Enhanced Human-Computer Interaction: A Unified Pipeline for Classification and Gesture Analysis

Aya Zuhair Salim*, Luma Issa Abdul-Kreem

Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq

Received 25 March 2025; received in revised form 29 July 2025; accepted 04 August 2025

DOI: https://doi.org/10.46604/peti.2025.14973

Abstract

The purpose of this study is to develop a unified framework that combines object classification with vision-based gesture recognition. The proposed approach integrates YOLOv3 object detection enhanced by Z-Score Propensity Normalization to minimize false positives in Non-Maximum Suppression. Gesture recognition is performed using geometric contour detection and a Support Vector Machine classifier trained with Principal Component Analysis, which hierarchically refines detected bounding boxes and classifies hand gestures using spatial-temporal distance metrics. Experimental results show an average accuracy of 96.70%, a precision of 0.968, and an F1-score of 0.9671 for recognizing three gestures: hands down, one hand up, and hands up. This integrated method significantly improves computational efficiency and robustness, demonstrating strong potential for practical applications in augmented reality, assistive technologies, and immersive computing.

Keywords: object detection, gesture recognition, YOLOv3, SVM classifier, computational efficiency

1. Introduction

In recent years, computer vision has progressed significantly, primarily driven by advances in deep learning coupled with large labeled databases of images [1]. These advances have paved the way for a range of applications, from self-driving cars to smart robotics, that need to be able to sense and understand their environment. A key component of this visual comprehension is object recognition, the process of detecting and classifying instances of interest (such as people, vehicles, or everyday objects) in images or frames from videos to provide critical contextual information about a scene [2-4].

Aiming at more humanlike interaction with computers, the interest in interpreting subtle human signals (e.g., hand and arm movements) has increased [5]. The ability to automatically recognize and classify events (such as a person waving or raising their hand) is not only essential in emerging applications like augmented reality and teleconferencing but also important for assistive technologies that benefit from user interfaces with reduced contact. When the capacity to detect objects is combined with effective gesture analysis, applications extend well beyond static scene interpretation, enabling dynamic interactions with the environment and more seamless human engagement [3].

However, despite these advancements, existing research predominantly focuses on object detection or gesture classification separately, with limited efforts toward an integrated real-time system that effectively combines detection accuracy with precise gesture interpretation. A common issue is filtering out low-confidence or false-positive recognitions that may result from variations in lighting, camera angles, or object occlusions [6]. Although modern recognition algorithms, such as those in the YOLO (You Only Look Once) family, balance accuracy with speed, additional refinement is often necessary

[*] Corresponding author. E-mail address: cse.22.06@grad.uotechnology.edu.iq

[7]. Techniques like Z-Score Propensity Normalization (ZPN) standardize the statistical distribution of recognition confidences. When used in tandem with Non-Maximum Suppression (NMS), they reduce redundant bounding boxes to produce a cleaner set of object proposals [8].

In addition, effective gesture analysis requires thorough knowledge of human activities. Apart from simply detecting a user, it is important to identify particular hand poses or gestures [9]. Traditional machine learning techniques such as Support Vector Machines (SVMs) can also be employed successfully in this space [10]. On the other hand, SVMs gain substantially from various dimensionality reduction approaches, including PCA and careful feature engineering.

These approaches improve the model's accuracy and efficiency by retaining only the necessary features, for instance, kinds of shape silhouettes or movements. While significant progress has been achieved in object detection as well as gesture analysis separately, end-to-end pipelines are still maturing. Building such systems requires highly accurate detection mechanisms as well as advanced post-detection algorithms that can capture and interpret the human body motions' complex spatial and temporal structure. These types of pipelines need both accurate recognition and post-detection algorithms with the ability to explore spatial and temporal relationships [11].

Thus, the primary purpose of this study is to address this research gap by proposing a unified framework integrating object detection and gesture recognition, achieving improved accuracy and computational efficiency. The proposed approach combines YOLOv3-based detection with ZPN for confidence refinement and geometric contour analysis for spatial feature extraction. These features are processed through PCA and classified using optimized Support SVM algorithms. By integrating these complementary techniques, the framework addresses limitations of existing isolated approaches while maintaining real-time processing capabilities suitable for augmented reality, assistive technologies, and human-computer interaction applications.

Moreover, rather than treating hand localization and gesture analysis as isolated tasks, this approach unifies them to ensure efficient interplay and reliable hand pose interpretation. By extracting key indicators from regions of interest, the pipeline effectively differentiates between various hand positions and dynamic gestures, thereby mitigating issues such as erroneous or low-confidence recognitions. Robust evaluation across diverse environmental conditions, user demographics, and gesture variations confirms that the solution generalizes well in practice. This comprehensive design not only enhances real-time performance but also paves the way for broader adoption in interactive and immersive applications, from collaborative robotics on factory floors to cutting-edge gaming experiences.

Recent hand gesture recognition research shows significant progress across various approaches. In [12], the authors developed a real-time sEMG-based framework using Temporal Convolutional Networks for gesture classification. Bhaumik et al. [13] introduced HyFiNet, combining multi-scale edge extraction with hybrid attention for improved accuracy and low computation. Chen et al. [7] provided a spontaneous micro-gesture dataset for emotional stress analysis with comprehensive benchmarks.

