AVE Trends in Intelligent Computer Letters

CovertSign AI: A Lightweight Vision Framework for Silent SOS Signals

Pavithra Babu[1,*], A. Shree Harini[2], S. Sowmiya[3]

[1,2,3]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India.
pavithra.babu.cse@sathyabama.ac.in[1], shreeharini027@gmail.com[2], sowmiyas0806@gmail.com[3]

*Corresponding author

Abstract: Hazard-to-first-responder time affects the security and survival of international individuals. Traditional emergency response systems work, but victims need independence, a voice amplifier, or a way to call for help. Domestic violence, human trafficking, and fast jail threaten victims. Lightweight, wide computer vision platform CovertSign AI wants smart CCTV cameras for emergency response. The global Signal to Help, a hand motion to avoid aggressors, is recognised in real time. The MediaPipe Hands 3D landmark extraction model converts visual hand movements into 63-dimensional feature vectors for 21 major joints. For reliability and cheap processing cost, Researchers use a machine learning classifier ensemble. Automated Hand Gesture Recognition and Performance Analysis: Our gradient-optimised Random Forest model outperformed SVM and K-Nearest Neighbour classification methods on over 1,000 samples across varying lighting and distances. GMM and SVM optimise 30 fps edge computing on consumer-grade hardware with 23% CPU consumption. Automatic multi-channel warnings are issued when a distress signal is at least 90% certain. Local visual detection and automated high-priority SMTP email warnings with live timestamps and high-resolution scene captures are used. From gesture start to emergency notice, the pipeline takes 5–10 seconds. With its hardware-agnostic, cheap solution, CovertSign AI can help vulnerable people in residential, business, and public spaces switch to passive monitoring and positive intervention. Improved gesture recognition pipelines enhance technological accessibility and detection precision to protect humans.

Keywords: Hand Gesture Recognition; Random Forest; Precise Detection; Technological Accessibility; Positive Intervention; Passive Monitoring; Live Timestamps; Edge Computing.

Cite as: P. Babu, A. S. Harini, and S. Sowmiya, "CovertSign AI: A Lightweight Vision Framework for Silent SOS Signals," *AVE Trends in Intelligent Computer Letters*, vol. 2, no. 1, pp. 26–41, 2026.

Journal Homepage: https://avepubs.com/user/journals/details/ATICL

Received on: 06/12/2024, **Revised on:** 25/01/2025, **Accepted on:** 20/04/2025, **Published on:** 03/01/2026

DOI: https://doi.org/10.64091/ATICL.2026.000253

1. Introduction

The current state of public safety and emergency management and intervention is undergoing a deep-seated change due to the blistering advances in artificial intelligence and computer vision. Conventionally, safety precautions have largely relied on Closed-Circuit Television (CCTV) systems, which have served as passive surveillance systems used mainly to investigate incidents after they occur, rather than to intervene in real time. In case a dangerous situation occurs, security officers can frequently review footage of the event they could not stop because they were unaware of the situation. Such natural latency between when an emergency occurs and when a response occurs can have catastrophic, and even fatal, effects. This is more so

among victims who are in high-stakes situations, whereby they cannot use their cell phones or call after them verbally due to their physical or social incapacity to do so, like in cases that involve them being restrained, being immediately threatened with violence, or being in a tight space. The urgent demand for silent, independent, and secret communication becomes an essential social and technical requirement. It deals with the basic safety of the vulnerable groups, especially the safety of women and those trapped by domestic violence or work-related accidents. Recent developments in computer vision and artificial intelligence have enabled the advancement of forms of communication that are entirely based on visual stimuli, without requiring any direct verbal or bodily interaction [7].

Hand-gesture recognition systems have risen to prominence as the potential solution to this issue, allowing pre-programmed hand gestures to serve as non-verbal cues that machine learning algorithms can directly read. Combined with smart monitoring systems, these systems have effectively differentiated between deliberate emergency gestures and everyday hand motions with very high accuracy. It is this technological base that makes CovertSign AI a new, lean-bodied vision framework aimed at transforming passive CCTV infrastructure into active, intelligent emergency detection systems. By identifying specific, unspoken hand gestures associated with seeking assistance, the system will fill the vital gap between a crisis and the delivery of assistance. The main task of the work is to develop a useful, real-time, camera-based, silent, autonomous emergency signalling system that can be smoothly integrated with hand-landmark feature classification and machine-learning automatic alert systems to provide an immediate response to distress. Compared to most available safety solutions, which require active hardware interaction, e.g., pressing a wearable panic button or performing a manoeuvre in a mobile application, CovertSign AI uses the already installed CCTV system. This makes it a very cost-efficient and scalable solution, as no additional hardware is provided to the end-user, and it uses equipment found in most of their public and private areas. The core of this framework is the combination of the MediaPipe Hands framework, a breakthrough in computer vision. MediaPipe enables real-time tracking of 21 three-dimensional hand landmarks on each hand, providing accurate x, y, and z coordinates that serve as the basis for complex feature extraction.

By analysing these landmarks, the system will be able to extract more intricate spatial and temporal features of hand gestures, necessary to differentiate between subtle signs of emergency and normal gestures. This level of detection accuracy enables the system to focus on the shape of the hand rather than on crudely presented landmarks, which may vary due to differences in hand size or the distance of the individual from the camera. Machine learning gives the requisite intelligence to convert this stream of information and make impulsive decisions in life-threatening situations. CovertSign AI's approach is based on an ensemble machine learning framework that has proven to identify emergency gestures with over 99 per cent accuracy and to reduce false positives considerably. The development process consisted of testing three machine learning classifiers: Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN). These algorithms are selected because each has demonstrated effective performance in classification tasks and can perform their work efficiently and quickly; that is, they can be properly implemented in real time on consumer hardware. The most useful one was the Random Forest algorithm, which achieved an impressive 99.3 per cent accuracy in distinguishing between emergency and non-emergency gestures across a collection of more than 1,000 samples. This is a very high level of accuracy, essential to a life-safety system. The main SOS gesture detection system had zero false negatives, meaning it would not miss any real cries for help, and the false-positive rate was very low at 0.2. The system achieves dependable performance by providing strong security protection while maintaining low false alarm rates that would disrupt emergency response teams.

The technology provides better real-world performance because its alerting system delivers information to users at high speeds. In case of an emergency gesture, with a confidence level of at least 90, the system triggers a multi-tiered alert process. This will involve presenting a visual alert on a monitoring platform and automatically sending an SMTP email notification to designated emergency contacts. These alerts include the date and time of detection, camera position, and live shots of the situation, enabling responders to evaluate the situation immediately. The speed at which it operates is one of this work's greatest contributions. On average, it takes between five and ten seconds between the first notice of a gesture and the message being sent as an email notification. This response time is almost real-time and a significant improvement over conventional security systems that use manual surveillance and verbal reporting. In addition, the system has undergone rigorous testing in a real-life setting. It has proven consistent in performance over distances of 0.5 to 2.0 meters, which closely aligns with typical CCTV environments. The system had an accuracy of more than 95 even when at 2 meters. In addition to technical accomplishments, CovertSign AI addresses society's more significant needs by creating new safety measures and a more receptive security environment. The technology offers hope to the victims of any cases where any apparent assistance would only expose them to further risks, e.g. in cases of personal threat or domestic violence by means of covert signalling. It represents a paradigm shift in how Researchers approach public safety, as reactive video recording has been replaced by proactive, intelligent monitoring.

