iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80029

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Homography Based Real-Time Intrusion Detection and Multi-Person Tracking System

Khushpreet Sindhu[1], Karishma Rathore[2], Deepika Sahu[3], Prof. Durgeshwari Sahu[4]
[1,2,3] *Student,* [4]*Assistant Professor, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Chhattisgarh, India*

Abstract: ***This research proposes a video intrusion detection system for detecting person and raising alarms on their entry into unauthorized zones. Five metrics have been used for evaluating this system. These include precision of homography calibration, precision in zone detection, effectiveness in tracking multiple persons, alarm generation, and database logging of events. According to experiment results, all above mentioned metrics exhibit efficient performances. The highest reprojection error for homography calibration was observed to be 0.58 inches and average error is 0.23 inches which is precise enough to perform spatial mapping with a minimum zone size of 20 inches. Zone detection was found to be efficient enough in classifying the person into one of the three predefined zones based on coordinates***
Keywords: YOLO, MediaPipe, R-CNN, Computer vision, PASCAL VOC, COCO, SVD, DLT, Trackelts, BYTE, IDF-1, RANSAC, Homography, Calibration, Bytetrack, Database logging.

I. INTRODUCTION

The term surveillance refers to the observation and monitoring of a person, group of people, or place, in order to collect information, influence, manage, or control. Governments and various organizations make use of surveillance for several purposes. With regards to security in the era of digitization, there is nothing better than the Surveillance System which can be used to ensure security as well as recording of events. Surveillance systems are not just limited to homes; rather they can be installed anywhere for the same purpose. In terms of surveillance systems, apart from the security purposes mentioned above, these systems are installed in places where they serve other purposes as well, namely at places of employment, manufacturing units, and in public places among others.

The most popular form of surveillance is closed-circuit television or CCTV. This technology collects, sends, and records video within the closed circuit. Although the technology was suitable enough for basic surveillance, this technology has some inherent weaknesses which make it incapable of being an effective tool for either detection or forensic purposes. Therefore, a huge amount of cameras should be used for one area which not only makes the system expensive but also does not improve detection performance at all. Moreover, old analog cameras do not have any analytical power and detection of incidents can only occur through human operators analyzing video images, either live stream or recorded ones.

This necessitates a situation whereby the collected visual information should be adequate enough to facilitate automated and reliable identification, rather than mere observation. It is henceforth that pixel-based detection emerges as the subsequent development. The pixel-based detection method utilizes the MediaPipe module for pose detection and pixel-based labeling for labeling of warning and restricted zones. Though it did better than basic CCTV systems, the results for assignment of zones were unsatisfactory; due to varying perspectives, the system failed to detect accurate zones, leading to false warnings.

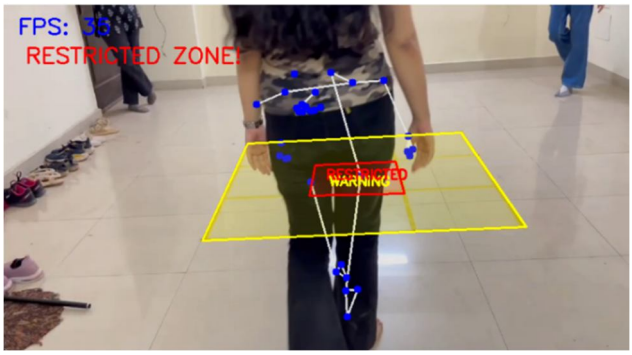


Fig. 1 Pixel based surveillance system's incompetence in detecting accurate zone. Source: Authors

As shown in Fig. 1, it can be seen that the individual has not even reached the warning zone when the system raises an alarm regarding a restricted zone violation, due to varying perspectives. It is henceforth that we introduce a planar homography-based detection system. The transformation is employed to map points on the image to the ground plane in reality. This process assists in converting the view captured by the camera into a more realistic representation of the world. This ensures that the system operates without distortion.

II. LITERATURE REVIEW

The need to have security systems arises from the alarming increase in breach of privacy and threat to protection by burglaries, intrusions and trespassing in both personal and professional environments. Be it small stores, houses or treasuries; these systems help both prevent such mishaps and also help solve them. These systems may differ depending on need, area to cover, targets to meet and geographical location too.

Some traditional technologies used in intrusion detection systems:

A. Passive Infrared Sensors

By measuring radiating Infrared light, this electronic sensor detects the presence of humans and performs required actions. For humans, the surface temperature is about 27 c - 36 c with its radiant energy in a range of 8um-12um. PIR receives infrared radiation from the human body rather than radiating it itself. [1]

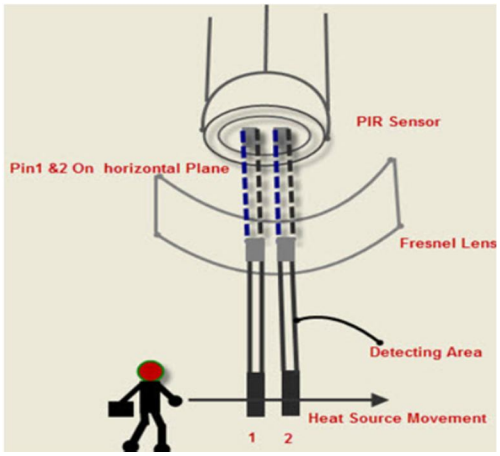


Fig.2 Passive Infrared Sensor. Source: [1]

B. Microwave Sensors

Microwave sensors do not use infrared radiation for thermally sensing human presence and motion, it utilises microwave frequency and the amount of energy bounced back towards it. Apart from it being large in scale and costly, one major flaw of this kind of sensor is its inability to distinguish between living and non-living objects. Its drawbacks range from not being able to count objects present to it being prone to raise false human alarms.

