Intuitive Human-Robot Interface: A 3-Dimensional Action Recognition and UAV Collaboration Framework

Akash Chaudhary[1][a], Tiago Nascimento[1,2][b] and Martin Saska[1][c]
[1]*Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague, Czech Republic*
[2]*Universidade Federal da Paraíba, Brazil*
{*chaudaka, pereiti1, martin.saska*}*@fel.cvut.cz*

Keywords: Human-Robot Interaction, Unmanned Aerial Vehicles, Gesture Recognition.

Abstract: Harnessing human movements to command an Unmanned Aerial Vehicle (UAV) holds the potential to revolutionize their deployment, rendering it more intuitive and user-centric. In this research, we introduce a novel methodology adept at classifying three-dimensional human actions, leveraging them to coordinate on-field with a UAV. Utilizing a stereo camera, we derive both RGB and depth data, subsequently extracting three-dimensional human poses from the continuous video feed. This data is then processed through our proposed k-nearest neighbour classifier, the results of which dictate the behaviour of the UAV. It also includes mechanisms ensuring the robot perpetually maintains the human within its visual purview, adeptly tracking user movements. We subjected our approach to rigorous testing involving multiple tests with real robots. The ensuing results, coupled with comprehensive analysis, underscore the efficacy and inherent advantages of our proposed methodology.

1 INTRODUCTION

In the rapidly evolving field of robotics, intuitive human-robot interaction (HRI) remains a pivotal challenge. The ability for robots to accurately interpret and respond to human actions is crucial for advancing their integration into diverse applications, from industrial automation (Vysocky and Novak, 2016) and healthcare (Mohebbi, 2020) to agriculture (Vasconez et al., 2019) and autonomous vehicles (Mokhtarzadeh and Yangqing, 2018). Traditional control interfaces, such as joysticks and remote controllers, often fail to provide the natural, seamless interaction that users require. This gap underscores the need for more intuitive and user-friendly methods to enhance human-robot collaboration, particularly in the context of Unmanned Aerial Vehicles (UAVs).

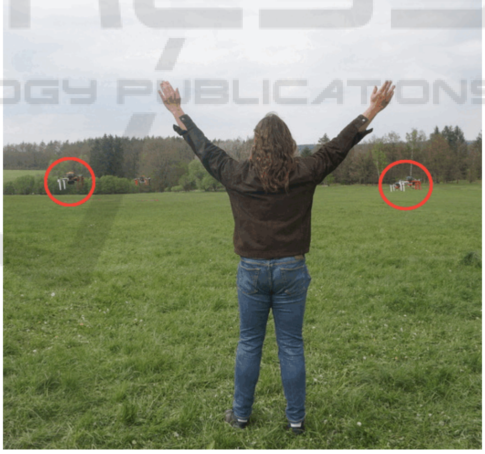


Figure 1: A Group of UAVs controlled by a human operator in an open field.

Recent advancements in action recognition and human-robot collaboration have shown significant promise in addressing these challenges. For instance, research on annotating human actions in 3D point clouds has demonstrated the importance of precise and flexible data for collaborative tasks with industrial robots, emphasizing the potential of 3D data to improve HRI systems (Krusche et al., 2023). Similarly, the integration of natural language instructions and 3D gesture recognition has enhanced the intuitiveness of human-robot interaction, making it more effective for industrial applications by facilitating a more natural communication interface (Park et al., 2024). Studies focusing on end-to-end systems for human-UAV interaction highlight the relevance of intuitive con-

[a] https://orcid.org/0000-0001-7857-7641
[b] https://orcid.org/0000-0002-9319-2114
[c] https://orcid.org/0000-0001-7106-3816

Chaudhary, A., Nascimento, T. and Saska, M.
Intuitive Human-Robot Interface: A 3-Dimensional Action Recognition and UAV Collaboration Framework.
DOI: 10.5220/0012921300003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 26-35
ISBN: 978-989-758-717-7; ISSN: 2184-2809

trol mechanisms in field applications, showing how 3D gestures can be effectively used to control UAVs in real-time scenarios (Jiao et al., 2020). Additionally, the use of intuitive interaction systems, such as RFHUI, highlights the significance of gesture recognition in enhancing the ease of operation and control of UAVs in 3D space (Zhang et al., 2018).

Despite these advancements, interpreting human gestures and translating them into robotic actions remain significant hurdles. The complexity of human movements and the variability in their execution pose challenges for robotic systems, particularly those with limited computational power, such as UAVs. Our research aims to bridge this gap by proposing a novel methodology for real-time, low computationally expensive, three-dimensional action recognition and UAV collaboration. By leveraging stereo cameras to capture both RGB and depth data, we can extract and classify three-dimensional human poses from continuous video feeds. This approach enables the UAV to accurately interpret human movements and respond appropriately 1, thereby enhancing the intuitiveness and effectiveness of human-UAV interaction. Our main contributions are:

1. A new method to estimate three-dimensional full-body pose from available 2D poses.
2. A proposed feature vector space tailored for Human Motion Recognition.
3. A unique, fast and lightweight human motion classifier suitable for UAVs with limited computing power.