It transforms the current cameras into proactive security tools that deliver a much-needed layer of safety to office areas, transportation systems, schools, and residential areas. The ethical and environmental nature of AI-guided surveillance is also considered in the development of CovertSign AI. As these systems become commonplace, privacy and data security are the foremost concerns that must be addressed. The planned future implementations will use on-device processing and encrypted

data storage to ensure the solution is responsible and protects individual privacy. The structure has been constructed to withstand extreme environments and, in the process, to evaluate various aspects, including camera angles and light brightness, which influence the clarity of what can be observed. This is meant to establish a system that will be technically excellent and socially acceptable through the exercise of ethical principles. The information gathered and processed by the researchers to produce training data with the various features of real-world scenarios was used in the system's training and testing. A dataset found in the UCA Dictionary and on the Kaggle site contained three emergency gesture classes and nine non-emergency sign language classes, along with other hand movements individuals commonly use, such as trembling, gesturing, and haphazard movements. This range enables the model to generalise across different hand sizes, lighting conditions, and viewing angles for hand gestures. The researchers used five-fold cross-validation and strong testing using precision, recall, and F1-score metrics, which ensured the validity and efficiency of the selected models before their implementation. The practical usefulness of the system can also be demonstrated by its low computational overhead.

The practical experience demonstrated that the system could sustain 30 frames per second with just 23 per cent CPU usage on a typical consumer-level computer, meaning a broad range of users could use it without requiring the highest computing capabilities. This portability, coupled with the fact that it is based on the infrastructure that is already in place, places the CovertSign AI as a better option than the current solutions, such as the use of SOS apps via cell phones or the use of costly wearable panic buttons, which tend to be more expensive and not as useful during silent emergencies. The next steps in the development of the technology will include increasing the number of gestures that can be recognised to represent a broader range of emergencies and using deep learning methods to enhance performance in extreme conditions. Plans to make versions of the system available on mobile and edge devices are also underway, making the system even more accessible. Finally, CovertSign AI is a giant step toward a realistic, silent, and universal emergency communication system. It is a technology made available to all people regardless of their circumstances and provides a needed safety feature that can save lives when other standard safety nets have collapsed. Building on the success of contemporary machine learning and the current infrastructure of the contemporary world, this paper will address a fundamental social need. It offers an active safety monitoring system that is both low-cost, scalable, and highly productive. However, with this piece, the possibility of AI changing the face of public safety is evident, and Researchers have taken one step closer to a society where all one must do is gesture.

2. Literature Survey

The history of the creation of covert emergency signalling systems based on computer vision is closely tied to advances across a variety of research areas, including real-time hand landmark detection, machine learning-based gesture recognition, intelligent surveillance systems, anomaly detection, human-computer interaction, and the ethical implementation of AI. The suggested CovertSign AI system integrates inputs from all these spheres to transform passive CCTV infrastructure into an active emergency response system. This section will provide a review of the most relevant literature that underlies the technical and theoretical background of the proposed system.

2.1. Real-Time Hand Landmark Detection and Tracking

The foundation of any vision-based gesture recognition system is accurate hand landmark detection. One of the major innovations in this field is MediaPipe Hands, a lightweight, highly optimised framework that detects 3D hand landmarks in real time [1]. The system predicts 21 three-dimensional hand keypoints in two stages of a pipeline: palm detection and landmark regression. MediaPipe was designed for on-device deployment and can be highly accurate while requiring low computational resources, making it suitable for real-time emergency scenarios where latency is a key concern. MediaPipe's efficiency is particularly applicable to systems such as the CovertSign AI, which do not require expensive hardware yet run continuously on CCTV feeds. It can be used to extract features because it reliably detects landmarks across different lighting conditions and hand positions, enabling subsequent classification steps. In addition to this paper, Cao et al. [2] introduced OpenPose, a framework for real-time multi-person 2D pose estimation that can simultaneously identify body, face, hand, and foot keypoints. The Part Affinity Fields (PAFs) introduced in OpenPose allow keypoints or parts of one individual in a multi-person image to be identified and assigned to that person. Although the main scope of the CovertSign AI is hand gestures, OpenPose demonstrates that it is possible to extract structured human pose data from unconstrained video streams. The fact that such systems can operate in a dense setting or one densely populated with surveillance systems justifies the feasibility of incorporating gesture detection into CCTV systems. The combination of these works confirms that landmark extraction for live video streaming is technically feasible and sufficiently trustworthy to serve as the basis for an emergency gesture-detection system.

2.2. Gesture Recognition Machine Learning

After extracting hand landmarks, the second difficulty is properly classifying gestures into meaningful groupings. The field of hand gesture recognition has advanced significantly, moving beyond handcrafted features and classical classifiers toward deep

learning architectures. Pisharady and Saerbeck [3] provide a thorough review of vision-based hand gesture recognition. The various methods are appearance-based and, as such, survey model-based, and describe how deep neural networks were developed to model spatial and temporal gestures. They emphasise that, despite the presentation of deep learning models, Convolutional neural networks (CNNs) and Recurrent neural networks (RNNs) are highly computationally efficient, and the classical machine learning methods remain so as well, useful in cases where feature extraction is robust. This observation directly justifies the architectural decision of CovertSign AI, which is based on derived features and deploys structured landmarks rather than unencoded pixel data. Computational with a great deal less cost, classical classifiers can compete with higher-dimensional inputs and underlying geometric correlations, e.g., Euclidean distances and inter-joint angles. It was also demonstrated by Köpüklü et al. [4] that a convolutional neural network is resource-efficient, capable of achieving real-time gesture recognition performance, and can be deployed in constrained settings. They stress the importance of balancing accuracy and computational efficiency in their work, particularly in services that require continuous monitoring. Classical machine learning classifiers are adopted because they are used to derive target-market derivatives.

This paper demonstrates that real-time gesture recognition is feasible and is called CovertSign AI in practical contexts. A seminal survey of gesture recognition methods, performed earlier, was conducted by Köpüklü et al. [5], who also discuss statistical classifiers such as Support Vector Machines (SVM), Hidden Markov Models (HMM), and K-Nearest Neighbours (KNN). The authors emphasise that SVM-based models are particularly useful in high-dimensional feature spaces, where classes are easily distinguishable. This hunch explains the existence of SVM in the CovertSign AI system, where 3D landmark-based feature vectors are being classified. In the same vein, Rautaray and Agrawal [6] discussed hand gesture recognition systems based on visual components and compared them with traditional machine learning models, including SVM and KNN. They are also discussing trade-offs between computational efficiency and classification accuracy, in which KNN is simpler and trains faster, whereas SVM offers better generalisation. These are comparable to the experimental comparison of Random Forest, SVM, and KNN performed by CovertSign AI to identify the most consistent model for use as a classifier in the same preprocessing environment to detect an emergency. All these studies, taken together, confirm that a structured feature should be used. A viable and effective method is the extraction-based, classical machine learning approach for real-time gesture recognition systems.

2.3. Intelligent Surveillance and Human Activity Recognition

The process of integrating gesture recognition into CCTV systems requires knowledge about human activity recognition research and surveillance analytics. The authors Aggarwal and Ryoo [7] provide an extensive guide to human activity analysis techniques, including motion-based systems, temporal segmentation methods, and various behaviour modelling approaches. Their work emphasises the role of contextual modelling in interpreting human behaviour in surveillance videos. Because emergency gestures are more localised than full-body activities, the concepts of structured motion representation and pattern classification are directly applicable. Sultani et al. [8] proposed a deep learning-based large-scale real-world detection framework of surveillance videos. Their paper indicates that video-based learning of normal and abnormal behaviours can be used to automate the detection of abnormal events. Although the conceptual framework of proactive surveillance fits within this body of work, unlike unsupervised anomaly detection, CovertSign AI focuses on predefined SOS gestures. The two systems are designed to help transform surveillance from recorded monitoring to live, intelligent monitoring. Cong et al. [9] proposed sparse reconstruction-based techniques for abnormal event detection, focusing on how surveillance systems can automatically detect abnormalities in behaviour patterns. Their strategy supports the notion that surveillance systems can serve as proactive safety mechanisms rather than strictly forensic ones. The extension of this paradigm is offered by CovertSign AI, which identifies specific emergency gestures as organised abnormal events that should be alerted to. All these studies demonstrate the transformation of CCTV systems into intelligent monitoring systems capable of processing and responding in real time.

