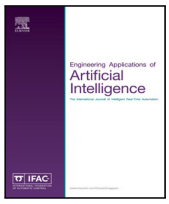
Engineering Applications of Artificial Intelligence 142 (2025) 109940

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Stampede detector based on deep learning models using dense optical flow

Antonio Carlos Cob-Parro *, Cristina Losada-Gutiérrez, Marta Marrón-Romera

University of Alcalá, Department of Electronics, Ctra. Madrid-Barcelona, km. 33600, 28805, Alcalá de Henares, Spain

ARTICLE INFO

Keywords:
Video surveillance
Stampede detection
Optical flow
Machine learning
Deep learning

ABSTRACT

The world's population has grown in recent decades, increasing social events and leading to more crowd situations with potential issues, such as bottlenecks, stampedes, or falls. In this context, this paper presents an approach for stampede detection from image sequences in low- and medium-crowd. It is based on a feature vector extracted from the dense optical flow, using the Gunner-Farneback method, and a deep learning-based classification model capable of determining, frame by frame, whether a stampede is happening. It has been evaluated on four different datasets: two widely used in the state-of-the-art— University of Minnesota (UMN) and Performance Evaluation of Tracking and Surveillance (PETS-2009)— and two new labeled datasets, Geintra-Behaviour-Analysis (GBA-Stampedes) and Geintra-Santander Multiple Actions Dataset in Cruises (GSMADC), which include realistic indoor and outdoor scenarios, as well as diverse crowd types and sizes (up to 6 people in GSMADC and a minimum of 15 in GBA). Both datasets have been made publicly available to increase the limited number of sequences for validating stampede detection in videos, with more than 43000 frames. The proposed method was evaluated across various training scenarios to test its adaptability to new environments. In the most challenging scenario, using a limited training set, our system achieved average metrics of around 99% on UMN and PETS-2009, 95% on GBA, and 91% on GSMADC. In comparison, other models achieved only 90% on UMN and PETS-2009, 80% on GBA, and below 80% on GSMADC, demonstrating the accuracy and robustness of the stampede detector across scenarios.

1. Introduction

Nowadays, the Earth's population has been growing exponentially all along the last humanity period. It reached eight billion people by the end of 2022, which means that the number of people has increased by around five million in the last seventy years, being the last billion increased in the last decade. Sociological experts detect that by the end of the century, the population on Earth will be eleven billion of person. This sort of increase of people carries out a high overcrowding in any social environment (tourism, events, quotidian actions, public transport, etc.). These crowds usually do not act chaotically, but in alarm situations, they can be risky, generating stampedes, falls or bottlenecks in dangerous places. For this reason, it is important to design detection systems to foresee these warning situations.

The advances in hardware and software in recent years, make it possible to design more powerful and faster systems with high accuracy, allowing them to save lives and early detect potentially risky situations. These advances have been proposed in embedded video surveillance systems for large vessels, such as Cob-Parro et al. (2021a), or in applications oriented to health that help to detect diseases before they occur, as Esteva et al. (2019), Hinton (2018), Ali et al. (2020), Zhou et al. (2020b) and Castiglioni et al. (2021), and in particular in the field of computer vision with works like (Wu et al., 2020; Howard et al., 2017), that is the core of all the previous ones.

It has now become possible to design powerful software architectures capable of analyzing large amounts of data in significantly reduced time. In the field of computer vision machine learning techniques and deep learning (DL) models, benefit from that advance generating comprehensive and precise information from images, surpassing the capabilities of human technicians, who may experience fatigue after long hours of work (Räty, 2010; Al-Nawashi et al., 2017).

These type of computer vision techniques are also employed in crowd analysis, enabling the detection of collective behavior and anticipation of dangerous situations. There identifying individual persons becomes challenging, so the approach needs to be modified. Hence, for detecting anomalous actions within groups of people, it is necessary to consider the crowd as a unified entity and study it as a whole (Tu et al., 2008; Mehran et al., 2009; Ali and Shah, 2007; Muhammed Anees and Santhosh Kumar, 2022).

Several proposals for crowd behavior analysis use end-to-end DL-based approaches (Luque Sánchez et al., 2020), such as convolutional neural networks (CNNs) (Tripathi et al., 2019) or transformers (Zuo

* Correspondence to: Escuela Politécnica Superior, Universidad de Alcalá, Ctra Madrid-Barcelona km. 33,600, 28805 Alcalá de Henares (Madrid), Spain.
E-mail addresses: antonio.cob@edu.uah.es (A.C. Cob-Parro), cristina.losada@uah.es (C. Losada-Gutiérrez), marta.marron@uah.es (M. Marrón-Romera).

https://doi.org/10.1016/j.engappai.2024.109940
Received 27 May 2024; Received in revised form 5 November 2024; Accepted 24 December 2024
Available online 4 January 2025

et al., 2023), which allow processing the entire image of the monitored area and extracting spatio-temporal information through convolutional layers. These proposals achieve good results in detecting crowds and analyzing their movements to prevent stampedes (Gupta et al., 2019; Haque et al., 2020), but none of them detects stampedes when they occur. Moreover, most of these works have a high computational cost, which prevents their use in real-time applications. Thus, the real-time application of DL-based methods for stampede detection is still an open issue.

As it is explained in Section 2, to reduce the computational requirements, there are other approaches for stampede detection that use traditional feature extraction and pattern recognition techniques. Since the stampedes are closely related to the movement in an image sequence, there is widely used the analysis of the optical flow to extract features (Pennisi et al., 2016). This involves measuring the movement quantity and orientation present in the image sequences, and usually computing the entropy values related to motion quantity and direction. The obtained features can be analyzed using classical methods (Pennisi et al., 2016), but it is also possible use it as the input of a DL architecture (Cob-Parro et al., 2021b). These methods allow results to be obtained at a lower computational cost compared to working with the full image. However, while these approaches can detect stamps in controlled environments, they rely on the use of thresholds, which drastically reduces its performance if there are changes in the environment or recording conditions.

Therefore, it is also worth noting the small number of datasets available for stampede detection, and the poor quality of labeling in some of them.

In this context, the present work aims to develop a robust stampede detection solution that is able to function correctly in different environments and conditions, but with a reduced computational cost that allows it to operate in real-time in a video surveillance system. Thus, this paper presents a novel approach for stampede detection with two different stages. First, there are extracted different features from a Farneback optical flow detector (Farnebäck, 2003). Then, we propose a novel feature vector that is used as input to a DL-based architecture that uses Recurrent Neural Networks (RNNs) to detect the presence of a stampede or an alarm situation regardless of the environment. The use of an ad hoc feature vector of small dimensions allows a significant reduction in computational cost compared to works that analyze the whole image with a neural network.

In addition, the approach has been designed to detect anomalous situations in various types of scenarios and realistic environments, i.e. has to be robust enough to behave efficiently facing scenes generalization. It has been proved by evaluating the proposal performance within four datasets: three of them widely used in other works related to stampede detection (Ferryman and Shahrokni, 2009; Mehran et al., 2009; Cob-Parro, 2023), and two novel ones that have been recorded and manually labeled for validating this work generalization capability. These last include stampedes in two different environments: a hall at the Polytechnic School in University of Alcalá (UAH), with more than 20 people simultaneously moving in the scene; and different parts of a cruise (including indoor corridors and a hall, and an exterior terrace) where alarm situations happen creating also stampede scenes. It is worth noting that both of them are available to the scientific community (Cob-Parro et al., 2023b).

In summary, the main contributions of this work are outlined below:

- A novel proposal leveraging the detected research gaps is here presented to determine people hazardous situations in video surveillance contexts regardless of the environment, based on optical flow.
- In order to investigate the proposed system behavior, various analyses have been conducted. These include an architectural ablation study, an examination of computational costs in comparison with alternative network structures, and a comprehensive comparison in performance metrics against other state-of-the-art stampede detectors.
- To validate the architecture's ability to detect stampedes independently of the scenario, four datasets within completely different environments have been employed. The aim behind this task is to demonstrate that the proposal is capable of detecting hazardous situations in highly controlled environments, as those shown in datasets such as PETS 2009 (Ferryman and Shahrokni, 2009) and UMN (Mehran et al., 2009; Cob-Parro, 2023), but also in more chaotic and realistic ones involving varying numbers of people and camera positions, as exemplified in GBA Stampedes (GEINTRA, 2022), recorded in a University of Alcala's hall, and GSMADC, acquired on a ferry.
- Finally, a novel dataset named GSMADC (Cob-Parro et al., 2023b) is also presented and made available for the scientific community. Recorded in a cruise, as mentioned, includes both, individual and group actions.

The rest of the paper is organized as follows: Section 2 presents a study of the main previous related works and software tools used in the stampede video surveillance context of interest. Then, Section 3 describes the hardware architecture and the developed algorithms contributed within the paper. After that, Section 4 presents and analyzes the main experimental results performed to validate the proposal. Finally, in Section 5, conclusions and further proposed work are presented.

2. Related works

The study of crowd behavior is a fundamental field in computer vision, with extensive research focusing on different methods for classifying and analyzing crowds.

Most of the works for crowd analysis can be divided in two main groups: macroscopic and microscopic approaches, as it is explained in detail in the work (Jebrane et al., 2019).

On one hand, the microscopic approaches focuses on studying individual behaviors and their interrelation, which collectively influence the crowd's behavior. In Matkovic et al. (2019) it is presented a DL model for features' extraction, combining information from histograms and optical flow to predict crowd trajectories. Besides, in Abdullah et al. (2021) individual trajectories within the crowd are correctly detected and predicted following the same idea. This type of analysis provide good results, but they typically involves computationally intensive systems and poses hardware scalability challenges, being difficult their application if the number of people increases.

On the other hand, the macroscopic approaches treats the crowd as a cohesive entity, resembling the behavior of a fluid. This type of analysis disregards individual actions within the crowd and instead examines how individuals respond to external stimuli. For instance, Liu et al. developed in Liu et al. (2019b) crowd simulations with hundreds of people to analyze crowd movement as a fluid. Moreover, in Matkovic et al. (2022) it is aimed to overcome the challenges of crowd segmentation, tracking, and region-of-interest detection by identifying predominant movement patterns within the crowd.

The proposal in this work is included in this second group, i.e. macroscopic crowd analysis, since there are not detected the individuals movement, but the whole crowd behavior.

The various works that address anomaly detection in crowds with a macroscopic approach can be further divided according to whether they perform the detection end-to-end, analyzing the whole image, or in two steps: typically features' extraction and subsequent classification. Some of the main works in each of these groups are described below, along with their advantages and disadvantages.

As it has been stated in the introduction, end-to-end approaches allow processing the entire image of the monitored area, extracting and classifying spatio-temporal information through convolutional layers. These proposals usually analyze aspects such as the movement of individuals and the nature of that movement whether abrupt or

smooth. The analysis extends to the flows of entry and exit within these crowds. In the work (Tyagi et al., 2022), a review is presented on various methods for stampede detection based on DL. Most of them, such as Gupta et al. (2019) and Haque et al. (2020), achieve good results in detecting crowds and analyzing their movements to prevent stampedes. However, end-to-end approaches usually come with computational cost implications, requiring high-performance hardware for running. Moreover, due to their end-to-end nature, scaling and diversifying them across different hardware setups can be complex, leading to bottlenecks. These drawbacks make most of those end-to-end works not suitable for real-time applications.

The second group includes works that process sets of images to extract information, generating a feature vector, usually related to the movement of people in the scene, and then use a classification stage to detect anomalous crowd behaviors. It should be noted that, the choice of descriptors and the definition of anomalous crowd behaviors' classes heavily influence the performance of the classification stage, being very dependent of those aspects. Regarding the classification stage, several models have been proposed. These include both classical classifiers (Pennisi et al., 2016; Cob-Parro et al., 2021b) as well as classifiers based on neural networks, including those using context location and motion-rich spatio-temporal volumes (Patil and Biswas, 2018), temporal convolutional neural network patterns (Ravanbakhsh et al., 2018), generative adversarial networks (Ravanbakhsh et al., 2017), and global event influence models (Pan et al., 2019).

The two-stage approaches provide good results in controlled environments, with a lower computational cost than end-to-end works, since the classifiers in the former group process the feature vector with a smaller size than the input images. However, the main drawback of those works is the lack of generality across different scenarios and contextual changes, suffering a rapid drop in their performance if the environment or recording conditions change, as happens with Pennisi et al. (2016), where it appears a threshold that needs an adjustment to make the proposal working effectively in varied scenarios.