2 RELATED WORKS

Human-robot interaction (HRI) offers a myriad of methodologies. Among these, the most intuitive is the teleoperation of a robot through a physical controller. Yamada et al.(Yamada et al., 2015) employ this strategy by integrating it with virtual reality to direct robots in construction scenarios. Conversely, Sathiyanarayanan et al.(Sathiyanarayanan et al., 2015) harness a wearable armband, translating its gestures into commands for robot systems. For individuals with disabilities, voice-controlled systems present an invaluable solution. Gundogdu et al.(Gundogdu et al., 2018) pioneered such a system, facilitating the operation of prosthetic robot arms. Our proposed action classification method adeptly amalgamates RGB video data with depth video output, enabling the classification of 3D human movements. Our prior research (Chaudhary et al., 2022) explored a similar domain, but was constrained to 2D data, thereby limiting the dominion of the user over the robot and impeding optimal performance. Our tailored feature space for k-Nearest Neighbor further enhances its effectiveness even amidst intricate actions by using depth information, a custom feature space, and fast lookup times during the classification process.

In the realm of sequence classification for human motion classification, Celebi et al. (Celebi et al., 2013) marked a significant advancement by introducing a weighted Dynamic Time Warping (DTW) methodology that achieved a remarkable accuracy of 96%, a substantial improvement from the preceding state-of-the-art's 62.5%. Following this, Rwigema et al. (Rwigema et al., 2019) refined the system by integrating a differential evolution strategy to optimize DTW's weightings. This enhanced approach achieved a stellar accuracy of 99.40%. However, its extended processing time posed challenges for real-time classification applications. In parallel, Schneider et al. (Schneider et al., 2019) melded DTW with the one-nearest neighbour technique for movement classification. Their methodology closely aligns with our approach, prompting a comparative analysis between their method and ours to discern our method's efficacy. Additionally, Yoo et al. (Yoo et al., 2022) demonstrated rapid processing capabilities in their classification system, although it necessitated an auxiliary IMU sensor for optimal performance. Additionally, their system is limited to palm-action classification. Our proposed approach endeavours to address the trifecta of challenges: processing speed, accuracy, and sensor dependency, offering a holistic solution in the domain of human motion recognition.

3 METHODOLOGY

Our overarching ambition is to architect a method that is apt for deployment on flying robots and proficient in human detection, action classification, and the subsequent translation of these actions into robotic tasks. Given the inherent computational limitations of UAVs, the challenge lies in devising a classifier that synergizes accuracy with computational efficiency.

A salient feature of our methodology is its prowess in classifying 3D actions. This capability augments the spectrum of detectable movements, offering an enriched, intuitive user interaction. It not only mitigates potential classification errors where 2D projections might be misleading but also empowers the system to discern directional nuances from actions, granting users refined control over the trajectory of the robot.

Structured methodically, our approach is seg-

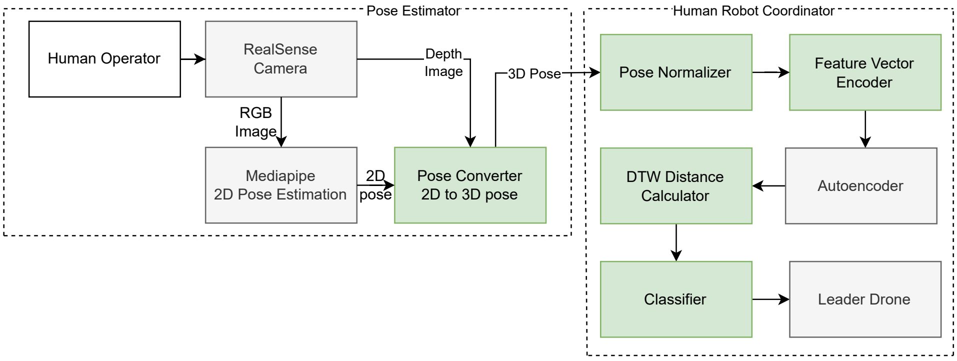


Figure 2: Flowchart depicting the action classifier and its use in UAV control, with the green blocks depicting our contribution.

mented into three core modules: Pose Estimation, Action Classification and UAV Control. An overview of our proposed approach can be seen in Fig. 2.

4 POSE ESTIMATION

The recognition of static gestures can be achieved either directly from RGB images via neural networks, or by distilling skeletal data and applying subsequent functions to this data for classification. For the 3D representation, we initially harness a pre-existing model to derive 2D poses from live video and subsequently use depth images to gauge the distance of the pose joints from the camera.

4.1 2D Pose Estimation

Our choice was to use MediaPipe, an implementation rooted in BlazePose (Bazarevsky et al., 2020). This Google-based pose estimator is adept at extracting 2D pose information from an RGB image in a live video stream, capturing 33 key landmarks of human anatomy. Calibration in MediaPipe is anchored on the 'Vitruvian Man', ensuring accurate scale and orientation recognition. This precision is paramount- especially in aerial robotics, where UAV movement can result in a non-horizontal camera orientation relative to the ground.

MediaPipe operates in a distinct fixed-rate loop, processing the concurrent image. If image callbacks outpace the MediaPipe loop rate, some frames might be skipped. Despite this, our onboard system managed a commendable 30 Hz MediaPipe loop frequency, ensuring near-real-time responsiveness.

4.2 3D Pose Estimation

Our methodology leverages 2D poses to predict the z-coordinate for each landmark. Assuming the camera plane as the X-Y plane, each body pose keypoint corresponds to a coordinate (x,y). Beyond just RGB, the Intel Realsense also provides depth images, albeit with a slight misalignment. By aligning the depth image with the RGB counterpart, we ensure a precise correspondence between them.