2.4. Covert Communication and Silent Emergency Signalling

The idea of silent emergency signalling overlaps with crisis communication, human behaviour, and studies of assistive safety technologies. The dynamics of communication in crises were studied by Marsen [10], who noted that victims usually use nonverbal or indirect communication when direct signalling is insecure. They found that the ability to facilitate a discreet communication process in an emergency is critical. CovertSign AI, a vision-based gesture-detection system, meets this need by enabling the victim to indicate distress without speaking or touching the device. Similarly, in related fields, Chan et al. [11] investigated the use of smart home and wearable technologies for emergency detection. Wearable panic buttons and biometric sensors are quick alert mechanisms but require specialised hardware and user interaction. Compared with these systems, vision-based systems built on the current CCTV infrastructure will minimise hardware-dependent requirements while remaining responsive in real time. The study by Hom and Shanley [12] examined the factors that prompt help-seeking behaviours in the context of emergency communication. Their contribution states that people who are threatened might be reluctant to engage in open help-seeking behaviour. Thoughtful layouts that react to nonverbal cues, which are pre-agreed upon, create less mental

load and are more usable when under high-stress situations. This mental and cognitive premise underlies the selection of the intuitive SOS gestures in the CovertSign AI.

2.5. Database, Benchmarking and Evaluation

Gesture recognition systems can be evaluated effectively using standardised datasets and benchmarking techniques. Mohammadi et al. [13] highlight the importance of organised datasets of hand gestures for training and evaluation. Models can be reproducible and benchmarked using publicly available datasets. The use of organised landmark-based representations in CovertSign AI is consistent with data-driven training procedures outlined in those publications. Garcia-Hernando et al. [14] also proposed benchmark datasets for first-person hand action recognition with accurate pose annotations. They focus their work on strict evaluation procedures such as cross-validation and latency measurements. In the case of real-time systems such as CovertSign AI, benchmarking should consider classification accuracy, computational efficiency, and response latency.

2.6. Ethics of Surveillance in AI Surveillance

As surveillance systems become more intelligent, ethical considerations should be taken into account. Mittelstadt et al. [15] analyse the ethical controversies surrounding algorithm-based decision-making, privacy, transparency, and responsibility in AI. They point out the dangers of abuse and the unintended consequences of implementing AI in surveillance settings. Regarding the use of the CovertSign AI, some ethical concerns include secure data storage, limited access to captured frames, reduced data retention, and explicit policy frameworks defining system deployment. By considering these issues, this approach will provide a responsible way to implement AI-driven safety systems and help people trust them more.

3. Methodology

The CovertAI system is an advanced computer vision-based interface that enables the identification of human hand gestures and converts them into instructions that a computer can execute. This system is intended to enable touch-free, natural interaction with a computer by using real-time computer vision techniques to interpret hand motions captured by a camera. In many instances, users interact with a computer using physical input devices (keyboard, mouse, touchscreen). While the use of these physical input devices is fine, they can have limitations in settings where physical interaction with the device is impractical, unsanitary, or impossible, including, but not limited to, medical environments, smart, automated homes, virtual reality, and assistive technology for physically disabled individuals. The CovertAI system will allow users to interact with a computer remotely via hand gestures detected by computer vision. By integrating technologies such as image processing, computer vision, and machine learning into a single application, CovertAI creates an efficient gesture-controlled interface. It works by analysing the position and movement of the hands to identify hand features, then maps those features to predefined system commands. The system uses gestures, such as an open palm or a closed fist. It moves in defined directions to perform actions such as navigating a presentation, controlling multimedia playback, changing system settings, or opening an application.

The system has been designed to provide a more intuitive and natural way for people to communicate with computers by allowing them to use their hands as the primary means of interaction. The entire system is based on a sequential data-processing pipeline that converts raw image data into a meaningful command. The first step in the pipeline is to take visual input from a camera. The memory storage holds pictures that depict the user's real hand movements in their current surroundings. Preprocessing operations should be applied, as the images exhibit significant noise, lighting variations, and background distractions, resulting in low-quality images that cannot facilitate the recognition of hand gestures. The hand detection process identifies the hand region in an image after pre-processing the input image. This operation is crucial to the application, as it allows it to concentrate its processing power on critical parts of the image rather than the entire image. Once the application identifies the hand region, it extracts the landmark points that shape the hand. The recorded positions will create a complete geometric model of the hand, including joint and fingertip positions. The extracted landmarks will be organised into a feature representation that preserves the original positions of various parts of the hand. The features will form a small numerical image that characterises the gesture, which will serve as a machine learning input. These features are input to the classification model, which determines which gesture the detected hand position belongs to.

After the gesture is classified, the application maps it to the corresponding computer command. The CovertAI framework has a key benefit: it can work in real time, providing very low latency for gesture recognition and command execution. The importance of real-time performance in interactive systems is that users receive immediate feedback based on their gestures. The use of efficient image processing techniques and lightweight machine learning models makes the system feasible, as they require minimal computational resources. Another factor in the design of the new system is its modular architecture, in which each stage of the processing pipeline is independent. This independent modular design enables the system to be scalable and flexible. For example, if a new, more advanced gesture classification module becomes available in the future, it can be added to the system without modifying the other components. Similarly, additional gesture commands can be added to the system to

expand its capabilities. Thus, the proposed system is a complete system for gesture-based interaction between people and computers. CovertAI combines computer vision algorithms with intelligent feature extraction and classification techniques to create a reliable platform for interpreting human gestures and converting them into meaningful actions by the computer system. The use of this method not only creates a better user experience but also enables the further development of new types of touchless interaction technologies across a variety of application domains.

3.1. System Architecture

This modular structure enables greater efficiency, maintainability, and scalability for the system. The architecture comprises the input layer, processing layer, recognition layer, and output layer. These four layers collectively provide a complete pipeline that converts raw photo data into actions in a system. The input layer collects photo data from the user's environment. A separate camera, or video camera, continuously captures the user's hand gestures. Each hand gesture is captured as a single digital photo, composed of numerous individual dots organised by coordinates in a 2D grid. The colour information in the captured photo depicts what is in the image. Mathematically, the captured photo of the user is represented as a function of spatial coordinates (x,y) and the value of each dot (intensity). In this Figure, the assigned pixel intensity at electron beam locations (x, y) is shown, coded across all three colour channels. Each electron beam position corresponds to a pixel on the image's physical surface. The pixel intensity at each of the three locations represents the colour displayed at that location by the channel data. If the image being analysed is full colour, there will be 3 pixels per physical position in the RGB colour space (Figure 1).