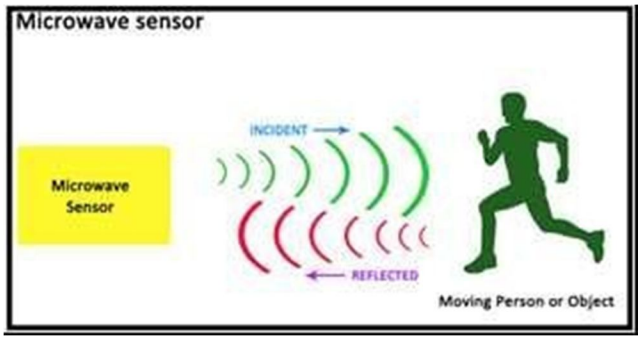


Fig.3 Microwave Sensor. Source: [2]

C. Photoelectric Beam Sensors

Photoelectric sensors also known as photo-eye sensors utilize light rays for detection purposes. Photoelectric beam sensors specifically use a laser beam using an emitter and detects obstruction in its path towards the receiver.[3]

Fig. 4 Photoelectric Beam Sensor. Source: [3]

One of its major drawbacks is its limitation to a single straight path thus inadequate for a larger and wider surface area.

These traditional systems fall short at being useful in real world scenarios. In contrast, our proposed system not only detects and tracks multiple people, it also successfully applies multi-region zone logic and automated action responses.

Earlier models like R-CNN were unfair to use in real time as they took a lot of time for prediction, whereas YOLO is drastically faster as it only needs a single look at the network to make a prediction. The first version of it i.e. YOLOv1 performs the prediction by first resizing the input image to 448x448 pixels and is then divided into 7 x 7 grids. The grids cell at the centre of the image holds preference when it comes to making predictions with the aim to attain minimum loss.[4] The entire process is completed in a single forward pass. It is trained end-to-end on the PASCAL VOC dataset.

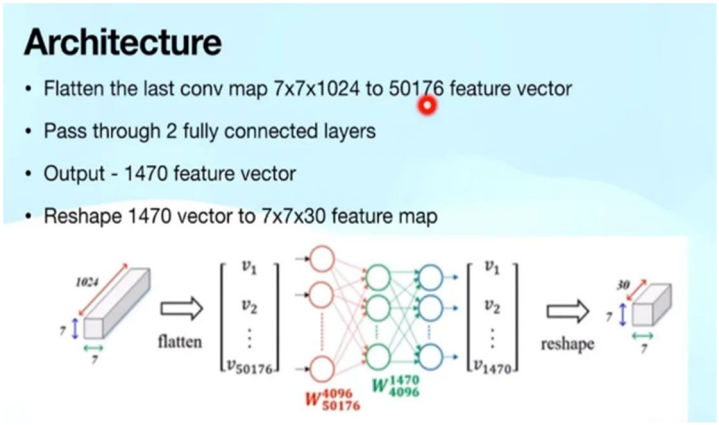


Fig. 5 YOLO architecture. Source: [4]

Loss is the difference between actual and predicted values and is calculated by considering the loss over all the grid cells. Localization loss penalizes error in predicted bounding box position and size:

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2 + \left(\sqrt{w_i} - \sqrt{\hat{w_i}}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h_i}}\right)^2 \right]$$

(x, y) (x, y) (x, y) — predicted vs actual box centre coordinates
(w, h) (w, h) (w, h) — predicted vs actual box width and height
Square root on w, w, w and h, h, h to penalize errors in small boxes more heavily than large ones
$\lambda_{coord} = 5$ *to give localization higher weight than classification*

Whereas, confidence loss penalizes error in objectivity score for both object and no-object cells:

$$L_{conf} = \sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i - \hat{C}_i\right)^2 + \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(C_i - \hat{C}_i\right)^2$$

$\hat{C}_i C_i$ – predicted confidence, actual confidence (IOU of predicted vs ground truth)
$\lambda_{noobj} = 0.5$ to down-weight empty cells, since most grid cells contain no object

Key limitations of version 1 were localization precision, prediction rigidity, lower accuracy compared to Two-Staged models. Each successive YOLO version had scalability, accuracy with speed and deployment flexibility advancements.[5] In our proposed system we have used YOLOv8 which is pre-trained on COCO datasets. It's utilized to detect and track multiple human presence simultaneously in our field of sight.

According to Yifu Zhang et al. (2021), in the typical methods of Multi-Object Tracking, the identity is established by matching only such detection boxes that meet the criterion of a sufficiently high confidence score. Such approaches are rather counterproductive since objects having a low detection score those being covered by occlusion, for example are eliminated from the dataset entirely. The researcher's response to the mentioned problems is their development of an innovative association mechanism named BYTE. In contrast to other trackers, BYTE retains almost all the detection boxes. The comparison of the similarities of low-scored detections and tracklets enables the efficient reconstruction of the actual object trajectory and separation of false positives from real objects. With the use of the new method added to nine top-notch trackers, an improvement in IDF1 scores was achieved consistently, leading to the creation of the ByteTrack model, which is now considered the state-of-the-art tracker. The proposed algorithm consists of two steps carried out hierarchically. The first step implies matching high-scored detections with tracklets via Kalman filters and using such criteria as IoU or Re-ID. In turn, the second step involves matching tracklets with low-scored detections via motion analysis. Taking into account the improvements made in ByteTrack, Longxiang You et al. (2024) added some improvements to make the tracker better suited for tracking cars. One of these improvements was connected to the fact that the initial Kalman Filter model was not very good at handling the non-linear dynamics of cars. For that reason, the state vector and the matrix were optimised. Another improvement added by these scientists was GSI, which stands for Gaussian Smoothing Interpolation. It allowed making the trajectory continuous without any interruption caused by detection loss, eliminating identity switch and preventing the vehicle from being lost while it is obstructed.[6]