The z-coordinate is deduced by overlaying the 2D poses on the depth image. An area surrounding each landmark is considered, factoring in the surrounding pixels for the best z-coordinate estimate. The area of consideration inversely correlates with the distance from the camera. This necessitates an initial estimation of this distance, achieved by superimposing the coordinates of the shoulders and hips on the depth image, creating a bounding quadrilateral. Given the potential for background inclusion, we opt for the first quartile average of the depth values, excluding background measurements.

Given the output from an RGB-D camera like Realsense, each pixel's correspondence to an actual physical area varies with the distance. At a distance of 1 meter, each pixel represents an area of 1.5mm × 1.5mm, whereas at 6 meters, it is 9mm × 9mm. With human proportions in consideration, we scale the number of pixels to match a 90mm × 90mm area. The subsequent z-coordinate estimation, combined with the MediaPipe output, yields a comprehensive 3D pose.

5 FULL-BODY ACTION CLASSIFICATION

Given the computational constraints on our fleet of UAVs, the emphasis is on a lightweight algorithm that retains accuracy. A balance is struck with a custom k-nearest Neighbor (kNN) classifier. Our approach to action classification is holistic, beginning with the careful design of a feature space. By leveraging the relationship between several joints in 3D space and in time, we derive a representative feature vector that encapsulates the dynamics of human posture. The core of our methodology adopts a two-stage classification approach. In the initial stage, Feature embedding translates pose landmarks into meaningful vector space. Autoencoders are then employed to reduce the dimensionality of the embeddings. Subsequently, Dynamic Time Warping (DTW) is used for a refined comparison, accounting for potential temporal variations and guaranteeing the precise alignment of sequences. This comprehensive strategy ensures a balance between computational efficiency and classification accuracy, providing a robust solution for human action recognition. The steps of classification are detailed in sections 6 and 7.

6 FEATURE SPACE DESIGN

In the realm of human pose estimation and analysis, the extraction of meaningful features from detected landmarks is of prime importance. The proposed methodology focuses on extracting embeddings from 13 key anatomical landmarks, which are represented by their 3D coordinates (x, y, z).

6.1 Landmarks Identification

The identified landmarks in the human body are as follows: Nose, Left and Right Shoulder, Left and Right Elbow, Left and Right Wrist, Left and Right Hip, Left and Right Knee, and Left and Right Heel.

6.2 Pose Normalization

To ensure consistency across different poses and individuals, the detected landmarks undergo a normalization process. The normalization is performed in three stages:

1. **Translation Normalization:** The landmarks are translated such that the centre of the pose (midpoint between the hips) is at the origin.
2. **Scale Normalization:** The landmarks are scaled based on the size of the torso or the maximum distance from any landmark to the pose centre, multiplied by a given torso size multiplier.
3. **Orientation Normalization (optional):** The landmarks are rotated to align the vector connecting the hip centre to the shoulder centre with a predefined target direction, ensuring an upright orientation of the pose.

6.3 Embedding Calculation

For the efficient extraction of features from these landmarks, an `EmbeddingCalculator` class has been utilized. In constructing the feature space for action classification, we prioritized features with inherent resilience to noise and occlusions, essential in 3D pose estimation. Selection focused on relative positions and orientations between joints, as these are less sensitive to occlusions and provide a stable reference in noisy data. Additionally, incorporating temporal features like joint velocity and acceleration helps smooth out noise over time. Depth information plays a crucial role in enhancing occlusion handling, allowing for a more accurate estimation of partially visible actions. This strategic selection ensures our classification remains robust across diverse and challenging conditions.

1. **Single Joint Operations:** Processes features derived from individual landmarks.
2. **Joint Pair Operations:** Processes features derived from pairs of landmarks.
3. **Tri Joint Operations:** Processes features derived from groups of three landmarks.

6.3.1 Single Joint Operations

For each landmark, the following are calculated:

- **Joint Vector**: Directly takes the 3D coordinates of the landmark. Provides the spatial positioning necessary for accurate pose recognition, crucial for interpreting directional UAV commands based on limb orientation.

  $$\text{Joint Vector} = \text{landmarks}[i] \tag{1}$$

  where i is the index of the landmark in the predefined list.
- **Joint Velocity (v):** The rate of change of the joint's position with respect to time. Both velocity and acceleration are essential for distinguishing dynamic gestures from static poses, enabling the UAV to interpret the urgency or intended pace of human commands. It is calculated as:

  $$v = \frac{\text{current joint vector} - \text{previous joint vector}}{\text{current timestamp} - \text{previous timestamp}} \tag{2}$$
- **Joint Acceleration (a):** The rate of change of a joint's velocity with respect to time. It is computed as:

  $$a = \frac{v - \text{previous } v}{\text{current timestamp} - \text{previous timestamp}} \tag{3}$$

- **Joint Vector Angle**: The angle between the joint vector and each of the coordinate axes (x, y, z). Offers insights into the limb orientation, critical for understanding gesture directionality and ensuring precise UAV response to commands like vertical takeoff or horizontal movement. For a given joint vector **v** and axis **a**, the angle θ is computed using the dot product:

$$\theta = \arccos\left(\frac{\mathbf{v}\cdot\mathbf{a}}{\|\mathbf{v}\|\|\mathbf{a}\|}\right) \tag{4}$$

- **Joint Angular Velocity:** The rate of change of a joint's vector angle with respect to time. Joint angular velocity and acceleration help the system gauge the smoothness or abruptness of movements, aiding in the interpretation of gesture urgency for immediate or deliberate UAV actions.
- **Joint Angular Acceleration:** The rate of change of a joint's angular velocity with respect to time.
- **Displacement Vector:** The change in position of the joint from its previous position. Indicates the trajectory of joint movements, guiding the UAV in adjusting its flight path to align with the operator's intended direction.