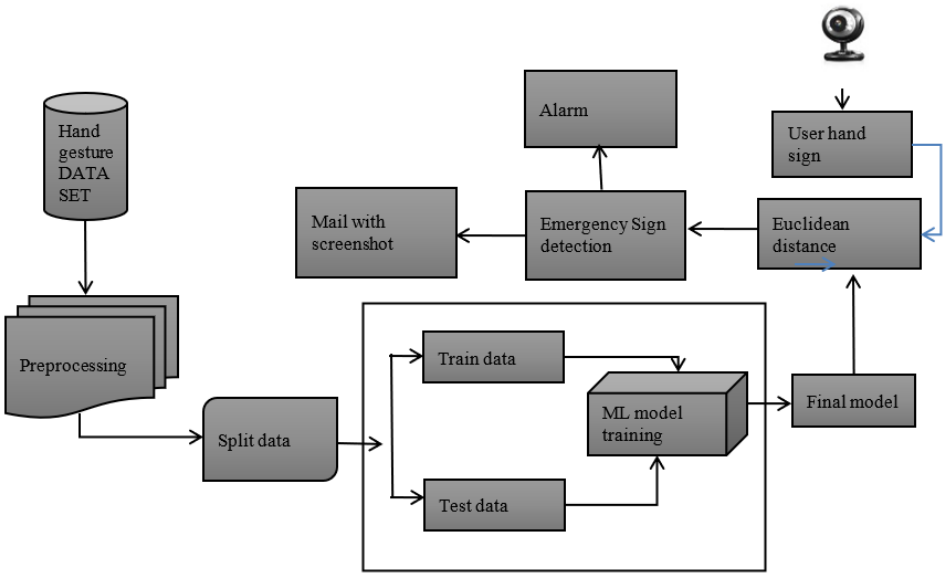

Figure 1: System architecture

Image capture occurs at the sensor, and the resulting images typically pass through a processing layer to prepare them for gesture recognition. The processing layer comprises several modules that perform preprocessing, noise removal, hand detection, and feature extraction. These operations transform raw image data into structured formats that enable machine learning methods to analyse them effectively. The hand detection process allows the processing layer to reduce noise and provide the hand detection region, thereby improving detection accuracy. Hand detection occurs after preprocessing; however, the hand-detection module identifies the region of the image containing the user's hand. This step is essential for gesture recognition, as the background may contain various objects or patterns that could interfere with it. When the user's hand is isolated from the rest of the image, the appropriate area can be used for all subsequent analyses. After locating and identifying a hand region, the hand recogniser will produce hand landmarks (typically the tips and joints of the fingers, and the centre of the palm) that represent the hand's critical anatomical reference points. These landmarks will serve as a skeletal representation of the hand, capturing its geometric structure. As a result, the landmark coordinates can be used to derive valuable information about finger placement and hand orientation, which will assist the recogniser in differentiating between gestures.

After the landmarks are extracted, the recognition layer converts the landmark data into numeric features, enabling the recogniser to describe the gesture. Some of the different types of features may include distances between landmarks, angles between finger segments, and the relative placements of different joints. The recognition layer will use a machine learning

model trained during the model's training phase to classify the detected gesture based on features created by the recogniser. The classifier will use the patterns learned during training to classify and/or group detected gestures in real time. Finally, the output layer will convert (translate) the detected gesture into a command that the computer can execute within the system environment, such as opening an application, controlling media playback, or navigating to a specific slide in a presentation. Several significant advantages arise from CovertAI's modular architecture. First, the use of separate modules for different tasks enables efficient data processing. Second, new models and functional features can be added easily at any time without requiring changes to the entire system. Third, the ability to isolate and address errors in a single module without affecting other elements improves overall system reliability. Overall, the architecture will enable accurate, responsive interaction through gestures, leveraging efficient image processing techniques and machine-learning-based classification.

3.2. Image Processing

Image preprocessing is a vital step in the new approach, as it addresses issues in raw camera images arising from factors such as lighting conditions, camera resolution, background clutter, and motion blur. If these problems aren't resolved, the result is due to the gesture-orientation recognition modules, which are inaccurate too often because of how the gesture is perceived. The first step in the preprocessor is image resizing; this step ensures that each image input to the system has a fixed width and height. Digital cameras capture images at high resolution (high definition), which increases the computational burden and limits real-time processing speed. Therefore, after resizing the image to the same width and height, the amount of data needed to process it will be reduced, as all meaningful visual information will still be present in the resized image. The resize function can be mathematically represented as transforming the original input image into an output with the same aspect ratio as the original, but with predefined Width and Height:

$$I_r = resize(i, w, h) \quad (1)$$

In this equation, I represents the original image captured by the camera, while I_r denotes the resized image. The parameters W and H represent the desired width and height of the output image. Standardising the image size simplifies subsequent processing steps by ensuring all frames share the same spatial dimensions. After resizing, the system performs colour space conversion, converting the image from RGB to grayscale. Colour images contain three channels corresponding to red, green, and blue components. The channels provide complete visual information, but they increase the system's computational burden. Converting the image to grayscale reduces data size while preserving the intensity changes needed for shape and edge detection. The grayscale conversion process uses a mathematical method that combines the red, green, and blue colour components:

$$Gray=0.299R+0.587G+0.114B \quad (2)$$

The formula uses human colour perception analysis to determine colour weight, as visual brightness depends mainly on green light. The process allows the system to extract essential structural details which help it identify hand positions. The next major step is to eliminate all unwanted sound elements from the system. Real-world images face brightness challenges because their pixels undergo random changes from sensor noise and environmental disturbances, making it difficult for computers to detect hand features. The problem requires Gaussian filtering to produce a smooth picture while removing high-frequency noise. The Gaussian filter is defined mathematically by this equation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

The Gaussian filter uses a bell-shaped distribution to assign different weights to each pixel in the image based on its distance from the filter's centre. When the Gaussian filter is applied to the image, it produces a blurred version by averaging pixel values. This process removes noise from the images while preserving their structural integrity. Normalisation can be performed during pre-processing to ensure pixel values fall within a specified range. By normalising pixel values, the effect of varying illumination levels can be reduced, providing a system with stable operating characteristics across different environmental conditions. Normalising pixel intensity scales ensures stable system operation. Performing the pre-processing step improves image quality and reduces differences introduced during image acquisition, thereby enhancing the accuracy and reliability of the gesture recognition system.

3.3. Feature Extraction

Extracting relevant features from the initial set of hand landmark (coordinate) data is an important step in converting raw landmark (coordinate) data into numerically meaningful features that the gesture recognition model can use. Once the system has detected the hand landmarks, the next step is to examine the quantitative (mathematical) relationships between these points to identify the most distinctive (discriminative) features of each gesture, which indicate the relative position of the hand. Rather

than providing the raw coordinates directly to the hand gesture classification model, the proposed system calculates multiple summary metrics that describe the structural arrangement of the fingers and palm, thereby preparing the data for classification. These summary metrics are derived from various data sets, including measurements of distances between landmarks, angles between joints, orientations of each finger relative to reference points on the hand, and the relative locations of the most important points on the hand.

By evaluating the derived measurements, it is possible to detect subtle differences (which the human eye might miss), enabling the hand gesture recognition model to classify the gesture accurately. The Euclidean Distance between landmarks, which are also referred to as Hand Landmarks, is one of the most important characteristics used by the proposed model. Essentially, finding the Euclidean distance between two Hand Landmarks can give you a great deal of useful data related to how far a given finger has been extended relative to the rest of the hand and the position of the hand itself. For example, when a person has their index finger extended so that the tip of their finger is farther away from the centre of their palm than when they have it bent, the Euclidean Distance between them will increase. By calculating distances between many pairs of landmarks, the proposed system can begin to define the structure of a Hand Gesture. The formula used to calculate the Euclidean Distance between two Landmarks is a standard geometric equation.

