Lavender autonomous navigation with semantic segmentation at the edge

Intelligent Systems with Applications 26 (2025) 200536

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

From detection to intervention: An end-to-end system for recognizing the "signal for help" gesture in real-time

Federico Buccellato *, Eleonora Vacca, Sarah Azimi, Corrado De Sio, Luca Sterpone

Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy

ARTICLE INFO

Keywords:
Signal for help
Real-time gesture recognition
Machine learning
Domestic violence intervention
Mobile application
Amazon S3
FCM
MongoDB

ABSTRACT

The "Signal for Help" is a simple hand gesture, internationally recognized, that enables individuals experiencing domestic violence to discreetly signal their need for help without alerting their aggressors. Developed during the COVID-19 pandemic to address the growing isolation of victims, it serves as a powerful tool to facilitate silent communication in dangerous situations. Despite its potential, its effectiveness has been impeded by limited public awareness, the risk of misinterpretation, and the lack of reliable automated detection systems.

To address these challenges, this paper introduces a framework consisting of two interconnected components: a real-time detection system of the "Signal for Help" gesture using a machine learning-based recognition system and a custom mobile application that receives notifications from the detection system and alerts security personnel in real-time.

During the development process, we faced several challenges, including detecting the gesture in crowded environments and keeping the computational load low to ensure the system could run efficiently on edge devices.

We overcame these challenges by designing a system that combines hand tracking and feature extraction, using tools such as MediaPipe and DeepSORT, followed by a final classification step. After testing various classifiers, Random Forest achieved the best results, reaching an accuracy of 94 % with a very low rate of false positives. The system was carefully optimized to minimize computational cost while maintaining real-time performance. In fact, as shown by the tests conducted on Apple M3, NVIDIA Jetson Orin Nano, and NVIDIA Jetson AGX Orin, the system achieved inference times of 0.067 s, 0.471 s, and 0.343 s respectively. These outcomes demonstrate the system's possibility for deployment in smart city environments, supporting both urban and non-urban areas. When a gesture is detected, the system immediately notifies the mobile application, which provides instant alerts, geolocation data, and a short video clip of the incident, enabling a rapid and informed response. Additionally, the app includes advanced features such as detailed notification history, real-time operator status monitoring, and an integrated team coordination chat, which optimize operations, enhance collaboration among security staff, and ensure timely and effective interventions in emergency situations. This research marks a step forward in real-time gesture recognition and intervention, setting a new benchmark for automated safety systems aimed at preventing domestic violence and other emergencies. By increasing awareness and ensuring a rapid response to the "Signal for Help" gesture, the system empowers individuals in distress and contributes to safeguarding those at risk.

1. Introduction

One in three women has experienced either physical or sexual violence in her lifetime. This is not just a statistic, it is a silent request for help that demands urgent attention and action. During the COVID-19 pandemic, this situation worsened significantly. Lockdowns and isolation turned many homes into places of danger, leading to a sharp increase in domestic violence cases. Victims were often trapped with no access to the support systems that might have helped them escape (World Health Organization, 2021).

In response to this crisis, the Canadian Women's Foundation developed an innovative solution to help victims communicate their need for assistance. This solution, known as the "Signal for Help", is a simple hand gesture that allows individuals to silently indicate they are in

* Corresponding author.
E-mail addresses: federico.buccellato@polito.it (F. Buccellato), eleonora.vacca@polito.it (E. Vacca), sarah.azimi@polito.it (S. Azimi), corrado.desio@polito.it (C. De Sio), luca.sterpone@polito.it (L. Sterpone).

https://doi.org/10.1016/j.iswa.2025.200536
Received 19 March 2025; Received in revised form 9 May 2025; Accepted 16 May 2025
Available online 17 May 2025

danger without alerting their abuser. The gesture involves folding the thumb into the palm and covering it with the fingers as represented in Fig. 1, creating a non-verbal signal that is both discreet and powerful. Since its introduction during the COVID-19 pandemic, the "Signal for Help" has continued to be a vital tool for victims, remaining widely used even in the post-pandemic period as a recognized means of seeking help in dangerous situations.

The "Signal for Help" has proven to be a critical tool in several cases, enabling victims to silently communicate their distress and ultimately escape dangerous situations. Recognized and adopted worldwide, this gesture has had a significant global impact in raising awareness and facilitating real-time interventions. These successful interventions highlight the potential of this gesture to save lives and provide immediate assistance in moments of crisis, as seen in cases like the United States, where a teenage girl was rescued after using the signal from a moving vehicle (The Guardian, 2021), in Malaysia, where a woman was saved after making the gesture during a public event (The Rakyat Post, 2023) or in Milan where a 19-year-old woman used the gesture to discreetly signal an employee, who immediately alerted the police, leading to the assailant's arrest (MilanoToday, 2025).

However, there have also been instances where the "Signal for Help" was overlooked (Ossino, 2023). In busy or distracted environments, or among people who were unaware of its meaning, victims' silent pleas for help went unnoticed, leaving them without the support they urgently needed. These situations emphasize the limitations of relying solely on human recognition and awareness for such a critical signal. This gap underscores the importance of developing a system capable of reliably detecting the signal and ensuring timely intervention. Current methods for detecting the "Signal for Help" primarily rely on 3D convolutional neural networks (3D CNNs) to analyze hand gestures in video footage (Azimi et al. 2023, 2023; Elliott, et al. 2021), but these systems often face challenges in real-time scenarios. A more advanced approach, described in Buccellato et al. (2024), incorporates AI-powered recognition models with person-tracking algorithms to improve detection accuracy. However, these methods are computationally demanding, making them unsuitable for resource-constrained environments, and they still fall short of delivering fast, real-time support.

This paper aims to address these challenges by presenting a system that enhances safety in both urban and non-urban environments combining surveillance cameras with advanced AI algorithms. Trained on a realistic dataset, the system detects the "Signal for Help" gesture in real time and relays alerts through a dedicated mobile app that instantly notifies security personnel. By ensuring rapid alerts and enabling prompt interventions, this approach, tested on different types of devices, offers people a discreet way to request help while contributing to the development of safer, more responsive smart cities environments.

1.1. Main contribution

While previous research has focused primarily on gesture detection using computationally intensive models or limited datasets, our approach introduces a novel, efficient, and deployable end-to-end pipeline. Specifically, we go beyond existing systems by:

- ***Dataset Refinement for Robust and Accurate Detection:*** We addressed a critical gap in existing research by refining the dataset used to train the detection model. By expanding the dataset to include more subtle and ambiguous variations of the "Signal for Help", the model's accuracy and reliability have been improved. This ensures robust performance in diverse and challenging real-world conditions, such as noisy environments or when gestures are less explicit.
- ***Two-Step Machine Learning Pipeline for Signal Detection*:** We have developed an advanced detection pipeline for the real-time recognition of the "Signal for Help" hand gesture, designed to be both efficient and highly accurate. The process begins with a real-time hand detection and tracking system, capable of performing effectively even with low-resolution video inputs. This system captures both spatial and temporal data, providing a representation of hand movements over time. In the second stage, a machine learning model processes this spatiotemporal data to reliably identify the "Signal for Help" gesture. Once detected, the system seamlessly triggers an alert to a connected mobile application, enabling immediate follow-up actions.
- ***Mobile Application for Real-Time Alerts and Verification:*** Our system integrates a dedicated mobile application designed specifically for security personnel. The app streamlines the response process by delivering video clips of detected gestures along with precise location data, allowing security teams to quickly assess the validity of the alert and take immediate action. Additional features include advanced tools to assist in identifying the individual requesting help, even in challenging environments such as complex or crowded scenes, ensuring timely and accurate intervention.
- **Testing and Validation:** We have tested and validated our system performance across edge and central platforms to ensure scalability and reliability under diverse operational conditions.

We are aware that some errors in this system may still lead to false positives and unwanted alarms. However, given the critical importance of this application, we have worked to reduce this issue as much as possible.

This is the first system capable of detecting the help signal and tracking multiple hands in crowded, low-quality surveillance footage with up to 94 % accuracy, all while operating in real time with minimal computational overhead. By seamlessly integrating advanced detection capabilities with a streamlined response mechanism, our solution offers

Fig. 1. The "Signal for help" hand gesture.

a practical tool for enhancing the safety and support of individuals in danger.

The paper is structured as follows: *Section 2* reviews related works on gesture recognition systems and their application in detecting the "Signal for Help." *Section 3* introduces the technical background and tools used in this study for both the development of the detection system and the mobile application. *Section 4* details the dataset used, its refinement process, and the integration of realistic scenarios to enhance model training. *Section 5* describes the system architecture, outlining the transition from the initial prototype to the optimized real-time detection pipeline. *Section 6* focuses on the mobile application, highlighting its features and role in providing real-time alerts and facilitating security interventions. *Section 7* explains the integration between the detection system and the mobile application using secure data management. *Section 8* presents the experimental results, including performance metrics, hardware evaluations, and comparisons with existing approaches. Finally, *Section 9* concludes the paper with a summary of findings and future directions, such as expanding the system to detect additional gestures or triggers for broader safety applications.

2. Related works

To develop a robust and practical solution, we began by analyzing existing approaches to emergency and gesture recognition detection. This review highlights current limitations and motivates the need for our system.

Detecting violent behavior through technological means has emerged as a critical area of focus within safety and security research. Over the years, methodologies for identifying potentially dangerous situations in real-time have significantly advanced, primarily through the analysis of visual and auditory indicators, including motion patterns, emotional states, facial expressions, and auditory signals such as screams.

For instance, in Traoré and Akhloufi (2020) Traoré et al. developed a hybrid architecture combining Bi-directional GRUs with 2D CNNs to capture spatial frame features and model temporal dynamics of violent actions.

Similarly, in Wang, et al. (2024) Wang et al. proposed a lightweight 2D CNN augmented with Bi-Directional Motion Attention and frame--grouping to optimize computation while sensitively detecting short-term violent activities in real time.

