarrow f_i$. does not have a person
    i = i+1
  else
    $f_i = p_1, p_2, \ldots p_i$ // person $p_1, p_2, \ldots p_i$ detected are stored in frame f_i .
    Check if the person p_i is a new or previously tracked person using D_{model}
      if $D_{model}(p_i)$ = new person
        $p_i \Leftarrow$ assign id_{new}
      else
        calculate the relative change (Δ) in position using D_{model}.
  Predict whether the person is in a safe or dangerous location based on the Δ in position.
  if Δ = safe location
    Update a non-decreasing probability value. (Φ)
    adjust_color(f_i)
  else
    Update Φ
    return "person is in danger"
}

3.3 Video Processing Techniques

The analysis of video data from security cameras using video processing techniques can be utilized to spot people who might be in danger of Self-Inflicted Violence [12]. To prevent Self-Inflicted Violence, the following typical video processing methods are employed.

Object detection is a method for locating particular individuals or objects in a video feed. Object detection can be utilized in Self-Inflicted Violence prevention initiatives to spot individuals perched precariously on a bridge or exhibiting other risky behaviors, detect coordinates of body parts in a given frame, and conclude.

3.4 Alert and Intervention Mechanisms

Usually, alert and intervention procedures are implemented to stop self-harm. As soon as it becomes clear that someone is in danger of injuring themselves, the appropriate treatments must be initiated. Notification systems are a critical component of this process. If someone is suspected of being suicidal, these systems can be put in place to promptly notify the relevant authorities, support groups, or caretakers.

- Emergency response teams could be asked to help people who might be socially inclined right away. Trained law enforcement officers, mental health specialists,
- Operators who see concerning indicators can manually trigger these alerts, or prediction algorithms can do it automatically.

3.5 Deep-sort Algorithm [13]

Videos are sequences of frames. We use a deep sort algorithm to extract the appearance and motion and connect different frames for real-time tracking with excellent performance and accuracy, even in crowded environments. Many industries like Tesla use this technique for building self-driving cars, etc.; the setup of parameters is crucial and is contingent upon the separation between the ROI and the camera.

Parameters used during the initialization of the Deep-Sort object tracker:

- *max_age:* The maximum number of frames a track can miss before it is deleted. In other words, this parameter determines the maximum age of a track (in frames) before it is considered inactive and removed from the tracker. For our research, it is set to 30 and taking 30 fps, typically indicating 1 sec.
- *n_init:* The minimum number of detections required to initiate a track. If a track does not have enough detections in its history to meet this threshold, it will not be initialized. For our research, a minimum of 5 detections is required to initiate a track.
- *nms_max_overlap:* The maximum allowed overlap between two detections to perform non-maximum suppression (NMS). This parameter helps to remove redundant detections and improve the quality of the tracking results. For our research, it was 0.5.
- *max_cosine_distance:* The maximum cosine distance between two embeddings to consider them part of the same track. This parameter helps determine whether two detections belong to the same object. In our research, it is taken as 0.12.
- *nn_budget:* The maximum number of embeddings that can be stored in memory at any time. The oldest will be discarded if the number of embeddings is within this budget. Used Default values.

- *override_track_class:* Allows the user to override the default track class used by the tracker. Used Default value.
- *embedder:* The name of the feature extraction network to use. In our research, the "mobilenet" architecture is used.
- *half:* Using half-precision floating point arithmetic for the feature extraction network helps speed up the computation and reduce memory usage. Used Default value.
- *bgr:* Using BGR image format for input to the feature extraction network. Used Default value.
- *embedder_gpu:* Whether to use GPU acceleration for the feature extraction network. Used Default value.
- *embedder_model_name:* The name of the pre-trained feature extraction network to use. If set to None, the default network for the chosen architecture will be used. Used Default value.
- *embedder_wts:* The path to the weights file for the pre-trained feature extraction network. Used Default value.
- *polygon:* Whether to use a polygon instead of a bounding box. The tracker will use a polygon instead of a bounding box for detections if a polygon is enabled. Used Default value False.
- *today:* The date to use as the reference date for the tracker. The current date will be used if set to None (as in our case).