3.4. Euclidean Distance Formula

In this example, the coordinates (x1, y1) and (x2, y2) represent the locations of the two Landmarks in an image or coordinate system. Once you multiply (x1-x2) by itself and (y1-y2) by itself and then add those two distances together, you arrive at a value that represents how far apart the two Landmarks are from each other. By performing this calculation on many pairs of Landmarks, a Feature Vector can be created that represents the general shape and configuration of the person's Hand. The distance-based characteristics of these measurements can also provide very useful data for determining if any Fingers are extended, folded, or Partially Bent. In addition, the system will consider the angles formed between the finger joints. These angles are significant for determining the position of the fingers. When the finger is bent, the angle between the two adjacent finger joints will vary significantly. This will help the system determine whether the finger is straight or bent. To determine the angle between two adjacent finger joints, three landmark points are used to form a triangle at the joint. Using the geometric relationships of the triangle formed by the three landmark points, the angle between the three points can be determined using the cosine rule. The angle between three landmark points can be determined as follows:

$$\emptyset = \boldsymbol{cos^{-1}}\left(\frac{\boldsymbol{a^2+b^2-c^2}}{\boldsymbol{2ab}}\right) \quad (4)$$

Where a, b, and c represent the distances between the three landmark points forming the triangle. In this context, a and b represent the segments connected to the joint under consideration, while c represents the distance between the outer landmark points. The resulting angle θdescribes the bending state of the finger joint. Smaller angles typically indicate a folded finger, while larger angles correspond to extended fingers. One important characteristic considered by the proposed method is the orientation of the fingers. Finger orientation is determined by how the fingers align relative to the palm or the camera frame. Knowing where the fingers are oriented can help differentiate between two different gestures that have similar finger positions (e.g., hand signals showing thumbs up vs. hand signals showing thumbs down) because the orientation of the thumb will be used to create a vector that can determine if the gesture was intended as a thumbs up or a thumbs down based on the calculated slope of the line between the base joint of the thumb and the tip of the thumb. Additionally, the relative positions of these landmarks are another key factor in determining gesture patterns. Rather than relying on the absolute positions of these pixels, relative positions are used to determine key points, such as the fingertips, palm, and wrist positions.

Using relative positions increases the system's reliability because they remain constant even with slight hand movement within the camera frame. For instance, the relative positions of the thumb and index-finger tips are used to detect gestures such as pinching or gripping objects. Combining all these features into a structured feature vector is crucial for numerically representing the detected gestures. This is achieved by creating a feature vector that includes all the distances, angles, and orientation values computed from the coordinates of all the landmarks. The proposed approach for feature extraction improves the reliability of the gesture recognition system by transforming landmark coordinates into useful geometric features. The use of various features in the proposed approach allows the system to detect the spatial arrangement and orientation of the hand components. This enables the classifier to differentiate between gestures, thereby improving the system's accuracy and reliability.

3.5. Gesture Classification Model

After extracting the hand landmark features, the next step in the proposed model is gesture classification, in which the numerical feature vectors are analysed to determine the most likely gesture performed by the user. In this model, the numerical feature vectors serve as input to a machine learning model that classifies the user's gesture. Assuming the numerical feature vectors are

represented as a multidimensional vector, the gesture classification model can be explained as follows: Let the numerical feature vector be represented as a multidimensional vector, where the numerical features are represented as:

F=[f1,f2,f3,...,fn] (5)

Where each represents individual numerical features of the hand landmarks, the numerical features are composed of the normalised coordinates of the hand landmarks, the Euclidean distances between the landmarks, and the angles of the fingers. The dimensionality of the numerical feature vector depends on the number of input landmarks and the additional geometric features used in the model. The gesture classification model represents a mapping function between the input numerical feature vector and the gesture class performed by the user, which can be represented as:

y=f(F) (6)

Where y represents the predicted gesture class label and f(F) is the classification function that has been learned during the training phase. This classifier will examine the relationship between the features and predefined gesture classes to determine the best class label for the hand posture. The classification model used in the proposed system is efficient for real-time applications. This is because the system's main purpose is to facilitate human-computer interaction. Therefore, the system should make predictions as efficiently as possible while maintaining high accuracy. During the classification phase, the system will process each frame individually to produce a gesture prediction. To ensure classification reliability, the system will be trained on a dataset containing multiple instances of each gesture type. This is because the classifier can differentiate between gesture classes based on features.

3.6. SoftMax Probability Estimation

To obtain probabilities from the classification results, the suggested system utilises the SoftMax function. The SoftMax function converts the classifier's outputs into probabilities for all possible gesture classes. By applying this function, it is guaranteed that the estimated probabilities are non-negative and sum to 1, which are requirements for multi-class classification. The estimated probabilities are guaranteed to be between 0 and 1 for all classes of gestures by applying the SoftMax function. The class of gestures with the highest estimated probability is taken as the system's result.

3.7. Model Training and Loss Optimisation

Efficient training of the gesture identification model depends directly on how well the training phase is conducted. The model will learn to associate a set of feature vectors generated for each gesture with the appropriate gesture label by adjusting its internal parameters to minimise errors in predicting output values (or gesture labels). Let X represent the training data; each case in the training data consists of the feature vector extracted from the hand image. The gesture label associated with each feature vector is defined by Y. The training (data) sample consists of an input feature vector and its corresponding gesture label, as represented:

$$l = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \quad (7)$$

The end goal of the training is to minimise the loss function, which reveals how close the output predictions are to the actual gesture labels. In this structure, the Cross-Entropy Loss function will be used to see how well the gesture labels were predicted. The use of Cross Entropy Loss to compute classification error is very common for solving multi-class classification problems, as it measures the dissimilarity between the predicted probability distribution and the true label distribution of the gesture class. Only the properties are similar; the values can differ (e.g., the outputs may all be above 0.5, so the probability distribution remains the same). In this equation, the true label for each training example is denoted by yi, and the predicted probability output by the classifier is represented by $y\hat{}i$. As the predicted probabilities approach the true labels, the loss value decreases, indicating a better-performing classifier. To reduce the loss during training, the classifier iteratively updates its parameters using an optimisation algorithm, such as gradient descent. Every time the classifier iterates over the training samples, it makes a small parameter update to reduce the model's prediction error. Therefore, the model improves its ability to classify various gesture classes. During training, the loss decreases while the classifier's accuracy increases. The training process ends when the model is stable enough that only slight improvements are possible. When the model is well trained, hand gesture classification is possible under various hand conditions.

3.8. Command Mapping and System Interaction

The next step in the process would be to map the recognised gestures to the computer's recognised commands. This mapping enables the recognition of visual hand gestures to be translated into functional tasks or commands within the computer system.

The command mapping module uses a mapping table that defines each gesture class and the command that triggers a function within the computer system. For example, a "palm open" gesture may map to the command to turn on the computer or start up an application. On the other hand, a “palm closed” gesture may correspond to a command that will stop a function or stop playing a media file. When the gesture classification module identifies a gesture class, the module searches the mapping table for the gesture's command. Once a command is found, it is sent to the system control interface for execution. The mapping mechanism enables the proposed system to serve as a completely touchless interface, allowing users to control a computer system solely through gestures in front of the camera, thereby eliminating the need for traditional touch-based input devices (e.g., a keyboard or mouse). The mapping approach provides flexibility when designing an individualised system, so if new gestures/commands are created, they can be added to the mapping table without any changes needed to the underlying functionality of the gesture recognition algorithm, which will allow many different domains of application, e.g., smart home automation, presentation control, gaming interfaces and assistive technologies for people with disabilities or limited mobility. The dual functionality (i.e., integrated gesture recognition and command execution) promotes the development of a more efficient, easy-to-use, and accessible human-computer interface for the public.

3.9. Output Generation and System Response

The last part of our proposed methodology involves generating a sound based on visual recognition and understanding of the user's gesture or input. Once gesture mapping determines the proper system function corresponding to the user's gesture, the system will execute it through the OS interface or the application control module. The system will generate output in real time to ensure it reacts to the user's gesture or input as quickly as possible. Every frame of video received from the video camera will go through the entire processing pipeline (hand detection, feature extraction, classification, and gesture mapping) to enable quick execution of the user-provided command. Each of these stages in the process has been optimised to minimise the time required to execute the command and to provide an effective interaction experience. The moment the gesture mapping module recognises a gesture as positive, the system immediately executes the command corresponding to that gesture. For example, if the user gestures to indicate "next slide," the presentation application will advance to the next slide; or if the user gestures to indicate "volume up," the system's audio level will increase. Optionally, to provide the user with additional visual feedback for their recognised gesture, the system will display the gesture label in text format on the screen. Gesture recognition systems must be responsive, or the end-user experience will be inefficient and unsatisfactory. Therefore, the proposed system includes lightweight components that enable rapid gesture detection and response. Additionally, the output generation module can process ambiguous or incorrectly predicted gestures. If a gesture's classification confidence falls below an acceptable threshold, the system may ignore it or prompt the user for confirmation. This helps to minimise accidental actions and increase system reliability.