According to Saad M. Khan and Mubarak Shah (2006), multi-view person tracking can be effectively executed using a novel planar homography constraint. This approach specifically addresses the challenge of resolving occlusions—instances where people are hidden behind one another—to accurately pinpoint ground-plane coordinates (i.e., where a person's feet are located). Their research focused on high-density environments where crowds are so congested that individuals are rarely visually isolated. Notably, their methodology eschews traditional colour modelling or individual shape cues. Instead, it relies on geometric constructs and the fundamental distinction between foreground and background. The core of this constraint lies in a simple but powerful observation: only pixels corresponding to the actual contact points on the ground (the feet) will consistently "warp" into foreground regions across every camera view. By utilising foreground likelihood maps rather than binary images, they intentionally deferred the thresholding process to maintain maximum data integrity. Furthermore, this homography constraint functions as a visual hull intersection on a 2D plane. Unlike traditional algorithms, this method is remarkably efficient as it bypasses the need for complex 3D camera calibration.[7] However, based on the work by Y Luo et al (2023), the common geometric projection transformation models consist of projection transformations, rigid body transformations, and affine transformations. Affine and rigid transformations can be applied to some deformations and minor perspective transformations. On the other hand, the projective transformation model is an advanced and complex geometric transformation model that can handle several changes and perspective transformations of viewpoints. Homography estimation is used to identify the projective transformation between two images. By this technique, one can effectively analyse and compensate for the geometrical differences in the images through image registration. Homography estimation relies on the homography matrix as the main component of its operations. The homography matrix can handle different types of geometric transformations, including rotation, translation, scaling, and projection. After decades of study, many techniques of homography estimation have been developed for different purposes, such as image registration, image fusion, and object detection.[8]

Homography estimation facilitates the development of self-driving cars and security surveillance systems by accurately identifying and monitoring objects in dynamic situations. By successfully correcting the visual aberration brought on by various photographic angles, in-camera calibration and perspective correction, homography estimation can significantly increase the identification, system's accuracy and dependability.

Zone-based security systems are frequently employed in autonomous monitoring applications, smart environments, and surveillance. In order to identify irregularities, unlawful access, or dangers, these systems partition a physical area into predetermined zones and keep an eye on activity within each area. Many strategies have been investigated over time, from conventional vision-based tracking to cutting-edge AI-driven methods.

Single-camera object identification and tracking techniques were the mainstay of early surveillance systems. These methods performed poorly in complicated settings, particularly when blockage was involved and high population density.

To increase tracking accuracy in congested settings, Khan and Shah, for instance, suggested a multi-view tracking system that integrates data from numerous cameras utilizing planar homography constraints. By merging foreground data from many viewpoints, their approach successfully addresses occlusion problems, improving individual localization. However, these systems are less useful for scalable zone-based deployments since they significantly rely on several calibrated cameras and unobstructed ground-plane visibility [7].

Pixel-based tracking techniques were the focus of subsequent developments in motion detection, especially for dynamic scenes with moving cameras. Sundaresan presented a technique that uses depth information and pixel deviation to separate object motion from camera motion.

These systems are computationally efficient, but because they mainly concentrate on motion detection rather than contextual comprehension of spatial regions, they have limited capacity to construct and enforce semantic zones [9]. From a geometric standpoint, zone mapping and spatial alignment have been made possible by the widespread application of homography-based techniques to map various camera perspectives into a shared plane. According to recent evaluations, homography estimation is essential for applications including picture registration, object tracking, and surveillance. However, substantial viewpoint variations, multimodal inputs, and environmental changes, all of which are prevalent in real-world zone-based security systems are present difficulties for conventional homography techniques [8].

An evaluation of the literature regarding contemporary security and surveillance systems recognizes limitations of any given zone-based surveillance system which hinders its efficacy in operational environments. Predominantly those systems that consist, in particular, of multiple camera systems, depend heavily on the use of complex hardware configurations, predominantly requiring that the camera(s) have an unobstructed - ground plane view, while necessitating precision calibration upon installation. Alternatively, although multi-view techniques improve movement-tracking capabilities (i.e., movement-tracking capabilities reduce false positives due to obstruction), their practical applications are impeded by the significant costs, limited availability, and challenges posed by the scalability of such techniques.

In contrast, single-camera techniques or pixel-motion techniques experience similar difficulties, as they fail to differentiate between the movement of an object that is actually present and changes in the background of an environment that is "busy" or "dynamic"[10]. Furthermore, it cannot be stressed enough that many existing systems do not provide real-time intelligence about zone/sub-zone definition; most systems specify the zone/sub zone geometrically, without any inclusion of contextual knowledge about the importance of each defined zone/sub zone or the type of behavioral activity occurring within each defined zone/sub zone [7]. Therefore, even though either presence/detection can occur, it will not be possible for the surveillance systems to differentiate or infer, via software, whether the activity being detected would be indicative of a crime or other "suspicious" behavior; i.e., typical activity. Furthermore, a large number of techniques exhibit high levels of sensitivity to environmental changes (e.g., ambient illumination, shadows, weather, and camera angles), which could significantly reduce the effectiveness of such techniques in an operational environment.[9]

III.METHODOLOGY

Since pixel-based detection algorithms have their own limitations, the suggested approach uses the homographic transformation between the imaging plane and the real floor to establish a one-on-one mapping. As a result, the algorithm will be able to run in real inches rather than screen pixels.