In Kang and Kwak (2015) Kang & Kwak integrated color and motion cues, such as blood detection and aggressive movement analysis, into a CNN framework to differentiate violent acts from non-violent behavior in dynamic surveillance environments. Furthermore, in Manias, et al. (2022) Manias et al. proposed a neural-network-based pipeline specifically to detect sentiments from text, applying multilingual sentiment analysis on Twitter streams to extract emotional cues directly from textual inputs. These methodologies demonstrate a high degree of efficacy in recognizing visible and physical manifestations of violence.

However, the "Signal for Help" presents a different challenge. Unlike traditional violent acts, it is a discreet hand gesture used to silently request assistance without drawing attention from potential aggressors. Its silent nature makes it difficult to detect using standard violence detection systems, which rely on clear signs such as sudden movements, or visible injuries. For this reason, rather than focusing on identifying violent actions, we shifted our attention to gesture recognition techniques, as the "Signal for Help" is a specific hand gesture that requires more precise detection methods.

In the field of gesture recognition, several advanced techniques have emerged. In Tran et al. (2020), for instance, a novel method for real-time fingertip detection and hand gesture recognition was developed using an RGB-D camera combined with a 3D convolutional neural network. This system effectively pinpoints fingertip locations, highlighting the benefits of depth information. Likewise, (Junping & Siping, 2021) explores the use of convolutional neural networks with the YCbCr color space to accurately recognize gestures from video data.

Moreover, the approach in Peng et al. (2020) introduced a dynamic gesture recognition method that employs feature fusion networks and variant ConvLSTM, emphasizing the importance of understanding ongoing human gestures in various contexts. Additionally, in Krishnan et al. (2021) was addressed the challenges of gesture recognition in degraded environments, highlighting the need for robust systems that can perform well under varying conditions. These advancements in gesture recognition underscore the ongoing evolution of technologies aimed at enhancing the detection of subtle gestures like the "Signal for Help", thereby improving safety and security measures in various contexts.

After reviewing the most effective gesture recognition approaches, we first conducted an analysis of existing datasets specifically aimed at the "Signal for Help" gesture. This initial analysis highlighted a critical limitation: the absence of realistic open-source datasets. Currently, the only publicly available dataset dedicated to this gesture is the SFH-Dataset (Shafique), consisting of approximately 7000 frames depicting hands performing the "Signal for Help." However, this dataset is constrained by its limited size and lack of diversity. These limitations significantly complicate the training of robust models and hinder their ability to generalize effectively.

Subsequently, we analyzed current solutions aimed at recognizing the "Signal for Help," identifying their strengths and limitations.

Early models for recognizing the "Signal for Help" have primarily relied on convolutional neural networks Among these, the work in Elliott, Meehan and Hyndman, (2021) proposed a solution involving the use of a small dataset to train a 3D-CNN model. However, given the limited size of the dataset, CNNs are not the most suitable choice, as they typically require a significant amount of diverse data to effectively learn complex patterns and generalize well. In this model, they attempted to address the limited amount of data by applying various data augmentation techniques. Nevertheless, even after these augmentations, the dataset remained low in variability. This lack of variance negatively affected the model's ability to generalize to new contexts. The reported accuracy results, showing 87.5 % without augmentation and a perfect 100 % with augmentation, appear unrealistic and raise concerns about the model's robustness.

Another CNN solution proposed in Azimi, et al. (2023) [7] leverages transfer learning by utilizing the Jester dataset during the pre-training phase. However, applying a pre-trained model on a general gesture dataset like Jester and subsequently fine-tuning it for the "Signal for Help" recognition task relies on a key assumption. It presumes that the features learned during pre-training are sufficiently adaptable to distinguish between "Signal for Help" and non-"Signal for Help" gestures. While transfer learning is often beneficial for related tasks, this assumption introduces a critical challenge for the 3D-CNN-based approach, particularly due to the differences in gesture semantics between the two datasets.

Some more advanced works abandoned the use of CNNs for gesture detection and instead focused on extracting hand features using MediaPipe, followed by training classifiers on those features. One such approach, described in Thejowahyono, et al. (2022), proposed a system where hand landmarks are extracted from video frames using OpenCV and MediaPipe. The system tracks hand positions by detecting the palm, and then extracting key points representing the fingers and palm structure. These key points serve as input for a Deep Neural Network (DNN) that recognizes if there is the presence of the "Signal for Help" gesture.

However, while this method shows promise, it has a significant limitation when applied in real-world scenarios. MediaPipe's hand tracking algorithm performs well in detecting up to two hands simultaneously, but it becomes unstable when dealing with three or more hands in a single frame. This issue is critical in public environments where the ability to distinguish multiple hands accurately is essential for recognizing the "Signal for Help" in a crowd. The model's reliance on

stable two-hand detection makes it impractical for more complex settings, where the gesture might be partially occluded or surrounded by other hands, thus reducing the robustness of the system.

A more recent approach, also based on MediaPipe (Buccellato, et al. 2024), involves a three-step pipeline that will be discussed in detail in Section 5. While this method improves hand gesture recognition accuracy by applying more sophisticated feature extraction and classification techniques, it suffers from high computational costs. Such requirements make it less suitable for deployment on devices with limited hardware capabilities such as some edge devices. Therefore, although the model enhances detection performance in controlled environments, its scalability and practicality for real-time applications remain constrained. Despite significant advancements in gesture recognition, existing solutions for detecting the "Signal for Help" fail to address a critical gap: the need to automatically call for help when the gesture is recognized. In fact, current models focus only on identifying the gesture, lacking the capability to trigger real-time intervention. For instance, the study conducted by authors in Mohd, et al. (2010) presents a low-cost system where a network camera detects motion and, upon capturing an image, sends an SMS alert via a GSM modem. Similarly, in Margapuri, et al. (2021) was proposes a modern IoT solution that integrates a Raspberry Pi with PIR sensors and a camera, employing Google Firebase Cloud Messaging to deliver push notifications to a mobile phone upon intrusion detection. However, none of these applications offer a mobile solution that works in real time with video surveillance systems to send instant alerts when subtle, hidden gesture, such as the "Signal for Help", are detected.

In the majority of these studies, the primary challenge has been finding an optimal way to detect the hand and extract the most relevant information from the available datasets, a task further complicated by various external factors, including the resolution of the camera and digital images, the inherent difficulty of distinguishing between similar gestures, and fluctuating environmental conditions.

To bridge this gap, our work presents a solution that goes beyond simple gesture detection. It provides a complete process, from recognizing the "Signal for Help" to triggering real-time intervention, offering an effective system for ensuring safety and security in both public and private spaces.

We developed a two-step pipeline model specifically designed to detect the "Signal for Help" efficiently and reliably. The first step of our model focuses on hand tracking and feature extraction, ensuring that only relevant hand gestures are processed. The second step applies gesture recognition to accurately identify the "Signal for Help".

Our approach not only ensures low computational costs and fast response times but also achieves high accuracy and robust performance across various environments. The two-step pipeline model addresses the limitations of existing systems by offering a reliable, scalable, and efficient method to recognize covert distress signals and ensure that help is delivered promptly.

Moreover, we created the first mobile application for security operators, which is installed on their devices to receive instant alerts when the "Signal for Help" is detected. This ensures that the gesture is immediately followed by a call for assistance, providing a practical and actionable solution to improve safety in real-world scenarios.

Table 1 shows a comparison of the main features of our work with those of several of the most significant solutions mentioned earlier. Specifically, it presents the implementation details of each system, such as whether video processing is applied or the maximum number of hands that can be detected, and the testing details, highlighting performances and the context in which each study's results were obtained.

3. Technical background

Based on the gaps identified in the literature, we selected technologies specifically suited to address the challenges of real-time gesture recognition and alert delivery. This section outlines the rationale behind these choices.

At the core of the detection system, DeepSORT and MediaPipe Hand work together to ensure precise hand tracking and gesture recognition, forming a solidfoundation capable of adapting to complex and dynamic environments.

Relatively to the mobile application, Amazon S3 and Firebase Cloud Messaging (FCM) ensure smooth operation and responsiveness, creating a cohesive and reliable system that supports efficient data management and communication.

- **DeepSORT:** DeepSORT (Simple Online and Real-time Tracking) enhances video-based object tracking by integrating deep learning with traditional tracking algorithms (Wojke, Bewley & Paulus, 2017). This hybrid approach allows for precise tracking of objects across sequential video frames. Unlike basic tracking methods, DeepSORT applies deep neural networks to maintain a consistent association of objects over time, even in the presence of occlusions, object overlap, or erratic motion. This capability is crucial for gesture recognition systems, as it ensures the reliable tracking of hands or other critical elements in dynamic and complex video environments. The enhanced tracking accuracy offered by DeepSORT strengthens the system's ability to handle complex movement patterns and maintain consistent recognition performance.
- **MediaPipe Hand:** MediaPipe Hand, a model developed by Google, specializes in the detection of human hands (Zhang et al., 2020). Unlike conventional object detection systems, MediaPipe Hand offers an advanced level of precision by identifying 21 specific key points for each hand, as illustrated in Fig. 2. These key points map the position of joints, fingertips, and other critical hand landmarks. This level of detail enables precise monitoring of hand movements, which is essential for the recognition of the "Signal for Help" gesture.
- **Amazon S3:** Amazon Simple Storage Service (Amazon S3) is a scalable cloud-based storage service that provides high availability, durability, and security for storing and retrieving data (Amazon Web Services 2025). Its ability to efficiently manage large volumes of data ensures reliable storage and quick access to essential files, making it a valuable component for handling application data efficiently and securely.
- **FCM:** Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that enables reliable delivery of push notifications and messages between servers and client applications. This feature is essential for applications that require immediate response and notification capabilities.

Table 1
Comparison of high-level features across existing systems and our proposed solution.

	Video Elaboration	Multi-Hand Detection	Dataset Size	Signal Notification	Real-Time Capability	Test Scenario
(Elliott, Meehan & Hyndman, 2021)	No	No	Very Small	No	Yes	Controlled Environment
(Thejowahyono, Setiawan, Handoyo & Rangkuti, 2022)	Yes	No (Maximum 2 hands)	Small	Yes (Email only, with location)	Yes	Controlled Environment
The Proposed	Yes	Yes	Medium	Yes (Full app: video, alerts, location)	Yes	Real-World Scenario

Fig. 2. Hand landmarks extracted by the MediaPipe (MediaPipe Hands solution, Hand landmarks detection).