The values of the above parameters will depend on the type and resolution of the camera, luminous condition, population density, the distance between the camera and the scene, and any other specific condition of a particular geography. Values to the above parameters can be referenced from our research and further tuned according to the needs.

3.6 Record updating

The system is designed to track multiple individuals, each with a unique identifier. It records each person's movement by tracking changes in their bounding boxes' horizontal and vertical coordinates (Fig. 4). When either coordinate changes, the system updates the movement direction. It counts the number of times the person changes direction. Additionally, it records how long a person has remained in frame sequences and safety/danger, i.e., each person's current situation. For example, suppose a person is moving to the right and suddenly changes their direction to the left. In that case, this will trigger an update of the record corresponding to that person. The number of direction changes will be incremented for other directions such as Left to Right, Up to Down, Down to Up, or a combination of these movements.

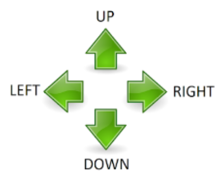

Fig 4 General 4-Direction of a person in motion

3.7 Display of Message

The program first verifies if the current individual being monitored has a unique identifier assigned by the tracker. Once the ID is identified, the program evaluates the person's movements to determine whether they are at risk. If the person's movements are suspicious, the program adds a message to the video frame displaying their ID and their time in danger. Conversely, suppose the person's movements are safe. In that case, the program generates a message indicating their ID, updates their status in the log to "safe," and removes them from the list of tracked IDs. The log keeps track of the status of each person, which can either be "safe" or "dangerous." The log is updated whenever a person's status changes from "danger" to "safe," the log is updated.

3.8. Curse of NMS

It is possible to generate different bounding boxes for the same person when the image is passed through the same network multiple times. This may happen due to various reasons, such as changes in the person's pose, lighting conditions, occlusions, and alterations in the background. If the lighting changes, NMS can suppress the original coordinates of bounding boxes and assign new ones. This would imply that the person is in motion, which is incorrect, as they might still be stationary. To address this issue, we introduced buffer regions around the points, indicating that a person would only be considered in motion when crossing a certain threshold. This approach dramatically reduces the risk of false motion detection and keeps the probability in check.

For the same reason, the Region of Interest (ROI) or the boundary of a highly elevated area had to be predefined. If trained through a neural network to identify the hazardous region, NMS can create varying bounding boxes, leading to ambiguous results.

3.9 Updating non-decreasing Probability and Color on the frame

The probability is calculated by distributing 85 to horizontal direction changes (a maximum of 10 direction changes), 85 to vertical direction changes (a maximum of 10 direction changes), and 40 to time spent (a maximum of 300 seconds spent in the frame) in the range of 0-255 (reserving 40 for future or additional geographic requirements).

Afterward, the effective color of bounding boxes is calculated, with higher probability indicating a redder color, and non-decreasing probability is assigned for individual unique IDs. The distribution of the pixel range, i.e., 255, can be adjusted differently for various purposes and based on different studies of behavioral prediction. The values used here are not derived from any scientific study but are instead utilized to illustrate the detection aspect of our research.

4. Results and Discussion

The system uses two models, YOLO v5, for object detection and deep sort for object tracking, which require much computation. Further improvements can be made in our work to gain optimization and better results with high accuracy. System testing is necessary to review the system after installation and configuration to ensure it operates as the system is intended and fulfils its objective. Setting up parameters is essential based on the distance between the camera and ROI. The following are some of the experimental results drawn from our experiments. In Fig. 5, the persons are localized, and their probability score is with respect to the hazardous boundary. Also, we localized the boundary from which self-inflicted violence may occur. The person's movement probability in Fig. 5 at the initial stages is around 0.04, which is only due to his movement and increases with time and movements. Fig. 6 is a scene processed from a video feed indicating the probability and color change with respect to Fig. 5. It shows when a person performs suspicious behavior in a high-risk area for a specific time. When the person approaches the boundary of the hazardous location, the probability starts to increase and becomes 0.39, as shown in Fig. 6 when he reaches a high distance from the boundary. Fig. 7 shows a scene from the processed video feed indicating probability and color change with respect to Fig. 6 and Fig. 5 when a person stands on the boundary of a hazardous location with notification of the person's ID and amount of time spent in that place. The probability rapidly increases when a person climbs the boundary, and it can be seen that now the probability is 0.71 in Fig. 7 when the person climbs the boundary wall. Fig. 8 shows a scene from the processed video feed indicating probability, time, and color change with respect to Fig. 7, Fig. 6, and Fig. 5 when the person continues to stand on top of the boundary wall, and notification concerning changed information. This probability will keep increasing with time to a certain threshold if the person remains in the frame, as in the case of Fig. 8. FPS, as shown in Fig. 5 to Fig. 8, represents the frames per second for processing the image.

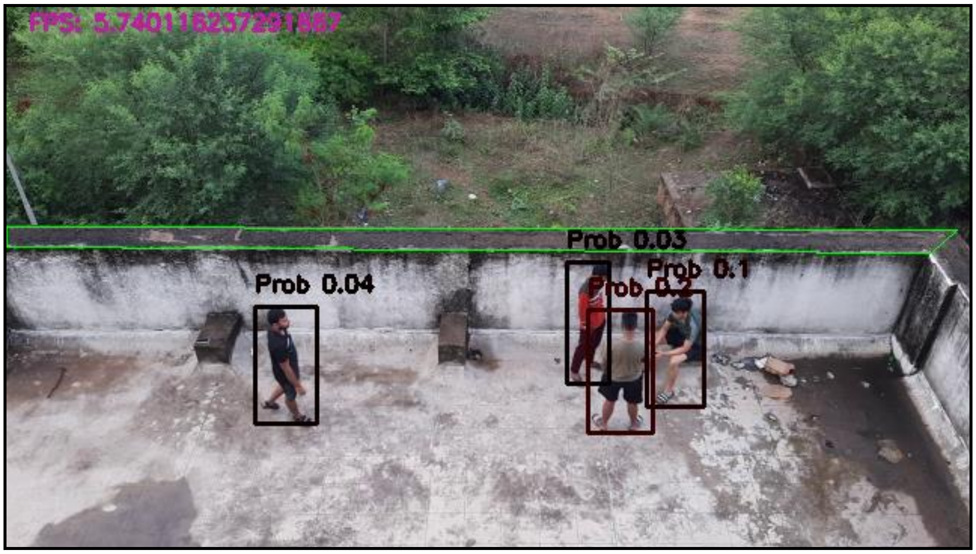

Fig. 5 Scene from processed video feed showing and localizing the person and their probability score w.r.t. Hazardous boundary.

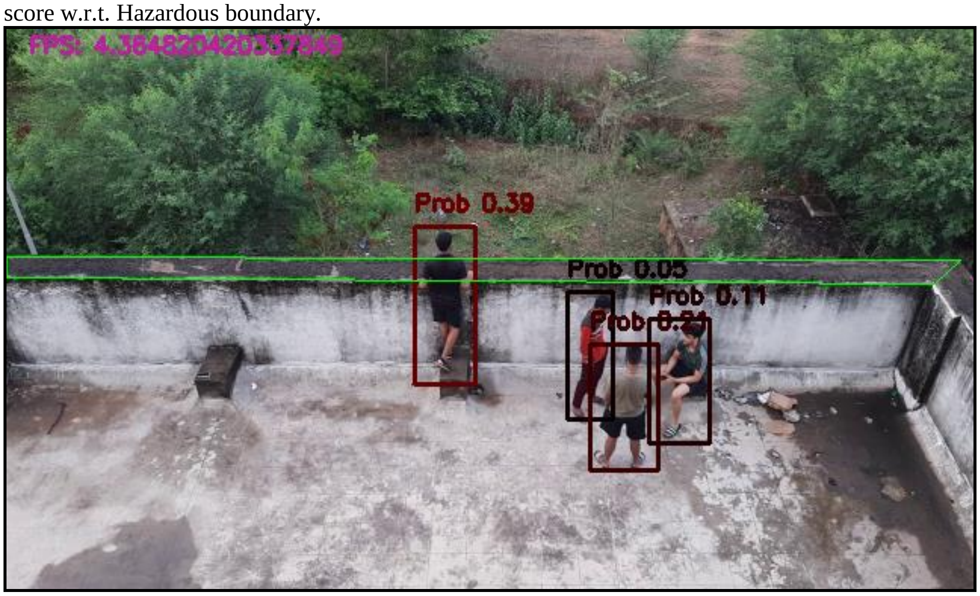

Fig. 6 is a scene processed from a video feed indicating the probability and colour change w.r.t. Fig.5 shows when a person performs suspicious behaviour at a high-risk area for a specific time.

Fig. 7 Scene from processed video feed indicating probability and color change w.r.t. Fig. 6 and Fig. 5 when a person stands on the boundary of a hazardous location.

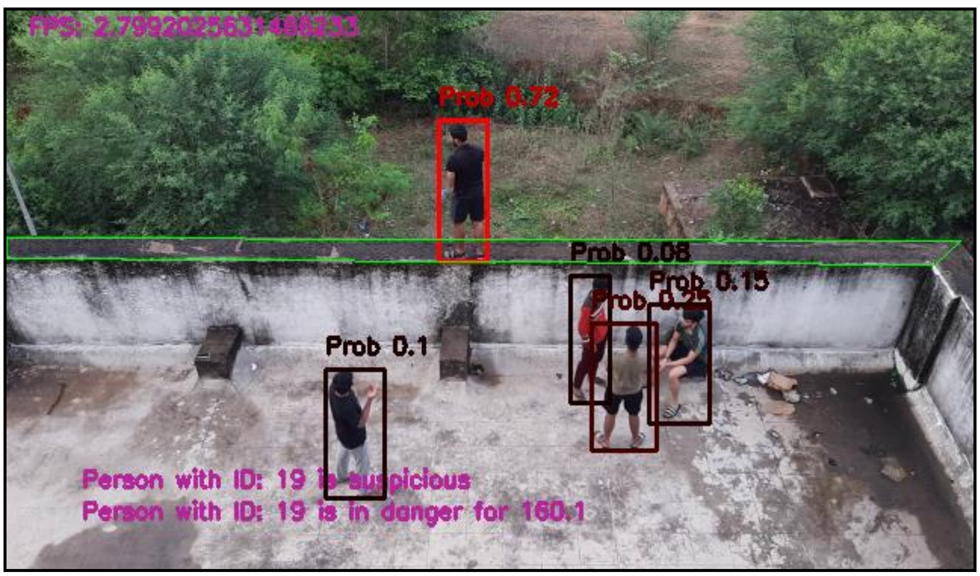

Fig. 8 Scene from processed video feed indicating probability, time, and color change w.r.t. Fig. 7, Fig. 6, and Fig. 5 when the person continues to stand on top of the boundary wall.

5. Conclusion

This paper aims to use self-inflicted violence detection as the primary objective of the research. It can save lives and avert tragedies. In this paper, we developed a model to predict a person's risk of self-inflicted violence correctly, using computer vision and deep learning methods. In our proposed model, we first used YOLO v5 for object (person) detection from the video collected from the real-time cameras. Then, the Deep sort mechanism was used to detect the person's suspicious behavior. Every person detected in the video is assigned a unique ID, and their positions are tracked. The person's ID and position are displayed with a black boundary. If a person's behavior is suspicious for a few seconds, the boundary changes to red and displays the time spent in the danger zone. Additionally, we predict the probability of the risk of self-inflicted violence based on the person's position track. The software in this system includes computer vision algorithms, machine learning, and other data preprocessing tools that will ultimately lead to higher accuracy. Hardware requirements comprise cameras, sensors, and other devices for data collection. Then, this system will be connected to any alarm system and response system, such as a notification system. Based on the experiment results, the developed model gives the best results when adequately tested and operates in real-time.

Limitations

There are a few limitations during our research:

- Limited data: Limited data leads to less accuracy and low performance. A significant amount of data can increase accuracy and performance
- Ethics: Using cameras, especially in public places, may raise ethical questions.
- Cost: system implementation and maintenance costs can be high. Installation of real-time cameras may be costly.

References

[1] Reisch, Thomas & Schuster, Ursula & Michel, Konrad. (2008). Suicide by jumping from bridges and other heights: Social and diagnostic factors. Psychiatry research. 161. 97-104. 10.1016/j.psychres.2007.06.028.