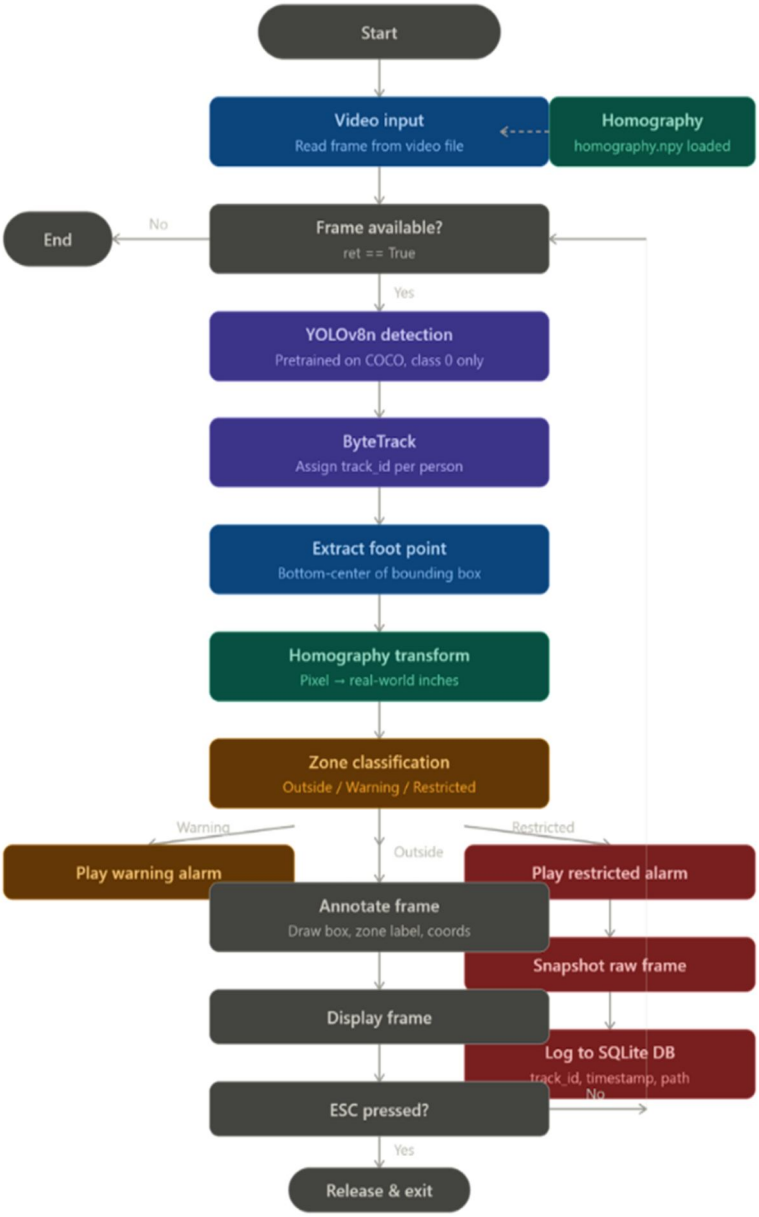


Fig. 6 Flowchart of the proposed system. Source: Authors

Architecture of the suggested solution is depicted on Fig.1. First, the system receives the stream of video frames as input data. Each frame is processed individually. For every frame individually, it goes through the pretrained YOLOv8n model and all possible individuals in the image are detected. Afterwards, the ByteTrack algorithm is executed in parallel with the object detection process, and each detected individual is provided with a persistent identity across frames. Then, after identifying the individual, their bottom centre point of the bounding box is chosen as the proxy of the foot position of the individual in inches in the physical world. This way, this point goes through the precomputed homography matrix, thus transforming from pixel-based to inches-based coordinates in the physical world. Such coordinates are evaluated against predetermined zones, thus defining what zone the person is currently at – out-of-bounds, warning or restricted zones. Based on this evaluation, an appropriate audio signal is generated by the alarm manager. Moreover, if a person enters a restricted zone, a raw frame image with track ID and timestamp is recorded into the SQLite DB. However, the accuracy of the coordinate transform is fully determined by the quality of the homography calibration, which is described in the following:

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A. Physical grid setup

The mapping of the digital movement to real-world coordinate space was achieved by using a precise calibration method through planar homography. It was achieved by creating a physical reference frame and aligning the camera's point of view mathematically. The surveillance area was created with the help of a 1.524m x 1.524m (60x60 inches) grid pattern laid out on the surface at (x) inches away from the camera with a step size of 0.508m (20 inches). For the grid reference origin, the upper left grid point, which is the opposite end of the grid with respect to the camera, was taken as the origin of the world frame. X coordinates run from left to right while Y runs from far to near to the camera, starting from 0 inches up to 60 inches. For the flat surface, the grid was placed on a levelled surface to meet the planarity requirement of homography operation in 2D space. Homography, according to Agarwal et al. (2005), is a non-singular mapping of a point in one plane onto another plane. The surveillance camera's image plane can be mapped to the ground plane using homography.

The relation between a point $\mathrm{x} = [\mathrm{u}, \mathrm{v}, 1]^{\mathrm{T}}$ in the image space and its real-world counterpart $\boldsymbol{x'} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{1}]^{\boldsymbol{T}}$ is governed by the 3x3 transformation matrix H as follows:

$$x' = Hx$$

Expressing this in matrix form, we have (as per the survey):

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

For degrees of freedom and constraints, H can be specified only up to a scalar factor and hence has 8 degrees of freedom. In order to determine these unknown variables, at least four points correspondences are needed. In the system under consideration, however, 16 points are purposely selected in order to enhance stability and reduce the sensitivity of the results to errors in individual point locations since each correspondence imposes two constraints. In order to estimate H from the actual physical arrangement of the grid, the Direct Linear Transformation (DLT) technique is commonly used. The following homogeneous equation is solved for each corresponding point i:

$$L_i h = 0$$

If L_i denotes the matrix formed based on the coordinates of the point and h denotes the vector having nine elements of H. The proposed method uses 16 points correspondences, and hence the solution is over-determined, and H is computed based on the minimization of the projection error by using SVD.[5] For the proposed approach, cv2.findHomography() function available in the OpenCV library is used to determine H, which implements the DLT method with RANSAC to reject the outliers. All 16 points correspondences have been considered as inliers.

B. Calibration

In order to obtain the correspondences between the image and world space, a special calibration routine was developed. The first frame of the input video was obtained and presented to the operator. The operator manually selected all 16 points where the lines intersected each other using a click on the point in a predefined sequence from the furthest row to the nearest one going from left to right in each row. Thus, pixel coordinates were collected for all 16 points, which, being combined with their real-world counterparts that were placed at 20 inches intervals, formed the set of correspondences. Next, cv2.findHomography() was called using these 16 points and their world coordinates as inputs; the library used the DLT algorithm enhanced with RANSAC outlier rejection to determine the matrix H. It turned out that all 16 pairs of correspondences were accepted as inliers. In order to verify the calculation, four corner points of the grid were transformed into world space using cv2.perspectiveTransform(); the difference between their calculated values and the expected ones was no more than 0.5 inches. Finally, the matrix H was serialized to homography.npy and deserialized only once during system startup.

C. Homography Computation

The algorithm for homography determination is based on projective geometry, using the cv2.findHomography() and cv2.perspectiveTransform() algorithms from the OpenCV library.cv2.findHomography() - computes a 3x3 matrix for the perspective transformation H of a list of corresponding point pairs using DLT (Direct Linear Transformation).