The seamless integration of these components has been crucial in building a robust and scalable solution. In our implementation, we chose a set of technologies that guarantee precision, fast execution, scalability and easy deployment even on edge devices with limited resources: MediaPipe provides lightweight yet highly accurate hand tracking thanks to its architecture optimized for both CPU and embedded platforms; DeepSORT delivers robust tracking of individual hands in video sequences, preserving each subject's identity even in crowded or dynamic scenes; and Amazon S3 together with Firebase Cloud Messaging ensure secure video storage and reliable push notifications, which are essential for managing alert workflows in live applications with the flexibility and robustness of cloud services. There are alternative options for each technology, such as OpenPose for hand detection, SORT for tracking, or Azure Blob for storage, but these prove suboptimal in terms of performance, making the system heavier and less effective on both CPU-based and edge systems. The components we selected work together to form a cohesive system capable of detecting with high reliability, reacting swiftly, and operating with a minimal hardware overhead.

4. "Signal for help" dataset

With the technical foundation defined, we turned our focus to the dataset, critical for training a reliable model. We describe how the dataset was built and refined to simulate realistic use cases.

Our project began with the analysis of the "Signal for Help" dataset, a substantial collection of videos created in collaboration with Alta Scuola Politecnica (ASP) as part of the S2CITIES project (S2CITIES - Toward Smart & Safe Cities: Exploiting Surveillance Videos for Real-time Detection of "Signal for Help), supported by the efforts of students from Politecnico di Milano and Politecnico di Torino. This dataset consists of around 4000 videos, evenly split between those demonstrating the "Signal for Help" gesture and those showcasing a range of everyday gestures. It served as afoundation for developing models capable of identifying the Signal for Help gesture with precision. One of the dataset's key strengths lies in its variety as it includes videos featuring diverse participants performing gestures in various conditions, such as different lighting, distances, angles, and backgrounds.

This diversity was important for training models to generalize effectively across different environments. However, while the dataset provided a strong starting point, our work also revealed different internal biases that required attention.

First is the presence of selection bias (Mavrogiorgos et al., 2024; Rodrigo-Ginés, et al. 2024); there was a noticeable inconsistency in the non-signal samples. These samples mostly contained gestures that were very different from the "Signal for Help." This lack of similar-looking gestures in the non-"Signal for Help" category made it challenging for the model to learn to distinguish between subtle differences, leading to potential misclassifications when encountering gestures closely resembling the signal.

Second is a coverage bias (Rodrigo-Ginés, et al. 2024), the dataset lacked realistic scenarios, such as videos resembling security-camera perspectives. This limitation hindered our ability to test the model's performance in real-world conditions effectively.

To address these issues, we expanded the original dataset by incorporating new videos and features specifically designed to fill the identified gaps. We established clear collection guidelines so that each new clip directly counteracts a measured bias. These enhancements were strategically tailored to improve the dataset in the following key areas:

- **Focused Gesture Representation:** We added videos that either clearly demonstrate the "Signal for Help" gesture or showcase gestures that closely resemble it, such as casual waves or other similar hand motions. This addition aims to reduce false positives by training the model to differentiate between gestures that might appear similar but do not convey the intended signal. Gestures significantly different from the "Signal for Help" were already well-represented in the original dataset, so our focus was on misleading and challenging cases.
- **Realistic Environmental Contexts:** The new videos prioritize realistic environments, simulating everyday scenarios where the "Signal for Help" gesture might naturally occur. These include a variety of settings with different lighting conditions and typical daily activities, such as walking near shops and crowds during the day or walking in the evening surrounded by other people, introducing background noise that better reflects real-world scenarios. Additionally, unlike the previous version of the dataset, where 60 % of the videos featured only a hand performing the gesture in isolation, 38 % included both the hand and the person making the gesture, and only 2 % realistically represented the full context with the person performing the gesture along with the aggressor, the updated dataset ensures that the person signaling for help is not alone in the scene but is most often depicted alongside the aggressor, creating more realistic and challenging recognition scenarios.
- **Diversity in Execution:** We ensured that the new videos capture the natural variations in how individuals perform the "Signal for Help" gesture. This includes differences in speed, hand positioning, and motion dynamics, reflecting personal habits and physical attributes. These variations make the dataset more representative and enhance the model's adaptability.
- **Realistic Framing and Distances:** To closely mimic real-world scenarios, we included videos filmed from a minimum distance of 2 m, replicating typical security camera perspectives. Many of these recordings also feature multiple individuals within the frame, creating more complex and dynamic environments. This allows the model to be tested and trained in conditions resembling real-life surveillance footage, improving its reliability and practical utility.

These targeted enhancements have not only minimized the risk of false positives but also significantly improved the model's ability to accurately recognize the "Signal for Help" gesture across diverse conditions. As part of this effort, we added approximately 500 new videos,

each recorded at various resolutions, later standardized to 12 fps, and lasting between 5 and 15 s to match the “Signal for Help” dataset, bringing the total dataset to around 4500 videos of which approximately 700 depict realistic situations recorded in varied environments and conditions. After this addition, the dataset continues to remain balanced, with half of the videos showing the "Signal for Help" gesture and the other half showing no gesture, allowing for fair and consistent model evaluation.

As shown in Fig. 3, the expanded dataset provides a reliable foundation for testing and implementing the model in practical, real-world applications, such as surveillance systems, where precision and adaptability are crucial.

5. Hands tracking and gesture detection system

Leveraging the dataset and technologies previously described, we developed our hand-tracking-based gesture recognition system. This section details the detection pipeline and its optimization for real-world deployment.

The system employs a streamlined two-step process. Firstly, it tracks the hands and extracts relevant features. Secondly, it performs real-time gesture detection. The design and performance of this optimized system are discussed in detail to highlight its efficiency and accuracy. To enhance the completeness of the proposed system, additional functionalities are introduced to improve its usability in real-world scenarios where detection is integrated with a surveillance system, enabling security operators to respond actively and promptly to alerts. Therefore, as detailed in the following sections, the video feed provided to security operators has been enhanced with features such as optional privacy preservation and intuitive visual feedback, thereby reinforcing the system’s effectiveness in practical applications.

5.1. Hand-Tracking-Based detection system

The proposed methodology for efficiently detecting the “Signal for Help” hand gesture employs a streamlined two-stage pipeline. Initially, video input captured by a camera is processed to perform hand tracking and extract the relevant features. In this stage, the system identifies hand movements and isolates the essential features required for gesture recognition. In the subsequent stage, these features are analyzed in real-time to determine the presence of the "Signal for Help" gesture. Fig. 4 illustrates an overview of the proposed methodology.

5.1.1. Hands tracking and feature extraction

The system starts with capturing frames in real-time with a security camera. Each frame is analyzed to identify the presence of hands using the MediaPipe tool. If no hands are detected in a frame, the system bypasses further processing for that frame, avoiding unnecessary computational effort. This ensures efficient operation, especially in scenarios where hands are absent for extended periods. On the other hand, when hands are present in a frame, the system employs a custom combination of MediaPipe’s Hand Landmarker and the Deep SORT algorithm for real-time hand tracking. MediaPipe identifies hands and their landmarks, while Deep SORT assigns each detected hand a unique ID to maintain continuity across frames. This tracking mechanism ensures the system can differentiate between multiple hands and accurately follow their movements over time, even in dynamic or overlapping scenarios. Once a hand is detected and tracked, MediaPipe at the same time extracts detailed information about its structure.

As mentioned in Section 3, MediaPipe collects 3D spatial coordinates (x, y, z) for 21 key landmarks, such as fingertips, knuckles, and the wrist. These coordinates are used to build a spatio-temporal representation of the hand over time. For each detected hand, the system accumulates landmark data across 20 consecutive frames, resulting in a feature vector of size $21 \times 3 \times 20$. The selection of 20 frames is based on an extensive experimental campaign conducted using videos recorded at 12 fps. In this context, 20 frames correspond to a temporal window of nearly 2 s, which was found to yield optimal accuracy for the “Signal for Help” recognition, as illustrated in Fig. 5. It is important to note that if videos with different frame rates are used, the number of frames should be re-evaluated to maintain the same real-time duration.

Consequently, the 20-frame setup has been adopted for our system.

Before proceeding to the next stage of the pipeline, the system normalizes the data by calculating the distance between the camera and the detected hand. This normalization adjusts the scale of the extracted hand features so that they match the scale used during training. By taking into account the hand’s distance from the camera, the process ensures that variations in apparent hand size do not adversely affect the system’s ability to recognize the gesture. To manage temporal data efficiently, the system uses a sliding window mechanism. Once a hand is present in 20 different frames, its normalized feature vector is forwarded to the next stage of the pipeline for classification. For hands that remain visible across additional frames, the sliding window ensures continuous processing while avoiding redundancy. After the initial batch of data is sent, the system discards the first half of the window’s data and retains the second half as illustrated in Fig. 6. This allows the system to incorporate new frame data while maintaining temporal continuity. When the updated window accumulates data from another 20 frames, a new feature vector is constructed and sent to the classification stage. By leveraging efficient tracking, robust feature extraction, and

Fig. 3. Examples of new images added to the previous “Signal for Help” dataset.