3.10. Novelty of the Proposed CovertAI System

The new CovertAI Gesture Recognition System introduces methods for gesture-based interface interaction that set it apart from conventional systems, greatly improving overall efficiency, accuracy, and usability while maintaining low computational complexity. A major innovation of the proposed system is its use of lightweight feature-extraction techniques to enable real-time gesture recognition. Rather than relying on computationally intensive deep learning models, which require expensive, powerful GPUs, the proposed system will use landmark-based geometric features derived from hand-tracking algorithms to minimise processing time and maximise gesture recognition accuracy. The development of a system that uses geometric features such as landmark coordinates, distances between landmarks, and angular relationships between fingers to produce a better representation of hand gestures greatly enhances the classifier's ability to differentiate between visually similar but physically distinct hand gestures. Additionally, this system provides a unified pipeline for hand detection, feature extraction, classification, and command execution, enabling seamless data transfer between components and facilitating real-time gesture interaction. Moreover, since contactless human-computer interaction is a growing field in today’s global technology market, the proposed approach helps to limit the need for physical input devices while supporting health and hygiene-based interactions in places like hospitals, labs, and public spaces by providing users with a way to control devices with gestures (i.e., hand movements). As a result, the system's adaptability makes it easily extendable to many different areas of application. This is primarily due to developers' ability to quickly and easily establish their own gestures and map them to commands using the gesture-command mapping module, thereby enabling use in areas such as virtual reality interfaces, smart home control systems, educational products, and assistive technologies. The proposed CovertAI system is an advancement in gesture-based interaction with computers and humans, enabling fast, accurate, and useful real-time gesture recognition.

4. Results and Discussion

The comprehensive experimental evaluation of the CovertSign AI gesture recognition system has yielded insights into its effectiveness in accurately classifying Emergency/Non-Emergency gestures and whether the manner in which these gestures are performed falls into the categories of ‘Emergency Signal’ and ‘Not An Emergency’. Random Forest, Support Vector

Machine (SVM), and K-Nearest Neighbour (KNN) were chosen as classifiers for these experiments due to their diverse machine learning paradigms and prior success with structured feature classification tasks. Gesture datasets were created for the experiments based on multiple samples of both Emergency and Non-Emergency gestures, with gestures performed in the same manner labelled as “No Detection”. After splitting into Training and Testing sets, the classifiers were trained to learn the relationships between the extracted gesture features and their corresponding classifications. Once training was complete, the classifiers were tested on their untrained evaluation data, and their performance and classification reliability were determined.

Multiple metrics were used to evaluate how well the algorithms performed in recognising an individual’s gestures of distress. They included multiple ways to calculate how accurately the gesture was recognised (including precision, recall rates, and a combined measure known as F1-score), thereby allowing one to evaluate both the overall proportion of correct recognitions and the relative numbers of false positives and false negatives. Specifically, in this scenario, for the detection of emergency gestures, the system must maintain high recall rates, as failing to recognise a distress signal that is in fact an emergency could result in serious consequences for an individual. The results from these four experiments suggest that all three algorithms achieved high levels of gesture recognition accuracy. However, they exhibited considerable variation in the consistency and robustness of their overall classifications. The models’ confusion matrices provided useful information about whether the algorithms exhibited significant differences in the numbers and types of classification errors. Further, they reveal the various types of strengths and weaknesses common to each.

4.1. Random Forest

When tested on these models, the Random Forest classifier performed best. It is an ensemble learning method that uses the collective opinion of multiple decision trees to produce a prediction by aggregating each tree's vote. Each tree is generated from a subset of the input data for each classification produced by the decision tree. The final result is produced by taking the vote from all the individual decision trees. The results from the Random Forest confusion matrix are presented in Fig. 2. The confusion matrix demonstrates that the Random Forest model has produced very high accuracy predictions across the majority of the gesture classes. The Random Forest model produced an overall accuracy of 99.3%, demonstrating excellent performance for a real-time gesture recognition system.

Most classes had precision, recall, and F1 scores near 1.0; therefore, the Random Forest model can correctly classify both emergency and non-emergency gestures with minimal error. There are many benefits of Random Forest, including the ability to model complex relational dynamics among independent variables without needing to tune numerous parameters. For the system being described, the types of hand gesture feature types extracted are comprised primarily of three elements: 1) spatial coordinates; 2) landmark distances; and 3) angular relationships between fingers. These 'hand gesture features' can interact with and affect one another through multiple complex relationships, defining a gesture pattern. Random Forest provides a tremendous benefit to gesture recognition systems, as it accounts for complex feature interactions by combining multiple decision trees that model a large number of feature combinations to capture the fingerprints of that gesture.

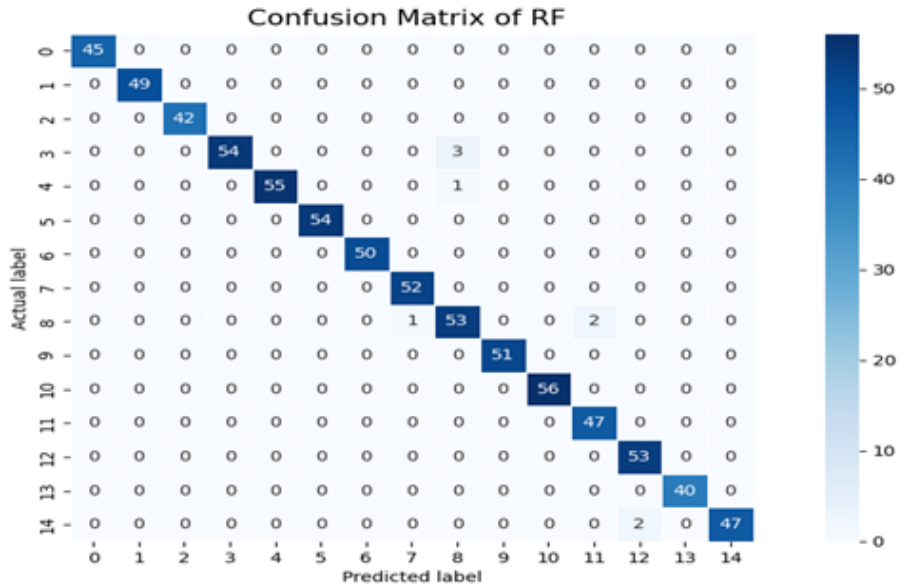

Figure 2: Random forest CM

Another very important characteristic of Random Forest is its robustness to noise and input data variability. Because gesture recognition systems will experience variability due to hand position and movement, lighting and camera angle, etc., it is beneficial as Random Forest aggregates predictions from all the decision trees used to classify the same gesture, thereby reducing any influence from one or a few misclassified metrics from the training set and increasing its ability to generalise to new input samples. The results show that all gesture patterns were correctly classified by the Random Forest model, with very few misclassifications across the dataset.

The misclassifications were generally at least partially attributable to the similarity between the hand position or gesture being modelled and another class defined as a non-emergency category. The emergency gesture category was consistently classified with a very high degree of accuracy, and given that the majority of users will be utilising the gesture recognition system for emergency applications (e.g., women's emergency signalling systems), this is a significant benefit of utilising the Random Forest model. The Random Forest classifier demonstrates strong classification performance, high reliability/stability, and high accuracy, thereby enabling it to handle complex features effectively (Figure 2). Therefore, this classifier is well-suited for implementation in gesture recognition systems that employ CCTV technology for use in real life. Accuracy: 99.3% achieved high PRF across all classes, with many scores at or near 1.0. Most predictions were correctly classified, with minimal errors across all classes. Random Forest performed well, showing strong classification capability with a balance of accuracy and reliability. It is particularly effective in situations that require straightforward decision-making.