Random Sample Consensus (RANSAC) is employed within to detect outlier point pairs in order to generate a robust estimate for the homography. cv2.perspectiveTransform() performs the homography transformation on any point by multiplying the computed H matrix with the point coordinates.

For matrix operations, storage, formatting and I/O purposes, we use the NumPy module. The H matrix computed from the calibration routine is stored to the disk using np.save() in homography.npy and then loaded on each program initialization using np.load(). OpenCV's cv2.findHomography() provides input in the form of 16 manual clicks on the image in pixels and 16 corresponding inches on the object to calculate the 3 × 3 homography matrix H using the Direct Linear Transformation (DLT) method with RANSAC.

The cv2.perspectiveTransform() Applies the stored H matrix every frame to convert the foot point pixel coordinate of each detected person into real-world ground-plane coordinates in inches.

```
pt = np.array([[[cx, cy]]], dtype=np.float32)
world_pt = cv2.perspectiveTransform(pt, H)
real_x, real_y = world_pt[0][0]
```

NumPy's np.save() and np.load() saves the computed H matrix to disk after calibration and reloads it once at runtime startup, separating the one-time calibration from the continuous inference loop.

```
np.save("homography.npy", H) # calibration
H = np.load("homography.npy") # runtime
```

Foot Point Selection Bottom-center of each bounding box (cx, y2) is used as the ground contact point rather than the box center, since it represents the actual position of the person on the floor plane.

$$c_x = \frac{(x_1 + x_2)}{2}$$

$$c_y = y_2$$

A homography maps a point **p** in image space to a point **P** in world space using projective transformation:

$$\begin{bmatrix} X \\ Y \\ W \end{bmatrix} = H \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Where the real-world coordinates are recovered as:

$$X_{real} = \frac{X}{W}, Y_{real} = \frac{Y}{W}$$

Then using matrix H is a 3x3 matrix with 8 degrees of freedom

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$

It is solved using Direct Linear Transform (DLT) with a minimum of 4 non-collinear point correspondences. In our system, 16 points are used (4×4 grid) for a robust RANSAC-based estimation.

D. Detection and Tracking Pipeline

The flow for human presence in our field of sight is explained in the diagram below.

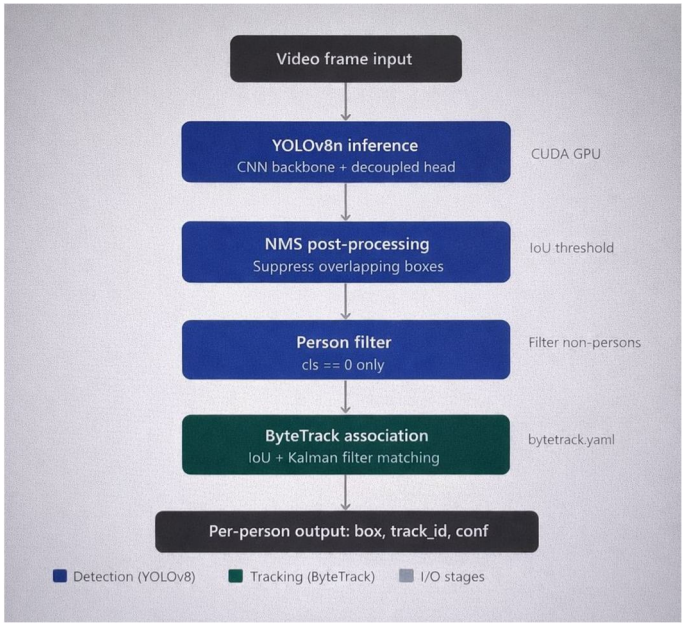


Fig.7 Detection and tracking pipeline flow chart. Source: Authors

1) YOLOv8

The YOLOv8 is the latest version of YOLO family developed by Ultralytics. It is a single stage anchor free detection network for object detection. Previous versions of YOLO utilized predefined anchor boxes, however, YOLOv8 is the first YOLO model that uses anchor-free detection head which directly predicts object's center and box dimensions, thus improving generalization and reducing hyperparameter sensitivity. YOLOv8 comprises of a feature extractor backbone model called CSPDarknet, a feature fusion module called PANet and a decoupled detection head model for classifying and regressing detection results. YOLOv8 Nano is chosen as the backbone in this work.

2) ByteTrack

It is an efficient and powerful multi-object tracker which solves one of the major problems in current trackers: discarding of low confidence detections. In conventional trackers, detections with low confidence score cannot be associated with any tracks. In case a pedestrian is partially occluded and hence gets detected with low confidence score, it will lead to switching of identity when the object is clearly visible again. ByteTrack solves this problem by performing two-stage association, first associating the confident detections and then recovering the low confidence detections. In terms of the video surveillance domain, person detection refers to locating each person in the captured scene by means of a bounding box. Trackings go further than that, as they provide an additional function of keeping the identity of every detected object in the scene constant throughout time, thereby allowing the identification of Person A entering the dangerous zone and Person B who is staying within the allowed zone. Without trackings, there is no way to identify who did what, as each frame sees everything anew without any knowledge about the past frames. The joint usage of YOLOv8 and ByteTrack allows tracking of multiple persons at once with unique track_ids, even if the persons are crossing paths and partly occluding each other or leaving/entering the scene. The pipeline of detection and tracking is the primary component of the system that acts as the first step and forms the core of the system. The output of the pipeline is used as input in every other module after the detection and tracking pipeline. The homography module utilizes the foot point of the bounding box to calculate the real-world coordinates. These real-world coordinates are then used by the ZoneManager to identify the zone membership of an individual. The Recorder module uses the track_id to ascertain whether there has been any previous snapshot stored for the individual in the same zone. The AlarmManager makes use of the dominant zone of all individuals being tracked to determine the alarm status.