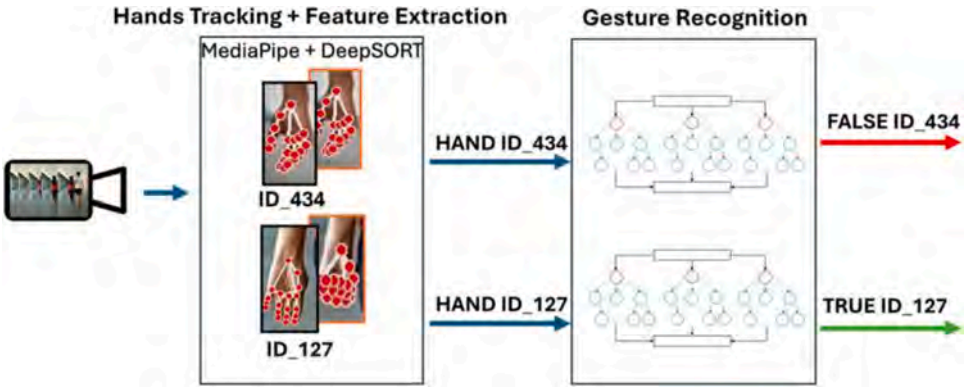

Fig. 4. An overview of our new two-step pipeline model based on hand tracking.

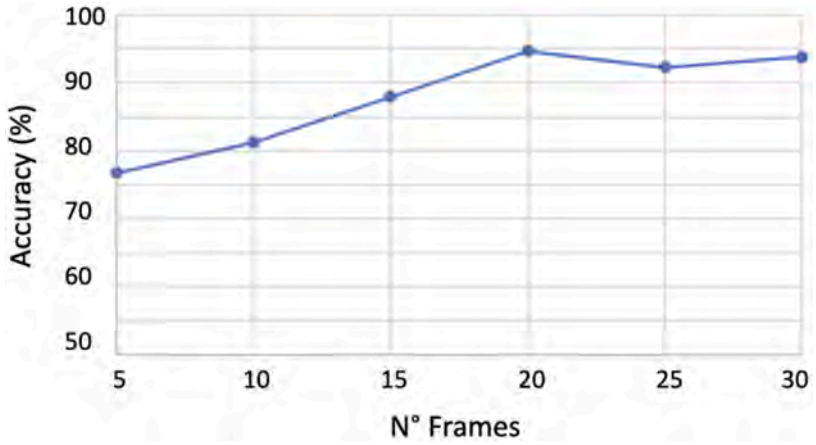

Fig. 5. Accuracy variations based on the number of frames accumulated for each tracked hand.

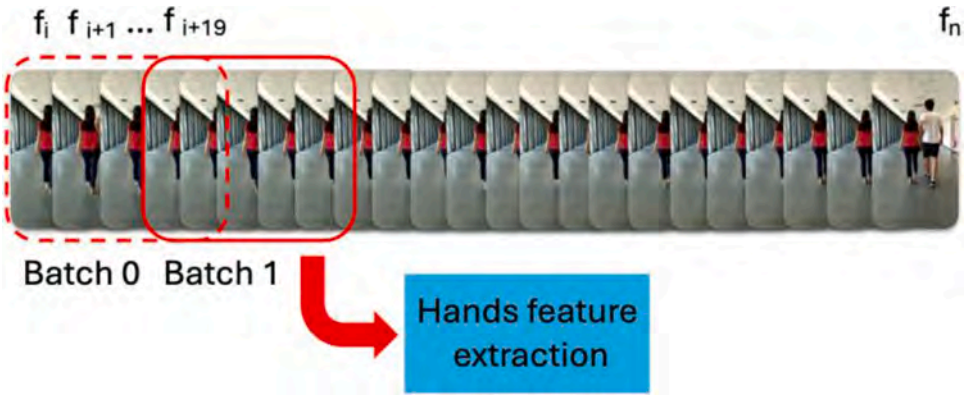

Fig. 6. Representation of the sliding window process on a video.

normalization techniques, this stage ensures that the system provides high-quality input to the second step of the pipeline while maintaining computational efficiency.

5.1.2. Real-Time gesture detection

The second and final phase of the proposed pipeline focuses on leveraging machine learning (ML) techniques to accurately identify the "Signal for Help" gesture from the hand data extracted during the previous step.

The performance of seven ML models, including K-Nearest Neighbors (KNN), Logistic Regression (LR), and many others, was evaluated for this task, with the results detailed in Section 8. Among these, the Random Forest (RF) model emerged as the most effective classifier for addressing the problem. The RF model is a powerful ensemble learning method that constructs multiple decision trees during training and aggregates their outputs to produce reliable predictions. This architecture allows it to handle diverse feature sets effectively, minimizing the risk of overfitting and enabling strong generalization to unseen data.

Another important aspect of the model's performance is determining the proper threshold used for classification. By adjusting this threshold, it is possible to optimize the balance between false positives and false negatives, tailoring the model to the specific requirements of the application. As shown in Fig. 7, a threshold of 0.5 was selected on the validation set as the decision boundary for classifying gestures as "Signal

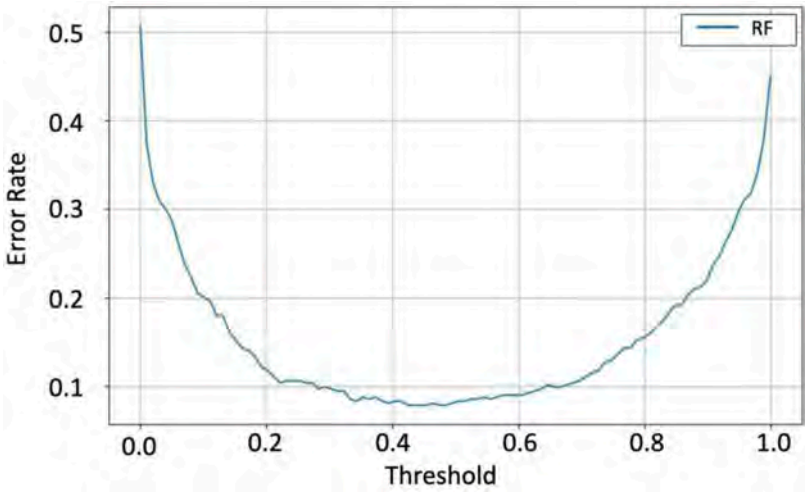

Fig. 7. Variation of error rate with changes in threshold using random forest classifier.

for Help." By setting the threshold at this value, the system effectively minimizes both false positives and false negatives, providing the best trade-off between accuracy and robustness.

To further enhance detection accuracy and minimize false positives, a temporal redundancy mechanism was integrated into the system. This mechanism ensures that an alert is triggered and sent to the mobile application only after a double-check process is satisfied. Specifically, the system verifies that the same hand, identified by an unique ID, consistently performs the "Signal for Help" gesture across two distinct temporal windows.

This approach adds an extra layer of reliability, preventing isolated detections from activating the alert, and ensures that only verified and consistent gestures are recognized and acted upon. Once the gesture is consistently detected, the system activates an alert and automatically sends a 10-second video of the event to the server connected with the mobile application.

By combining the RF model's ability to handle complex feature sets with the temporal double-check mechanism, the final stage of the pipeline achieves high reliability and adaptability. These enhancements optimize the pipeline's performance, making it a practical and dependable solution for recognizing the "Signal for Help" gesture in diverse real-world application.

5.2. Additional features

The "Signal for Help" detection system incorporates advanced visual representation and privacy-preserving features into the video sent to the security operator. These enhancements ensure that operators receive clear and intuitive visual cues, enabling them to quickly identify and respond to the detected gestures. At the same time, the system addresses privacy concerns by offering an optional feature that obfuscates personal details, making it ideally suited for deployment in environments where robust data protection is essential.

This design balances the need for actionable information with the preservation of individual anonymity, making the system suitable for deployment in sensitive environments.

5.2.1. Visual representation of hand bounding boxes

In crowded environments, identifying the source of an alert can become challenging due to the presence of multiple tracked hands. To address this, we implemented a real-time visual feedback mechanism that highlights the hand performing the "Signal for Help" gesture. The system dynamically displays bounding boxes around each detected and tracked hand. When a hand executes the gesture, its bounding box changes to green, offering an immediate visual cue for the operator to identify the person requesting help. This intuitive color change simplifies the identification process in complex scenes, reducing response time and enhancing situational awareness.

The bounding boxes remain synchronized with hand movements, providing a seamless and dynamic view of tracked hands as represented in Fig. 8. This feature ensures that operators can focus on the relevant individual without confusion, even in scenarios with multiple overlapping alerts.

5.2.2. Face blurring option for privacy preservation

To address privacy concerns, especially in public or sensitive environments, the system includes an optional face-blurring feature that can be enabled during operation. This feature integrates a dedicated neural network for real-time face detection and applies a blurring algorithm to obscure facial features before storing the video data.

When the face-blurring feature is enabled, the system processes each video frame to detect faces using a specialized detection model. Once a face is identified, it is automatically blurred, ensuring that individuals in the video remain unidentifiable. This privacy-focused design, as highlighted in Fig. 8, makes the system suitable for deployment in environments with stringent data protection requirements.

However, enabling this feature introduces an additional processing step, which slightly increases the computational load. Detecting and blurring faces adds to the system's workload, potentially reducing performance. To address this trade-off, the system offers a configurable setting that allows users to prioritize either privacy or performance based on the application's needs.

6. Mobile application

Once completing the recognition framework for the "Signal for Help" gesture, we identified the need for a final step: crafting a secure solution that goes beyond simple alerts by establishing a direct communication channel between those in distress and security authorities.

This need led to the creation of a mobile application designed specifically for security personnel, enabling them to receive real-time notifications and respond to emergencies with minimal reaction time. To ensure efficiency and ease of use, the mobile application was built with a user-friendly interface, prioritizing clarity and accessibility. Every interaction was carefully designed to provide immediate access to critical information, allowing security staff to quickly assess situations and take appropriate action without unnecessary delays. This chapter explores the various aspects of the mobile app's development, including the architectural choices made and the key functionalities implemented.

Fig. 8. Visualization of the Green Bounding Box Highlighting the Detected Signal and Face Blurring for Enhanced Privacy Protection.

6.1. System architecture

The architecture of the mobile application was designed to facilitate seamless communication between surveillance devices and security personnel, ensuring rapid and efficient response to critical incidents. By facilitating real-time transmission of alerts, the system minimizes delays between the detection of an emergency and the necessary intervention.