4.2. Support Vector Machine

In addition to the gesture-detection methods, SVM classifiers were assessed within the gesture-detection framework and evaluated for classification accuracy. The SVM algorithm is a very effective supervised learner that finds an optimal hyperplane that separates the different classes of data points by maximising the margin between classes (i.e., the separation of classes) to improve classification accuracy and maximise generalizability. The SVM classifier achieved a classification accuracy of 95.3%, only slightly lower than that of the Random Forest method, demonstrating strong classification performance. Overall, the SVM classifier produced high precision and recall scores for most classes of gestures, and particularly for the emergency gesture that was described as 'EMERGENCY! Women's Safety Call. The results suggest that SVM classifiers distinguish well between emergency gesture types and other hand movement types (Figure 3).

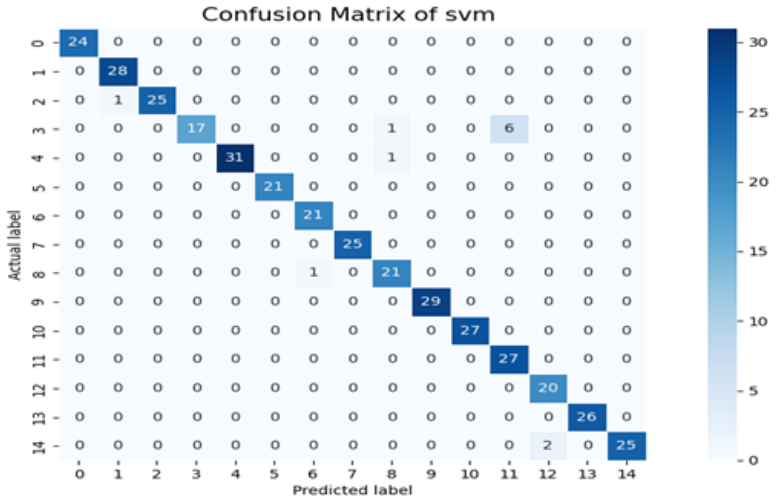

Figure 3: Support vector machine CM

On the other hand, the results also indicated slightly lower precision and recall for some classes that were classified as 'No Detection'. For example, the 'No Detection 11' class yielded a precision score of 0.71 and a recall of 0.89; thus, there were instances of the 'No Detection 11' class being classified incorrectly, and instances predicted to be of the 'No Detection 11' class being predicted as different gesture types. Certain gestures that are not classified as emergencies may be misclassified due to overlapping feature types with other non-emergency gestures. Since SVMs are based on separating boundaries between classes, they may struggle to distinguish between classes when feature types overlap significantly. With respect to gestural recognition, small variations in a user's hand position and/or finger position can produce similar feature types and thus mislead a linear or kernel classifier into producing less than perfect separations between the two classes. Although the SVM classifier struggled to distinguish between non-emergency gestures, it excelled at detecting emergency gestures, which is ultimately what matters most in this paper.

In addition, the SVM classifier demonstrated generalizable predictive capabilities and produced consistent, reliable predictions across most classes. Overall, SVM can be considered an appropriate classification method for gesture classification tasks that have sufficient structure and clear feature distributions. However, when compared with ensemble-based classification methods such as Random Forests, additional SVM model tuning and kernel selection may be necessary for optimal performance. Accuracy: 95.3%. Lower precision and recall for certain classes, e.g., "No Detection11" (p = 0.71, r = 0.89). This implies misclassifications for these classes. Higher scores for certain classes, e.g., "EMERGENCY! Women's safety call." This implies overall robustness. There were minor misclassifications in certain classes; e.g., "No Detection11" was misclassified as other classes. Emergency gestures were classified correctly. It works well for datasets with simple structures and few feature interdependencies.

4.3. K-Nearest Neighbor

The KNN algorithm is a straightforward yet effective method for classifying data based on the labels of the training dataset's nearest neighbours in a multidimensional feature space. KNN stores all training instances during the training phase and classifies new input instances by analysing their closeness to the training data during the prediction phase, using different distance metrics. The KNN classifier achieved an overall classification accuracy of 97.7%, placing it between the Random Forest and SVM classifiers. The KNN model generally achieved high precision and recall across most gesture classes, particularly for the emergency gesture classes, with no other gesture classes achieving correct classification.

An observation from the results included a relatively low recall for the "No detection" class, which had a recall of 0.83, indicating that there were misclassifications for some true instances of this class (i.e., gestures in this class were misclassified as gestures in other classes); however, this only occurred for non-emergency gesture types and did not impact emergency gesture detection. The KNN model demonstrated an impressive ability to predict accurately, as only 1 case of misclassification was produced by the model in the 'No Detection6' class. In addition, the KNN model performed exceptionally well on gesture datasets with clear clusters in the gesture space. Simplicity and adaptability are the two greatest advantages of the KNN algorithm. The fact that KNN doesn't require extensive training procedures allows it to quickly learn new information and update classifications each time new data is received (Figure 4).

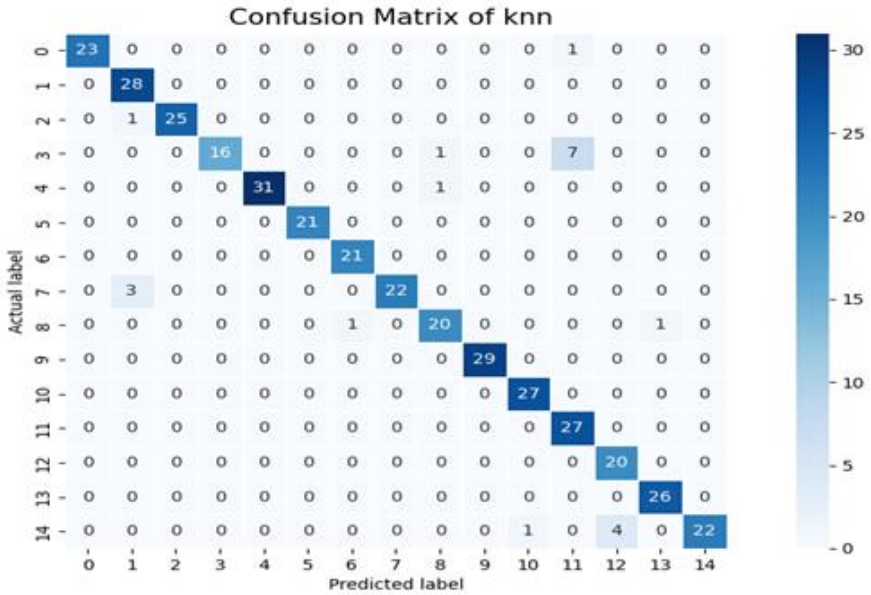

Figure 4: K-nearest neighbour CM

Nonetheless, when dealing with a very large training set, the KNN algorithm can consume significant computational resources to compute the distance between the new input instance and all stored instances. However, the KNN algorithm had a major flaw in gesture recognition, which was overcome, and it can thus be used as an alternative to other complex algorithms. Accuracy: 97.7% high marks were scored in most classes. There was a slight decrease in recall for class "No Detection3" (r = 0.83) (Figure 5).