In this project, a pretrained YOLOv8 model by Ultralytics is adopted to detect humans in real time. As the model has been trained on huge datasets such as COCO, it directly executes the inference process on individual frames and provides the bounding boxes in the form (x1, y1, x2, y2) together with their confidence values and labels. Confidence scores of each detection are determined by multiplying the probabilities of object and class (Confidence Score = P(object) x P(class|object)). Only those detections related to the class "person", which corresponds to class index 0, and have a confidence score greater than a certain threshold will be kept. In order to avoid multiple detections of the same person, Non-Maximum Suppression (NMS) is employed, which uses Intersection over Union

$$(\text{IoU} = \text{Area}(A \cap B)/\text{Area}(A \cup B))$$

to determine the amount of overlapping and retains the most confident bounding box only. Then, the filtered detections will be processed by ByteTrack to serve the purpose of tracking. ByteTrack uses Kalman filtering to estimate the state of the object to be tracked, where the state vector contains details about the centre, width, height, and velocity of the object's bounding box ($x = [cx, cy, w, h, vx, vy, vw, vh]^T$). Using the formula for the next state, which is

$$(x_k = F \cdot x_{k-1} + \text{noise})$$

ByteTrack can predict the object's state and hence track the object even when there is no detection. Matching detections to existing tracks will be done through the process of matching using IoU with Cost being (1 – IoU). This involves matching high-confidence detections in the first step and then matching low-confidence detections to unmatched tracks in the second step. The optimal solution to this problem is achieved via the Hungarian algorithm. The use of the pre-trained YOLOv8 detection model and ByteTrack tracking model will ensure the tracking of multiple people accurately and effectively.

E. Zone classification

The zone classification logic uses the object's real-world coordinates to put it in one of three spatial areas: outside, warning, or restricted. The first thing that happens in real-world surveillance systems is finding the positions of things in image coordinates. Then, geometric transformations like homography are used to turn those positions into real-world coordinates. After mapping, the system uses set spatial boundaries to figure out what zone it is in. This method checks for boundaries in a hierarchical way. To start, the system checks to see if the object is in the overall monitored grid. Then it checks to see if it is in a restricted area that is very important. Mathematical representation of zone classification is done using the following steps:

1) Coordinate mapping

It is done from camera to real-world using

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Where: (x,y) = image coordinate

(x_r, y_r) *= real-world coordinates*

2) Grid boundary condition

For outside zone check it is:

$$(xr, yr) \in G \iff Xmin \leq xr \leq Xmax,\ Ymin \leq yr \leq Ymax$$

Which if not satisfied is considered outside. For restricted zone, the condition is:

$$(xr, yr) \in R \iff Xrmin \leq xr \leq Xrmax,\ Yrmin \leq yr \leq Yrmax$$

And the zone is considered warning if it is inside the grid but outside the restricted region.[11]

F. Alarm Management

The alarm management system sends out the right alerts based on the object's classified zone. It acts as a decision layer that changes zone data into real-time responses, such as audio alerts. The system remembers the current state so that only the right alarms go off and sound doesn't play more than once. This method uses a state-based control system, which means that the alarm only goes off when the zone changes. This makes things run more smoothly and stops unnecessary interruptions. It is mathematically represented as:

$$A(Z) = \begin{cases} 0, Z = Outside \\ A_w, Z = Warning \\ A_r, Z = Restricted \end{cases}$$

Z = *Detected zone*
A_w = *Warning alarm sound*
A_r = *Restricted alarm sound*
0 = *No alarm*

The assigned state is updated only if $Z_t \neq Z_{t-1}$. This ensures no repeated playback if the zone remains the same and efficient alarm handling. Logic used by alarm control is:

$$A_{current} = A(Z_t)$$

$$A(Z) \rightarrow Play\ in\ loop\ if\ Z \in \{Warning, Restricted\}$$

G. Database Logging

The Recorder module is a key part of the zone-based security system that makes sure that intrusion events are always saved. This module uses an event-driven logging system instead of traditional surveillance systems that record video streams all the time. These systems take up a lot of space and make duplicate data. It only records and saves data when something important happens, like when an object being tracked enters a restricted area. This method makes systems more efficient, scalable, and easy to use by only keeping data that is important for security. The recorder performs three main tasks:

1) Event detection - Detects when an object transitions into a restricted zone**,** uses track ID + zone history to avoid duplicate logging
2) Snapshot capture - Captures the current video frame as evidence, Saves it with a unique timestamp-based filename.
3) Database logging - Stores metadata in a structured database, Enables future querying, auditing, and analysis

Zone transition is detected using $\Delta Z = Z_t - Z_{t-1}$ and logging is triggered only when $\Delta Z \neq 0$.

$$E_t = \begin{cases} 1, Z_t = Restricted \wedge Z_t \neq Z_{t-1} \\ 0, otherwise \end{cases}$$

Where $E_t = 1 \rightarrow intrusion\ event, E_t = 0 \rightarrow no\ logging$

Then real-time video frame is converted into stored image and acts and visual proof of intrusion as

$$R = (TrackID, Timestamp, ImagePath)$$

Where each record represents $who \rightarrow TrackId, when \rightarrow Timestamp, what\ evidence \rightarrow Image\ file$

Redundancy is avoided by:

$$Log_{new} = \begin{cases} 1, Z_t \neq Z_{last(ID)} \\ 0, otherwise \end{cases}$$

This ensures each object is logged only once per zone entry and thus prevents duplicate database entries.

IV. RESULT AND EVALUATION

In this part, the experimental results of the suggested intrusion detection model are discussed. For the experiment, the system was applied to indoor recorded video captured by the monocular camera. To calibrate the camera's homography transformation and define the zones, a physical grid having 60×60 inches size was positioned in the camera's field of view. In total, there are five aspects considered in the evaluation phase, including homography calibration accuracy, detection of zones, multi-person tracking, alarm behavior, and logging to a database.

A. Homography calibration accuracy

For calculating the homography matrix accuracy, we back-projected four points of the physical grid using cv2.perspectiveTransform(). Table 1 shows both real-world coordinates and our system's output along with the computed Euclidean distance error using the following equation.

$$Error = \sqrt{\left(x_{\exp} - x_{got}\right)^2 + \left(y_{\exp} - y_{got}\right)^2}$$

Corner	Expected (inches)	Got (inches)	Error (inches)
Top-left	(0,0)	(0.1,-0.2)	0.22
Top-right	(60,0)	(59.7, -0.5)	0.58
Bottom-left	(0,60)	(-0.0, 60.0)	0.00
Bottom-right	(60,60)	(60.0, 60.1)	0.10

Table 1: Error between expected and coordinate we got.