In summary, end-to-end approaches usually provide better results than two-stage approaches, but at a much higher computational cost, making them unsuitable for real-time applications. While two-stage proposals work well in controlled environments, with lower computational cost. However, they suffer from a lack of generalization capability, so that their performance degrades when the environment of the recording conditions changes.

In this context, the present work posits a new approach that is robust to changes in the environment, allowing to obtain a good accuracy in different scenarios, but with a lower computational cost than end-to-end approaches. It is a two-stage alternative. In the first stage, it is presented a novel descriptor. Then, it is proposed a DL model for stampede detection from the extracted features. It allows detecting anomalous situations as stampedes in different realistic environments, such as large vessels, without fine-tuning.

As it has been explained before, in this work, the analysis of stampedes is addressed based on macroscopic models, in order to analyze such crowd behavior rather than the individual one, and aiming to generate a safe and robust alarm and warning signal in realistic environments.

Consequently, the first stage involves generating descriptors. For such task optical flow has demonstrated to be really useful. It refers to the study of image motion (O'Donovan, 2005) within a video sequence, and it is widely used in various video surveillance activities, including motion detection (Shafie et al., 2009; Lee et al., 2020; Paredes-Vallés et al., 2019), sparse occlusion (Ayvaci et al., 2012; Liu et al., 2019a; Ren et al., 2020), and event detection (Brosch et al., 2015; Shiba et al., 2022; Lee et al., 2020).

To incorporate the temporal component into the second classification task within the neural networks context, it is proposed the use of RNNs (Hibat-Allah et al., 2020; Almiani et al., 2020; Li et al., 2018), as it has been mentioned in the introduction. RNN architectures vary in complexity, with classical ones being the simplest as they provide output feedback. Gated Recurrent Units (GRU) (Ullah et al., 2021) and Long Short-Term Memory (LSTM) (Sun et al., 2021) are modern architectures that introduce memory capabilities to RNNs and can effectively predict time series with limited periods. Finally, Transformers (Delvigne et al., 2022), which exhibit superior capabilities compared to GRU and LSTM but are computationally expensive, are considered pioneering in signal analysis for similar applications.

All mentioned architectures consider previous moments in a series to predict future moments (Raj et al., 2019) in different fields, such as analyzing stock market values in economic series (Naik and Mohan, 2019) or predicting strokes in healthcare (Chantamit-o pas and Goyal, 2018). Consequently, these networks are usually combined with CNNs to provide spatial and temporal response (Khaki et al., 2020; Zhou et al., 2020a; Nasir et al., 2021; Ajao et al., 2018), yielding improved results compared to models using only CNNs.

Based on this state-of-the-art analysis, the study here included proposes a system capable of extracting entropy from videos through optical flow. Dense optical flow techniques are used to quantify motion within the video. Subsequently, the entropy is vectorized, and an LSTM-based architecture is employed to obtain a classifier that can early detect stampedes in videos. The proposal demonstrates robust performance and outperforms existing approaches in stampede detection. Additionally, as the system is divided into an optical flow generation and a DL classification modules, it facilitates high parallelization and enables its execution on embedded systems or low computational platforms.

3. Proposal for stampede detection

This paper presents a system designed for stampede detection. As stated in the previous section, they have been proposed, in the scientific literature, several methods for detecting groups of people running in hazardous situations. However, this work proposes at the same time, a combination of optical flow and DL techniques to extract relevant information about the danger from a sequence of images.

The objective of the proposed architecture is to develop a system that alerts about dangerous situations in locations susceptible to congestion, such as hallways, hotel lobbies, or other spaces where bottlenecks can easily occur.

In fact, one of the main challenges in stampede detection systems is their dependence on the environment in which they are deployed. This implies that an artificial intelligence (AI) model is generally not applicable to the different environments of interest, and requires network adaptation to obtain accurate results. Additionally, the video-preprocessing system must be adjusted within numerous hyperparameters according to the camera's location in each situation. Therefore, one of the objectives of the proposal is to develop a robust system that is location-independent. To test this, different datasets recorder within diverse environments and camera-configurations were used. Furthermore, the system here proposed is capable of detecting stampedes in small and medium-sized groups of people (between 5 and 50 individuals).

The proposed architecture for stampede detection is depicted in Fig. 1. As it can be seen, it is divided into three main blocks. The first block is responsible for extracting images from the video surveillance system and performing essential preprocessing steps, including resizing, grayscale transformation, and blurring. The size of the original image depends on the dataset used, are downscaled to 320×240 to optimize computational efficiency while preserving key details necessary for analysis.

The second block focuses on analyzing these preprocessed images by generating optical flow magnitudes and angles using the Farneback method (Farnebäck, 2003). From the optical flow magnitude, two essential features are derived: entropy and Temporal Occupancy of Vision (TOV). Entropy captures the disorder within the scene, while

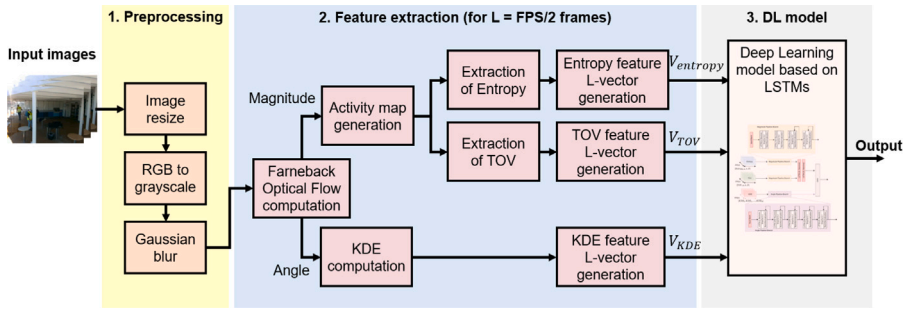


Fig. 1. General block diagram of the proposed system for stampede detection in crowd video sequences.

Fig. 2. Preprocessing. Left: Original image. Right: Preprocessed image.

TOV tracks the temporal dynamics of movement within the space. The optical flow angles are utilized to compute Kernel Density Estimation (KDE), providing insights into the directional distribution of movement. These three core metrics (entropy, TOV, and KDE) serve as the primary indicators for detecting potential stampede events. These indicators are grouped into vectors of size L ($L = FPS/2$), representing 0.5 s of image, which is enough time to detect an anomalous event. The extracted metrics are further represented as activity maps that visualize changes over time, serving as input for anomaly detection.

Finally, the third block integrates these features into a deep learning classification model based on Long Short-Term Memory (LSTM) networks. This model assesses the temporal dependencies across frames, classifying whether a stampede event is occurring. The combination of optical flow features and the LSTM network enables the system to effectively detect and classify abnormal crowd behaviors, such as stampedes. Detailed explanations of the feature computations and the LSTM model design are provided in subsequent sections.

In the following subsections the functionality at each of these blocks is described in detail, as well as some considerations related to their configuration.

3.1. Preprocessing

The initial phase of the system comprises image preprocessing, the single step intricately linked to the video characteristics in the surveillance aim. Within this preprocessing module, three distinct tasks are undertaken. These tasks are visually represented in Fig. 2, which illustrates an example of the input and output images in the preprocessing stage.

Therefore, the first action is to capture the video sequence and resize the images to 320 × 240 pixels. This resize allows reducing the computer effort for optical flow process.

After the resizing, the image is converted to grayscale, leaving only one channel. Given the limited utility of RGB channels for the purpose of motion analysis, their exclusion has negligible effects in the results accuracy, and similar benefits as those stated for the mentioned resize process within this context.

In the final step of preprocessing, a Gaussian blur is applied, in order to focus posterior processes on analyzing the crowd as a whole, and not on specific individuals. By applying blurring, the crowd is viewed as a single unit without distinct individuals.

The output image is then inserted in the next step for optical flow computation.

3.2. Optical flow extraction

Once the images are preprocessed, the next step in the proposed architecture is to extract the video optical flow. In computer vision, optical flow is defined as the measurement of intensity and direction differences between temporary consecutive frames' pixels (Beauchemin and Barron, 1995).

In the related works, it is assumed that the displacement of an object in the image $(\Delta x, \Delta y)$ does not alter its intensity, as stated in Eq. (1). So the goal in optical flow extraction from two consecutive frames Δt is to determine the motion vector $(\Delta x, \Delta y)$ so that:

$$I[x, y, t] \approx I[x + \Delta x, y + \Delta y, t + \Delta t] \tag{1}$$

This is achieved by rewriting the intensity equation using Taylor series as in Eq. (2):

$$\begin{aligned} I[x, y, t] - I[x + \Delta x, y + \Delta y, t + \Delta t] = 0 \\ I'xu + I'yv = -I't, \end{aligned} \tag{2}$$

where $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, and $I'x$, $I'y$ represent the image gradients.

This reformation of the intensity equation is crucial in the field of optical flow, as there is a linear co-dependence between spatial displacements (u and v) and temporal changes ($I't$).

The extraction of optical flow often results in a single equation with two unknown variables $[u, v]$, posing a direct solvability challenge. Consequently, indirect methods such as Lucas and Kanade (1981) and Farnebäck (2003) are commonly employed. In this work, the Gunnar-Farneback method is utilized, which approximates a neighborhood around each pixel with a polynomial to enhance precision, and calculates both the magnitude and direction of optical flow.

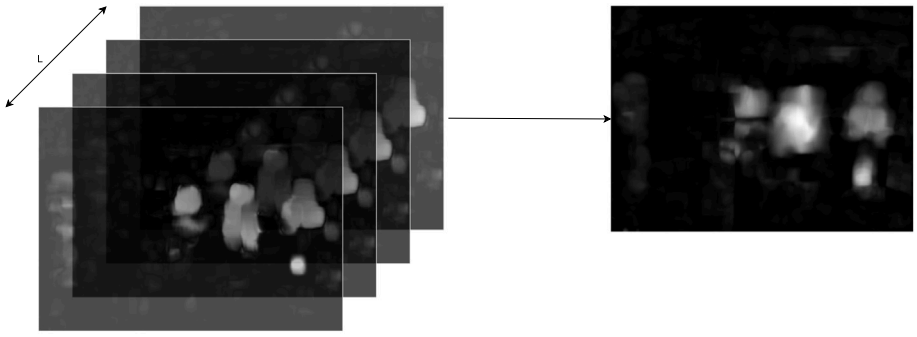


Fig. 3. Example of the generation of activity maps (right) from optical flow magnitude's images (left).

In comparison to Lucas–Kanade, Gunnar-Farneback offers a distinct contextual approach to image analysis. Such method focuses on extracting optical flow from specific pixels, potentially losing information if movement occurs in another section of the image. In response to this limitation, the dense Farneback method was introduced. The new method analyzes both the magnitude and movement between pixels on a pixel-by-pixel basis. Despite incurring higher computational costs, this approach significantly improves precision in the obtained optical flow.

After computing the magnitude and direction of optical flow, it can be obtained vectors' features to be subsequently integrated into the proposed DL model for the early detection of stampedes. Specifically, the optical flow magnitude contributes to the computation of entropy and Temporal Occupancy of Vision (TOV), while the angle-related values are used to derive the Kernel Density Estimation (KDE).

To compute both entropy and TOV, in this work it is proposed as initial step the generation of an *"activity map"* using the obtained magnitude values. This activity map serves as a visual representation of the optical flow magnitude variation within a set of images, taking the form of an image that visually encapsulates movements across a sequence of L frames.

In this work a value of $L = FPS/2$ consecutive frames has being chosen, achieving a 0.5 s frame by frame sliding window for completing the needed activity maps. This value is chosen as has proven to provide enough temporal resolution to detect the onset or end of a sudden movement, such as a stampede. While a larger window could be used, it did not yield improved results, and opting for a smaller window would result in information loss.

Rendered in grayscale, the activity map assigns pixel values within the range of 0 to 255, with higher values indicating more pronounced movements. A concrete example of an activity map is depicted in Fig. 3.

The activity map is obtained by aggregating the magnitude values in intervals of L and then computing the average value for each pixel. Resultant values thus undergo normalization, restricting them within the range of 0 to 1. To achieve normalization that scales according to light contrasts, the division for normalization is performed not by 255 but by the highest pixel value. This approach softens transitions and makes the system more adaptable to different environments.