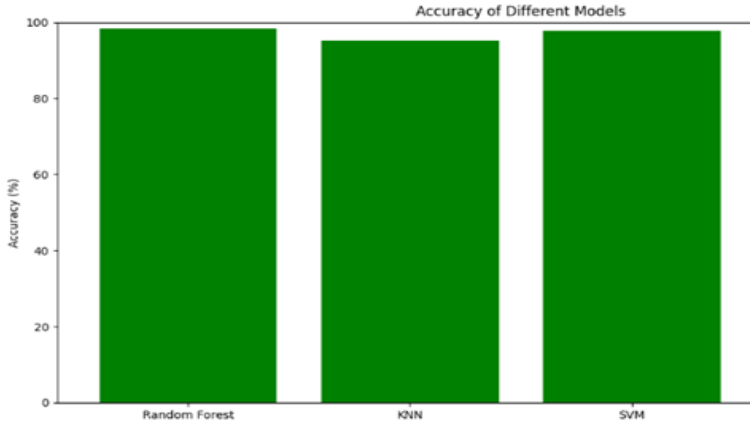

Figure 5: Comparative analysis

Only 1 instance of "No Detection6" was misclassified, indicating strong performance. K-Nearest Neighbour demonstrated exceptional performance through its iterative learning process, keeping errors low. It is particularly effective for data sets with minor feature complexities (Table 1).

Table 1: Presents the comparative performance of three classifiers

Classifier	Accuracy
Random Forest	99.3%
SVM	95.3%
KNN	97.7%

4.4. Comparative Performance of Three Classifiers

Random Forest achieved 99.3% accuracy, significantly higher than SVM and KNN. The high accuracy of Random Forest indicates its effectiveness in distinguishing emergency from non-emergency gestures, making it suitable for real-time CCTV systems (Table 2).

Table 2: Presents the comparison with existing systems

System Type	Hardware	Real-Time	Silent	Infrastructure	Cost
Phone SOS Apps	Phone	Yes	No	Individual	Low-Med
Wearable Panic	Wearable	Yes	No	Individual	High
CovertSign AI	CCTV	Yes	Yes	Existing	Low

The standard laptop webcam system was tested at several distances (0.5 M, 1.0 M, 1.5 M, and 2.0 M) to represent a typical CCTV environment accurately. Results showed consistent performance across all tested distances, with completed tasks achieving accuracy greater than 95% even at 2 meters.

4.5. Comparison with Existing Systems

Table 3 lists the accuracy percentages by distance from the camera. The inference from Table 3 is that the closer the person is, the better the accuracy, with 99.5% at 0.5m.

Table 3: Accuracy at different camera distances

Distance from Camera	Detection Accuracy
0.5 m	99.5%
1.0 m	99.2%

1.5 m	98.7%
2.0 m	95.8%

5. Conclusion

The proposed system of CovertSign AI can be considered an efficient computer vision and machine learning-based intelligent real-time emergency detection system. This proposed paper seeks to shift CCTV cameras from a passive to an active state, enabling them to monitor for silent SOS hand gestures and alert authorities accordingly. The proposed system's result can be considered extremely accurate and reliable. If the proposed system detects a distressed hand gesture with at least 90% confidence, it can generate an alert in various ways, including visual, email, and image alerts. The proposed system's response time is expected to be between 5 and 10 seconds. This can be considered not only a measure of the proposed system's accuracy but also its response time. The results obtained from the Tables and Figures clearly indicate the effectiveness of the proposed model. The comparative analysis table above clearly shows that the Random Forest classifier has the highest accuracy among the classification algorithms, i.e., 99.3%, compared to the Support Vector Machine (95.3%) and K-Nearest Neighbour (97.7%) algorithms. Moreover, the confusion matrices also clearly indicate the superiority of the Random Forest classifier. In addition, the Table results, which show accuracy at various distances, clearly demonstrate the system's robustness, as accuracy remains above 95% even at 2 meters.

Furthermore, the Table results clearly indicate the superiority of the proposed system, which is silent, cost-effective, and leverages existing infrastructure, unlike existing systems such as mobile SOS applications and wearable devices. The proposed work aims to provide an efficient system for gesture recognition using classical machine learning algorithms. The proposed system is designed to recognise hand gestures by converting them into vectors based on hand distances, angles, and orientations. The novelty of this paper lies in its potential to provide a silent, non-intrusive emergency communication solution. This is in sharp contrast to conventional solutions that require the victim to be active in responding to emergencies. In this paper, the victim can respond to emergencies in a silent, non-intrusive manner. This is achieved through hand gestures. There is no need for additional hardware since it uses the existing CCTV infrastructure. In addition, the use of classical machine learning for high accuracy instead of deep learning is a novelty. This is because deep learning is a complex and computation-intensive approach. Possible Improvements for Future Work: The paper could be improved. This is achieved by incorporating a variety of emergency gestures. Deep learning methods can be incorporated to improve performance in extreme conditions, such as low light. In addition, privacy issues can be handled. CovertSign AI is a significant improvement in the field of intelligent surveillance. This is because it has the potential to transform conventional CCTV surveillance into a proactive safety solution.

5.1. Future Work

Although 99.3% accuracy was achieved, there are some limitations:

- **Environmental Factors:** The effectiveness of the software is dictated by the placement and angles of the CCTV cameras and available light. Future development will include infrared technology.
- **Visibility of the Emergency Gesture:** The gesture must be clearly recognised as an emergency gesture. If either viewing angle is extreme (>3 meters) or very wide-angle, it may reduce accuracy.
- **Management of False Positives:** Future versions of the software will also require that gestures be sustained over time (5 seconds or longer) before generating an alert.
- **Privacy Issues:** Continuous tracking of individuals raises privacy concerns. Future development will include on-device processing and encrypted data storage.

The following are included in the future development: (1) Development of multiple gestures, (2) Integration with video management systems, (3) Development for edge computing, (4) Multiple cameras to be used in tracking, and (5) Conducting pilot studies in a controlled environment.

Acknowledgement: The authors would like to express sincere gratitude to Sathyabama Institute of Science and Technology for providing the necessary support and resources to carry out this work. The authors also extend their appreciation to the faculty members and guides for their valuable guidance and encouragement throughout the study.

Data Availability Statement: The data supporting this study are available from the authors upon reasonable request to maintain transparency and enable reproducibility.

Funding Statement: The authors confirm that this research and manuscript were completed without any external financial assistance.

Conflicts of Interest Statement: The authors declare that there are no conflicts of interest that could have influenced the outcomes of this research.

Ethics and Consent Statement: The authors have approved the publication of this work and consent to its accessibility for readers' reference and learning purposes.

References

1. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv preprint*, 2020. [Accessed by 18/10/2025].
2. Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.
3. P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, no. 12, pp. 152–165, 2015.
4. O. Köpüklü, A. Gündüz, N. Köse, and G. Rigoll, "Real-time Hand Gesture Detection and Classification using Convolutional Neural Networks," in *Proc. 14th IEEE Int. Conf. Automatic Face & Gesture Recognition*, Lille, France, 2019.
5. O. Köpüklü, N. Kose, A. Gunduz, and G. Rigoll, "Resource Efficient 3D Convolutional Neural Networks," *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, California, United States of America, 2019.
6. S. S. Rautaray and A. Agrawal, "Vision-Based Hand Gesture Recognition for Human Computer Interaction: A Survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.
7. J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–43, 2011.
8. W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado, United States of America, 2018.
9. Y. Cong, J. Yuan, and J. Liu, "Sparse Reconstruction Cost for Abnormal Event Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado, United States of America, 2011.
10. S. Marsen, "Navigating Crisis: The Role of Communication in Organizational Crisis," *International Journal of Business Communication*, vol. 57, no. 2, pp. 163–175, 2020.
11. M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes—Present state and future challenges," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.
12. M. A. Hom and I. H. Shanley, "Considerations in the assessment of help-seeking and mental health service use in suicide prevention research," *Suicide Life Threat Behav.*, vol. 51, no. 1, pp. 47–54, 2021.
13. Z. Mohammadi, A. Akhavanpour, R. Rastgoo, and M. Sabokrou, "Diverse hand gesture recognition dataset," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 50245–50267, 2024.
14. G. Garcia-Hernando, S. Yuan, S. Baek, and T. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado, United States of America, 2018.
15. B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi, "The ethics of algorithms: Mapping the debate," *Big Data & Society*, vol. 3, no. 2, pp. 1–21, 2016.