6.3.2 Joint Pair Operations

For each pair of landmarks, the following are computed:

- **Joint Pair Vector:** The difference in the 3D coordinates of the two landmarks. Provides a relational understanding of body posture by examining vectors between pairs of joints, aiding in the nuanced differentiation of gestures for accurate UAV command interpretation.

$$\text{Joint Pair Vector} = \text{landmark}[j] - \text{landmark}[k] \tag{5}$$

  where j and k are the indices of the two landmarks.
- **Joint Pair Velocity, Acceleration, Vector Angle, Angular Velocity, and Angular Acceleration:** These are calculated similarly to the single joint operations, but are applied to the joint pair vector.

6.4 Tri Joint Operations

Given the three landmarks A, B, and C, we can define two vectors:

$$\vec{AB} = B - A \tag{6}$$

$$\vec{BC} = C - B \tag{7}$$

6.4.1 Tri Joint Angle

The angle θ between two vectors $\vec{AB}$ and $\vec{BC}$ is given by:

$$\theta = \arccos\left(\frac{\vec{AB}\cdot\vec{BC}}{\|\vec{AB}\|\|\vec{BC}\|}\right) \tag{8}$$

where $\vec{AB}\cdot\vec{BC}$ is the dot product of the two vectors.

The normal to the plane containing A, B, and C is given by the cross product of the vectors $\vec{AB}$ and $\vec{BC}$:

$$\vec{N} = \vec{AB} \times \vec{BC} \tag{9}$$

The unit normal vector $\hat{N}$ is then:

$$\hat{N} = \frac{\vec{N}}{\|\vec{N}\|} \tag{10}$$

The tri joint angle Θ (or the feature we are considering) is then a combination of θ and $\hat{N}$, which could be represented as:

$$\Theta = \hat{N} \times \theta \tag{11}$$

Our selection of the feature vector was grounded in its capability to uniquely represent the anatomical structure and dynamics. It captures the geometric configuration of poses involving bends or twists, enabling complex gesture recognition for sophisticated UAV manoeuvre commands. The unit normal vector distinctively identifies the plane in which rays connecting a landmark to its neighbouring joints reside. Concurrently, the cosine of the angle effectively captures the relative positioning of these rays. By taking the product of these two entities, we obtain a singular, robust feature vector. This vector augments the feature space, bolstering our ability to discern and classify sequences with heightened precision.

6.4.2 Tri Joint Angular Velocity

The angular velocity ω for the tri joint angle is the rate of change of Θ with respect to time:

$$\omega = \frac{\Delta\Theta}{\Delta t} \tag{12}$$

6.4.3 Tri Joint Angular Acceleration

The angular acceleration α for the tri joint angle is the rate of change of ω with respect to time:

$$\alpha = \frac{\Delta\omega}{\Delta t} \tag{13}$$

6.5 Feature Vector Extraction and Normalization

Upon processing the normalized landmarks with the embedding calculator, we obtain the primary feature vectors, as explained in the previous sub-sections. These vectors, imbued with the dynamics of human movement, are central to our classification scheme.

To ensure a consistent representation across the dataset, we calculate certain parameters. Specifically, for each embedded sample, we ascertain its minimum (min) and maximum (max) values. These extremities are extracted from a concatenated array of embeddings, which is aggregated frame by frame from each sample.

Post parameter estimation, we normalize the feature vectors using min-max scaling. This normalization is pivotal in ensuring that no specific feature overshadows others during the classification process. By mapping the features to a range between -1 and 1, we achieve uniformity in their magnitudes while maintaining their sign.

Integrating these features into our classification framework allows for an advanced, nuanced understanding of human motions, ensuring the UAV actions are tightly coupled with the operator's intent. This synergy between human gestures and UAV response is fundamental for applications requiring intuitive, real-time robot control.

7 ENHANCED K-NEAREST NEIGHBOR CLASSIFIER THROUGH DIMENSIONALITY REDUCTION

Our innovative approach to action classification combines the strengths of dimensionality reduction via autoencoders and an augmented k-nearest Neighbor (kNN) algorithm integrated with Dynamic Time Warping (DTW). This two-step methodology is tailored for varying computational efficiency and accuracy requirements.

7.0.1 Dimensionality Reduction with Autoencoders

In the first pathway, we deploy a deep autoencoder for significant dimensionality reduction of the input space. The autoencoder architecture comprises a series of dense layers that form an encoding phase, transitioning from an input dimension D_{input} to a reduced latent dimension D_{latent}, where $D_{\text{latent}} \ll D_{\text{input}}$. Formally, the encoder function $E : \mathbb{R}^{D_{\text{input}}} \to \mathbb{R}^{D_{\text{latent}}}$ compresses the data, and the decoder function $D : \mathbb{R}^{D_{\text{latent}}} \to \mathbb{R}^{D_{\text{input}}}$ aims to reconstruct the original data. The reconstruction loss is minimized, $L_{\text{reconstruction}} = \|X - D(E(X))\|_2^2$, where X denotes the input data.