As illustrated in Fig. 9, the process begins with User and Device Registration, an initial step where a unique registration token is generated for the security personne''s device. This token is created using Firebase Cloud Messaging (FCM) services and acts as an identifier for push notification delivery. The registration process is required only during the use''s first login to the application or if the previously generated token has expired. The token is securely transmitted to the server and stored in a relational database. This step ensures that the system can accurately identify and communicate with each device registered to the application.

Following registration, the User Authentication phase is initiated. During this phase, the application requires the user to authenticate their credentials, including a username, password, and the previously

Fig. 9. Mobile application architecture flow.

generated token. This information is transmitted to the backend for verification. Using a secure authentication protocol, the server validates the credentials and associates the registration token with the authenticated user. This ensures that only authorized personnel can access the system and receive sensitive notifications. Additionally, the server performs further checks to validate the use''s session. It ensures that the token matches the stored records and verifies that it is not obsolete.

Once the verification is complete, the system moves to the Notification Monitoring phase, a state where the security guar''s device maintains an active connection to FCM servers. The use of FCM ensures minimal resource consumption on the device while maintaining readiness to receive notifications in real-time. This mechanism is essential for achieving the low-latency alert delivery required in critical security applications.

When an alarm is detected, the process advances to the Alert Generation phase. At this stage, the web server processes incoming distress signals from surveillance devices. Upon receiving a request containing a JSON payload with metadata, the system generates an alert object that includes critical details about the incident. This alert object is stored in the database for future reference and traceability. The connection between the mobile application and the detection system will be discussed in greater detail in Section 9.

Finally, in the Real-Time Alert Delivery phase, the server forwards the alert data to the FCM backend. Using the registration tokens of the intended recipients, FCM generates a message and distributes the notification to the selected devices. Upon receiving the notification, the mobile application extracts and displays the alert details, allowing the guard to take swift and informed action.

This detailed process not only ensures robust communication between system components but also maintains high reliability and scalability, making it suitable for handling multiple simultaneous alerts across large-scale deployments.

6.2. Key features of the mobile application

Building on the system architecture, the mobile application integrates a set of features designed to optimize security operations and enhance coordination among security staff. Each functionality has been carefully implemented to ensure rapid access to critical information, seamless communication, and efficient alert management.

The system starts with secure user authentication, allowing access through credentials provided by system administrators. During registration, the user is assigned a unique password, which is sent via email and must be reset upon the first login. In case of lost or forgotten credentials, a recovery mechanism allows users to initiate a password reset by entering their registered email after which the application sends a secure link for password reset.

Once authenticated, users are directed to the main screen shown in Fig. 10, which consolidates all essential features. The home screen provides real-time information on the user's service status (e.g., "On Duty," "Busy," "Off Duty") and displays prominent notifications for unresolved alerts or actions requiring immediate attention. Through intuitive navigation options, users can access features such as messages, contacts, historical records, and service configurations. The design emphasizes accessibility, ensuring that key features are within a single tap's reach.

At the core of the mobile application is its real-time notification system, where emergency alerts are delivered directly to users' devices, regardless of whether the app is open, running in the background, or closed. Notifications are accompanied by a concise summary, including details such as the alarm's location, the time of the event, the surveillance device that triggered the signal, and a 10-second video showing the situation in which the request for help occurred. Users can interact with these notifications by choosing to accept or dismiss them. Accepted notifications are logged as new actions within the application, while dismissed alerts are archived with a "dismissed" status. The alert

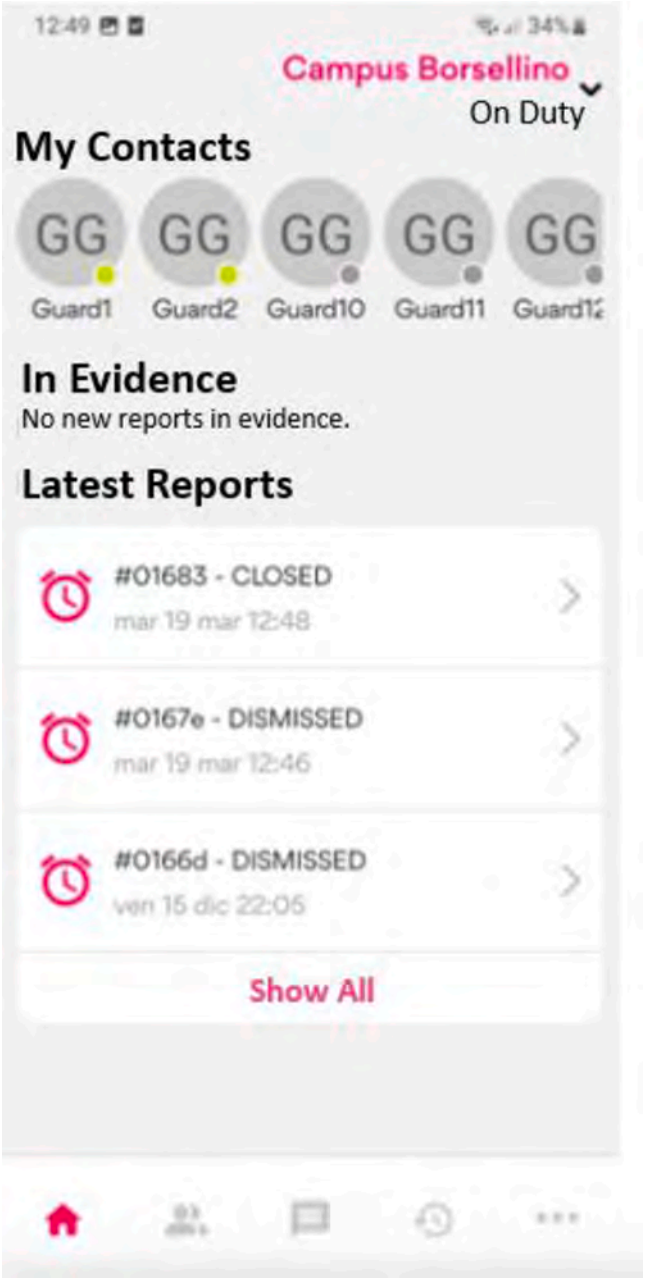

Fig. 10. Home screen overview displaying user status, notifications, and quick access to key features.

management module allows users to explore the details of each alert. As illustrated in Fig. 11, the flow begins with the notification and progresses to detailed information displayed within the application.

All the received alerts are moved to the "History" section where they are categorized into three distinct statuses: New, Dismissed, or Closed. This system ensures a complete and verifiable record of all interactions with emergency alerts. The *New* status identifies a newly received notification that has not yet been addressed by the user. The *Dismissed* status is used for alerts that have been recognized as false positives or irrelevant notifications, which the user has intentionally dismissed. The *Closed* status is applied to alerts that have been resolved by the user. To close an alert, the user updates its status to *Closed* and selects a corresponding resolution category such as *Handled Internally* when the issue was resolved without external intervention, *False Alarm* when the alert was triggered by a non-critical event or technical error, or *Police Involved* when law enforcement was required to intervene. Users also have the

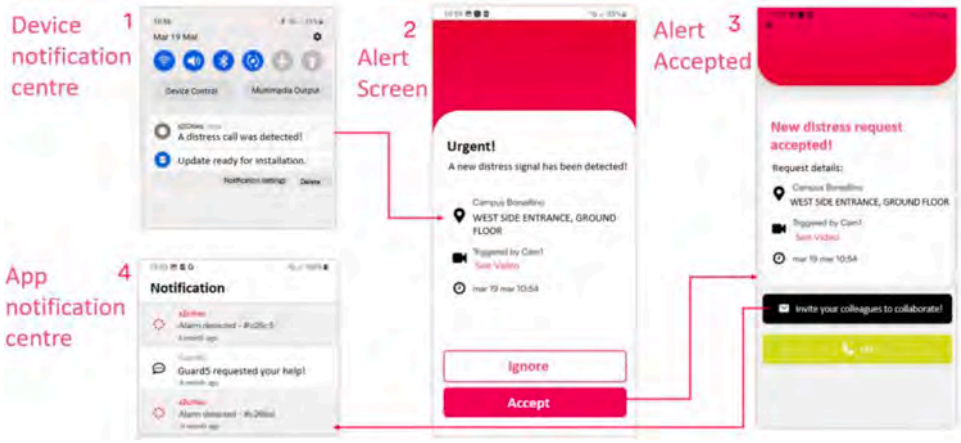

Fig. 11. Workflow of the real-time alert system, from notification delivery to alert acceptance, including event details and user actions.

option to attach a detailed report to the closed alert, ensuring that all relevant information is documented and accessible for future reference or audits.

The application also includes a robust contact management system that allows users to access a directory of colleagues working within the same organization or building. Each contact is displayed with their current service status (e.g., "On Duty," "Busy") and key details such as phone number and email. Fig. 12 illustrates the searchable directory, which includes filters by name, phone, or email, simplifying the process of identifying and contacting colleagues for assistance during emergencies.

Another key collaboration feature is the messaging system, which serves as a centralized hub for all communications. Users receive emergency messages generated by the system, as well as peer-to-peer messages from colleagues. Emergency messages are flagged for immediate attention, while messages from colleagues facilitate coordination among team members. Fig. 13 highlights how unread messages are visually marked until opened, ensuring that critical communications are not overlooked.

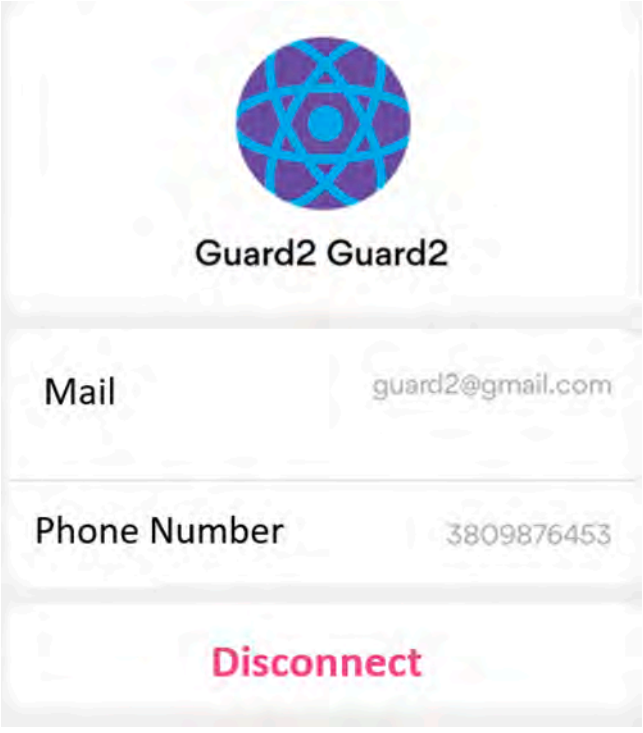

Fig. 12. Contact interface showing user details, including email and phone number.