The activity map $Am(n)$ defined as in previous paragraphs, is formally expressed through Eq. (3), where X_i describes a frame of the collection of $i \in \{n, \ldots, n+L\}$ ones, containing the magnitude values derived from the optical flow process. Moreover, each X_i frame in Eq. (3) is given from the optical flow in a temporal window encompassing L frames. To enhance the obtained result and mitigate the effects of salt and pepper noise, a median filter is applied to homogenize the activity map.

$$Am(n) = median\left(\frac{\frac{1}{n}\sum_{i=1}^{n} X_i}{max(\frac{1}{n}\sum_{i=1}^{n} X_i)}\right) \tag{3}$$

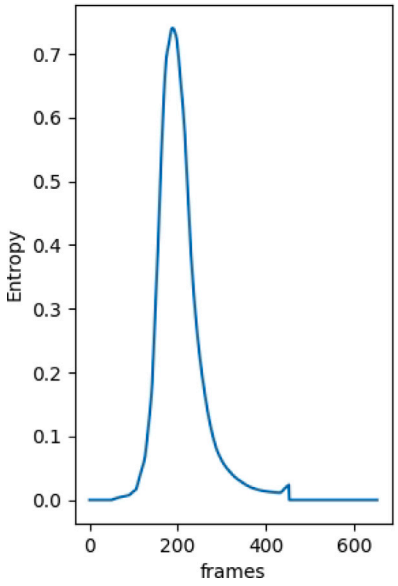


Fig. 4. Example of entropy signal corresponding to a stampede.

3.3. Feature vector generation

Once angle values from optical flow and the activity map $Am(n)$ derived from its magnitude values are obtained, the next step involves extracting vectors for their later use in the DL model that effectively detects the stampede. Therefore, the following two subsections focus on feature extraction from both magnitude and angle vectors.

3.3.1. Magnitude related features

Upon obtaining the activity map $Am(n)$, which visually encapsulates the image's motion patterns, the following phase involves codifying the magnitude information there included via entropy and TOV.

Entropy can be used to measure the basic characteristics of motion: smooth and regular motions result in lower entropy values, while sudden and fast movements lead to higher ones. This effect is exemplified in Fig. 4, with the x-axis showing the frame count and the y-axis showing the entropy value. As it can be seen, it appears a clear peak in the last frames indicating the occurrence of a stampede event, against the firsts parts of the graph that, with consistent entropy value, suggest regular and continuous motion.

In the entropy computation process, a histogram of the activity map $Am(n)$ is initially generated, considering values greater than 0. The total count of these nonzero values is denoted as m. Subsequently, each pixel value in the activity map is normalized by dividing it by its size ($dims = W \times H$). This normalization results in a single component

Fig. 5. Example of the TOV (right) generated from original image (left).

value, represented as p_i for each pixel i in $Am(n)$. The pseudocode corresponding to the algorithm that computes the entropy is shown below in Algorithm 1.

Algorithm 1 Pseudocode to compute the entropy value

```
function Entropy(Am(n))
    Input: Am(n) values as uint8
    counts, bins ← ComputeHistogram(Am(n), range(256))
    pos ← Where(counts > 0)
    p_pos ← counts_pos / dims
    Entropy(n) ← −∑_i p_i × log2(p_i)
    Return Entropy(n)
end function
```

Following this, an additional operation is executed across each activity map $Am(n)$ for these normalized values, each of them multiplied by the base-two logarithm of itself, as expressed in Eq. (4).

$$Entropy(n) = \sum_{i=1}^{n} p_i log_2(p_i) \tag{4}$$

Another crucial parameter in creating the magnitude features' vectors, used for the stampede detection proposal, is the TOV. This parameter is also derived from the activity map and captures the amount of motion from one frame to the next one, similarly to entropy. To compute it, the difference between the current activity map $Am(n)$ and the previous one $Am(n-1)$ is determined (as show Fig. 5).

TOV is defined in Eqs. (5) and (6). Its computation involves pixel-wise determination of the difference between the current activity map $Am(n)$ and the prior one $Am(n-1)$. This assessment is conducted pixel by pixel, with $u = 1, \ldots, W$ representing pixels along the x-axis and $v = 1, \ldots, H$ those one along the y-axis. In cases where the pixel discrepancy yields negative values it is set to 0.

$$TOV_{image}(n)[u,v] = \begin{cases} Am(n)[u,v] - Am(n-1)[u,v] & \text{if} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This sequence of operations yields a representative value indicating the proportion of motion between successive frames. Such a measurement effectively captures the temporal intricacies in crowd behavior dynamics.

After obtaining the image of TOV, the next step is to extract a single numerical value representing it. To achieve this, the positive values of $TOV_{image}(n)$ are summed up and then normalized by the image size with $dims$ value, as shown in Eq. (6).

$$TOV(n) = \frac{\sum_{i=0}^{u} \sum_{j=0}^{v} TOV_{image}(n)[u,v]}{dims} \tag{6}$$

To obtain additional information about entropy and TOV, a set of their features are compiled to form input vectors for the DL model. Specifically, three values have been considered: mean (μ), standard deviation (σ), and the difference between them (D). Eq. (7) illustrates the structure of the magnitude vectors for both entropy ($V_{entropy}$) and TOV (V_{TOV}).

$$\begin{aligned} V_{\text{entropy}}(n) &= [Entropy(n), \mu_{\text{entropy}}(n), \sigma_{\text{entropy}}(n), D_{\text{entropy}}(n)] \\ V_{\text{TOV}}(n) &= [TOV(n), \mu_{\text{TOV}}(n), \sigma_{\text{TOV}}(n), D_{\text{TOV}}(n)] \end{aligned} \tag{7}$$

The last parameter in the two vectors ($D_{entropy}$ and D_{TOV}) is derived from the combination of the mean (μ) and standard deviation (σ): when we represent these values they exhibit distinct oscillations, and by adding and subtracting them the distance signal ($D_{entropy}$ and D_{TOV}) is generated. This distance is completed as a vector with two values: the first being its maximum point, and the second with its minimum, as outlined in Eq. (8).

$$D_j(n) = [\mu_j(n) + \sigma_j(n), \mu_j(n) - \sigma_j(n)] \tag{8}$$

An example of entropy and TOV values, as well as their features, are presented in Fig. 6. Entropy is shown in the top left graph, and TOV in the bottom left one. On the right side of the image, the values of mean μ, standard deviation σ, and the previously defined distance signal D, for both entropy and TOV, are depicted. μ is represented in dark blue, σ in yellow, and the maximum and minimum of D in dark green. Additionally, D is visualized by shading the area between maximum and minimum with a gray shade. These values are computed using a sliding window of size L, with a stride of one.

This set of features enables the DL model to effectively detect stampedes by capturing various temporal patterns and movement behaviors in the crowd.

3.3.2. Angle related features

After extracting the features related to the optical flow magnitude values, the next step involves computing the ones related to its angle values.

Various methodologies are explored to transform the angle matrix from the optical flow into a vectorial representation. Initially, flattening the matrix was considered, but this resulted in a vector with dimensions of 320 × 240, rendering it excessively long to be then inserted into the DL architecture. As a result, an alternative approach is adopted, involving the probability density of the angles in the matrix.

In order to estimate this probability density, a technique rooted in KDE is employed. This method constitutes a parametric approach that facilitates the estimation of that function. The mathematical formulation is described by Eq. (9), wherein $Ang(n)$ denotes the angles from the optical flow, K embodies the different kernel functions that serve as smoothing estimators, and l signifies the overall set of discrete angle's values extracted from the optical flow. To reduce the number of features introduced to the DL network, it has been empirically tested that $l = 50$ angular sectors are enough to provide the necessary information about the crowd movement.

$$KDE(n) = \frac{1}{l} \sum_{i=1}^{l} K\left(Ang(n) - Ang(n)_i\right) \tag{9}$$

This angle related features' vector encapsulates the distribution of movement directions within the image, resulting in a succinct representation manageable by the AI architecture utilized in the stampede detector proposed.

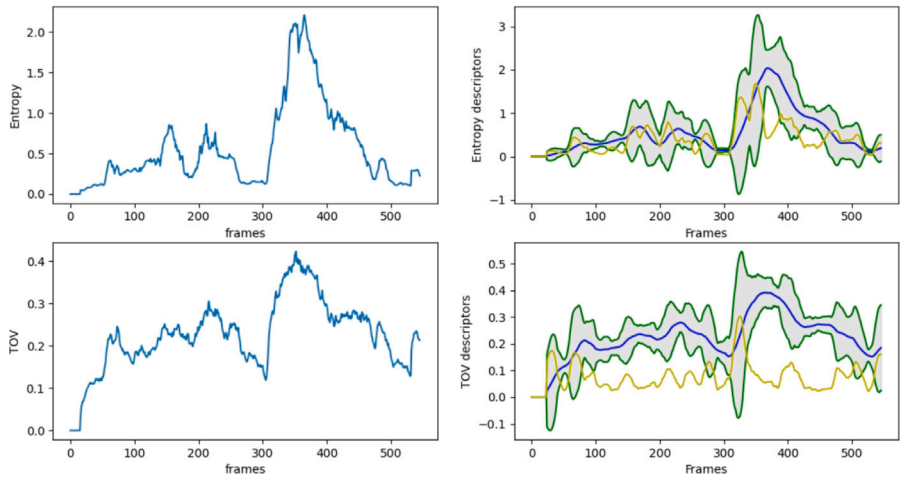
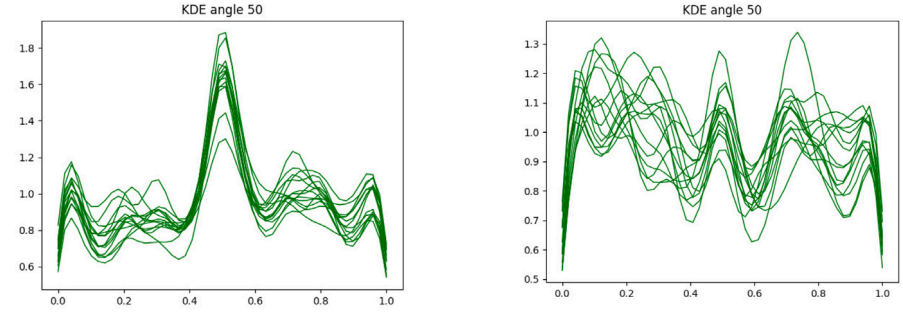


Fig. 6. Example of extracted features: $\mu_j(n)$ (blue), $\sigma_j(n)$ (dark green) and $D_j(n)$ (light green) descriptors (right column) from entropy (upper images) and TOV (bottom images) (left column).



(a) Probability density of a normal situation. (b) Probability density of a stampede.

Fig. 7. Example of the probability density extracted through a KDE.

Finally, these KDE vectors, each comprising 50 components, are grouped into sets of L time steps (as also done with entropy and TOV) to be introduced into the DL model. This allows for the analysis of temporal variations in this feature. Eq. (10) shows the format of this input vector.

$$V_{KDE} = \begin{bmatrix} KDE(n)_0 & KDE(n)_1 & \cdots & KDE(n)_{49} \\ KDE(n+1)_0 & KDE(n+1)_1 & \cdots & KDE(n+1)_{49} \\ \vdots & & & \vdots \\ KDE(n+L-1)_0 & KDE(n+L-1)_1 & \cdots & KDE(n+L-1)_{49} \end{bmatrix} \tag{10}$$

In Fig. 7, two instances of this probability distribution derived from angle data are presented. In these figures, the corresponding normalized vectors with a set of $l = 50$ kernels and grouped into slots of L vectors.

Fig. 7(a) shows the mentioned probability distribution recorded scene without a stampede. In this image is shown the L KDE-vectors grouped together, it can be noticed that movements' directions are more similar, appearing only changes with low probability, and just one angularly synchronized for the whole movement in the scene.

On the other hand, Fig. 7(b) displays a significant difference between the different L KDE-vectors, suggesting large changes of direction and movement from one frame to the next.

3.4. Deep learning model for stampede detection

After obtaining the signals from the optical flow magnitude and angle values, and grouped into vectors, they are ready to be processed by the DL model, as shown in Fig. 1. The main goal of the model designed, depicted in Fig. 8, is to make easier identifying temporal patterns that, at its time, help in classifying stampede events at its output.

Therefore, the model has three separate branches to analyze the temporal patterns of each of the features' vectors previously obtained. Moreover, to determine the architecture final size, an ablation study was conducted (presented in Section 4), so this section just presents the final architecture.