Maximum reprojection error was measured at the upper right corner, being 0.58 inches with a mean error of 0.23 inches for all four corners. Since the smallest width of the zone boundary is 20 inches, this accuracy is sufficient enough for proper zoning of the entire observed area.

B. Zone Detection Accuracy

Zone detection accuracy was assessed based on the analysis of how a person moving through all three zones (outside, warning and restricted) would be detected by the system. In particular, coordinates displayed by the system were compared visually with the actual coordinates of the person's location according to the reference grid. In all observed cases, the correct zone was properly detected by the system with corresponding coordinate values falling into expected areas of the zone: outside the grid (outside zone), 0–60 inches (warning zone) and 20-40 inches in both directions (restricted zone).

Test Case	Person	Coordinates (inches)	Zone
Casse 1	Person 1	(98.2,-32.4)	Outside
Case 1	Person 2	(24.5,31.8)	Restricted
Case 2	Person1	(6.5, 53.5)	Warning
Case 2	Person2	(53.4, 55.1)	Warning
Case 2	Person 3	(27.7, -16.2)	Outside

Table 2: Table showcasing different zones assigned based on coordinates the person is present in

In case one, the algorithm has been successful in recognizing that one individual is not within the monitoring area while the second individual is within the restricted area because of his/her coordinate values of (24.5, 31.8), lying within the boundary range of restricted area from 20-40 inches in both directions. As far as the second example is concerned, the detection of two individuals is accurate since their coordinates lie within the boundaries from 0-60 inches in both directions.

C. Alarm Response Behaviour

To test the alert response mechanism of the system, observation of audio output was conducted as subjects moved between zones during the multi-subject test process. The alarm manager kept track of the dominant zone for all tracked individuals every time step, favouring the restricted zone over the warning zone.

In the case of Case 1 where only one subject occupied the restricted zone, the restricted alert played immediately upon entry into the zone until the exit from the zone occurred. On the other hand, in Case 2, when there was no occupant in the restricted zone but two occupants in the warning zone, the warning alert responded correctly. Once the subject had exited the grid, the alarm status was reassessed according to the rest of the tracked subjects in the grid.

Fig.8 Case 1. Source: Authors

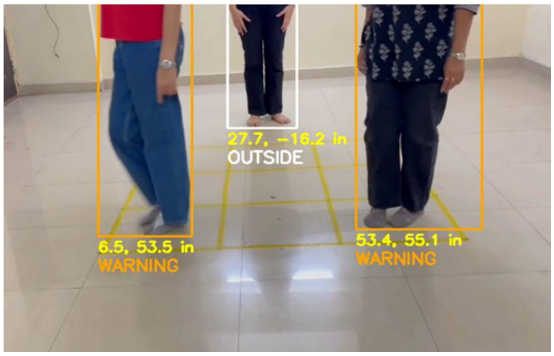


Fig.9 Case 2. Source: Authors

D. Database Logging Verification

To test the ability of the evidence logging feature, the SQLite database and snapshots folder were checked after running the experiment with multiple persons. From Figure 10, it can be observed that the database accurately logs all the events of intrusion into the restricted zones, with information about the track ID of the person entering the restricted area, timestamp of intrusion, and file path of the raw frame snapshot. Figure 10 presents the snapshots folder where the raw frame snapshots were automatically saved by the system, and the names of the files were generated using the zone name, track ID, and timestamp of the intrusion.

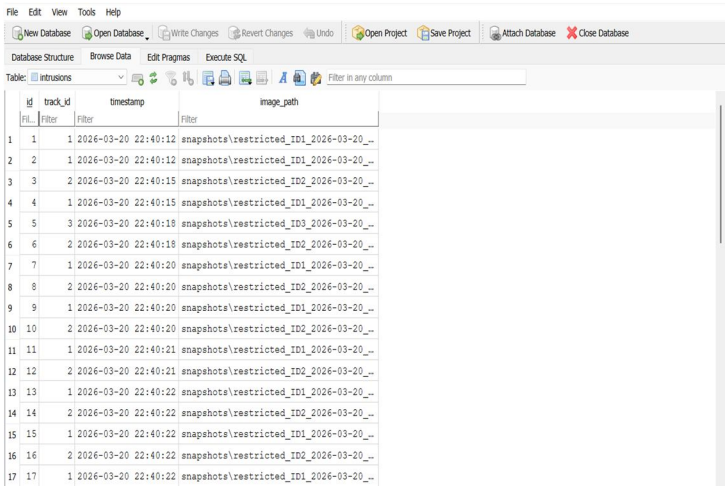


Fig.10 Database logs. Source: Authors

When opening the logged image where the restricted zone was intruded we can view that instance like shown in Fig.11