By seamlessly integrating authentication, real-time notifications, alert tracking, contact management, and messaging into a single platform, the mobile application provides a robust and intuitive tool that enhances situational awareness, reduces response times, and fosters effective communication. These key features not only improve security operations but also ensure that every alert is logged, every action is accounted for, and every emergency is addressed with precision and efficiency.

7. Connection between "Signal for help" recognition system and mobile app

Ensuring seamless communication between the detection system and mobile app is key to timely intervention. This section explains how we established a secure and efficient integration. The primary goal was to ensure seamless data exchange, enabling a fully integrated and functional solution. This connection is built on a resilient Amazon S3 infrastructure, ensuring fast, reliable communication and secure storage for sensitive data.

When the system detects the "Signal for Help" gesture, it generates a 10-second video clip capturing the key moment. This clip, along with associated metadata, is automatically uploaded to a private and secure storage bucket on Amazon S3.

Once the upload is complete, the system triggers a notification to the mobile application. This notification contains a JSON payload with key details, including the video file ID and the serial number of the surveillance camera that captured the footage. The serial number is linked to the camera's geographic location, providing accurate information about the incident's localization.

At the same time, the mobile application's backend processes the video file ID and retrieves the footage directly from Amazon S3. Security personnel can then review the video within the app, assess its relevance, and decide whether intervention is necessary.

A critical aspect of this connection is the strong emphasis on data security. The video files stored in Amazon S3 are accessible only to authorized personnel, ensuring controlled access and safeguarding sensitive information. Furthermore, video files are automatically deleted 24 h after upload unless flagged as important by authorities for further review or investigation. This automatic deletion mechanism balances privacy concerns with operational requirements, ensuring minimal retention of sensitive data.

By integrating Amazon S3 with the backend of the mobile application, this architecture provides a scalable, secure, and efficient solution for managing video and location data. The result is a reliable connection between the gesture recognition system and the mobile app, enabling

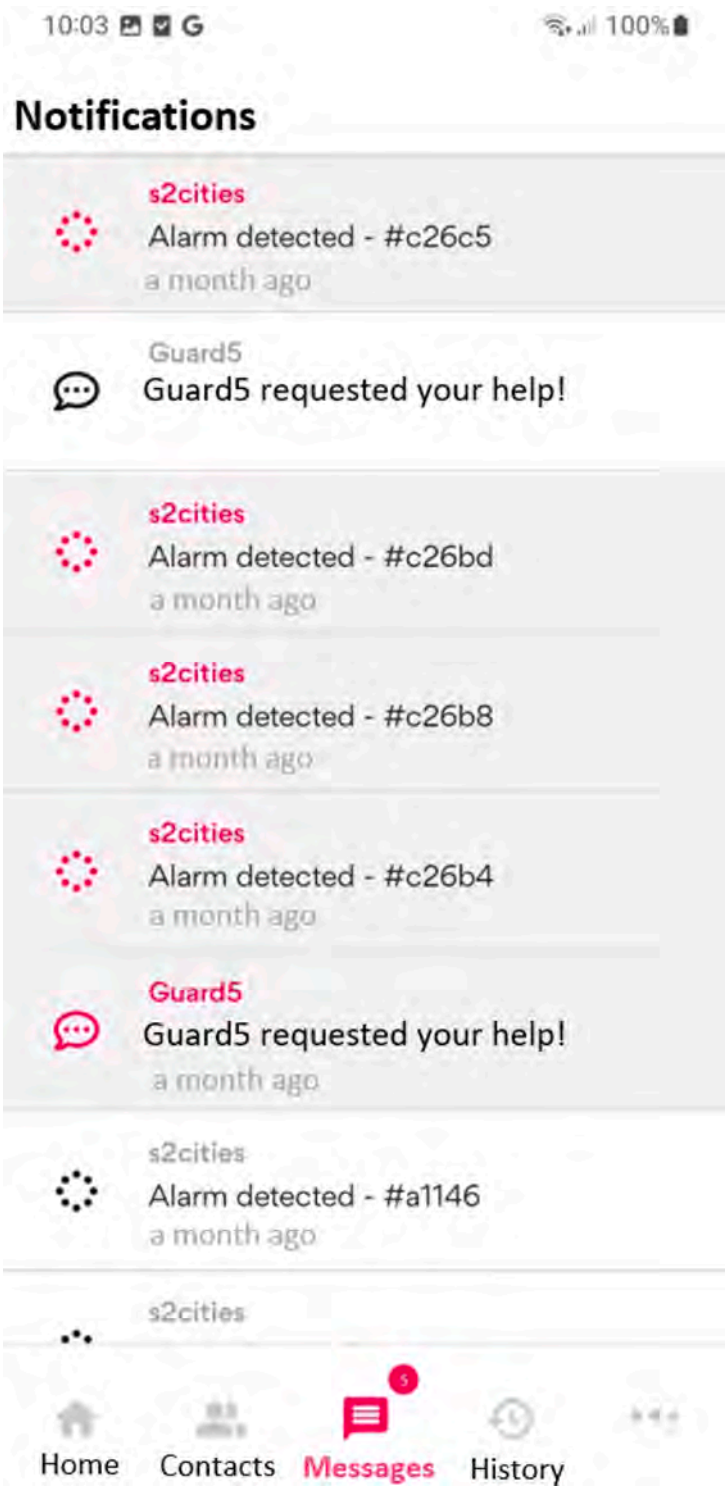

Fig. 13. Messaging system interface displaying emergency alerts and peer-to-peer communications, with visual indicators for unread messages to prioritize critical updates.

security personnel to receive actionable alerts with precise information, supporting swift and informed decision-making.

In Fig. 14 it is possible to observe the complete flow diagram of the process. This sequence of steps was chosen to keep the system optimized, balancing accuracy, speed, resource usage, and operational reliability under diverse operational conditions.

8. Experimental results

The proposed approach was subjected to a evaluation test structured in two key stages. The first stage focused on selecting the most suitable machine learning classification model from a pool of diverse algorithms, including Random Forest (RF) (Breiman, 2001), Support Vector Machine (SVM) (Ben-Hur, et al. 2001), Logistic Regression (LR) (Bishop, 2006), K-Nearest Neighbors (KNN) (Altman, 1992), Multilayer Perceptron (MLP) (Singh & Sachan, 2014), AdaBoost (Ada) (Collins, et al.

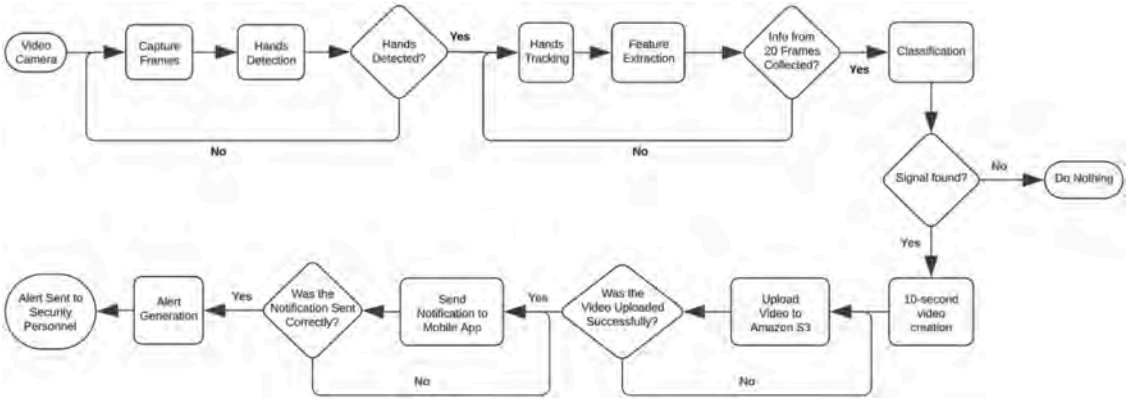

Fig. 14. Complete flow diagram of the "Signal for Help" detection and notification process.

2002), and Gaussian Naive Bayes (GNB) (Rish, 2001). Each model's performance was analyzed based on its ability to maximize classification accuracy while minimizing false positives and false negatives. Once the optimal model was identified, the second stage involved hardware testing to assess system adaptability across different computational platforms. This step included evaluations on devices ranging from high-performance machines equipped with Apple M3 chips to resource-constrained environments like NVIDIA Jetson Orin Nano and NVIDIA Jetson AGX Orin. This dual-stage approach ensured both the robustness and portability of the system.

To quantify and compare the performance of the models, the following performance metrics were employed:

- ***Accuracy:*** It measures the model's overall ability to correctly classify instances from all available classes (Fawcett, 2006; Brown & Davis, 2006). Accuracy is important to ensure reliable performance across both gesture and non-gesture instances, minimizing incorrect classifications and enhancing overall trust in the system.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

- ***Precision:*** This measure evaluates how accurately the model predicts positive instances for a given class (Powers, 2011). Good precision is crucial in surveillance applications, as it reduces unnecessary alerts and helps security operators focus their attention only on genuine distress signals.

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

- ***Recall:*** It measures the model's capability to correctly identify all positive instances for a specific class (Powers, 2011). High recall is critical in surveillance scenarios, as failing to recognize gestures (false negatives) may delay or even prevent necessary intervention, potentially placing individuals at greater risk.

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

- ***F1-Score:*** This measure provides a balanced measure that takes into account both precision and recall, offering a single, unified metric to evaluate classifier performance (Sasaki, 2007). It is essential for achieving an optimal trade-off between reducing false alerts and ensuring real gestures are not overlooked.

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

- ***Area Under the Curve (AUC):*** This indicator is used to assess the performance of binary classification models (Fawcett, 2006)[37]. It measures the model's overall capacity to distinguish between genuine distress signals and non-threatening gestures across different classification thresholds. A higher AUC indicates greater reliability and robustness of the model under varied real-world conditions.

Another metric that could be interesting to include is the "easiness" discussed in Kishida and Nakayama, (2019). However, this measure depends both on the bias present in the dataset, which we have already tried to minimize, and on the use of SGD, which our models did not employ, making this metric inapplicable in this work. However it remains a valuable indicator to consider in future evaluations.

8.1. Detection system results

The first experimental evaluation is conducted on the ML models aiming at finding the optimal gesture classifier. Indeed, the detection system for the "Signal for Help" gesture incorporates a classifier that processes the features extracted from the hand detection and tracking stage to determine the presence or absence of the gesture. To identify the best-performing classifier, seven different ML models were evaluated. The performance of each model was assessed using key metrics introduced earlier, including accuracy, AUC, precision, recall, and F1-score, as reported in Table 2. The results indicate that the Random Forest (RF) model consistently outperformed the others, achieving an accuracy of 94.72 %, an F1-score of 0.9529, and a recall of 0.9707. These findings underscore the RF model's balance between precision and recall, which is crucial for detecting the "Signal for Help" gesture, as minimizing both false positives and false negatives is essential for ensuring reliable recognition in real-world scenarios.

Other classifiers, such as SVM and MLP, also produced competitive results. However, they showed slight reductions in precision and recall, resulting in lower F1-scores compared to RF. In addition to these metrics, further analysis was performed by examining the error rate of each

Table 2
Performance metric comparison for various models.

Classifier	Accuracy	AUC	Precision	Recall	F1-score
RF	0.9472	0.9489	0.9355	0.9707	0.9529
KNN	0.9101	0.9333	0.8615	0.8960	0.9357
LR	0.9101	0.9105	0.8872	0.8987	0.9253
Ada	0.9078	0.9101	0.8821	0.8958	0.9279
MLP	0.9194	0.9211	0.8974	0.9091	0.9319
GNB	0.8157	0.8443	0.7231	0.7790	0.8977
SVM	0.9241	0.9264	0.9021	0.9141	0.9232

classifier as a function of the threshold used for classification. As shown in Fig. 15, almost all models exhibit a lower error rate when the threshold is set around 0.5. In particular, RF, SVM, Logistic Regression, and K-Nearest Neighbors show significantly reduced error rates as the threshold approaches 0.5, maintaining low and stable error rates. This further supports the robustness of these models, particularly RF, across different classification thresholds.

The confusion matrix for the RF model, shown in Fig. 16, further illustrates its classification performance, showing minimal misclassifications. This outcome highlights the RF model's ability to reduce errors, ensuring greater accuracy and robustness in detecting the "Signal for Help" gesture.

Compared to current state-of-the-art solutions (Buccellato, De Sio, Vacca & Azimi, 2024)[19], our system demonstrates a more practical and deployable approach, ensuring reliability in real-world applications. This makes it a robust solution adaptable to various surveillance scenarios.

8.2. Detection system performance

The efficiency of the model was also tested on different hardware platforms, specifically the Apple M3, NVIDIA Jetson Orin Nano, and NVIDIA Jetson AGX Orin.

The experimental scenario simulates the deployment of the detection system in various real-world environments. The Apple M3 is well suited in centralized infrastructures. In these scenarios, its high computational power allows for real-time processing of large volumes of video data, ensuring thorough analysis and the timely activation of alarms in case of an emergency. This configuration is perfect when cameras are connected to central servers that synergistically manage the information flow, optimizing resources for recognizing the signal for help.

Conversely, for implementations where processing must occur directly on the device, such as in surveillance cameras installed in resource-limited environments or remote locations, NVIDIA Jetson platforms offer significant advantages. In particular, the Jetson Orin Nano and the Jetson AGX Orin are designed to operate on the edge, reducing latency and minimizing costs associated with transmitting data to centralized infrastructures. This embedded approach is essential for ensuring an immediate and reliable response in critical situations, guaranteeing that the recognition of the "signal for help" occurs without delays. Moreover, the native support for MediaPipe on Jetson devices optimizes task management by fully leveraging the hardware architecture, ensuring that even complex applications like the recognition of "Signal for Help" operate efficiently in demanding operational scenarios.

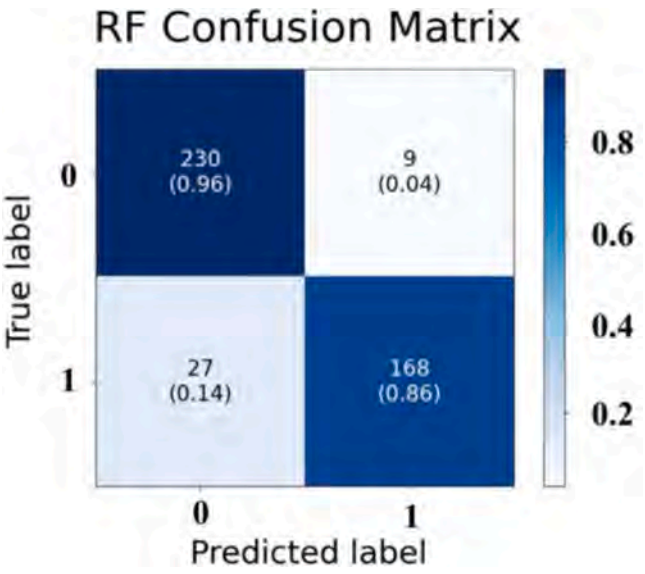

Fig. 16. Confusion matrix of Random Forest model.

Table 3 provides a summary of the hardware specifications for each platform, including details such as the GPU, CPU, clock frequency, memory, and power consumption.

On the Apple M3, the model demonstrated strong computational efficiency, delivering high performance compared to the other devices.

Table 3
Hardware specifications comparison.

Device	GPU	CPU	Clock Frequency (GHz)	Memory (GB)	Power Consumption (W)
Mac M3	10-core Apple Silicon	8-core Apple Silicon	3.2	16	20
Jetson Orin Nano	NVIDIA Ampere, 512 cores	6-core Arm Cortex-A78AE	1.5	8	7–15
Jetson AGX Orin	NVIDIA Ampere, 2048 cores	12-core Arm Cortex-A78AE	2.2	64	15–60

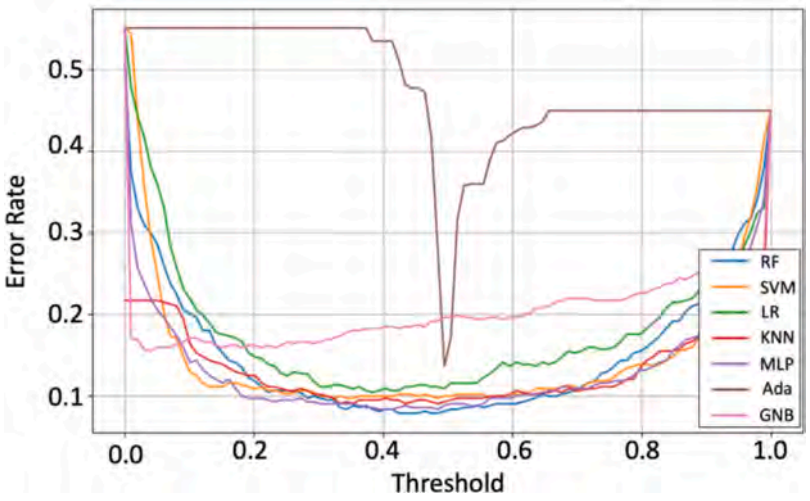

Fig. 15. Comparison of error rate variation with changes in threshold across classifiers.

As shown in Table 4, the Jetson Orin Nano and Jetson AGX Orin exhibited worse results. This is primarily due to the inability to fully leverage the computing power of the two Jetson devices, as the necessary NVIDIA libraries for running MediaPipe on their GPUs are not available, which resulted in less promising performance.

As noticeable, the Apple M3 excelled with the fastest inference time and lowest memory usage, making it highly suitable for real-time edge applications or remote processing. Although the Jetson Orin Nano and Jetson AGX Orin faced limitations due to CPU-based processing, the AGX Orin outperformed the Nano, demonstrating shorter inference times and lower memory usage. Despite these limitations, the hand-tracking model remained efficient and adaptable across the platforms, confirming its potential for deployment in various real-world environments.

Table 4
Comparison of inference time and memory consumption across devices for the hand-tracking-based model.

Device	Inference Time (s)	Memory (GB)
Mac M3	0.067	0.4
Jetson Orin Nano	0.471	1.2
Jetson AGX Orin	0.343	0.6

9. Conclusions and future works

This work has introduced an innovative real-time system for detecting the "Signal for Help" gesture, providing an effective and robust solution to support victims of violence.

In this system, cameras operating at a minimum of 12 fps were used for testing, and we assume that individuals in danger are aware of and able to perform the "Signal for Help" gesture.

After facing several challenges during development, such as detecting multiple hands that occupy only a few pixels in crowded scenes, reliably processing low-resolution camera feeds, and consistently tracking each hand over time, our system still runs in real time with very low computational cost.

To achieve this, we employ an advanced two-stage pipeline, combining a hand-tracking system with a Random Forest–based classifier, which has demonstrated superior performance compared to other methodologies. The system achieves 94 % accuracy and delivers inference times between 0.067 s and 0.5 s, depending on the device, ensuring computational efficiency and the capability to operate on resource-constrained hardware. Through the expansion and refinement of the dataset, the model effectively adapts to realistic scenarios, significantly reducing false positives and enhancing the reliability of gesture recognition even under complex environmental conditions. Additionally, the integration with a dedicated mobile application enables seamless notification flows and real-time interventions, further strengthening the system's effectiveness in protecting victims.