After dimensionality reduction, the latent representations are processed through a kNN classifier augmented with DTW as the similarity metric, enhancing accuracy and making it suitable for precision-critical scenarios.

7.1 Dynamic Time Warping (DTW)

Following the encoding of the candidate set C, we refine our matches employing the DTW algorithm. The DTW distance between two sequences S and S' is computed as:

$$D_{\text{DTW}}(S,S') = \min \sum_{(i,j)\in\text{path}} d(s_i, s'_j) \qquad (14)$$

Here, $d(s_i, s'_j)$ denotes the Euclidean distance between the respective feature vectors, while the ”path” symbolizes the optimal alignment between the sequences.

7.2 Classification

Post the DTW filtering, the sequences in our final shortlist dictate the classification outcome. Given the frequency distribution of each class within the shortlist, the class exhibiting the highest prevalence is designated to the incoming sequence. Mathematically, for an incoming sequence S, the assigned class C^* is:

$$C^* = \arg\max_{c\in\text{Classes}} \text{Frequency}(c, \text{Shortlist}) \qquad (15)$$

The selection of k in the kNN classifier is critical for the balance between noise sensitivity and the classifier's generalization ability. We determined the optimal k empirically using a cross-validation approach on various dataset segments to achieve a balance that maximizes classification performance while minimizing error rates. The augmentation of kNN with Dynamic Time Warping (DTW) further enhances its sensitivity to the temporal dynamics of actions, ensuring that the dimensionality reduction does not compromise the classifier's ability to distinguish between similar movements.

7.3 Mathematical Formulation and Empirical Evaluation

Our methodology was empirically evaluated against the UTD-MHAD(Chen et al., 2015) dataset to quantify the performance metrics of accuracy and computational efficiency. The autoencoder-based method focused on reconstruction loss and classification accuracy, utilizing the formula *Accuracy* $=$

$\frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

8 UAV CONTROL

The UAV continuously monitors the position of the human and does so in two steps. In the first step, it tracks the operator's position on the video feed and perpetually corrects its heading, such that the human remains in the centre of its field of view. The next step, it monitors the distance of the human from the drone and tries to maintain a set distance, thereby eliminating the need for the human to stay in one place. The UAV is able to do this while simultaneously receiving commands from the human's actions and performing them.

9 RESULTS

9.1 Method Verification and Benchmarking

To ensure the robustness and reliability of our proposed method, we initially subjected it to rigorous benchmarking. Several performance metrics were employed to verify the efficiency and accuracy of our classifier.

Performance Metrics:

- **Accuracy:** The overall rate of correctly classified human movements among all classifications.
- **F1 Score:** A weighted average of precision and recall, providing a balance between false positives and false negatives.
- **Confusion Matrix:** A detailed breakdown of true positives, false positives, true negatives, and false negatives.
- **Computational Time:** The time taken for the classifier to process an input and generate an output.

The performance metrics are summarized in Table 1. These tests were run on the same Intel NUC that is present in our UAVs (Hert et al., 2023), and therefore accurately reflect the ground reality. The approach was tested on the UTD-MHAD Dataset (Chen et al., 2015), which consists of 27 classes. We also conducted tests with a reduced number of classes to directly compare our method with those in (Schneider et al., 2019). Six classes were chosen for this comparison, with one additional class (a8) added to evaluate its effect on algorithm performance. These 7 actions were, a1: Arm swipe to the left, a6: Cross arms in the chest, a7: Basketball shoot, a8: Hand draw x, a9: Hand draw circle (clockwise), a24: Sit to stand, a27: Forward lunge.

Two versions of the proposed approach were tested on the dataset. The Heavy version utilizes the full set of encodings and is expected to be very accurate, but considerably slower. The encoded version incorporates dimensionality reduction, providing faster classification at slightly lower accuracy. This version offered a good balance between accuracy and computational time for real-time classification. The insights drawn from this evaluation illuminate the trade-offs between accuracy and computational demands, guiding the selection of the optimal configuration for specific application scenarios.

Heavy Configuration: Precision at the Cost of Computational Efficiency. The Heavy configuration (characterized by its utilization of the full set of encodings) demonstrated remarkable accuracy and F1 scores across all evaluated class groupings. It achieved a pinnacle of classification precision in the 6-class setup, with an accuracy of 98.72% and an F1 score of 99. However, this high degree of precision comes at a significant computational cost. The total time for processing 27 classes was recorded at 360.44 seconds, with a per-case time exceeding 2000 ms. This considerable computational demand demonstrates the Heavy configuration's limited applicability in real-time or resource-constrained scenarios.

Encoded Configuration: A Pragmatic Balance. Emerging as the balanced contender, the Encoded configuration significantly reduces computational time without drastically compromising on accuracy. For 6 classes, it maintained an impressive accuracy of 97.44% and an F1 score of 97, with a markedly reduced per-case time of approximately 38 ms. This configuration adeptly balances computational efficiency and precision, making it an ideal candidate for real-time applications. While notable, the dip in accuracy to 83.24% for 27 classes still positions the Encoded configuration as a robust option, capable of handling a diverse range of movements with considerable accuracy.

Discussion on Trade-offs and Configuration Selection. The analysis of the three configurations highlights a fundamental trade-off between computational efficiency and accuracy. The Heavy configuration, while highly accurate, may not be feasible for real-time applications due to its significant computational demands. The Encoded configuration stands out as the optimal choice for applications requiring a harmonious balance between accuracy and computational speed, offering high performance without substantial

Table 1: Performance metrics of the proposed method.

Configuration	Classes	Test/Train cases	Accuracy (%)	Total Time (s)	Per case Time (ms)	F1 Score
Heavy	27	173/688	98.27	360.44	2083	98
	6	39/153	98.72	59.03	1513	99
	7	45/178	95.56	71.71	1594	96
Encoded	27	173/688	83.24	23.82	138	84
	6	39/153	97.44	1.4896	38	97
	7	45/178	86.67	1.7458	39	87

sacrifices. For real-time applications, the Encoded configuration's balanced performance profile makes it exceptionally suitable. It provides a viable solution that accommodates the need for quick processing times, while still maintaining a high level of accuracy. This balance is crucial for deploying efficient and responsive systems in dynamic environments where both precision and speed are essential.

Table 2: Comparison of accuracy with state-of-the-art methods.

Method	Accuracy (%)
Classical	60
Schneider et. el. (Schneider et al., 2019)	63-76
Rwigema et al. (Rwigema et al., 2019)	99.4
Celebi et al. (Celebi et al., 2013)	96
Proposed method (Encoded)	86-97

Additionally, the confusion matrices of the Encoded variant of the approach for 6 and 7 gestures respectively are displayed in Fig 3 and Fig 4. The model exhibits a high degree of accuracy for gestures, such as a24_sit_to_stand, a26_lunge, and a7_basketball_shoot, which are likely to have distinct starting and ending poses or unique motion patterns that are easily distinguishable. Confusions are primarily seen with a1_swipe_left which is sometimes mistaken for a8_draw_X and vice versa, suggesting that the horizontal component of the swipe is similar to part of the "draw X" motion. Similarly, misclassifications between a1_swipe_left and a9_draw_circle_cw imply that certain segments of the swipe and circular gestures may be indistinguishable to the model. The model's difficulty in differentiating between a1_swipe_left and gestures involving complex hand trajectories (a8_draw_X, a9_draw_circle_cw) indicates a potential area for improvement. Refinement of the feature set and the inclusion of more granular temporal data could enhance the model's ability to discern between these gestures with overlapping features.

9.2 Comparative Analysis with State-of-the-Art Methods

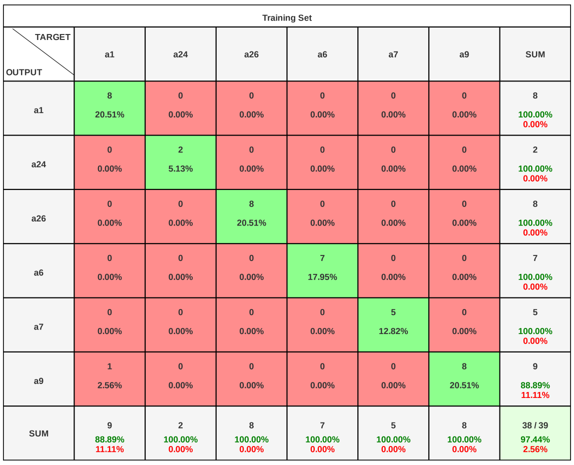
In this section, we juxtapose the performance of our proposed Encoded configuration against various state-of-the-art methods, as summarized in the accompanying table. The comparative analysis is crucial to positioning our work within the broader landscape of action classification methodologies, emphasizing its competitive advantages and identifying areas for further refinement.

Overview of Comparative Performance

1. Classical Methods: These approaches, typically involving hand-engineered features and classical machine learning algorithms, show a base accuracy of 60%. Our method significantly surpasses this benchmark, demonstrating the efficacy of modern, data-driven approaches in handling complex classification tasks.
2. Schneider et al.: With accuracies ranging between 63% to 76%, the work by Schneider et al. closely aligns with the initial performance metrics our study aimed to exceed. By achieving accuracy between 86% to 97% in the Encoded configuration, our method not only surpasses Schneider et al.'s performance, but also showcases the potential of embedding calculators and dimensionality reduction techniques in enhancing classification accuracy.
3. Rwigema et al.: Although Rwigema et al.'s method achieves an impressive accuracy of 99.4%, it is noted for its unsuitability for real-time applications due to substantial computational requirements. This highlights a critical aspect of our research focus—balancing high accuracy with computational efficiency to enable real-time classification.
4. Celebi et al.: The method by Celebi et al. presents a high accuracy of 96%, situating it as a leading approach within the field. Our proposed method's performance falls within this high-accuracy bracket while emphasizing real-time applicability.

9.3 Real-World Deployment

Upon verification, we proceeded to deploy our system on our UAV platform(Hert et al., 2022)(Hert et al., 2023) and our UAV control system(Baca et al., 2021),



Training Set

TARGET OUTPUT	a1	a24	a26	a6	a7	a9	SUM
a1	8 20.51%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	8 100.00% 0.00%
a24	0 0.00%	2 5.13%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	2 100.00% 0.00%
a26	0 0.00%	0 0.00%	8 20.51%	0 0.00%	0 0.00%	0 0.00%	8 100.00% 0.00%
a6	0 0.00%	0 0.00%	0 0.00%	7 17.95%	0 0.00%	0 0.00%	7 100.00% 0.00%
a7	0 0.00%	0 0.00%	0 0.00%	0 0.00%	5 12.82%	0 0.00%	5 100.00% 0.00%
a9	1 2.56%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	8 20.51%	9 88.89% 11.11%
SUM	9 88.89% 11.11%	2 100.00% 0.00%	8 100.00% 0.00%	7 100.00% 0.00%	5 100.00% 0.00%	8 100.00% 0.00%	38 / 39 97.44% 2.56%

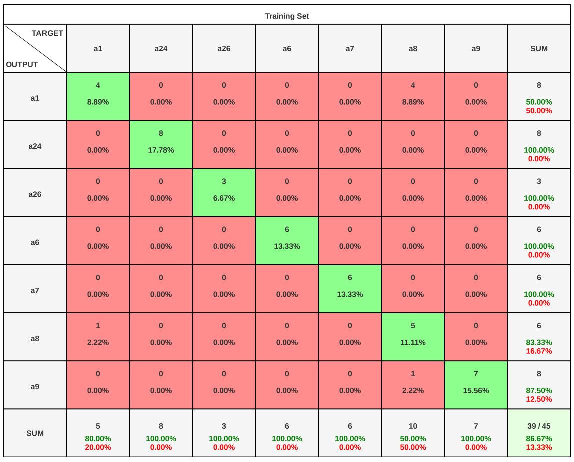
Figure 3: Confusion Matrix of Encoded Variant with 6 Gesture Classes.

to assess its real-world applicability. The primary focus of this phase was to determine how well our classifier could translate laboratory results into practical, actionable commands in an outdoor environment.

During the trials, the UAV was subjected to a series of predefined human actions. The 6 gestures that were used for lab validation (a1, a6, a7, a9, a24, a26), were performed for testing, with the human standing between 4-8 meters away from the UAV. The UAV correctly recognized and responded to 19 out of 20 actions, yielding a real-world accuracy rate of 95%. Notably, the proposed approach adeptly handled dynamic environmental factors, such as changing light conditions and background noise due to the presence of clouds, occluding sunlight sporadically throughout the experiment, showcasing its adaptability and robustness. Additionally, we encountered no false positives, which is crucial as the performance of unintended actions is undesirable. The one action that was not correctly identified was classified as a null action, leading to no command being sent to the drone. This is the intended behaviour that we want our approach to adopt. Missing an action is preferable to misidentifying an action and behaving erratically. Fig 5 shows a UAV being controlled by a human operator using gestures.

10 CONCLUSION

In conclusion, the feature space design offers a comprehensive approach to extracting rich embeddings from human pose landmarks. These embeddings, grounded in both anatomical significance and mathematical rigour, are poised to enhance the capabilities of pose-based analysis systems. The comparative analysis underscores the Encoded configuration as the preferred choice for a wide range of applications, particularly those necessitating real-time processing. It embodies a practical compromise, delivering high accuracy and F1 scores with considerably lower computational times compared to the Heavy configuration. The comparative analysis also elucidates the positioning of our proposed method within the action classification domain. By offering a substantial improvement over classical methods and some contemporary approaches, as well as by providing a viable alternative to high-accuracy, computationally intensive methods, our work carves out a niche in the pursuit of real-time, efficient, and accurate action classification. It underscores the importance of methodological advancements that do not sacrifice practical applicability for theoretical precision, thereby aligning with the evolving needs of real-world applications. The proposed method performed exceptionally when deployed on a real UAV, proving its capability in real-world applications.



Training Set

TARGET OUTPUT	a1	a24	a26	a6	a7	a8	a9	SUM
a1	4 8.89%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	4 8.89%	0 0.00%	8 50.00% 50.00%
a24	0 0.00%	8 17.78%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	8 100.00% 0.00%
a26	0 0.00%	0 0.00%	3 6.67%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	3 100.00% 0.00%
a6	0 0.00%	0 0.00%	0 0.00%	6 13.33%	0 0.00%	0 0.00%	0 0.00%	6 100.00% 0.00%
a7	0 0.00%	0 0.00%	0 0.00%	0 0.00%	6 13.33%	0 0.00%	0 0.00%	6 100.00% 0.00%
a8	1 2.22%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	5 11.11%	0 0.00%	6 83.33% 16.67%
a9	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	1 2.22%	7 15.56%	8 87.50% 12.50%
SUM	5 80.00% 20.00%	8 100.00% 0.00%	3 100.00% 0.00%	6 100.00% 0.00%	6 100.00% 0.00%	10 50.00% 50.00%	7 100.00% 0.00%	39 / 45 86.67% 13.33%

Figure 4: Confusion Matrix of Encoded Variant with 7 Gesture Classes.

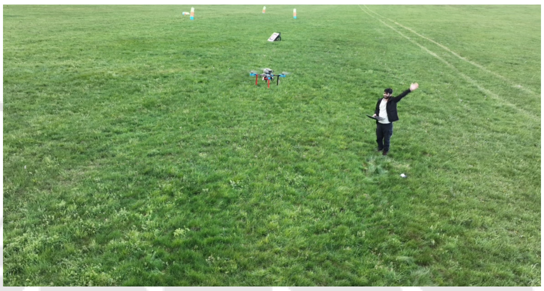


Figure 5: A UAV being controlled by a human operator in an open field.

ACKNOWLEDGEMENTS

This work has been supported by CTU grant no SGS23/177/OHK3/3T/13, by the Czech Science Foundation (GAČR) under research project No. 23-07517S, and by the Europen Union under the project Robotics and advanced industrial production (reg. no. *CZ*.02.01.01/00/22_008/0004590), by TACR project no. FW03010020, by the National Council for Scientific and Technological Development – CNPq, by the National Fund for Scientific and Technological Development – FNDCT, and by the Ministry of Science, Technology and Innovations – MCTI from Brazil under research project No. 304551/2023-6 and 407334/2022-0, by the Paraiba State Research Support Foundation - FAPESQ under research project No. 3030/2021.

REFERENCES

Baca, T., Petrlik, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M. (2021). The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 102:26.

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., and Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking.

Celebi, S., Aydin, A. S., Temiz, T. T., and Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. volume 1, pages 620–625.

Chaudhary, A., Nascimento, T., and Saska, M. (2022). Controlling a swarm of unmanned aerial vehicles using full-body k-nearest neighbor based action classifier. pages 544–551. IEEE.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. pages 168–172. IEEE.

Gundogdu, K., Bayrakdar, S., and Yucedag, I. (2018). Developing and modeling of voice control system for prosthetic robot arm in medical systems. *Journal of King Saud University - Computer and Information Sciences*, 30:198–205.

Hert, D., Baca, T., Petracek, P., Kratky, V., Penicka, R., Spurny, V., Petrlik, M., Vrba, M., Zaitlik, D., Stoudek, P., Walter, V., Stepan, P., Horyna, J., Pritzl, V., Sramek, M., Ahmad, A., Silano, G., Licea, D. B., Stibinger, P., Nascimento, T., and Saska, M. (2023). Mrs drone: A modular platform for real-world deployment of aerial multi-robot systems. *Journal of Intelligent & Robotic Systems*, 108:64.

Hert, D., Baca, T., Petracek, P., Kratky, V., Spurny, V., Petrlik, M., Vrba, M., Zaitlik, D., Stoudek, P., Walter, V., Stepan, P., Horyna, J., Pritzl, V., Silano, G., Licea, D. B., Stibinger, P., Penicka, R., Nascimento, T., and Saska, M. (2022). Mrs modular uav hardware platforms for supporting research in real-world outdoor and indoor environments. pages 1264–1273. IEEE.

Jiao, R., Wang, Z., Chu, R., Dong, M., Rong, Y., and Chou, W. (2020). An intuitive end-to-end human-uav interaction system for field exploration. *Frontiers in Neurorobotics*, 13.

Krusche, S., Al Naser, I., Bdiwi, M., and Ihlenfeldt, S. (2023). A novel approach for automatic annotation of human actions in 3d point clouds for flexible collaborative tasks with industrial robots. *Frontiers in Robotics and AI*, 10.

Mohebbi, A. (2020). Human-robot interaction in rehabilitation and assistance: a review. *Current Robotics Reports*, 1:131–144.

Mokhtarzadeh, A. A. and Yangqing, Z. J. (2018). Human-robot interaction and self-driving cars safety integration of dispositif networks. pages 494–499. IEEE.

Park, S., Wang, X., Menassa, C. C., Kamat, V. R., and Chai, J. Y. (2024). Natural language instructions for intuitive human interaction with robotic assistants in field construction work. *Automation in Construction*, 161:105345.

Rwigema, J., Choi, H. R., and Kim, T. (2019). A differential evolution approach to optimize weights of dynamic time warping for multi-sensor based gesture recognition. *Sensors (Switzerland)*, 19.

Sathiyanarayanan, M., Mulling, T., and Nazir, B. (2015). Controlling a robot using a wearable device (myo).

Schneider, P., Memmesheimer, R., Kramer, I., and Paulus, D. (2019). Gesture recognition in rgb videos usinghuman body keypoints and dynamic time warping.

Vasconez, J. P., Kantor, G. A., and Cheein, F. A. A. (2019). Human–robot interaction in agriculture: A survey and current challenges. *Biosystems Engineering*, 179:35–48.

Vysocky, A. and Novak, P. (2016). Human – robot collaboration in industry. *MM Science Journal*, 2016:903–906.

Yamada, H., Muto, T., and Ohashi, G. (2015). Development of a telerobotics system for construction robot using virtual reality. pages 2975–2979. Institute of Electrical and Electronics Engineers Inc.

Yoo, M., Na, Y., Song, H., Kim, G., Yun, J., Kim, S., Moon, C., and Jo, K. (2022). Motion estimation and hand gesture recognition-based human–uav interaction approach in real time. *Sensors*, 22:2513.

Zhang, J., Yu, Z., Wang, X., Lyu, Y., Mao, S., Periaswamy, S. C., Patton, J., and Wang, X. (2018). Rfhui: An intuitive and easy-to-operate human-uav interaction system for controlling a uav in a 3d space. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '18, page 69–76, New York, NY, USA. Association for Computing Machinery.