The branches for TOV and entropy are designed equally because they have similar input data types, thus having both the same number of LSTM layers. It has to be pointed out that we have selected LSTMs layers due to their good analysis of time sequences. This selection enables tracking optical flow magnitude changes and pulling out important patterns about the crowd behavior there represented. In the architecture here proposed this branch has 4 LSTM layers: the first two are unidirectional to perform an initial time analysis reorganizing the input vector values; the next two layers are bidirectional, giving a deeper look at time elements here inserted. Both magnitude LSTM-based branches come together in two additional final LSTM layers, to pull out shared time features from TOV and entropy. In any case, each LSTM layer has $2L$ neurons as this option empirically showed to give a detailed analysis of the input features related to entropy and TOV.

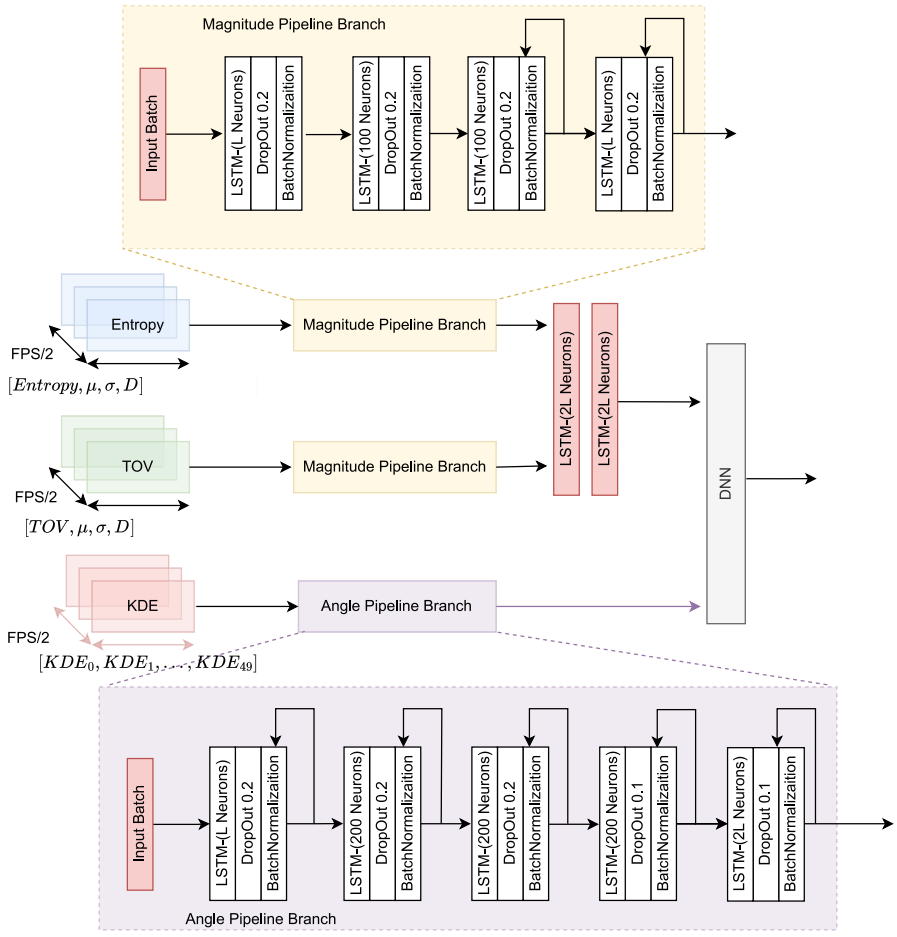


Fig. 8. Proposed DL architecture with three branches, for stampede detection.

The third branch looks at optical flow angle, comprised in KDE vector. Since the vector from KDE has more components than those for entropy and TOV, this network needs to be bigger: it has 5 bidirectional LSTM layers to pull out all information from the optical flow angle values represented in KDE vector.

At the end, the branches from TOV and entropy and the one from the angle come together into a final dense network, that has 4 dense layers with $4L$, 100, 50, 20, 1 neurons, respectively, and use *tanh* activation function. Every layer in the design has a 20% dropout feature, added to prevent overfitting. There is also a batch normalization layer after each dropout step, to normalize each layer inputs and aid in accelerating training, by reducing dependence on weights initialization.

The global architecture was tuned with Adam optimizer (Kingma and Ba, 2014) due to two primary considerations. Firstly, Adam aligns with the recurrent nature of LSTM networks, repeatedly adjusting the network weights to minimize the loss function. Secondly, the system's intended deployment on embedded platforms necessitates stringent computational efficiency. In this context, Adam, as a first-order gradient stochastic optimization algorithm, offers an optimal solution. This characteristic not only minimizes its computational burden but also ensures accuracy in loss function optimization, setting it apart from alternatives requiring higher computational costs or delivering less precise optimization outcomes.

The training of the final model encompasses specific hyperparameters: a learning rate set at 5×10^{-6}, with exponential decays of 0.9999 for both β_1 and β_2, a decay of 1×10^{-6}, and a total of 50 000 epochs. The batch size, an important factor in training efficiency, is configured at 50. All these values were determined through empirical experimentation to strike a balance between accuracy and avoid overfitting.

Each layer within the architecture is trained with a loss function based on binary cross-entropy, which can be mathematically defined as shown in Eq. (11). Here, y represents the true label (0 or 1) of each train instance, and $\hat{y}$ denotes the model's prediction.

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{11}$$

Given that there are 3 parallel LSTM-based architectures, their global loss function can be expressed as Eq. (12).

$$L_{\text{total}} = L_{TOV}(y, \hat{y}) + L_{Entropy}(y, \hat{y}) + L_{KDE}(y, \hat{y}) \tag{12}$$

It is worth highlighting that, Section 4 includes an ablation study that carefully examines the effects of the proposed architecture and its hyperparameter choices, on the success of stampede detection. This study thoroughly investigates changes in LSTM layer setups, sizes of dense layers, dropout percentages, and training elements. This detailed review of design decisions and training methods is a key aspect for enhancing the system's overall efficiency.

4. Experimental results

In this section, we present the methodologies and results that assess the feasibility of the system proposed. We provide a detailed description of the experimental setup, aiming to promote experiment repeatability. The section also explains and justifies the use of different datasets selected to evaluate the architecture, including specific details about their characteristics. Lastly, an ablative study of the architecture within these different datasets is conducted, comparing the obtained results with other given by similar state-of-the-art methods in terms of precision and computational cost.

Fig. 9. Different scenes corresponding to the three scenarios in UMN.

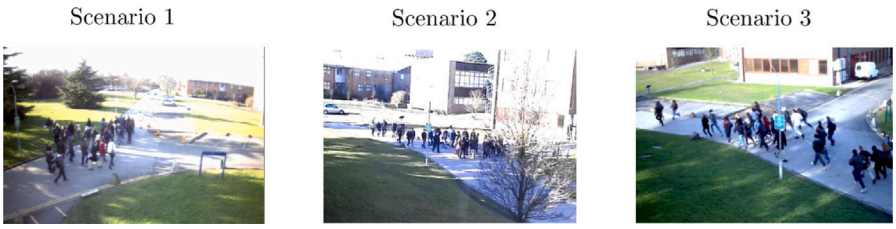


Fig. 10. Example images corresponding to the three PETS 2009 scenarios.

4.1. Experimental set-up

The experiments were conducted on a PC equipped with the following hardware specifications: an Intel Core i7-9700K Central Processing Unit (CPU) @ 3.60 GHz, 32 GB of RAM, an NVidia GeForce RTX 2080 Ti Graphics processing unit (GPU), and a 500 GB SSD ROM memory. As for the software environment, a Unix operating system with Ubuntu 20.04.3 LTS distribution was chosen. CUDA version 12.0 was installed to optimize and utilize the integrated GPU on the PC, along with OpenCV version 4.5.1.

4.2. Datasets

One of the main challenges in stampede detection pertains to the limitations of available datasets. State-of-the-art works often rely on the same datasets, which consist of a very limited number of videos. Moreover, they are not specifically tailored for stampede detection and typically lack of proper annotations, making it challenging to extract meaningful metrics. Furthermore, the nature of stampedes can vary significantly based on camera distance and the number of individuals within the crowd.

As highlighted in Section 1, this work proposes a system capable of detecting stampedes by analyzing crowds of 3 to 50 people. However, this also means that the number of suitable datasets for evaluation is limited. UMN (Mehran et al., 2009), and PETS 2009 (Ferryman and Shahrokni, 2009) are among the most well-known in this domain, encompassing individuals in controlled environments with varying camera positions and scenarios. These datasets were used as comparative ones with the state-of-the-art and to conduct the ablation study of the proposed system.

Due to the scarcity of appropriate datasets, two additional ones were constructed to evaluate the system's performance in real-world scenarios. The first, GBA Stampedes, is an extension of GBA2016 (Baptista-Ríos et al., 2016), comprising recordings of stampedes occurring in the hall of the Polytechnic School, at UAH. The second one, GSMADC, was recorded on a ferry, simulating an environment where accurate and robust stampede detection is crucial to mitigate potential risks.

For all datasets, ground truth annotations were manually adjusted per frame, indicating the exact onset of a stampede. To precisely determine this moment, it was empirically stated that when approximately 20% of individuals begin to run, it triggers a chain reaction causing the rest of the crowd to flee, thus marking the beginning of the stampede. Regarding the definition of the end of a stampede, two conditions were established: either no one remains in the scene, or more than half of the crowd members slow down or come to a stop.

The main characteristics of all mentioned datasets are explained below.

- **UMN**: is a comprehensive collection created and recorded in the University of Minnesota, showcasing diverse crowd behaviors in both indoor and outdoor environments. This dataset consists of a total of 11 videos captured in 3 distinct scenarios, with two of them being outdoors and the third one indoors. Fig. 9 illustrates scenes corresponding to each of the three scenarios provided in UMN. Overall, the dataset comprises 7502 frames with a resolution of 240 × 320 at 30 frames per second (FPS).
- **PETS 2009**: this dataset has been specifically curated for the development and evaluation of the stampede detection algorithm here proposed. It was created for the PETS 2009 Workshop at the Whiteknights Campus, University of Reading. PETS 2009, and comprises multi-sensory sequences containing crowd scenarios with an increasing level of scene complexity. The dataset is thus divided into three subsets (Fig. 10): S1, focused on person count and density estimation; S2, dedicated to people tracking; and S3, centered around crowd flow analysis and abnormal event detection.
  For our experiments, we have specifically utilized the S3 subset of the PETS 2009 for stampede detection. This subset is characterized by presenting crowd situations with a growing scene complexity, which is crucial for evaluating the efficacy of our algorithm in realistic and challenging environments.
- **GSMADC**: this dataset was recorded at Santander shipyards as part of a collaboration in the European project PALAEMON (Palaemon-H2020, 2023). Its key feature lies in the meticulous recording and annotation for Human Activity Recognition (HAR) and Crowd Activity Recognition (CAR) in real-world settings, encompassing multiple individuals engaging in diverse actions within the scene. The dataset goes beyond focusing on a single individual performing a singular action throughout the sequence. This initial version of the dataset comprises a total of 37 sequences, with an average duration of 40 s per video. All videos exhibit a resolution of 1080 × 980 pixels and a frame rate of 30 FPS. It is worth noting that the dataset includes both color and infrared images, as an Intel D435i camera was utilized for the recordings, but only RGB videos are used in this work.

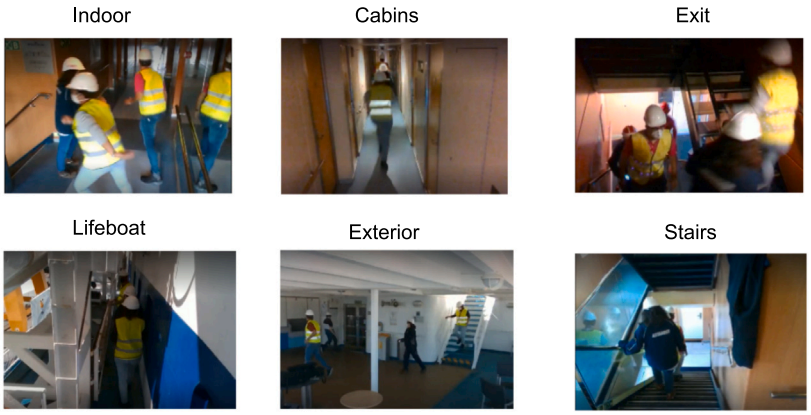


Fig. 11. Sample images of the six scenarios from the GSMADC.