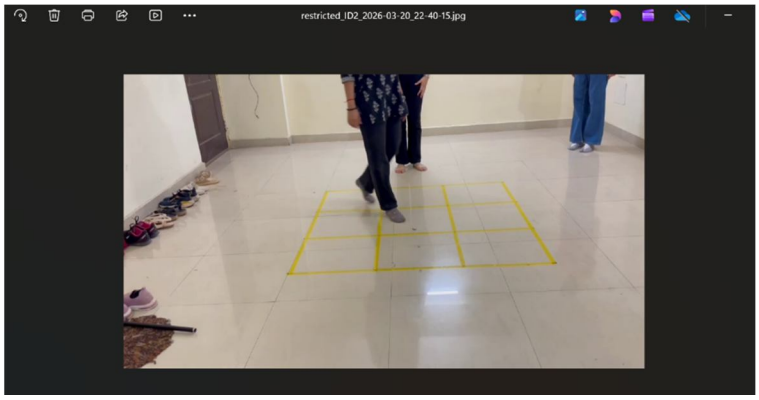


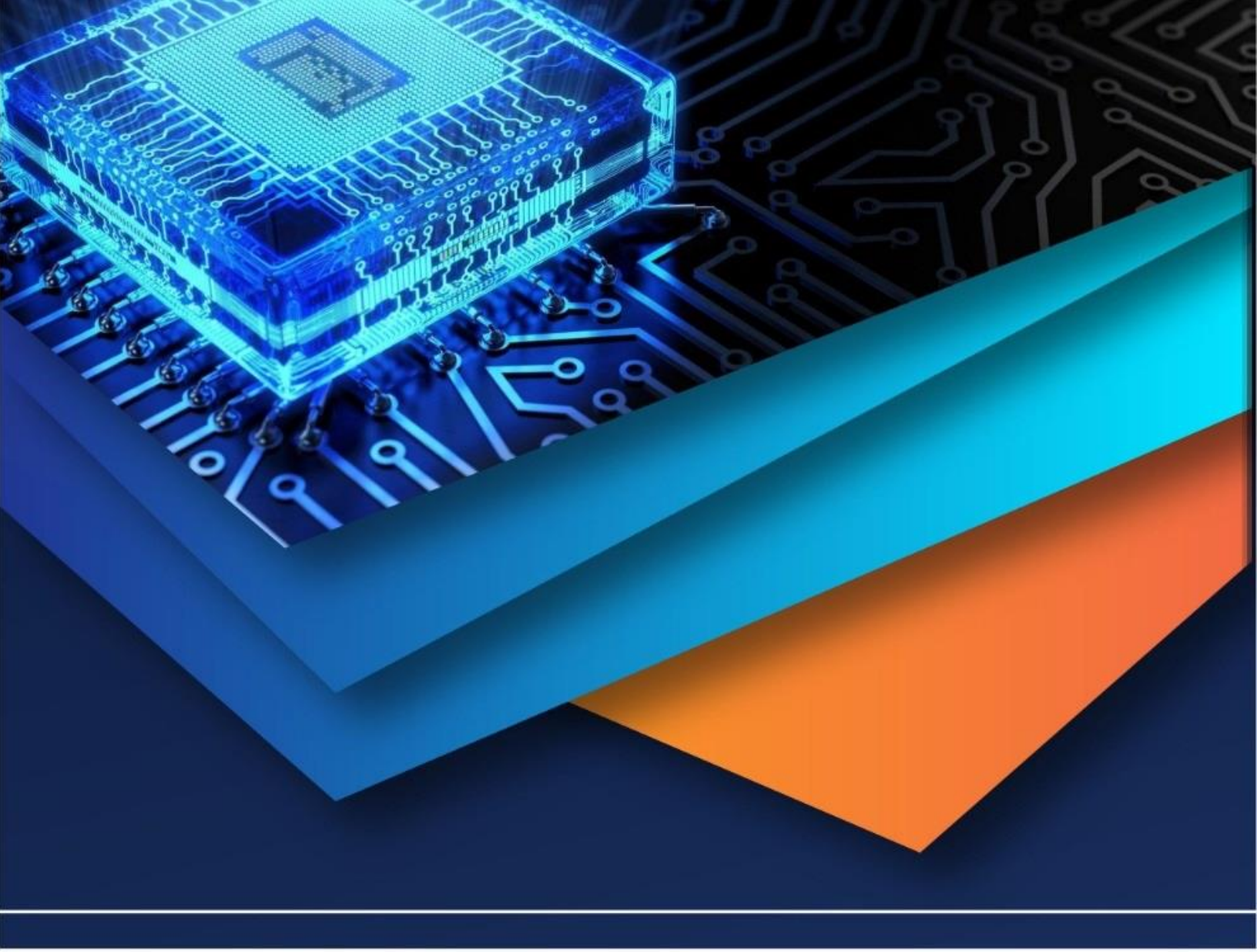
Fig.11 Instance where the restricted zone was intruded. Source: Authors

V. CONCLUSION

The developed intrusion detection system has been able to fulfill the design requirements effectively and can be put into action for monitoring applications inside buildings. It was able to provide great accuracy in homography calibration, where the highest error in terms of the maximum reprojection error was just 0.58 inches and the average reprojection error was 0.23 inches, well under the 20-inch limit for minimum distance between any two zone boundaries, thereby ensuring reliability of spatial mapping throughout the area. As far as the function of detecting zones was concerned, there were no errors made by the algorithm, as it was able to classify each individual according to their coordinate locations correctly in all cases. The system worked smartly for the purpose of alerting by taking priority for restricting zone over warning zone, along with providing real-time updating of zones in terms of the highest-priority one for each tracked individual. Additionally, logging of all incidents in the form of data about the tracks made and images of corresponding frames was also carried out properly by the system.

REFERENCES

[1] Robu.in, “PIR Sensor Working Principle”, 2020.
[2] M. Baballe et al., “The Various Types of sensors used in the Security Alarm system. International Journal of New Computer Architectures and their Applications”, 2020. 10. 50-59. 10.17781/P002618.
[3] REALPARS.com, “Photoelectric Sensor Explained (With Practical Examples)”, 2021.
[4] S. Soni, “Concept of YOLOv1: The Evolution of Real-Time Object Detection”, Medium, 2023.
[5] M. Kotthapalli et al., “YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and challenges”, arxiv.org, 2025.
[6] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in Proc. European Conf. Computer Vision (ECCV), 2022.
[7] S. Khan and M. Shah, “A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint”, in Proc. European Conf. Computer Vision (ECCV), Lecture Notes in Computer Science, vol. 3954, 2006, pp. 133–146.
[8] Y. Luo et al., "A Review of Homography Estimation: Advances and Challenges," Electronics, vol. 12, no. 24, p. 4977, 2023.
[9] R. Sundaresan, "Pixel-Based Object Motion Detection and Tracking with a Moving Camera," M.Eng. thesis, Dept. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, May 2020.
[10] Eshel, Ran and Yael Moses. “Homography based multiple camera detection and tracking of people in a dense crowd,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2008): 1-8.
[11] DigiMRO, “Zone System in Security: complete Guide to Smarter Protection”, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY