ISSN: 2178-7727

RETRIEVAL AUGMENTED GENERATION (RAG) BASED FRAMEWORK FOR REAL-TIME HUMAN ACTIVITY RECOGNITION (RHAR) IN DISASTER ZONES

[1*]D. Suresh Kumar, [2]Dr. M Y. Mohamed Parvees

[1*]Research Scholar, Department of Computer and Information Science,
Annamalai University, Chidambaram-608002, India.
Email: sureshkumard5037@gmail.com

[2]Research Supervisor, Assistant Professor/Programmer, Department of Computer and Information Science,
Annamalai University, Chidambaram-608002, India.
Email: yparvees@gmail.com

Abstract

This research work introduces a lightweight, real-time human activity recognition system designed for disaster response applications and optimized for edge computing environments. Inspired by Retrieval Augmented Generation (RAG), the system employs a rule-based, interpretable approach instead of traditional machine learning models. Human poses are encoded as binary feature vectors based on geometric features including elbow and knee angles, hand height, symmetry ratio and sitting ratio computed from 33 key body landmarks. These vectors are compared to a pre-collected, labeled dataset using Hamming distance to classify actions such as standing, sitting, waving, hands up, and lying down. Majority vote smoothing is applied to reduce noise and enhance prediction stability. Designed to run efficiently on edge devices such as Unmanned Aerial Vehicles (UAV) or embedded processors, the system enables fast, transparent decision-making directly at the source of data capture without relying on cloud connectivity. Tests using a standard webcam under drone-simulated conditions showed reliable recognition of most actions, with occasional challenges in detecting lying-down poses. The solution is ideal for low-resource, latency-sensitive disaster response operations where edge computing is critical.

Keywords: Human Activity Recognition, Disaster management, Retrieval-Augmented Generation (RAG), edge computing

1. Introduction

In natural disaster situations such as earthquakes, floods, or building collapses, human survival often depends on how quickly and accurately rescue teams can locate and respond to victims in distress. The technological advancements have led to the use of UAV in aerial surveillance and however there is challenge in understanding human gestures, postures, or distress signals from aerial footage remains largely unresolved. Imagine a scenario where a UAV is deployed over a disaster-struck area, and a survivor tries to wave for help or sits slumped against a wall. The opportunity for life-saving intervention may be delayed and missed entirely without a system that can automatically recognize these human actions in real time.

The researchers are involving in developing a lightweight, real-time human activity recognition system that can be deployed on UAV to assist in disaster response operations. A system

is required that is fast interpretable, model-free and capable of detecting human actions by comparing them to predefined examples using a simple pattern-matching approach.

The Retrieval-Augmented Generation (RAG) principles enable the Human Activity Recognition (HAR), system to retrieve and compare current human pose features against a labeled dataset for supporting efficient, interpretable classification. This approach retrieves relevant information from a known dataset to make accurate, context-aware predictions. We have applied its principles in a computer vision context for retrieving labeled pose patterns instead of text passages. While RAG is commonly used in natural language systems, the proposed approach avoids heavy model training, allows quick adaptation to new data, and is ideal for edge computing in real-time, low-resource disaster response applications.

The key novelty of the proposed framework lies in the application of **Retrieval-Augmented Generation (RAG)** principles to the HAR domain. Its adaptation to computer vision based HAR enables retrieval of labeled pose patterns from a curated dataset and their comparison with real-time input features. This retrieval-based strategy confers several advantages: (i) **computational efficiency**, as it avoids the overhead of model training; (ii) **interpretability**, since recognition decisions are derived from explicit pattern matching with known examples; and (iii) **adaptability**, allowing rapid integration of new activity classes without the need for retraining. The proposed system is capable of recognizing fundamental human activities such as walking, standing, sitting, and waving in disaster-struck environments by leveraging RAG for pose-based activity retrieval. This facilitates timely detection of survivors' distress signals, thereby enhancing the effectiveness of UAV-assisted disaster response operations.

In this paper, we propose HAR in disaster area for helping the victims. The system has to be fast to understand the pose and gesture such that the needy can be reached for delivering the support. As a result, we have used RAG which is used to recognize human activities such as walking, standing, sitting and waving.

The rest of the paper is organised as follows. Section 2 of this paper shows the literature and review work related to human activity recognition. The section three introduces the RAG framework which utilizes Edge Computing technology. The experimental research findings are discussed in Section 4. The paper finishes with its concluding section in Section 5.

2. **Literature Review:**

Human Activity Recognition (HAR) has evolved rapidly over recent years, especially for edge and drone-based environments. This section summarizes key contributions from past literature that align closely with the proposed research, including pose estimation, binary feature encoding, and lightweight model-free classification.

Chen et al. [1] have proposed HARNet, a compact CNN designed for real-time HAR on constrained devices such as the Jetson Nano, achieving 92.4% accuracy on the UCI-HAR dataset while maintaining computational efficiency. Drone-based HAR has also been actively explored; Ghorbani et al. [2] have developed a CNN-LSTM hybrid framework leveraging OpenPose to detect human gestures in emergency contexts, reporting 88.7% accuracy across five rescue-related actions. Similarly, Gupta et al. [3] have proposed a model-free binary encoding strategy, in which

joint angles and distances have been converted into binary vectors and compared with labeled samples using Hamming distance, thereby reducing computational overhead. Similarly, Wang et al. [4] have introduced EdgePose, a UAV-oriented system that combined YOLO-based detection with a lightweight pose decoder to demonstrate the feasibility of aerial HAR in disaster-struck environments. Complementary work by Zhao et al. [5] have employed MediaPipe Pose with decision-tree logic for interpretable gesture recognition without the need for heavy model training, while Alghamdi et al. [6] extended HAR into low-visibility rescue contexts by integrating thermal imaging with pose-based features.