[2] Carleton, R. Nicholas, et al. "Suicidal ideation, plans, and attempts among public safety personnel in Canada." Canadian Psychology/Psychologie canadienne 59.3 (2018): 220.

[3] Lönnqvist, Jouko, H. Aro, M. Heikkinen, Hannele Heilä, M. Henriksson, E. Isometsä, K. Kuurne, et al. "Project plan for studies on suicide, attempted suicide, and suicide prevention." *Crisis* 16, no. 4 (1995): 162-175.

[4] Wu, Wentong, et al. "Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image." *PloS one* 16.10 (2021): e0259283.

[5] Lin, Guijuan, et al. "An efficient and intelligent detection method for fabric defects based on improved YOLO v5." Sensors 23.1 (2022): 97.

[6] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257–276, March 2023, doi: 10.1109/JPROC.2023.3238524.

[7] Chan, Jacob, Jimmy Addison Lee, and Qian Kemao. "Border: An oriented rectangles approach to texture-less object recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[8] Liu, Wenyao, et al. "Applications of computer vision in monitoring the unsafe behavior of construction workers: Current status and challenges." *Buildings* 11.9 (2021): 409.

[9] Alzubaidi, Laith, et al. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." *Journal of Big Data* 8 (2021): 1-74.

[10] KAIST. (2019). Title of the research project for preventing self-inflicted violence. Journal of Suicide Prevention Research, 10(2), 123-145.

[11] B. Li, W. Bouachir, R. Gouiaa and R. Noumeir, "Real-time recognition of suicidal behavior using an RGB-D camera," *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Montreal, QC, Canada, 2017, pp. 1-6, doi: 10.1109/IPTA.2017.8310087.

[12] V. R. Chiranjeevi and D. Elangovan, "Surveillance-Based Suicide Detection System Using Deep Learning," *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Vellore, India, 2019, pp. 1–7, doi: 10.1109/ViTECoN.2019.8899360.

[13] Meimetis, D., Daramouskas, I., Perikos, I. et al. Real-time multiple object tracking using deep learning methods. Neural Comput & Applic 35, 89–118 (2023).

[14] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." *multimedia Tools and Applications* 82.6 (2023): 9243-9275.

[15] Fang, Wei, Lin Wang, and Peiming Ren. "Tinier-YOLO: A real-time object detection method for constrained environments." *IEEE Access* 8 (2019): 1935-1944.

[16] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." *2018 IEEE international conference on big data (big data)*. IEEE, 2018.

[17] Park, Y., Dang, L.M., Lee, S., Han, D. and Moon, H., 2021. Multiple object tracking in deep learning approaches: A survey. Electronics, 10(19), p.2406.

[18] Kalake, Lesole, Wanggen Wan, and Li Hou. "Analysis based on recent deep learning approaches applied in real-time multi-object tracking: a review." IEEE Access 9 (2021): 32650-32671.

[19] Chai, Junyi, et al. "Deep learning in computer vision: A critical review of emerging techniques and application scenarios." Machine Learning with Applications 6 (2021): 100134.

[20] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: Challenges, architectural successors, datasets and applications." multimedia Tools and Applications 82.6 (2023): 9243-9275.

[21] Heller, Travis S., Jacinta L. Hawgood, and Diego De Leo. "Correlates of suicide in building industry workers." Archives of Suicide Research 11.1 (2007): 105-117.

[22] Hemmer, Alexander, Philipp Meier, and Thomas Reisch. "Comparing different suicide prevention measures at bridges and buildings: lessons we have learned from a national survey in Switzerland." PLoS One 12.1 (2017): e0169625.

[23] Beautrais, Annette. "Suicide by jumping: A review of research and prevention strategies." Crisis 28.S1 (2007): 58-63.

[24] Nguyen, Huy Hoang, et al. "Yolo based real-time human detection for smart video surveillance at the edge." 2020 IEEE Eighth International Conference on Communications and Electronics (ICCE). IEEE, 2021.

[25] Putra, Muhammad Hadi, et al. "Convolutional neural network for person detection using yolo framework." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.2-13 (2017): 1-5.