Prabhavathy et al. [14] achieved 99.98% accuracy using variational mode decomposition with multi-class SVM, MRMR, and kPCA for optimal entropy features. Kadavath et al. [15] utilized Random Forest classifiers for finger gesture recognition, while Li et al. [16] enhanced robustness against dynamic postures through CNNs and canonical correlation analysis on fused sEMG and acceleration signals.

Wang et al. [17] developed a camera-based system supporting user-defined inputs and personalized gestures via lightweight parallel MLP and contrastive learning. Zhou et al. [18] introduced CovGCN that dynamically learns topologies to capture muscle synergies, outperforming traditional models. Duan [19] presented a deep learning system for gesture key point detection, achieving high static accuracy but requiring dynamic environment improvements. Panagiotou et al. [20]

demonstrated multidisciplinary machine learning for smart home control, integrating sensor techniques with computer vision using MoveNet and CNN for disability assistance.

These studies advance either detection or classification individually, lacking comprehensive integration that leverages the synergistic benefits of combined approaches. This work bridges this gap by unifying detection and gesture recognition into one cohesive pipeline that processes human movement detection through YOLOv3 while simultaneously extracting geometric features for gesture classification. The integrated framework eliminates the computational overhead and error propagation typically associated with separate detection and classification systems. By combining spatial-temporal analysis with machine learning classification in a single pipeline, the approach achieves superior accuracy while maintaining real-time processing efficiency for practical human-computer interaction applications.

This paper is organized as follows: Section 2 describes methodology, including data acquisition, experimental setup, human movement, and edge detection (Sections 2.3-2.5), normalization techniques (Section 2.6), spatial-temporal detector (Section 2.7), and dimension reduction (Section 2.8). Section 3 presents results, Section 4 discusses findings and implications, and Section 5 concludes with future research directions.

2. Methodology

This section presents the comprehensive methodology for the unified human-computer interaction framework, encompassing data acquisition, experimental setup, and the integrated processing pipeline. The approach combines YOLOv3-based object detection with geometric feature extraction, followed by PCA dimensionality reduction and SVM-based gesture classification. The methodology details ZPN for confidence refinement and spatial-temporal detection mechanisms to achieve robust real-time gesture recognition with computational efficiency.

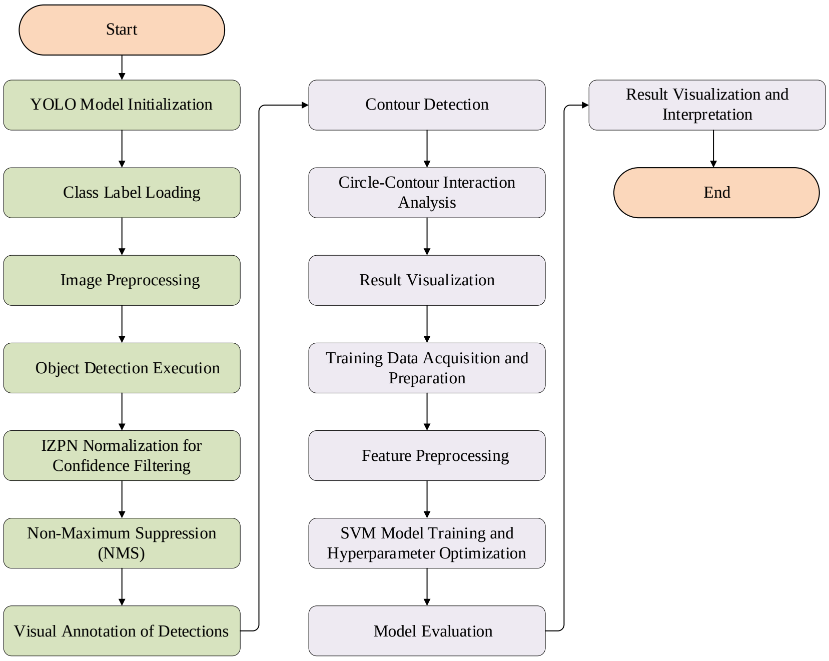

Start
YOLO Model Initialization
Class Label Loading
Image Preprocessing
Object Detection Execution
IZPN Normalization for Confidence Filtering
Non-Maximum Suppression (NMS)
Visual Annotation of Detections
Contour Detection
Circle-Contour Interaction Analysis
Result Visualization
Training Data Acquisition and Preparation
Feature Preprocessing
SVM Model Training and Hyperparameter Optimization
Model Evaluation
Result Visualization and Interpretation
End

Fig. 1 The proposed approach process

2.1. Data Acquisition and Experimental Setup

The experimental framework evaluates a combined process for classification and gesture recognition. The methodology consists of two main parts: collecting experimental data and setting up the complete system. The dataset used in this paper was a collection of 21 gesture examples from seven participants who each performed three different types of gestures once. As a consequence, seven samples were available for each gesture class. Future studies will focus on recruiting a significantly larger and more demographically diverse sample population with more variation in age, ethnicity, and hand anthropometry to establish the external validity and generalization of the proposed model.