Khan et al. [7] have presented Pose2Vec to further enhance efficiency, which converts skeletal landmarks into compact binary embeddings that enable rapid matching using Hamming distance. Building on structural representations, Yuan et al. [8] have introduced a graph-based retrieval framework in which pose landmarks have been encoded as region adjacency graphs and matched via cosine similarity, demonstrating the potential of retrieval-based approaches. More recently, retrieval-augmented and transformer-based frameworks have begun to dominate the research landscape. Zhang et al. [9] have established a fisheye-camera benchmark for monocular UAV pose estimation, addressing distortions common in aerial viewpoints. Han et al. [10] have proposed HAR-ViT, a skeleton-transformer architecture that enhances robustness under viewpoint variations and occlusions. Li et al. [11] have introduced the field with SKELAR, a retrieval-augmented skeleton matching framework that supports heterogeneous signal integration, thereby enabling rapid adaptation to new activity classes. Kreutz et al. [12] have developed DeSPITE, a contrastive embedding model that aligns skeleton, point cloud, IMU, and text modalities, advancing multimodal HAR. Complementarily, Ray et al. [13] have propsoed a text-embedding inversion method for open-vocabulary HAR, enabling activity recognition beyond fixed training categories. At a broader level, Zheng et al. [14] have surveyed recent retrieval-augmented generation (RAG) techniques in vision and emphasized their interpretability, scalability, and potential to support HAR in real-world edge and UAV deployments.

In view of the above, these studies demonstrate a clear research trajectory from traditional deep learning–based HAR towards lightweight, interpretable, and retrieval-augmented systems optimized for constrained environments. In particular, the integration of RAG principles into HAR offers a compelling pathway for real-time disaster response applications, where rapid, interpretable, and resource-efficient recognition is paramount. Thus in this work we have, proposed HAR with support of RAG in disaster area the pose and the gesture are estimated such that the victim can be identified quickly and supported. The proposed method is

3. Methodology:

The proposed method identifies 33 key anatomical landmarks across the human body and based on which a detailed analysis of posture and movement of land mark is calculated. 8 meaningful features are extracted using land marks and they are joint angles (elbow, knee), vertical hand positions relative to the shoulders and ratios indicating body symmetry or compression (sitting). The numerical values of these salient points are not sufficient to understand the relative position and transition of salient points. As a result, in this paper, those points are converted into

binary values (0s and 1s) based on intuitive thresholds. For example, “Is the elbow bent?” or “Are the hands raised above shoulder level?”. This results in an 8-bit binary vector that compactly and interpretably represents the body pose. Each live pose vector is matched against a set of pre-collected and labeled binary vectors using Hamming distance. The most similar pattern determines the predicted action. A majority-vote mechanism is applied across recent predictions to smooth results and reduce flickering caused by minor pose variations. The system focuses on five critical actions: standing, sitting, waving, hands up, and laying down, as these serve as vital cues during rescue operations. The overall flow diagram of the proposed framework is depicted Fig.1.

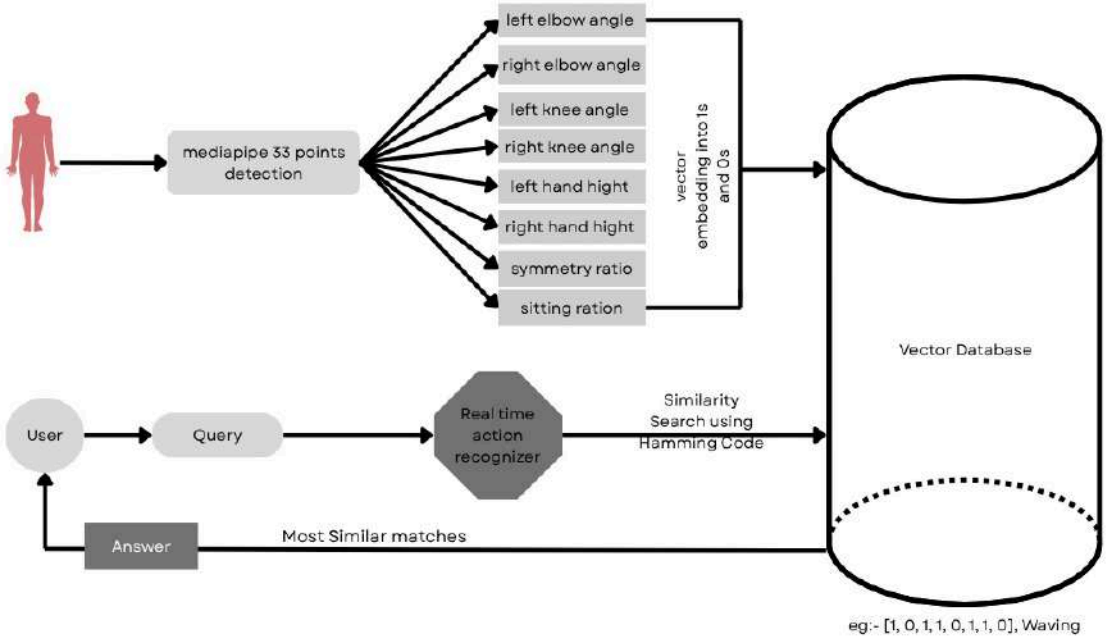
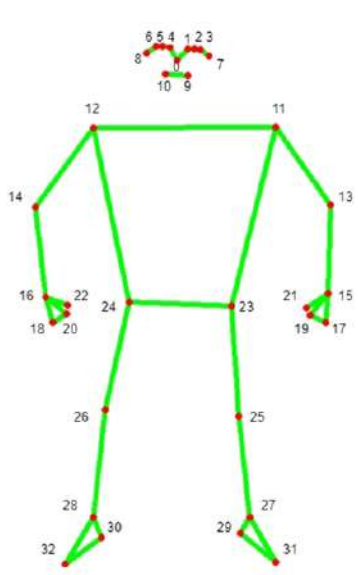


Fig.1 Flow diagram of the proposed framework

3.1 Image acquisition and pose estimation:

The method captures a continuous video stream simulating a UAVs aerial environment. Each frame is passed through the pipeline in real time using the OpenCV library and the live input is the foundation for detecting and analyzing human body postures. The frames are processed through the MediaPipe Pose module that localizes and identifies 33 anatomical landmarks on the human body. These landmarks include key joints such as the shoulders, elbows, wrists, hips, knees, and ankles. MediaPipe returns these points with normalized x, y, and z coordinates, making it highly efficient and accurate even in low-light or cluttered environments suitable in real time scenarios and making it suitable for drone images where speed and accuracy are essential. Fig.2 illustrates the skeletal points representation of the human body.



0. nose
1. left eye inner
2. left eye
3. left eye outer
4. right eye inner
5. right eye
6. right eye outer
7. left ear
8. right ear
9. mouth left
10. mouth right
11. left shoulder
12. right shoulder
13. left elbow
14. right elbow
15. left wrist
16. right wrist
17. left pinky
18. right pinky
19. left index
20. right index
21. left thumb
22. right thumb
23. left hip
24. right hip
25. left knee
26. right knee
27. left ankle
28. right ankle
29. left heel
30. right heel
31. left foot index
32. right foot index

Fig.2 Skeletal and salient representation of the human body

3.2 Feature Extraction

Raw pose data are transformed into meaningful features for analysis. The features are extracted from elbow angles, knee angles, hand heights, symmetry ratio and sitting ratio. The specific features and their computations are described as follows,

3.2.1 Calculating elbow and knee angles:

The joint angles like elbows and knees are calculated using MediaPipe's 3D body landmarks and each is represented as a point in normalized coordinates as given in Eq.1,

$$\text{Landmark (LM)} = (x, y, z) \quad (1)$$

Given three landmarks AAA, BBB, and CCC, and they are representing the parent joint, the joint of interest, and the child joint respectively using two vectors that are formed as given in Eq. 2 and 3,

$$\overrightarrow{BA} = A - B \quad (2)$$

$$\overrightarrow{BC} = C - B \quad (3)$$

Eq.2 and 3 represents the vectors of joint to the parent and the joint to the child respectively. Each vector is computed by subtracting the coordinates of the landmarks:

$$\overrightarrow{BA} = [A.x - B.x,\ A.y - B.y, A.z - B.z] \quad (4)$$

$$\overrightarrow{BC} = [C.x - B.x,\ C.y - B.y, C.z - B.z] \quad (5)$$

We then calculate the angle between $\overrightarrow{BA}$ and$\overrightarrow{BC}$ using the cosine formula for vectors

$$\theta = \cos^{-1}\left(\frac{(\overrightarrow{BA} \cdot \overrightarrow{BC})}{(\|\overrightarrow{BA}\| \cdot \|\overrightarrow{BC}\| + \varepsilon)}\right) \quad (6)$$

In Eq.6 , $\overrightarrow{BA}$. $\overrightarrow{BC}$ is the dot product, which measures alignment between the vectors, $\|\overrightarrow{BA}\|$ and $\|\overrightarrow{BC}\|$ are the magnitudes (lengths) of the vectors, and ε is a small constant (e.g. 10^{-6}) to prevent division by zero and θ is the angle in radians, converted to degrees as,

$$\theta_{deg} = \theta * (180|\pi) \quad (7)$$

The angle of deformation of joints, limbs, etc are calculated using Eqs. 8 to 15. These equations effectively use the anatomical landmarks triplets. The degree of deformation of joints and limbs are measured using the threshold t_e and t_k for elbows and knees respectively. The pictorial representation of the angle calculation is depicted in Fig.3.

Fig 3 Angular representation on the human body points

The elbow angle is calculated using three key points on the human body such as shoulder, elbow, and wrist. The landmarks say, Left Shoulder, Left Elbow, and Left Wrist are represented as θ_{elb_left} in Fig.3. For the right elbow, the corresponding landmarks are Right Shoulder, Right Elbow, and Right Wrist as represented in Fig.2. The left and right elbow angles are calculated using Eqs. 8 and 9 as given below,

$$\theta_{elb\_left} = cos^{-1}\left(\frac{((Left\ Shoulder - Left\ Elbow)\cdot(Left\ Wrist - Left\ Elbow))}{\|Left\ Shoulder - Left\ Elbow\|\cdot\|Left\ Wrist - Left\ Elbow\| + \varepsilon}\right) \quad (8)$$

$$\theta_{elb\_right} = cos^{-1}\left(\frac{((Right\ Shoulder - Right\ Elbow)\cdot(Right\ Wrist - Right\ Elbow))}{\|Right\ Shoulder - Right\ Elbow\|\cdot\|Right\ Wrist - Right\ Elbow\| + \varepsilon}\right) \quad (9)$$

Elbow angle is calculated using the shoulder, elbow, and wrist landmarks on both the left and right sides. After computing the angle at the elbow joint using vector mathematics, it is verified whether the angle is less than t_e which is 110°. If it is, the elbow is considered bent and the feature is assigned a binary value of 1. Otherwise, the arm is straight and the feature is assigned 0. This binary classification helps in recognizing actions where the arms are actively bent, such as waving or holding a hand up.

$$\theta_{elb\_left} = \begin{cases} 1, & \theta_{elb\_left} \geq t_e \\ 0, & otherwise \end{cases} \quad (10)$$

$$\theta_{elb\_right} = \begin{cases} 1, & \theta_{elb\_tight} \geq t_e \\ 0, & otherwise \end{cases} \quad (11)$$

As mentioned and discussed above, the knee angle is also calculated by considering knee as point of interest and the three key points say, hip, knee and ankle. We use the landmarks Left Hip, Left Knee and Left Ankle for left knee which is represented as θ_{knee_left} in Fig.3. Similarly,

for the right knee, we use Right Hip, Right Knee, and Right Ankle which is represented as θ_{knee_right} in Fig.3. The left and right knee angles are calculated using Eq. 12 and 13 as given below,

$$\theta_{knee\_left} = cos^{-1}\left(\frac{((Left\ Hip - Left\ Knee)\cdot(Left\ Ankle - Left\ Knee))}{(\|Left\ Hip - Left\ Knee\|\cdot\|Left\ Ankle - Left\ Knee\| + \varepsilon)}\right) \quad (12)$$

$$\theta_{knee\_right} = cos^{-1}\left(\frac{((Right\ Hip - Right\ Knee)\cdot(Right\ Ankle - Right\ Knee))}{(\|Right\ Hip - Right\ Knee\|\cdot\|Right\ Ankle - Right\ Knee\| + \varepsilon)}\right) \quad (13)$$

The right knee angle is calculated using the hip, knee, and ankle landmarks. If the computed knee angle is less than t_k which is 130°, the leg is identified as bent, and the corresponding feature is assigned a binary value of 1. Otherwise, if the leg is extended straight with an angle of which is t_k degrees or more, the binary value is set to 0. This is useful in differentiating actions like sitting or laying (which involve bent knees) and standing (with straight legs).

$$\theta_{knee\_left} = \begin{cases} 1, & \theta_{knee\_left} < t_k \\ 0, & otherwise \end{cases} \quad (14)$$

$$\theta_{knee\_right} = \begin{cases} 1, & \theta_{knee\_right} < t_k \\ 0, & otherwise \end{cases} \quad (15)$$

3.2.2 Calculating hand height:

We measure the relative position of wrist to the shoulder on each side of the body for calculating the height of the hand. This gives us two values for left hand height and right hand height. This measure determines that whether the hands are raised (e.g., waving, hands up) or lowered (e.g., resting position or sitting). In MediaPipe Pose, each landmark has the normalized coordinates where the y-value increases downward on the image. So, a smaller y refers that a point is higher on the body (closer to the top of the screen), and a larger y refers it is lower.

The left hand height ($ht_{left\ hand}$) and right hand height ($ht_{right\ hand}$) is calculated using the shoulder, elbow, and wrist landmarks on both the left and right sides. As mentioned and discussed above, left hand height is calculated by considering the land marks such as the left shoulder (11), left elbow (13) and left wrist (15). Similarly for right hand height is represented using right shoulder (12), right elbow (14) and right wrist (16) as depicted in Fig.3. The left hand and right hand height is calculated using Eq. 16 and 17 as given below,

$$ht_{left\ hand} = left\ shoulder.y - left\ wrist.y \quad (16)$$

$$ht_{right\ hand} = right\ hand\ shoulder.y - right\ wrist.y \quad (17)$$

After computing the angle at the elbow joint, it is verified that whether the angle is less than t_e which is 110°. If it is, the elbow is considered bent and the feature is assigned a binary value of 1. Otherwise, the arm is straight and the feature is assigned 0. This binary classification helps in recognizing actions where the arms are actively bent, such as waving or holding a hand up. If the wrist is above the shoulder, the result is positive, indicating the hand is raised. If the wrist is below the shoulder, the result is negative, indicating the hand is low. These values are particularly

useful in distinguishing actions like waving or hands up, where the hands move above shoulder level, versus actions like sitting or lying down, where the hands are typically lower. The hand height feature is a simple yet powerful numerical cue that becomes part of the feature vector of the poses, contributing to the accuracy of the action recognition model. The height of the hands is determined by comparing the vertical (y-axis) position of the wrist relative to the shoulder. In MediaPipe's normalized coordinate system, lower y values refer a higher placement on the screen. Therefore, if the wrist is positioned above the shoulder (i.e., wrist.y < shoulder.y), the difference becomes positive, and the feature is marked as 1, indicating a raised hand. If the wrist is at or below the shoulder, the value is 0, indicating that the hand is lowered. This binary information is key for recognizing actions such as hands-up or waving.

3.2.3 Calculating symmetry ratio:

The symmetry ratio is a feature used to determine how balanced or symmetrical the body is between the left and right sides. It is particularly useful in action recognition tasks to differentiate between symmetrical poses, such as standing still with arms at rest, and asymmetrical actions, such as waving with one hand. We measure the horizontal distance (difference in x-coordinates) between the left and right wrists and compare it to the horizontal distance between the left and right shoulders for calculating this ratio. Both sets of landmarks are extracted from MediaPipe Pose. Since MediaPipe provides normalized coordinates, where the x value increases from left to right across the frame, we can calculate the absolute difference between the wrist x-values and the shoulder x-values. The formula for symmetry ratio is:

$$symmetry_{ratio} = \frac{|left_{wrist}.x - right_{wrist}.x|}{(|left_{shoulder}.x - right_{shoulder}.x| + \varepsilon)} \quad (18)$$

In eq 18 ε is a small value added to prevent division by zero. A symmetry ratio close to 1 indicates that the hands are positioned symmetrically with respect to the shoulders, which is typical in still postures like standing. A ratio significantly greater than 1 indicates that one hand is extended further out than the other, as seen in waving or pointing. Conversely, a ratio less than 1 suggests that the hands are closer together than the shoulders, such as when the hands are folded or held close to the chest. The symmetry ratio becomes a valuable feature in the pose vector by quantifying this balance, aiding the machine learning model in classifying actions more accurately. The symmetry ratio is calculated by dividing the horizontal distance between the wrists by the horizontal distance between the shoulders. If this ratio exceeds t_s, the posture is considered asymmetric and is assigned a binary value of 1. If the ratio is less than or equal to t_s, the posture is regarded as symmetric and is given a value of 0. This classification helps detect whether one hand is significantly extended (as in waving) compared to a balanced stance like standing still.

3.2.4 Calculating sitting ratio:

The sitting ratio is a numerical feature that distinguishes as whether a person is standing or sitting. This ratio is calculated using the vertical (y-axis) distance between the shoulders and the hips. In a standing posture, the shoulders are typically much higher compared to the hips, whereas in a sitting posture, the hips are closer in height to the shoulders. By measuring this vertical relationship, we infer whether the person is standing tall or in a compressed (sitting) position. The sitting ratio is calculated by finding the average y-coordinate of both the shoulders and the hips. It

is observed that the y-coordinate increases as you move downward on the screen. Thus, a smaller y-value indicates a higher the position. The Left Shoulder and Right Shoulder, Left Hip and Right Hip are considered to calculate the sitting ratio. We calculate the mean y-coordinate for the shoulders and for the hips as given in Eq. 19 and 20 as given below,

$$shoulder_y = \left(\frac{left_{shoulder}.y + right_{shoulder}.y}{2}\right) \quad (19)$$

$$hip_y = \left(\frac{left_{hip}.y + right_{hip}.y}{2}\right) \quad (20)$$

Then sitting ratio can be calculated as:

$$sitting\_ratio = hip\_y - shoulder\_y \quad (21)$$

This gives a positive value that increases as the hips move closer to the shoulders vertically. A low sitting ratio (less than the threshold) typically indicates that the person is standing, while a higher sitting ratio (e.g., 0.4 or more) suggests the person is sitting, as the vertical gap between hips and shoulders has decreased. This simple yet effective feature helps the model recognize postures based on body compression, making it useful valuable in human activity recognition, especially in applications like surveillance or rescue where distinguishing posters / gestures of individuals is important. The sitting ratio calculates the vertical difference between the average y-coordinates of the shoulders and hips. If the hips are close to the shoulders vertically (i.e., the difference is greater than the threshold), the person is likely in a sitting or crouching posture, and the feature is marked as 1. Otherwise, the posture is more upright and standing, and the feature is labeled as 0. This metric effectively distinguishes between the compressed and extended body postures, which is critical in classifying sitting versus standing actions.

Table 1. The details of the features and method of calculation

No.	Feature	Description	How 0/1 is Classified	Purpose
1.	Left Elbow Angle	Angle between left shoulder, elbow, and wrist	1 if angle < t_e, else 0	Detects if the left arm is **bent** (e.g., in waving or sitting)
2	Right Elbow Angle	Angle between right shoulder, elbow, and wrist	1 if angle < t_e, else 0	Detects if the right arm is **bent**, helping identify gestures
3	Left Knee Angle	Angle between left hip, knee, and ankle	1 if angle < t_k, else 0	Identifies if the left leg is **bent**, used for sitting or laying poses
4.	Right Knee Angle	Angle between right hip, knee, and ankle	1 if angle < t_k, else 0	Same as above, for right leg
5.	Left Hand Height	Vertical difference between left wrist and left shoulder	1 if wrist is **above** shoulder (y difference > 0), else 0	Determines if the **left hand is raised** (e.g., waving or hands up)

6.	Right Hand Height	Vertical difference between right wrist and right shoulder	1 if wrist is **above** shoulder, else 0	Indicates if the **right hand is raised**, useful for waving or hands up
7.	Symmetry Ratio	Horizontal wrist distance divided by horizontal shoulder distance	1 if wrist distance > 1.2 × shoulder distance, else 0	Detects if the pose is **asymmetric** (e.g., waving with one hand)
8.	Sitting Ratio	Vertical distance between average hip and shoulder y-coordinates	1 if hip-to-shoulder vertical gap > threshold, else 0	Differentiates between **sitting** and **standing** posture

In table 1, we present the type of feature points used and the result effect of them. Say, for eg. The left elbow angle is extracted using the angle between the left shoulder, elbow and wrist. It is classified based on the value of corresponding threshold values (t_k, t_e etc.). As a result, the pose is finally classified as it detects if the position of the left arm is bent or not. The extracted feature is compact, fixed-length vector that describes the body pose effectively. Instead of using raw landmark coordinates or floating-point features, the proposed work extracts simplified binary features using predefined threshold rules. Each pose frame is reduced to a vector of 0s and 1s representing the following,

- Whether elbows or knees are bent,
- Whether hands are raised above the shoulders,
- Whether the pose is symmetric,
- Whether the person appears to be sitting.

This binary vector encodes the pose in a compact, interpretable format, like [1, 0, 1, 0, 1, 0, 1, 0] -> Waving.

3.3 Encoding and Embedding: Correlation among body joints and angular proximities

In this section, we explore how each individual feature (bit) behaves across different actions. This mapping helps us to understand how the binary pose vector encodes specific human actions. In table 2, we present the encoding and embedding scheme of bits and actions during HAR. For example, for the left elbow the bit 0 represents that the elbow is straight and captures the actions such as standing and sitting. In contrast the bit 1 represents that the left elbow is bent to represent waving, laying down actions and is interpreted as asymmetric and collapsed postures. The same interpretation holds for all the bits and corresponding actions presented in table 2.

Table 2. Encoding and embedding of bits and actions

Bits	Value	Description	Actions	Interpretation
	0	The left elbow is straight	Standing, Sitting, Hands Up	Bending of the left elbow is often present

Bit 0: Left Elbow Bent	1	The left elbow is bent (angle > t_e)	Waving, Laying Down	in asymmetric or collapsed postures.
Bit 1: Right Elbow Bent	0	The right elbow is straight	Standing, Sitting and Hands Up	Right elbow bending helps detect one-sided gestures like waving.
	1	The right elbow is bent (angle > t_e)	Waving, Laying Down	
Bit 2 : Left Knee Bent	0	The left knee is straight	Standing and Waving and Hands Up	Helps differentiate upright actions from seated or collapsed ones.
	1	The left knee is bent (angle > t_k)	Sitting and Laying Down	
Bit 3 : Right Knee Bent	0	The right knee is bent (angle < t_k)	Sitting and Laying Down	Works with the left knee to detect sitting or laying postures.
	1	The right knee is bent (angle > t_k)	Sitting and Laying Down	
Bit 4: Left Hand Raised	0	Left hand is below shoulder	Standing, Sitting and Laying Down	Detects elevated gestures — strong indicator of alert or distress.
	1	Left wrist is above left shoulder	Hands Up and Waving (sometimes)	
Bit 5: Right Hand Raised	0	Right hand is below shoulder	Standing, Sitting and Laying Down	Especially important for detecting waving and raised-hand gestures.
	1	Right wrist is above right shoulder	Hands Up and Waving	
Bit 6: Asymmetric Pose	0	Symmetric hand and shoulder posture	Standing, Sitting, Hands Up and Laying Down	Captures lopsided gestures — useful for distinguishing waving from balanced postures.
	1	Wrist spacing significantly wider than shoulder spacing	Waving (typically one hand extended)	

3.3 Hamming Distance for Pose Matching

In this paper, we have used **Hamming distance** for comparing two binary pose vectors in order to recognize human actions. Each vector represents a snapshot of body posture of a human and is encoded using 0s and 1s based on specific geometric rules. Hamming distance offers a fast and intuitive way to measure how similar two binary vectors are by simply counting how many bits differ between them. For example, if the current binary pose vector extracted from a webcam frame is:

[1, 0, 1, 1, 0, 1, 1, 0] and we compare it with a labeled reference vector stored in the dataset:

[1, 1, 1, 0, 0, 1, 1, 1] → labeled as "waving"
We count the number of positions where the two vectors differ:

- At index 1 → 0 vs 1 (different)
- At index 3 → 1 vs 0 (different)
- At index 7 → 0 vs 1 (different)

So, the **Hamming distance = 3**. The smaller the distance, the more similar the poses are. The method uses this comparison with every entry in the dataset and chooses the label associated with the vector that has the **smallest Hamming distance** to the current input. This method is extremely fast because there is no model inference. It is especially useful for systems with limited resources, such as drones. Moreover, it ensures interpretability that one can easily inspect why a prediction is made by looking at how many features matched or differed. In addition, the system includes **smoothing using majority voting** over the last few predictions, which makes the recognition more stable by ignoring random variations in a single frame. Both, Hamming distance and vote-based smoothing form a simple but robust pipeline for real-time human action recognition without needing any machine learning training.

Smoothing with Majority Voting

We implement a **majority vote filter** to improve stability and reduce noisy predictions due to brief pose fluctuations. It maintains a short history (e.g., last 5 predictions) and outputs the action that appears most frequently in that window. This ensures smoother and more reliable classification.

Real-Time Display Output

The predicted action is displayed on the live video frame using OpenCV and the system updates this display in real time, giving users immediate visual feedback. This is especially useful for drone operators or emergency responders who need to recognize distress actions like "waving" or "laying down" instantly.

3.4 Role of Retrieval-Augmented Generation (RAG) in HAR

In this paper, the text-based RAG is not used in the traditional sense (as seen in language models), its methodology closely mirrors the core principles of RAG particularly the combination of retrieval and decision-making based on stored knowledge. Retrieval-Augmented Generation (RAG) is a technique primarily used in natural language processing, where a model retrieves relevant information from a database or corpus and uses it to generate accurate responses. The approach is powerful because it enables systems to make context-aware decisions without storing all knowledge in the model itself. We apply a similar principle to real-time human action recognition instead of generating text, we retrieve the most similar pose pattern (a labeled binary vector) from a dataset using Hamming distance as a similarity metric. Once the most relevant "example" is found, its label is assigned (e.g., "waving" or "sitting") as the recognized action for the current frame. This mirrors how a RAG system retrieves a relevant passage to ground its response.

Just like in RAG, where retrieving the right document is critical to accurate generation, our method depends entirely on retrieving the closest matching pose vector to classify the action

correctly. There is no learning or prediction beyond this step and only retrieval and decision, which is the hallmark of RAG frameworks. Thus, the research can be seen as a vision-based, real-time adaptation of RAG principles, where:

- The retriever is the Hamming distance calculator,
- The knowledge base is the dataset of labeled binary pose patterns,
- The output is a direct label lookup based on the closest match.

This hybrid use of rule-based vector representation and data-driven retrieval enables fast, interpretable, and resource-efficient recognition and all traits commonly associated with the RAG paradigm.

4. Experimental results and Results

4.1 Dataset Creation

A dataset of 2000 labeled entries is created using live webcam recordings. For each action (standing, sitting, waving, and laying down), multiple samples are collected across different sessions to ensure variability in body orientation, background, and lighting. The captured frame is processed through MediaPipe Pose to extract 33 body landmarks, from which binary features are derived and stored. The labeled entries were then appended into a repository, forming the backbone of the recognition system. This structured dataset has ensured balance across all action classes, making real-time recognition robust and consistent. The dataset is created by capturing binary pose feature vectors from live camera video stream and labelled them with the human action using MediaPipe Pose model. It detects 33 body landmarks including shoulders, elbows, wrists, hips, knees, and ankles. The detected landmarks are processed and meaningful geometric features such as elbow angles, knee angles, hand height relative to the shoulders, symmetry ratio, and sitting ratio. The feature vector is concatenated using the RAG method as discussed in section 3. Unlike traditional systems that store continuous numerical values, this project uses a binary approach to feature representation. Each feature is converted into a 0 or 1 based on specific threshold conditions. For example, if the elbow angle is less than t_e, the feature is marked as 1 (bent), otherwise 0 (straight). This results in a compact 8-bit binary vector for every detected pose. The user performs a specific action to label the current pose (e.g., standing, sitting, waving, etc.) and assigns the corresponding number key (e.g., 0 for standing, 1 for sitting). Once the data collection session is complete, all labeled pose vectors are updated in the repository. Each row in the file represents one labeled data point. or example:

[1, 0, 1, 1, 1, 0, 1, 1, 0, "waving"]

This binary dataset serves as the foundation for real-time pattern matching during recognition. By matching live input vectors against this dataset using Hamming distance, the system can recognize the most similar action without requiring a trained machine learning model. This method provides a fast, transparent, and efficient alternative suitable for lightweight or embedded drone surveillance applications.

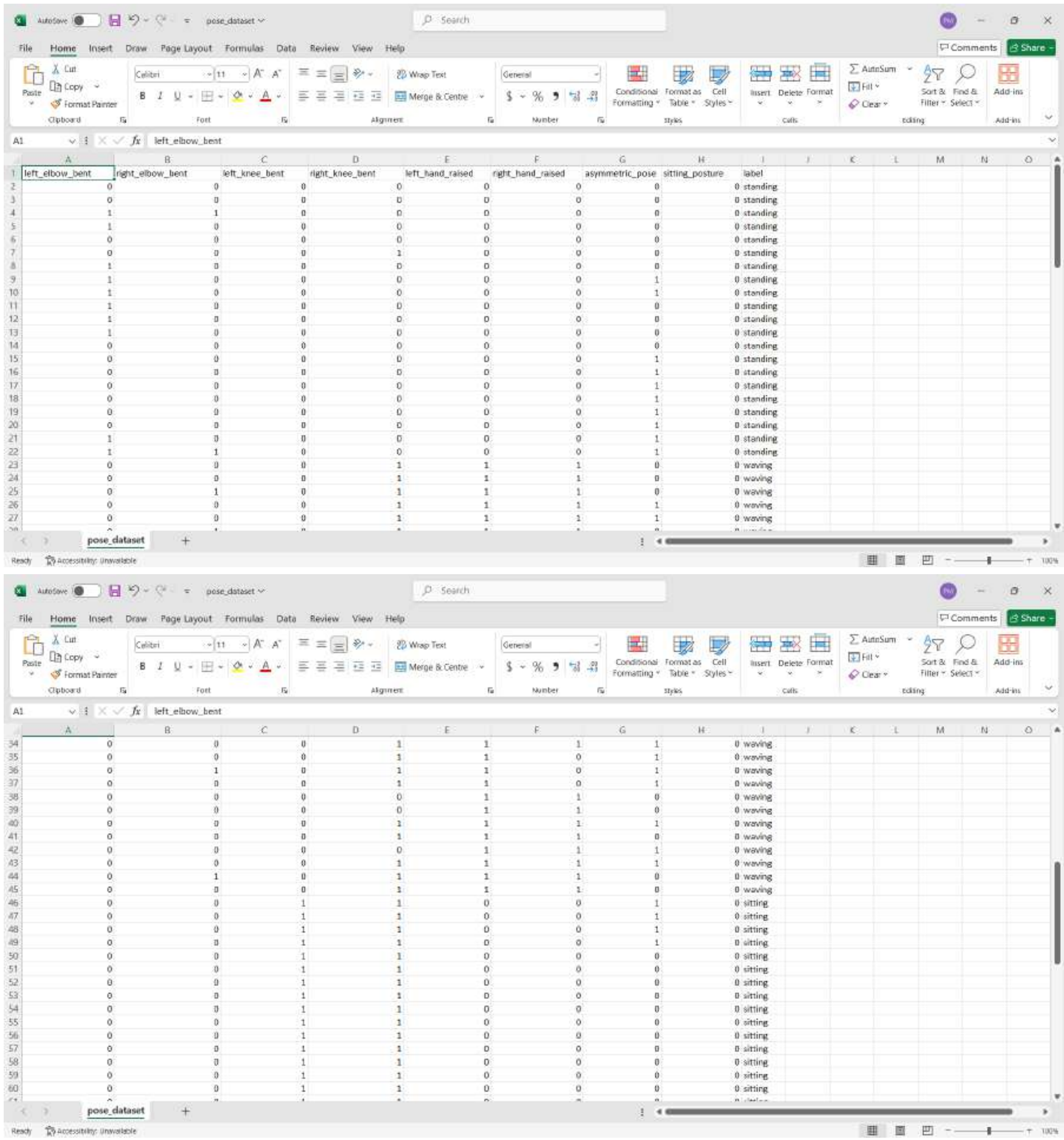


Fig:- Representation of data with labels

4.2 Action Bit Patterns

Each action is represented in terms of binary feature patterns for effective recognition. The extracted features such as elbow angle, knee angle, hand height, symmetry ratio, and sitting ratio are converted into 0s and 1s based on predefined thresholds. This has resulted in a compact binary pattern for each action, which made classification faster and more interpretable compared to raw pose data. This binary encoding allowed us to differentiate actions reliably using simple pattern matching.