Table 1
Summary of the main characteristics of analyzed datasets.

Dataset	Scenario	Resolution	Illumination	#people	#frames	#videos	FPS
UMN	Lawn	240 × 320	Variable	15	1950	3	30
	Plaza			12	1980	3	
	Indoor		Constant	18	3420	5	
PETS 2009	Scenario-1	576 × 768	Variable	41	864	4	30
	Scenario-2			41	1300	4	
	Scenario-3			42	1216	4	
GBA Stampedes	Indoor	1920 × 1080	Variable	15	9300	12	60
GSMADC	Indoor	1080 × 980	Constant	4	8628	10	30
	Cabins			4	6874	5	
	Exit		Variable	6	2739	5	
	Exterior			4	7413	6	
	Lifeboat			4	2336	5	
	Stairs			4	5854	6	

The videos were acquired from two perspectives: individual and group levels. The individual level showcases the position and actions taken by each person, while the group level indicates whether the crowd is in a state of calm or running. For this study, the focus is on the information related to crowd behaviors.

The videos feature crowds with an equal distribution of men and women. The number of individuals in each video varies, ranging from 3 to 12 people. The recorded scenarios on the ferry (Fig. 11) (indoor, cabins, exit, exterior, lifeboat, stairs) are characterized by complexity in terms of action, number of people, overlapping, and varying brightness. As a result, this dataset serves as a valuable resource for evaluating algorithms in real-world environments.

All the sequences in the dataset are labeled, using two formats: `.XML` and plain text. It is worth highlighting that this dataset has been made available to the scientific community (Cob-Parro et al., 2023b).

- **GBA Stampedes**: as explained in Cob-Parro et al. (2023a), GBA Stampedes is a HAR-oriented dataset with large crowds of people performing different actions. For this paper, we proposed an extension of the original dataset where we recorded groups of people (between 10 and 30 individuals) recreating stampedes in common areas of the Polytechnic School at UAH (Fig. 12). This data consists of 12 new videos that have been recorded using a GoPro camera with a resolution of 1920 × 1080 and a frame rate of 60 FPS. Additionally, we need to highlight that the acquired information is in RGB format, and the camera optics has a fish-eye effect.

  The variety of individuals in the images is 50% men and 50% women of different sizes and complexity. Due to the nature of the recording location, there are spontaneous situations that have been labeled in the ground truth of the dataset. The dataset is characterized by constant illumination, and has been labeled in both `.xml` and plain text formats.

  The diversity of individuals and consistent lighting make this dataset a valuable resource for the development and evaluation of stampede detection algorithms in realistic and challenging environments.

Table 1 includes a summary of the main characteristics of all datasets used in this work and described in this section. The table includes the number of classes, number of actors, FPS, resolution and number of sequences.

4.3. Performance for stampede detection

To evaluate the system performance, we have analyzed two key parameters: the first one is the accuracy to detect stampedes, and the second one the system computational cost.

The effectiveness of the DL model for stampede detection has been evaluated using precision (*Pre*), accuracy (*Acc*), and recall (*Rec*) metrics, based on the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Precision (*Pre*) assesses the model's ability to avoid FP, accuracy (*Acc*) provides an overall measure of correctness, and recall (*Rec*) gauges the model's capability to identify actual stampede events. This comprehensive evaluation ensures a thorough assessment of the model's performance and its suitability for real-world stampede detection scenarios.

To validate such DL-based proposal, it was also sought to perform a comparison among different architectural possibilities based on non-parametric Bayesian tests, typical to assess different models in the field of Machine Learning. The Wilcoxon signed-rank (Benavoli et al., 2017) test is employed from a Bayesian perspective, to evaluate whether one model is statistically superior to another, or if both will probably fall within a region of practical equivalence, concerning some measure of performance. This approach provides a valuable tool for making

Fig. 12. GBA Stampedes scenario of the Alcalá University hall during a stampede.

informed decisions in model selection and comparison within Machine Learning problems, and thus, it is here used.

4.4. Ablation study

In this section, an ablation study of the DL-based architecture has been conducted. This phase aims to investigate the number of layers comprising the architecture and ascertain the significance of each of the three constituent branches. Through this analysis, the goal is to comprehend how the presence and configuration of these branches impact the overall model performance. Experiments have been carried out by selectively removing or modifying certain layers or components of the architecture to assess their influence on precision and system performance. The ultimate objective is to identify the most optimal and meaningful configurations for the DL model design.

As depicted in Fig. 8, the first two branches corresponding to TOV and entropy analysis are twin branches. For this purpose, each branch was trained with an increasing number of LSTM layers until the minimum number of layers that achieved maximum accuracy was determined. This analysis was performed exclusively using the UMN and PETS 2009, with 60% of the videos used for training the networks and 40% for conducting the tests.

Table 2 shows the results of the analysis to examine how adding more layers to each of the branches affects the metrics. For each used metric, there are shown both, the metric value in percentage, and its relative increase, denoted as Δ, computed as the difference between the value obtained with the current number of layers and that achieved with one less,divided by the value of the previous number of layers. The analysis starts including just 1 layer and progresses up to having 6 layers.

In this analysis, the first and last layers are sized L, while the intermediate ones have a larger size, specifically 100 neurons. This value, empirically set, has been chosen to be sufficiently large for extracting input features but not so large as to require a heavy computational cost.

It is observed that as the complexity of the network increases, the metrics also show improvement. This trend continues until the 4th layer, beyond which the metric values do not exhibit significant further improvement. The rationale behind studying a relatively low number of layers is driven by the limited dataset availability. It is crucial to maintain a compact network to ensure stable training, as increasing the layers with a small dataset runs the risk of overfitting and exponential computational costs. Therefore, the design decision is to employ 4 layers of LSTMs with uniform depth, both for TOV, achieving an accuracy of 99.36%, and for entropy, attaining a precision of 99.75%. This choice aims to strike a balance between model performance and computational efficiency. It can be observed that the improvements plateau when adding layers, starting from the fourth layer. This is why the utilized branch consists of 4 layers.

Table 2
Layer analysis of TOV and entropy branches. All metrics are in percentages (%).

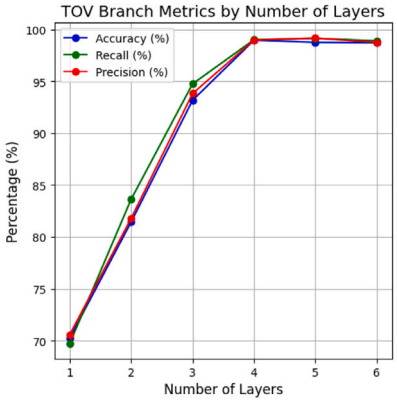
Number of	Branch-TOV					
layers	*Acc*	(ΔAcc)	*Recc*	(ΔRec)	*Pre*	(ΔPre)
1	70.23	(–)	69.71	(–)	70.57	(–)
2	81.43	(+15.95)	83.62	(+19.95)	81.78	(+15.88)
3	93.17	(+14.42)	94.75	(+13.31)	93.82	(+14.72)
4	**98.96**	(+6.21)	**99.01**	(+4.50)	**98.98**	(+5.50)
5	98.76	(−0.20)	99.15	(+0.14)	99.17	(+0.19)
6	98.72	(−0.04)	98.91	(−0.24)	98.75	(−0.42)
Number of	Branch-entropy					
layers	*Acc*	(ΔAcc)	*Rec*	(ΔRec)	*Pre*	(ΔPre)
1	69.82	–	68.74	–	70.69	–
2	81.11	(+16.17)	82.39	(+19.86)	82.56	(+16.79)
3	93.23	(+14.94)	93.61	(+13.62)	94.14	(+14.03)
4	**99.17**	(+6.37)	**99.10**	(+5.86)	**98.71**	(+4.85)
5	99.12	(−0.05)	99.27	(+0.17)	99.23	(+0.53)
6	99.21	(+0.09)	99.97	(+0.71)	99.05	(−0.18)

Due to the substantial disparity in information extracted from the optical flow angle between TOV and entropy, a segregated analysis has been conducted within the framework of the DL architecture. The investigation culminates in Table 3, which presents a meticulous per-layer examination to discern the optimal quantity of layers constituting the KDE segment and the improvement when adding extra layers. This improvement is calculated by subtracting the value of the current layer from the previous one and dividing by the number of layers in the current configuration.

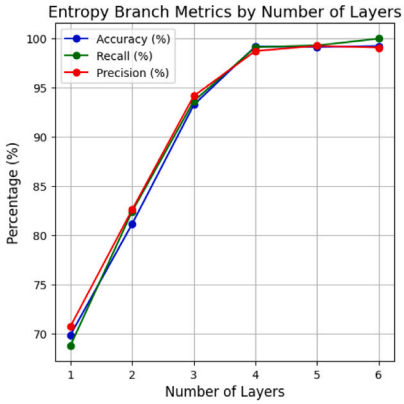
In configuring this architecture, the initial layer consistently maintains a dimension of L, the terminal layer assumes $2L$, while the intermediary layers are characterized by a size of 200 neurons. This approach aims again to maximize information extraction while minimizing computational costs.

In the context of optimizing the architecture's size for computational efficiency and robustness, again the number of layers can vary, but an optimal configuration of 5 has been identified, as exceeding this threshold does not yield significant improvements, as shown in Table 3: it can be observed that improvements show substantial gains when the number of layers is still low, but after adding the fifth layer, no further improvement is apparent.

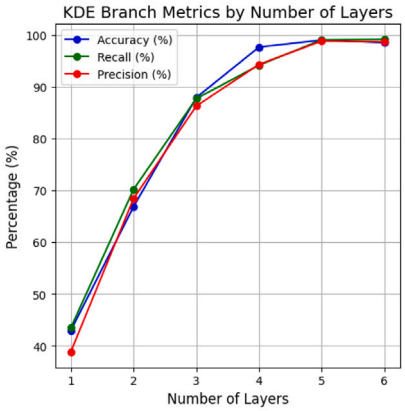
As a result, precision notable reaches at 98%, although slightly below TOV or entropy benchmarks, it remains substantial. Furthermore, the integration of this third branch enhances the architecture's adaptability across diverse datasets and scenarios.



(a) TOV Branch Metrics



(b) Entropy Branch Metrics



(c) KDE Branch Metrics

Fig. 13. Comparison of the effect of the number of layers in the different metrics, for each of the branches of the proposal.

Table 3
Layer analysis of KDE branch. All metrics are presented as percentages (%).

Number of	Branch-KDE					
layers	*Acc*	(*ΔAcc*)	*Rec*	(*ΔRec*)	*Pre*	(*ΔPre*)
1	42.91	–	43.44	–	38.81	–
2	66.82	(+55.72)	70.16	(+61.51)	68.39	(+76.22)
3	87.88	(+31.52)	87.67	(+24.96)	86.25	(+26.11)
4	97.61	(+11.07)	94.12	(+7.36)	94.27	(+9.30)
5	**98.97**	(+1.39)	**99.01**	(+5.20)	**98.79**	(+4.79)
6	98.47	(−0.51)	99.10	(−0.09)	98.61	(−0.18)

Fig. 13 shows how precision, accuracy, and recall metrics improve as more layers are added to the TOV 13(a), Entropy 13(b), and KDE 13(c) branches respectively.

As it can be seen in Fig. 13, up to the fourth layer, all branches see significant improvements, with TOV and Entropy reaching nearly 100%. After the fourth layer, the improvements stabilize or slightly decrease. In the KDE branch, although it starts with lower values, it achieves 98.97% precision by the fifth layer, confirming that five layers provide an optimal balance between performance and computational efficiency.

After analyzing each branch separately and optimizing metrics there, the next step performed involves examining how well the final architecture performs across different datasets. Two experiments were conducted: the first focused on training and testing within a single dataset, while the second involved initial training on UMN and PETS 2009 followed by fine-tuning with a 10% subset of GBA Stampedes and GSMADC. This latter experiment aimed to highlight the architecture's capacity for generalization.

Each branch and their combinations were thus systematically examined, as detailed in Table 4. The experiments involved training each branch with distinct datasets, extracting relevant metrics for stampede detection. Due to the limited number of frames in UMN and PETS 2009, both branches were jointly trained with 60% of sequences shared for training and 40% for evaluation. For GBA Stampedes and GSMADC datasets, a distribution of 80% for training and 20% for test was adopted, due to their larger number of frames and training complexity.