This framework is intended to serve as a universal safety tool, available at any hour and in any situation, so that, with a single intuitive gesture, anyone can summon assistance immediately. Moreover, as our system is designed around the detection of hand gestures, its underlying architecture is broadly applicable to multiple domains where gesture-based communication, safety, or intervention is relevant, such as healthcare and elderly care, by monitoring gestures indicating distress or assistance requests, especially in care facilities or smart homes; industrial safety, enabling workers in noisy manufacturing or construction environments to signal danger or request support through specific hand gestures; and education and inclusivity, by supporting sign language translation or non-verbal communication tools for students with hearing or speech impairments, enhancing accessibility in classrooms.

As future work, we aim to expand our system in five main directions: (1) introduce additional distress signals, such as gestures to call an ambulance or alert critical services, covering a broader range of emergency scenarios; (2) fully leverage GPU acceleration to achieve real-time processing and improved efficiency under intensive workloads; (3) enhance detection robustness for challenging conditions, such as low-light and long-range environments; and (4) release an iOS-compatible application version, supporting cross-platform accessibility and continuous feature improvements for security personnel.

Our development plan includes:

- Phase 1: Expand the gesture set and retrain the system on new distress signals, alongside initial integration of sentiment analysis components.
- Phase 2: Deploy GPU-accelerated pipelines for optimized real-time processing, particularly targeting edge-device performance.
- Phase 3: Develop robust algorithms to ensure reliable detection under adverse environmental conditions, including extensive low-light and long-range testing.
- Phase 4: Complete the development, testing, and launch of an iOS application, validated in collaboration with local safety organizations.

These steps will be addressed through an agile development cycle, in collaboration with partners from academia and public safety institutions, ensuring progress remains both focused and impactful.

In its current form, the system is reliable, robust, and ready for real-world use. It performs well on devices with limited resources, making it practical and adaptable for diverse scenarios. These strengths ensure that the system can provide real value in protecting individuals and responding effectively to emergencies. This research demonstrates the meaningful role technology can play in addressing urgent social challenges, continuing to develop and refine this system, we can ensure that no "Signal for Help" goes unnoticed, contributing to a safer and more supportive world for those in need.

CRediT authorship contribution statement

Federico Buccellato: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Eleonora Vacca:** Resources, Writing – review & editing, Visualization, Supervision. **Sarah Azimi:** Conceptualization, Resources, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Corrado De Sio:** Resources, Writing – review & editing, Visualization, Supervision. **Luca Sterpone:** Writing – review & editing, Visualization, Supervision.

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

We wish to express our sincere appreciation to the members of the S2CITIES project at Alta Scuola Politecnica for their dedicated work in assembling the first version of the "Signal for Help" dataset. Additionally, we would also like to extend our thanks to Alessandra Palma, a student from the Politecnico di Torino, for her valuable assistance in developing the mobile application.

Data availability

The data that has been used is confidential.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.

Amazon Web Services, "Amazon Simple Storage Service (Amazon S3) - secure, durable, & scalable storage." Accessed January 2025. Available at: Https://aws.amazon.com/s3/.

Azimi, S., De Sio, C., & Sterpone, L. (2023). Enhanced video surveillance systems for 'signal for help' Detection on edge devices. In *IEEE International Symposium on Technology and Society (ISTAS)*.

Azimi S., De Sio C., Carlucci F., and Sterpone L. "Fighting for a future free from violence: A framework for real-time detection of 'signal for help'," ISSN 2667-3053, 2023.

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research, 2*, 125–137.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Brown, C. D., & Davis, H. T. (2006). Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems, 80*(1), 24–38.

Buccellato, F., De Sio, C., Vacca, E., & Azimi, S. (2024). *Enhancing security of smart cities with 'Signal for help' recognition system*. IEEE International Smart Cities Conference.

Collins, M., Schapire, R. E., & Singer, Y. (2002). Logistic regression, AdaBoost and Bregman distances. *Machine Learning, 48*, 253–285.

Elliott, G., Meehan, K., & Hyndman, J. (2021). *Using cnn and tensorflow to recognize 'Signal for help' hand gestures*. IEEE.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Junping, H., & Siping, X. (2021). Gesture recognition based on YCbCr color space and neural network. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)* (pp. 738–741). https://doi.org/10.1109/ICSP51882.2021.9408765

Kang, J., & Kwak, S. (2015). Violent behavior detection using motion analysis in surveillance video. *Journal of Broadcast Engineering, 20*(3), 430–439. https://doi.org/10.5909/jbe.2015.20.3.430

Kishida, Ikki, & Nakayama, Hideki (2019). Empirical study of easy and hard examples in cnn training. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019, Proceedings, Part IV 26*. Springer International Publishing.

Krishnan, G., Joshi, R., 'Connor, T. O., Pla, F., & Javidi, B. (2021). An overview of hand gesture recognition in degraded environments using three-dimensional integral imaging and deep neural networks. *OSA Imaging and Applied Optics Congress*.

Manias, George, Kiourtis, Athanasios, Mavrogiorgou, Argyro, & Kyriazis, Dimosthenis (2022). Multilingual sentiment analysis on Twitter data towards enhanced policy making. In *18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)* (pp. 325–337). https://doi.org/10.1007/978-3-031-08337-2_27. Junhal-04668667.

Margapuri, V., Penumajji, N., & Neilsen, M. (2021). PiBase: An IoT-based Security System using Raspberry Pi and Google Firebase. *arXiv preprint*. arXiv:2107.14325.

Mavrogiorgos, K., Kiourtis, A., Mavrogiorgou, A., Menychtas, A., & Kyriazis, D. (2024). Bias in machine learning: a literature review. *Applied Sciences, 14*, 8860. https://doi.org/10.3390/app14198860

MediaPipe Hands solution, Hand landmarks detection. Available at: Https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker?hl=it.

MilanoToday. Violenza sessuale in centro a Milano: Ragazza si salva col segnale anti-abuso. Available at: Https://www.milanotoday.it/cronaca/violenza-sessuale-piazza-scala.html. January 2025.

Mohd, N. H., Abd Wahab, M. H. B., & Ariffin, S. K. (2010). Motion Detection Notification System by Short Messaging Service Using Network Camera and Global System for Mobile Modem. *arXiv preprint*. arXiv:1006.2798.

Ossino, Andrea (2023). Il mistero del segnale in codice non capito dal cameriere, la lite e l'assedio in bagno. Così è morta Martina Scialdone. *La Repubblica*. January 15 https://roma.repubblica.it/cronaca/2023/01/15/news. January 2025.

Peng, Y., Tao, H., Li, W., Yuan, H., & Li, T. (2020). Dynamic gesture recognition based on feature fusion network and variant ConvLSTM. *IET Image Processing / IET, 14*, 2480–2486. https://doi.org/10.1049/iet-ipr.2019.1248

Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Research, 2*(1), 37–63.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI workshop*.

Rodrigo-Ginés, Francisco-Javier, Carrillo-de-Albornoz, Jorge, & Plaza, Laura (2024). A systematic review on media bias detection: What is media bias, how it is expressed, and how to detect it, expert systems with applications. *Part C, 237*, Article 121641. https://doi.org/10.1016/j.eswa.2023.121641. ISSN 0957-4174.

S2CITIES - Toward Smart and Safe Cities: Exploiting Surveillance Videos for Real-time Detection of "Signal for Help" Retrieved from https://www.asp-poli.it/s2cities/.

Sasaki, Y. (2007). The truth of evaluation metrics: Precision, recall, and F-measure. *Journal of Machine Learning Research, 2*, 1–10, 3.

Shafique, U. Signal for help - Hand Gesture detection (SFH-Dataset). Kaggle. https://www.kaggle.com/datasets/umairshafique/sfh-dataset.

Singh, G., & Sachan, M. (2014). Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition. In *IEEE International Conference on Computational Intelligence and Computing Research* (pp. 1–5).

The Guardian. (2021). Teen rescued after showing domestic violence hand signal known on TikTok, police say. https://www.theguardian.com/us-news/2021/nov/08/teenager-rescued-after-showing-domestic-violence-hand-signal-to-passing-motorist-police-say.

The Rakyat Post. (2023). Woman rescued after making 'Signal for Help' hand gesture at a carnival in PJ. https://www.therakyatpost.com/news/2023/09/15/watch-woman-rescued-after-making-signal-for-help-hand-gesture-at-a-carnival-in-pj-found-to-be-a-victim-of-domestic-abuse.

Thejowahyono, N. F., Setiawan, M. V., Handoyo, S. B., & Rangkuti, A. H. (2022). Hand gesture recognition as signal for help using deep neural network. *International Journal of Emerging Technology and Advanced Engineering, 12*(2), 1–5. February.

Tran, Dinh-Son, Ho, Ngoc-Huynh, Yang, Hyung-Jeong, Baek, Eu-Tteum, Kim, Soo-Hyung, & Lee, Gueesang (2020). Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network. *Applied Sciences, 10*(2), 722. https://doi.org/10.3390/app10020722

Traoré, A., & Akhloufi, M. A. (2020). 2d bidirectional gated recurrent unit convolutional neural networks for end-to-end violence detection in videos. *Lecture Notes in Computer Science*, 152–160. https://doi.org/10.1007/978-3-030-50347-5_14

Wang, J., Zhao, D., Li, H., & Wang, D. (2024). Lightweight violence detection model based on 2d cnn with bi-directional motion attention. *Applied Sciences, 14*(11), 4895. https://doi.org/10.3390/app14114895

Wojke N., Bewley A., and Paulus D. "Simple online and real-time tracking with a deep association metric," arXiv:1703.07402, 2017.

World Health Organization. (2021, March 9). Devastatingly pervasive: 1 in 3 women globally experience violence. Retrieved March 15, 2025, from https://www.who.int/news/item/09-03-2021-devastatingly-pervasive-1-in-3-women-globally-experience-violence.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., Grundmann, M. "MediaPipe hands: On-device real-time hand tracking," arXiv: 2006.10214, 2020.