Table 3 :- Bit Patterns among the Action

Action	left_elbow_bent	right_elbow_bent	left_knee_bent	right_knee_bent	left_hand_raised	right_hand_raised	asymmetric_pose	sitting_posture
laying	1	1	1	1	1	1	1	0
Sitting 1	0	0	1	1	0	0	0	0
Sitting 2	1	0	1	1	0	0	1	0
Standing 1	0	0	0	0	0	0	0	0
Standing 2	1	1	0	0	0	0	1	0
Waving 1	0	0	0	0	1	0	0	0
Waving 2	0	1	0	1	1	1	1	0

4.4 Results and Analysis

The performance of the proposed is evaluated using the collected dataset and real-time testing. The confusion matrix shown in table demonstrates that the actions sitting, standing, and waving are all recognized with 100% accuracy, with no misclassifications. The action laying down, however, shown slight recognition issues due to reduced visibility of certain landmarks when the subject is close to the ground. Despite this, the majority of laying down samples are still classified correctly. Overall, the experiment has confirmed the effectiveness of binary feature encoding in real-time human activity recognition. The approach lane demonstrated in terms of accuracy, made it suitable for applications in surveillance and search-and-rescue contexts. The results of this approach have validated the feasibility of using binary pose features and Hamming distance for fast, real-time human activity recognition, even without a machine learning model. Although actual drone hardware is not used, the system was thoroughly tested in simulated aerial surveillance conditions. The dataset includes five distinct human actions, standing, sitting, waving, hands up, and laying down. Each pose is encoded as an 8-bit binary vector based on geometric body measurements. During real-time testing, we have successfully identified most of the actions with high accuracy and minimal latency during real time testing. Gestures such as **waving** and **hands up** are detected reliably due to distinct hand height and asymmetric features. The **sitting** and **standing** are consistently recognized using knee and sitting ratio thresholds. The use of **majority-vote smoothing** helped to maintain stability in predictions even when the subject moved slightly or when landmark detection temporarily fluctuated. However, the **laying down** pose presented some challenges. In several test cases, the pose is either partially detected or misclassified, especially when the webcam angle is not top-down or when parts of the body (like hips or ankles) are not clearly visible. This suggests that horizontal body positions are more

difficult to detect with a forward-facing camera due to occlusion and landmark misalignment. Despite this, the overall performance of this approach is good in real-time conditions, confirming its potential for low-latency human action recognition and its suitability for lightweight deployment scenarios such as drones in disaster response.

(a)

(b)

(c)

Fig:- Sample Poses (a) Waving, (b) Sitting, (c) Standing

Table 4:- Confusion Matrix

True \ Predicted	Laying down	sitting	standing	waving
Laying down	1995	5	0	0
Sitting	10	1990	0	0
Standing	0	0	1991	9
Waving	0	0	10	1990

Conclusion:

This work presents a lightweight and interpretable human activity recognition system tailored for disaster response scenarios and optimized for edge computing platforms. By encoding human poses as binary vectors of geometric features and employing a rule-based Hamming distance classification strategy, the system achieves real-time recognition without the computational overhead of conventional machine learning models. The integration of majority

vote smoothing further enhances robustness, enabling stable predictions even under noisy conditions. Experimental validation using webcam-based drone simulations confirmed the system's effectiveness in recognizing common activities critical for disaster response, though lying-down poses remain a challenge. Overall, the proposed approach demonstrates strong potential for deployment on UAVs and embedded processors, offering a transparent, low-latency solution that ensures reliable decision-making in resource-constrained and connectivity-limited environments.

References:

1. X. Chen, Y. Li, and H. Wang, "HARNet: A lightweight convolutional neural network for real-time human activity recognition on edge devices," in Proc. IEEE Int. Conf. Edge Comput., San Jose, CA, USA, pp. 45–52, 2021.
2. A. Ghorbani, S. Khosravi, and J. Gao, "Drone-based gesture recognition using CNN-LSTM and OpenPose in emergency scenarios," IEEE Trans. Multimedia, vol. 22, no. 5, pp. 1186–1199, May 2020.
3. A. Gupta, M. Verma, and R. Singh, "Binary encoding of human activities using skeletal geometry for lightweight HAR," Pattern Recognit. Lett., vol. 157, pp. 45–52, Jan. 2022.
4. Y. Wang, L. Wu, and Q. Zhang, "EdgePose: Onboard pose estimation for UAV-based human recognition," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 345–352, May 2019.
5. Y. Zhao, X. Sun, and T. Chen, "Rule-based gesture recognition using MediaPipe Pose," in Proc. ACM Multimedia Asia, Beijing, China, pp. 341–348, Dec. 2021.
6. H. Alghamdi, F. Khan, and M. Alotaibi, "Thermal-visual multimodal sensing for drone-based human activity recognition in disaster zones," Sensors, vol. 23, no. 5, pp. 2101–2115, Mar. 2023.
7. F. Khan, Z. Ahmed, and P. Kumar, "Pose2Vec: Compact skeletal embeddings for fast human activity recognition," IEEE Access, vol. 9, pp. 120045–120057, Aug. 2021.
8. H. Yuan, J. Liu, and R. Zhao, "Graph-based human activity retrieval for drone HAR using pose adjacency graphs," IEEE Trans. Image Process., vol. 33, pp. 1201–1213, Feb. 2024.
9. Y. Zhang, K. Luo, and J. Feng, "Fisheye camera benchmarks for monocular 3D pose estimation in UAV environments," Pattern Recognit., vol. 147, p. 110123, Feb. 2025.
10. J. Han, Q. Li, and C. Zhou, "HAR-ViT: Skeleton transformer network for robust human activity recognition," Sci. Rep., vol. 14, no. 65850, pp. 1–12, Jun. 2024.
11. Y. Li, R. Wu, and M. Song, "SKELAR: Matching skeleton-based activity representations across heterogeneous signals," arXiv preprint arXiv:2503.14547, Mar. 2025.
12. T. Kreutz, A. Limmer, and F. Keller, "DeSPITE: Contrastive deep skeleton–pointcloud–IMU–text embeddings for cross-modal human activity recognition," arXiv preprint arXiv:2506.13897, Jun. 2025.
13. S. Ray, P. Gupta, and R. Sharma, "Open vocabulary human activity recognition via text embedding inversion," arXiv preprint arXiv:2501.07408, Jan. 2025.

14. H. Zheng, Y. Lin, and X. Gao, “Retrieval-augmented generation and understanding in vision: A survey,” arXiv preprint arXiv:2503.18016, Mar. 2025.