The obtained results are shown in Table 4, where the last two rows correspond to the combination of Entropy+TOV (ET), and Entropy+TOV+KDE proposal (ETK) respectively.

The results, outlined in Table 4, showcase the UMN + PETS 2009 datasets' strong performance, achieving an accuracy exceeding 95% across all architecture branches. Metrics indicate their decrease when dealing with more complex datasets like GBA Stampedes and GSMADC. Additionally, combining all three branches (with ETK proposal) escalates architectural complexity, resulting in accuracy reaching 95% in any dataset. Notably, the combination of Entropy + TOV (ET proposal) produces values comparable to the complete ETK architecture in the

Table 4
Study of the proposals within each of the analyzed datasets. All metrics are presented as percentages (%). Best values in bold.

Proposal	UMN + PETS 2009			GBA Stampedes			GSMADC		
	Acc	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*
Entropy	99.43	98.65	99.32	96.07	96.82	96.31	86.04	83.37	82.22
TOV	99.05	99.37	98.63	95.35	95.77	95.83	82.11	80.92	80.27
KDE	98.71	99.05	99.17	96.21	97.57	**98.55**	91.63	91.05	92.77
ET	**99.50**	**99.42**	98.95	98.00	97.11	96.72	90.23	88.65	86.86
ETK	99.36	99.19	**99.58**	**98.28**	**97.66**	97.39	**94.84**	**95.77**	**93.86**

Table 5
Study about the generalization ability of stampede detection proposals. All metrics are shown as percentages (%). Best values en bold.

Proposal	UMN + PETS 2009			GBA Stampedes			GSMADC		
	Acc	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*
Entropy	99.12	99.20	99.09	86.17	96.42	95.91	70.54	73.37	72.09
TOV	99.01	**99.33**	98.88	85.61	95.12	95.83	72.33	71.27	70.48
KDE	98.56	99.00	99.14	83.47	91.68	92.62	74.84	71.94	74.18
ET	99.51	90.88	99.23	90.23	**97.32**	**96.89**	79.93	80.35	79.56
ETK	**99.14**	99.30	**99.72**	**93.95**	95.66	93.38	**91.74**	**90.64**	**90.35**

Table 6
Computational cost analysis of different proposals according the DL model used.

	Time (ms)	Accuracy (%)	MFLOPS
Networks with **RNN**	0.11	71.12	1.10
Network with **GRU**	0.18	82.15	2.08
Network with **LSTM**	0.24	99.36	2.58

GBA Stampedes, with metrics exceeding the 95% threshold. However, when analyzed individually, TOV, entropy, and even KDE branches show relatively subdued performance within the GSMADC, yielding metrics below the 90% mark.

A thorough evaluation of the architecture's ability to generalize has been undertaken, as depicted in Table 5. In this assessment, half of the frames from UMN and PETS 2009, along with a 10% subset from GBA Stampedes and GSMADC datasets, were employed to train the model, and the evaluation is then carried out in all the datasets without fine-tuning. This rigorous analysis seeks to assess the architecture's effectiveness in extrapolating results beyond the training datasets.

As indicated by Table 5, a decline in metrics is observed for the GBA Stampedes and GSMADC, whereas metrics for UMN and PETS 2009 remain relatively stable. This performance variation can be ascribed to the substantial portion of the training set used from UMN and PETS 2009, with only 10% of GBA Stampedes and GSMADC being utilized for such training. This distinct shift is particularly noticeable within the GSMADC, where individual branches obtain results under 80% in accuracy. The KDE branch, however, stands as an exception, exhibiting metrics not significantly differing from those in Table 4.

It is noteworthy that the complete architecture attains 90% accuracy on the GSMADC. This underscores the architecture's capacity to generalize training by incorporating all optical flow features (magnitude and angle). This further validates that the proposed architecture can extend its applicability to other datasets, having undergone training on a diverse range of them.

To conclude this study, the computational cost of the proposed DL architecture for stampede detection has been analyzed in the Table 6. To achieve such analysis, various types of recurrent networks have been examined, replacing the previously shown LSTMs with GRU and RNN cells. The processing time per frame, accuracy, and the number of FLOPs for each architecture have been evaluated. The datasets utilized include UMN and PETS 2009, where all frames were collected, and the average value per frame was computed to extract the execution time for each architecture.

The challenges associated with studying stampede events, as highlighted in the introduction, primarily arise from two key factors. Firstly, these occurrences often take place within video surveillance systems, where computational power is constrained, posing significant challenges due to the limitations of hardware in such systems. Secondly, the scarcity of labeled datasets specifically representing stampede situations adds to the complexity.

Initially, there was consideration for Transformer-based architectures to tackle these challenges. However, such models require a substantial amount of data for optimal functioning and entail high computational costs. Even with the use of lightweight Transformers, as discussed in Ek et al. (2022) where the lightest model consumes 15 MFLOPS, the computational burden remains substantial.

Architectures upon RNNs and GRUs offer efficient execution times of 0.11 ms and 0.18 ms, respectively. RNNs consume 1.1 MFLOPS, while GRUs require 2.08 MFLOPS. However, their precision levels do not surpass the 90% threshold, unlike LSTMs.

Ultimately, the LSTM-based architecture strikes a commendable balance between computational load and performance. This architecture demands 2.58 MFLOPS, significantly less than the 15 MFLOPS required by Transformers, while maintaining high precision and an acceptable processing time of 0.24 ms.

4.5. Computational cost analysis

To evaluate the computational cost of the complete system, two types of experiments were carried out: one using a GPU and the other one using a CPU. These experiments are consistent with the previous analysis of the optimal RNN layer type, utilizing the UMN and PETS datasets. The UMN dataset consists of a total of 7502 frames with a resolution of 240 × 320 at 30 fps, divided into three scenarios: two outdoor and one indoor. The PETS dataset, on the other hand, includes a single outdoor scenario with 2876 frames and a resolution of 768 × 576 at 30 fps.

To better explain the computational cost, two detailed analyses were performed, measuring the two key components of the system: the optical flow (OF) block and the DL block. In both experiments, the DL model was executed on the system's GPU, while the OF one was executed both on the GPU and the CPU for comparison purposes.

The reason for executing the OF block on the GPU is that it is computationally more expensive compared to other methods, such as Lucas–Kanade. However, with current hardware advances, it is possible to execute these systems in real-time, which allows for more robust results compared to previous works, such as the study by Pennisi, which will be discussed in the system metrics section.

In order to optimize the optical flow analysis, we aimed to boost the execution speed by utilizing the GPU. For this, OpenCV2 was employed, which includes libraries for communicating with CUDA (Compute Unified Device Architecture), a platform used to interface

Table 7
CPU and GPU times (in ms) in PETS and UMN datasets.

Datasets		OF over CPU			OF over GPU		
		OF block	DL block	Total	OF block	DL block	Total
UMN	Lawn	18.34	0.25	18.59	4.01	0.48	4.40
	Indoor	17.21	0.22	17.43	3.54	0.39	3.89
	Plaza	18.17	0.19	18.36	3.98	0.38	4.28
PETS 2009	video-1	20.23	0.26	20.49	4.65	0.60	5.17
	video-2	21.88	0.23	22.11	5.01	0.57	5.51

with the GPU. This transformation of the code execution was based on the study by Pulli et al. (2012), which, although published in 2012, clearly outlines the computational advantages of using GPU over CPU, and also helps to offload work from the CPU.

In Table 7, the temporal measurements of the DL model on the GPU are presented, along with the measurements of the OF block on both the CPU and GPU, and finally, the total end-to-end application time. These measurements were obtained by processing the complete UMN and PETS datasets in each block and dividing the results by the number of frames in each dataset to obtain the average processing time.

From the experiments conducted, it can be seen that OF block involves a higher computational cost compared to the DL one. An interesting point is that executing optical flow on the GPU is significantly more efficient than on the CPU, achieving a speed approximately five times faster on the former.

However, when OF block is executed on the GPU, an overload is generated, which causes a slight slowdown in the performance of the DL model compared to when the OF block was executed on the CPU. Although this impact is minimal compared to the total execution time of the OF block, it is important to consider it in systems with lower computational capacity.

In terms of average execution time, the worst case (when the optical flow is computed on the CPU) is around 20 ms per frame, allowing the whole system to run in real-time at 30 fps. Furthermore, if the optical flow is computed on the GPU, the average time reduces significantly, being under 6 ms per frame.

It is noteworthy that the computational cost of the OF block is slightly higher for the PETS 2009 dataset compared to the UMN one. This is because the preprocessing of PETS 2009 includes an additional resizing step, which is unnecessary for the UMN dataset.

4.6. Comparison with previous works

In this section, a comprehensive comparison has been conducted between the method described in this paper and those presented in Pennisi et al. (2016) and Cob-Parro et al. (2021b). The primary aim is to illustrate how the system developed in this work not only extends but also enhances the two previous approaches, particularly in its ability to generalize with limited training data.

The method of Pennisi et al. (2016) was aimed at developing an optical flow-based classifier, specifically using Lucas–Kanade, to detect anomalous events by employing a threshold for the raw values of entropy and TOV. This threshold must be manually added after analyzing the average values. The challenge of this system lies in its lack of ability to generalize, as each environment requires a specific threshold, failing in identifying entropy or TOV peaks.

In contrast, the method of Cob-Parro et al. (2021b) utilizes Farneback's flow and a more refined signal of entropy, akin to the one applied in this paper. The TOV signal is not used, leading to a loss of information in the scene. Additionally, the classifier is based on a stack classifier composed of a random forest and a support vector classifier. Although it has better ability to generalize, it finds issues when analyzing datasets not initially used for training.

These two methods have been compared with the branch corresponding to Entropy+TOV (ET) and with the complete architecture of this work (ETK). Two experiments were conducted, similar to those in the ablation study, and in the case of Pennisi et al. (2016), the average of thresholds used for each dataset was taken to measure its ability to generalize.

To assess the systems, the metrics shown in Section 4.3 were used, including accuracy, recall, and precision. Furthermore, the methods were compared using the Gaussian method of classifier comparison in Benavoli et al. (2017), which graphically represents a triangle where each vertex reflects the relative performance of each method.

Table 8 presents the results obtained from individually training each classifier with each dataset. For the classifier of Pennisi et al. (2016), the threshold was adjusted to achieve the optimal result. Additionally, UMN and PETS 2009 were trained simultaneously due to the similarity of the data and the fewer number of frames, thus allowing an increase in the number of frames for model training.

To conduct the training and testing more precisely, a 10-fold cross-validation was performed on the dataset, and the mean of each fold was calculated and represented in the table. It is observed that, in UMN and PETS 2009, results for all classifiers are similar, above 95%. The proposed ETK classifier stands out, but with very similar results to ET classifier and those presented in Cob-Parro et al. (2021b), all three being above 98.5%.

In the case of GBA Stampedes, the classifier of Pennisi et al. (2016) is under 98% in any of its metrics. This is attributed to the inadequate detection of the initial and final peaks of entropy and TOV, suggesting that the classifier may be slower in anomaly detection.

In GSMADC, the most challenging of the four, a greater difference is observed among the classifiers. All four classifiers reduce their metrics, but Pennisi et al. (2016) suffers a more significant decline, with metrics reaching a maximum of 80%. Cob-Parro et al. (2021b) and ET classifier experience a decrease in their metrics, with a slight improvement in ET classifier, but the result is similar. Conversely, ETK classifier, which includes the angular component of optical flow, suffers the smallest decline in metrics, reaching values of up to 95%.

These observations highlight the differences and similarities in the performance of the classifiers across different datasets and underscore the importance of parameter selection and tuning in anomaly detection.

In Fig. 14, the Gaussian method has been employed to compare machine learning models (Benavoli et al., 2017). This method provides a statistical description to discern which models perform better relative to others. The comparison was conducted using the GBA Stampedes, representing an intermediate level of difficulty in detecting stampedes, and there are studied the four approaches compared in Table 8: Pennisi et al. (2016) named as PEN in the figure, Cob-Parro et al. (2021b) named as IPIN, and finally ET and ETK proposed in this work. The figure illustrates several triangles that represent comparisons between different models.

To compare the different classifiers, this method utilizes the metric values extracted from each of them. To obtain a more generalized value and a larger set of output metrics, a K-fold analysis with a value of $k = 10$ has been performed. The metric used to determine which classifier has better performance is accuracy, since it considers correctly predicted positive outcomes.

In the first three upper triangles, in which there are shown the comparison of Pennisi et al. (2016) against the other works, it is observed

Table 8
First study of architectures within different datasets. All metrics are presented in percentage (%). Best values are in bold.

Architecture	UMN + PETS 2009			GBA Stampedes			GSMADC		
	Acc	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*
Pennisi et al. (2016)	97.87	97.75	97.93	96.18	**97.54**	**97.32**	79.05	80.20	78.23
Cob-Parro et al. (2021b)	98.49	98.67	98.83	**98.54**	97.22	96.14	90.96	88.29	87.92
ET	99.29	99.22	98.73	97.89	97.00	96.61	90.18	88.60	89.80
ETK	**99.53**	**98.98**	**99.30**	98.09	97.46	97.19	**94.74**	**95.67**	**93.76**

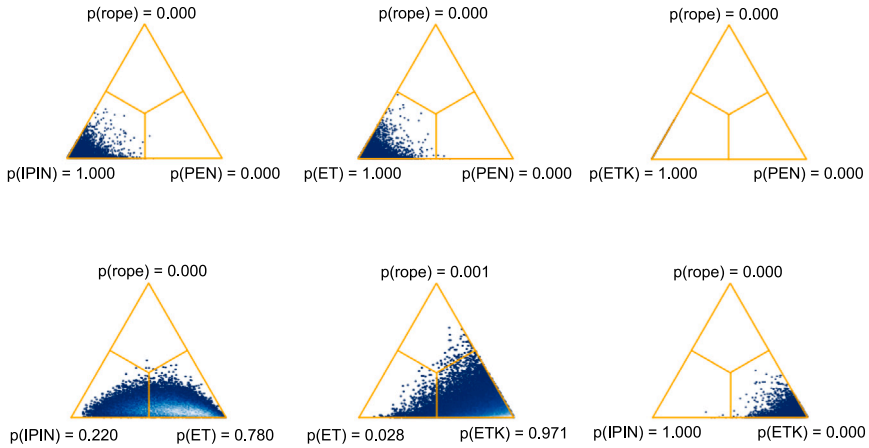


Fig. 14. Comparison of the different proposed classifiers performance using the Bayesian method (Benavoli et al., 2017) within GBA Stampedes.

that the method of Pennisi et al. (2016) is not statistically chosen when compared with the other models. This highlights a significant difference in the efficacy of this method relative to others.

When comparing the ET classifier with Cob-Parro et al. (2021b) (first triangle of the second row), the Gaussian dispersion indicates that ET classifier is more favorable 77% of the times, compared to 24.2% for Cob-Parro et al. (2021b). This quantitative comparison provides a clear insight into the relative superiority of ET classifier in this context.

Undoubtedly, the classifier that demonstrates the best results is the one formed by the three branches of optical flow (KDE). This observation underscores the effectiveness of integrating multiple components of optical flow in stampede detection.

The next study was conducted by training the models primarily with 60% of UMN and PETS 2009, 20% of GBA Stampedes, and 15% of GSMADC. This study was designed to analyze the ability to generalize to other datasets where there have been few samples for training. As in the previous case, a 10-fold cross-validation was performed, and the average of obtained metrics was calculated. For the classifier of Pennisi et al. (2016), the mean of thresholds used in the previous experiment was computed, as this classifier does not require training but rather a manual adjustment of the entropy and TOV thresholds.

In Table 9, the metrics corresponding to the various datasets used are presented. It is noteworthy that the metrics for the classifier of Pennisi et al. (2016) decline across all datasets.

In UMN and PETS 2009, precision decreases slightly, remaining in the 95%, as this metric evaluates the number of TP and TN, and what has increased is the number of FN. This situation arises because, when averaging the thresholds, these values have increased, leading to frames where the anomalous situation has begun, and the system has not yet detected it, resulting in an FN. Consequently, the metric related to recall decreases to 89%, and the accuracy metric is also affected, though to a lesser extent, settling at 92%.

In the case of GBA Stampedes, the obtained values follow the same pattern as in UMN and PETS 2009, but due to more abrupt changes in lighting between frames and oscillations in entropy and TOV, the metrics are further reduced, with precision at 86%, accuracy at 82%, and recall at 79%.

Finally, in the case of GSMADC, where lighting varies by scenario and the camera's focus and angle also change, entropy variations accumulate much more error. This results in a considerable decrease in the metrics compared to the previous experiment, where the threshold was adjusted for the dataset, with precision at 69%–70%, accuracy at 65%, and recall at 62%.

With regard to Cob-Parro et al. (2021b), it is observed that metrics have deteriorated across all datasets. In the case of UMN and PETS 2009, the results, although worse than in the previous experiment, are similar at around 95%–96%. This is attributed to the use of a greater number of data from these datasets to train the model. For GBA Stampedes, there is a significant decline in metrics (80%–82%), and the same trend is observed with the GSMADC dataset (75%–76%). This exposes that the method requires specific data to function correctly according to the environment.

In contrast, ET classifier, which in the previous experiment obtained metrics very similar to Cob-Parro et al. (2021b), has demonstrated a better ability to adapt to different environments, specifically in the GBA Stampedes(90%–91%), UMN and PETS 2009 (96%–97%). Although metrics decrease, they do not do so as drastically as with the classifier shown in Cob-Parro et al. (2021b). It is noted that the GSMADC is more complex (84%–85%), and the ET classifier struggles more to adapt to abrupt changes in scenarios compared to ETK classifier.

Finally, ETK classifier has achieved the best results across all datasets, even maintaining values in UMN and PETS 2009, and having very similar metrics in GBA Stampedes. Regarding GSMADC, metrics have declined but remain above 90% in the three proposed metrics, demonstrating that the system is capable of generalizing regardless of the scenario. This also confirms that the study of the angle generated by optical flow adds stability and robustness to the system in DL models that use this vector as an input feature.

In conclusion, the method demonstrating the greatest adaptability is the one that integrates both magnitude values (TOV and entropy) and angle values (KDE). This approach has proven to be both effective and robust across various types of datasets and experiments. It is particularly suitable for the detection of anomalous events in crowds, even with limited training data, making it a computationally efficient

Table 9
Second study of architectures within different datasets. All metrics are shown in percentage (%). Best values in bold.

Architecture	UMN + PETS 2009			GBA Stampedes			GSMADC		
	Acc	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*	*Acc*	*Rec*	*Pre*
Pennisi et al. (2016)	92.14	89.67	94.83	82.39	79.93	86.31	69.93	62.36	65.17
Cob-Parro et al. (2021b)	95.67	96.61	95.90	80.61	81.99	81.40	76.27	75.44	76.67
ET	96.93	97.23	95.94	91.41	90.14	92.33	85.64	84.58	85.11
ETK	**98.61**	**98.46**	**97.88**	**95.97**	**96.14**	**96.11**	**91.14**	**90.68**	**91.93**

and versatile system, regardless of the nature of the data analyzed.

While other classification methods have performed relatively well when trained on a single dataset, some, such as those proposed in Pennisi et al. (2016) and Cob-Parro et al. (2021b), lose their ability to generalize depending on the scenario. Compared to state-of-the-art methods, the approach that employs only the magnitude branch of optical flow exhibits better adaptability. However, in the face of abrupt changes in the scenario, it does not respond in the same manner as the classifier that inserts information of optical flow's angle.

5. Conclusions and future work

This paper describes a proposal for the detection of anomalous events within a people crowd in different video surveillance in-the-wild scenarios. The proposed system includes to modules: the first one extracts the optical flow from the analyzed scenario using the Gunner-Farneback technique, distinguishing between two output values, its magnitude (what allows computing its entropy and TOV) and the angle (from what KDE is extracted). Input vectors are grouped into temporal windows of $L = FPS/2$ and then fed into the DL network, that conforms the second module in the proposed system: an architecture that processes in parallel the three mentioned input vectors (TOV + entropy + KDE) and is able to determine frame by frame whether a stampede is happening.

The global proposal has been exhaustively evaluated on four datasets, two of them widely used in the state-of-the-art (UMN and PETS 2009) and two other recorded for this aim (and released for this publication) which are focused on realistic environments (GBA Stampedes and GSMADC). The reason for the design of these two last is in the scarcity of datasets labeled with groups of people having anomalous behavior as in stampede.

To assess the system's performance, a series of experiments have been conducted to observe its behavior across different scenarios. Initially, an ablation study of the DL network was carried out to ascertain the most efficient configuration with the fewest possible layers, thereby minimizing the proposal computational cost. This endeavor involved employing the UMN and PETS 2009 datasets to train each of the three branches of the architecture. Optimal configuration was achieved using four layers of LSTMs in the TOV branch, resulting in entropy that surpasses the 99.0% mark across all analyzed metrics. In the case of KDE, a five-layer configuration was utilized, yielding an average of 98% marks across the three metrics. It has also been demonstrated the improvement of using LSTMs instead of other RNNs such as classical ones or GRU.

A comparative study of the proposal was conducted. Similar results were observed across all methods for UMN and PETS 2009 datasets, yielding metrics around 98%. However, within GBA Stampedes and GSMADC, with higher complexity, all the metrics reduce their values. Notably, Pennisi et al. (2016) obtained in such context metrics around 79%–80%, where Cob-Parro et al. (2021b) and ET achieved around 88%–90%, and ours ETK around 94%. Thus, demonstrating the robustness of the proposal for different scenarios, and also highlighting the proposal's limitation within complex environments with changing lighting conditions, and broad acceleration movements within the crowd.

Nevertheless, it is worth highlighting that the proposal also outperforms the rest of approaches in terms of both performance and generalization capability. The system is capable of adapting to new datasets with minimal adjustments after initial training in UMN and PETS 2009. Thus, with an slight fine-tuning of only 10% of the dataset GSMADC, the proposal achieved an accuracy of 93.95%, a recall of 95.66 and a precision of 93.38%. This demonstrates that the contributed architecture operates effectively in different environments, showing its ability to apply previous learning to new situations and environments, maintaining the performance rates. Furthermore, while our system consistently exceeds 90% in all the metrics when exposed to new environments, previous works (Pennisi et al., 2016; Cob-Parro et al., 2021b), typically achieve metrics around 80%.

The analysis of the processing time has shown that, although the calculation of the optical flow has a considerable computational cost, the system is capable of performing frame-by-frame stampede detection in real time at 30 fps. However, this requires dedicated processing hardware, so the computational cost of dense optical flow computation can be a limitation. This may be particularly relevant for processing systems with limited computational capacities and within really complex environments, as stated in the paper, which might necessitate the use of specialized hardware to maintain efficient operability.

This limitation opens a line of future work focused on deploying and adapting the proposal to embedded platforms like NUC, UpSquared, Raspberry Pi, or CCTV systems. The aim would be to validate the real-time operation of the proposed solution on these devices, typical from commercial video surveillance products, where industrial processors do not rely on high-performance GPUs.

On the other hand, although our proposal has demonstrated a higher generalization capacity than state-of-the-art works, there is a dependence of the camera's perspective on the scene being analyzed. This is due to the optical flow, that can be noticeable in scene parts where the camera is close to individuals, doing that small movements may be detected as significant variations of it. This can introduce noise into the process, finally creating false predictions. Such issue is limited thanks to the entropy vectors and the DL block designed. However, it may appear in other environments with more dynamic lighting changes or more complex camera perspective.

This can also be addressed as a future work by incorporating dimensionality to the image using depth data. This would allow measuring the distance to the camera, which could be used as an input feature for the model, then increasing the robustness against changes in lighting and the flexibility across different scenarios of the system. This process requires a dataset that includes depth information in addition to RGB. In this regard, the dataset GSMADC, created for this project, includes both color and depth information, although it has not been introduced as the other available datasets do not incorporate it.

Despite these limitations, as discussed above, the proposal outperforms the state-of-the-art work both in terms of precision and generalizability. Moreover, Despite these limitations, as discussed above, the proposal outperforms the state-of-the-art work in terms of both accuracy and generalizability. In addition, several lines of future work are considered to improve its capabilities.

The proposal only detects if a stampede is happening. Thus, a potential enhancement could be to evolve from the detection system to a classification one, where different types of anomalous crowd behaviors implying high entropy (lateral shifts, crowd dispersal, etc.) could be distinguished. This would require video datasets with a ground truth

including these crowd anomalies.

Finally, although this work is focused on small and medium-sized crowds, with less than 100 people per scene, given the generalization capability of the proposal, another possible line of future work is to analyze this architecture or evolve it for events involving a larger number of individuals.

CRediT authorship contribution statement

Antonio Carlos Cob-Parro: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Cristina Losada-Gutiérrez:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Marta Marrón-Romera:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has received funding from the European Union's Horizon 2020 Research and Innovation Program under PALAEMON project (Grant Agreement n° 814962), by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 under projects EYEFUL-UAH (PID2020-113118RB-C31) and ATHENA (PID2020-115995RB-I00) and by CAM under project CONDORDIA (CM/JIN/2021-015).

Data availability

Data will be made available on request.

References

Abdullah, F., Ghadi, Y.Y., Gochoo, M., Jalal, A., Kim, K., 2021. Multi-person tracking and crowd behavior detection via particles gradient motion descriptor and improved entropy classifier. Entropy 23 (5), 628.

Ajao, O., Bhowmik, D., Zargari, S., 2018. Fake news identification on twitter with hybrid cnn and rnn models. In: Proceedings of the 9th International Conference on Social Media and Society. pp. 226–230.

Al-Nawashi, M., Al-Hazaimeh, O.M., Saraee, M., 2017. A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments. Neural Comput. Appl. 28 (1), 565–572.

Ali, F., El-Sappagh, S., Islam, S.R., Kwak, D., Ali, A., Imran, M., Kwak, K.-S., 2020. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. Inf. Fusion 63, 208–222.

Ali, S., Shah, M., 2007. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–6.

Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., Razaque, A., 2020. Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory 101, 102031.

Ayvaci, A., Raptis, M., Soatto, S., 2012. Sparse occlusion detection with optical flow. Int. J. Comput. Vis. 97 (3), 322–338.

Baptista-Ríos, M., Martínez-García, C., Losada-Gutiérrez, C., Marrón-Romera, M., 2016. Human activity monitoring for falling detection. A realistic framework. In: 2016 International Conference on Indoor Positioning and Indoor Navigation. IPIN, IEEE, pp. 1–7.

Beauchemin, S.S., Barron, J.L., 1995. The computation of optical flow. ACM Comput. Surv. (CSUR) 27 (3), 433–466.

Benavoli, A., Corani, G., Demšar, J., Zaffalon, M., 2017. Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J. Mach. Learn. Res. 18 (1), 2653–2688.

Brosch, T., Tschechne, S., Neumann, H., 2015. On event-based optical flow detection. Front. Neurosci. 9, 137.

Castiglioni, I., Rundo, L., Codari, M., Di Leo, G., Salvatore, C., Interlenghi, M., Gallivanone, F., Cozzi, A., D'Amico, N.C., Sardanelli, F., 2021. AI applications to medical images: From machine learning to deep learning. Phys. Medica 83, 9–24.

Cob-Parro, C., 2023. Git hub repository including UMN dataset with the updated grountruth. https://github.com/CarlosCobParro/UMN-groundtruth-update.

Cob-Parro, A.C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., Bravo-Muñoz, I., 2021a. Smart video surveillance system based on edge computing. Sensors 21 (9), 2958.

Cob-Parro, A.C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., Bravo-Muñoz, I., 2023a. A new framework for deep learning video based human action recognition on the edge. Expert Syst. Appl..

Cob-Parro, C., Losada-Gutiérrez, C., Marrón-Romera, M., Vidal Pinto, R., Bravo, I., Gardel, A., 2023b. The GEINTRA multiple actions dataset in cruises (GSMADC). https://geintra-uah.org/index.php?q=datasets#gsmadc. (Last Accessed 04 March 2024).

Cob-Parro, A.C., Losada-Gutiérrez, C., Romera, M.M., Vicente, A.G., Muñoz, I.B., Sarker, M.I., 2021b. A proposal on stampede detection in real environments. In: IPIN-WiP.

Delvigne, V., Wannous, H., Vandeborre, J.-P., Ris, L., Dutoit, T., 2022. Spatio-temporal analysis of transformer based architecture for attention estimation from EEG. arXiv preprint arXiv:2204.07162.

Ek, S., Portet, F., Lalanda, P., 2022. Lightweight transformers for human activity recognition on mobile devices. arXiv preprint arXiv:2209.11750.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J., 2019. A guide to deep learning in healthcare. Nature medicine 25 (1), 24–29.

Farnebäck, G., 2003. Two-frame motion estimation based on polynomial expansion. In: Scandinavian Conf. on Image Analysis. Springer, pp. 363–370.

Ferryman, J., Shahrokni, A., 2009. Pets2009: Dataset and challenge. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. IEEE, pp. 1–6.

GEINTRA, 2022. GEINTRA behaviour analysis dataset. https://geintra-uah.org/en/node/2309. (Last Accessed 19 February 2024).

Gupta, T., Nunavath, V., Roy, S., 2019. CrowdVAS-net: A deep-CNN based framework to detect abnormal crowd-motion behavior in videos for predicting crowd disaster. In: 2019 IEEE International Conference on Systems, Man and Cybernetics. SMC, IEEE, pp. 2877–2882.

Haque, S., Sadi, M.S., Rafi, M.E.H., Islam, M.M., Hasan, M.K., 2020. Real-time crowd detection to prevent stampede. In: Proceedings of International Joint Conference on Computational Intelligence: IJCCI 2018. Springer, pp. 665–678.

Hibat-Allah, M., Ganahl, M., Hayward, L.E., Melko, R.G., Carrasquilla, J., 2020. Recurrent neural network wave functions. Phys. Rev. Res. 2 (2), 023358.

Hinton, G., 2018. Deep learning—a technology with the potential to transform health care. Jama 320 (11), 1101–1102.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Jebrane, A., Argoul, P., Hakim, A., El Rhabi, M., 2019. Estimating contact forces and pressure in a dense crowd: Microscopic and macroscopic models. Appl. Math. Model. 74, 409–421.

Khaki, S., Wang, L., Archontoulis, S.V., 2020. A cnn-rnn framework for crop yield prediction. Front. Plant Sci. 10, 1750.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Lee, C., Kosta, A.K., Zhu, A.Z., Chaney, K., Daniilidis, K., Roy, K., 2020. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In: Euro. Conf. on Computer Vision. Springer, pp. 366–382.

Li, S., Li, W., Cook, C., Zhu, C., Gao, Y., 2018. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5457–5466.

Liu, P., King, I., Lyu, M.R., Xu, J., 2019a. Ddflow: Learning optical flow with unlabeled data distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33, pp. 8770–8777.

Liu, S., Liu, J., Wei, W., 2019b. Simulation of crowd evacuation behaviour in outdoor public places: A model based on shanghai stampede. Int. J. Simul. Model. 18 (1), 86–99.

Lucas, B.D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th International Joint Conference on Artificial Intelligence. Vol. 2, pp. 674–679.

Luque Sánchez, F., Hupont, I., Tabik, S., Herrera, F., 2020. Revisiting crowd behaviour analysis through deep learning: Taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects. Inf. Fusion 64, 318–335. http://dx.doi.org/10.1016/j.inffus.2020.07.008, URL https://www.sciencedirect.com/science/article/pii/S1566253520303201.

Matkovic, F., Ivasic-Kos, M., Ribaric, S., 2022. A new approach to dominant motion pattern recognition at the macroscopic crowd level. Eng. Appl. Artif. Intell. 116, 105387.

Matkovic, F., Marčetic, D., Ribaric, S., 2019. Abnormal crowd behaviour recognition in surveillance videos. In: 2019 15th Int. Conference on Signal-Image Technology & Internet-Based Systems. SITIS, IEEE, pp. 428–435.

Mehran, R., Oyama, A., Shah, M., 2009. Abnormal crowd behavior detection using social force model. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 935–942.

Muhammed Anees, V., Santhosh Kumar, G., 2022. Identification of crowd behaviour patterns using stability analysis. J. Intell. Fuzzy Systems 42 (4), 2829–2843.

Naik, N., Mohan, B.R., 2019. Study of stock return predictions using recurrent neural networks with LSTM. In: International Conference on Engineering Applications of Neural Networks. Springer, pp. 453–459.

Nasir, J.A., Khan, O.S., Varlamis, I., 2021. Fake news detection: A hybrid CNN-RNN based deep learning approach. Int. J. Inf. Manage. Data Insights 1 (1), 100007.

O'Donovan, P., 2005. Optical flow: Techniques and applications. Int. J. Comput. Vis. 1, 26.

Palaemon-H2020, 2023. Official page of PALAEMON project. http://dx.doi.org/10.3030/814962, https://cordis.europa.eu/project/id/814962/es. (Last Accessed 13 February 2024).

Pan, L., Zhou, H., Liu, Y., Wang, M., 2019. Global event influence model: integrating crowd motion and social psychology for global anomaly detection in dense crowds. J. Electron. Imaging 28 (2), 023033.

Paredes-Vallés, F., Scheper, K.Y., De Croon, G.C., 2019. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. IEEE Trans. Pattern Anal. Mach. Intell. 42 (8), 2051–2064.

Chantamit-o pas, P., Goyal, M., 2018. Long short-term memory recurrent neural network for stroke prediction. In: International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, pp. 312–323.

Patil, N., Biswas, P.K., 2018. Global abnormal events detection in crowded scenes using context location and motion-rich spatio-temporal volumes. IET Image Process. 12 (4), 596–604.

Pennisi, A., Bloisi, D.D., Iocchi, L., 2016. Online real-time crowd behavior detection in video sequences. Comput. Vis. Image Underst. 144, 166–176.

Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V., 2012. Real-time computer vision with OpenCV. Commun. ACM 55 (6), 61–69.

Raj, J.S., Ananthi, J.V., et al., 2019. Recurrent neural networks and nonlinear prediction in support vector machines. J. Soft Comput. Paradigm (JSCP) 1 (01), 33–40.

Räty, T.D., 2010. Survey on contemporary remote surveillance systems for public safety. IEEE Trans. Syst. Man Cybern. C (Appl. Rev.) 40 (5), 493–515.

Ravanbakhsh, M., Nabi, M., Mousavi, H., Sangineto, E., Sebe, N., 2018. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In: 2018 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 1689–1698.

Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., Sebe, N., 2017. Abnormal event detection in videos using generative adversarial nets. In: 2017 IEEE International Conference on Image Processing. ICIP, IEEE, pp. 1577–1581.

Ren, Z., Luo, W., Yan, J., Liao, W., Yang, X., Yuille, A., Zha, H., 2020. Stflow: Self-taught optical flow estimation using pseudo labels. IEEE Trans. Image Process. 29, 9113–9124.

Shafie, A.A., Hafiz, F., Ali, M.H., 2009. Motion detection techniques using optical flow. Int. J. Electr. Comput. Eng. 3 (8), 1555–1557.

Shiba, S., Aoki, Y., Gallego, G., 2022. Secrets of event-based optical flow. In: European Conference on Computer Vision. Springer, pp. 628–645.

Sun, L., Xu, W., Liu, J., 2021. Two-channel attention mechanism fusion model of stock price prediction based on CNN-LSTM. Trans. Asian Low-Resour. Lang. Inf. Process. 20 (5), 1–12.

Tripathi, G., Singh, K., Vishwakarma, D.K., 2019. Convolutional neural networks for crowd behaviour analysis: a survey. Vis. Comput. 35, 753–776.

Tu, P., Sebastian, T., Doretto, G., Krahnstoever, N., Rittscher, J., Yu, T., 2008. Unified crowd segmentation. In: Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10. Springer, pp. 691–704.

Tyagi, B., Nigam, S., Singh, R., 2022. A review of deep learning techniques for crowd behavior analysis. Arch. Comput. Methods Eng. 29 (7), 5427–5455.

Ullah, A., Muhammad, K., Ding, W., Palade, V., Haq, I.U., Baik, S.W., 2021. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. Appl. Soft Comput. 103, 107102.

Wu, D., Lv, S., Jiang, M., Song, H., 2020. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput. Electron. Agric. 178, 105742.

Zhou, X., Li, Y., Liang, W., 2020a. CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (3), 912–921.

Zhou, X., Liang, W., Kevin, I., Wang, K., Wang, H., Yang, L.T., Jin, Q., 2020b. Deep-learning-enhanced human activity recognition for Internet of healthcare things. IEEE Internet Things J. 7 (7), 6429–6438.

Zuo, Y., Hamrouni, A., Ghazzai, H., Massoud, Y., 2023. V3Trans-Crowd: A video-based visual transformer for crowd management monitoring. In: 2023 IEEE International Conference on Smart Mobility. SM, IEEE, pp. 154–159.