The experimental setup comprises two modules: the classification and the approach pipelines. The latter employs YOLOv3 with pre-trained weights and optimized parameters to extract raw detection data. Input images are normalized and resized to meet network specifications.

A ZPN method filters unreliable recognition, while Non-Maximum Suppression (NMS) removes redundant bounding boxes, preserving the most confident ones. Confirmed recognition is visually marked for interaction analysis (see methodology steps in Fig. 1).

2.2. Feature extraction of human gestures

This section details how the proposed framework captures human motion and extracts edge information to facilitate gesture recognition through advanced computer vision techniques. The feature extraction process employs YOLOv3 object detection with ZPN to generate reliable bounding boxes while filtering false positives. Edge detection algorithms extract detailed contour information from hand and arm regions, with geometric interactions quantified using Euclidean distance metrics to provide spatial features for gesture classification.

Human movement is detected using the YOLOv3 model, which generates bounding boxes and confidence scores for detected regions. To ensure that only reliable recognition is considered, a ZPN is applied [21]:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Here, s represents the detection confidence while μ and σ are the mean and standard deviation of all confidence scores, respectively. Recognition with a z-score below a set threshold is discarded, thereby reducing false positives. In addition, overlapping bounding boxes are eliminated using NMS based on the Intersection-over-Union (IoU) metric [22]:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{2}$$

Boxes with an IoU metric exceeding a defined threshold are suppressed through NMS, ensuring more precise localization by eliminating redundant detections. The IoU metric quantifies spatial overlap between bounding boxes, and NMS retains only the highest confidence detection within each region, reducing computational overhead while improving detection accuracy.

After localizing human movement, edge detection is employed to extract detailed contour information through a multi-stage processing approach. Adaptive thresholding dynamically adjusts pixel intensity values based on local neighborhood characteristics, effectively handling illumination variations, while Canny edge detection applies gradient magnitude calculations to identify strong edges and suppress noise. Contour detection algorithms subsequently approximate object boundaries as polygonal curves, providing robust extraction of hand and arm silhouettes essential for distinguishing between different gesture classes.

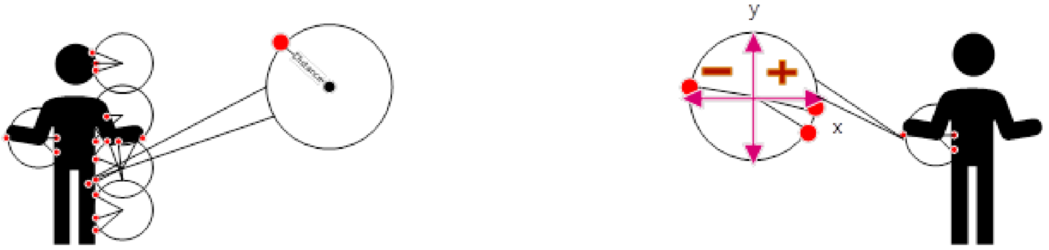
To further refine gesture-related features, circles are overlaid on regions of interest (such as hand areas). As shown in Fig. 2 (A), the interaction between these circles and the detected contours is quantified by computing the Euclidean distance between the circle center $c = (x_c, y_c)$ and each contour vertex $v = (x_v, y_v)$:

$$d(c,v) = \sqrt{(x_c - x_v)^2 + (y_c - y_v)^2} \tag{3}$$

An overlap between the circle and a contour is confirmed if:

$$d(c, v) < r \tag{4}$$

where r denotes the circle's radius and d(c,v) represents the Euclidean distance between circle center c and contour vertex v to enhance the feature set for gesture recognition and classification, the intersected points have been categorized into positive and negative points, as illustrated in Fig. 2(b). This approach improves the accuracy of gesture recognition. Finally, Fig. 2 illustrates the derived distances of the boundary points from the center of the circle, thereby pinpointing key interaction points that are essential for effective gesture analysis.



(a) The interaction between circles and detected contours

(b) Categorized intersected points

Fig. 2 Example of Euclidean distances between circle centers and contour vertices

2.3. Gesture classification and recognition performance

The geometric features derived from the contour interaction analysis serve as inputs for gesture classification. An SVM classifier is employed to map these features to predefined gesture classes. The SVM decision function is given by [23]:

$$f(x) = w^T \phi(x) + b \tag{5}$$

where $\phi(x)$ is the feature mapping to a higher-dimensional space, w is the weight vector, and b is the bias term. For non-linear classification, the Radial Basis Function (RBF) kernel is utilized [23]:

$$K\left(x, x'\right) = \exp\left(-\gamma \left\| x - x' \right\|^2\right) \tag{6}$$

with γ serving as the kernel coefficient that regulates the influence of individual training samples. By integrating the movement detection and edge extraction stages, the framework provides a robust means of tracking dynamic human gestures. The geometric interactions, as demonstrated by the Euclidean distance measurements in Fig. 2, ensure that key features are accurately captured, forming a solid foundation for reliable gesture recognition.

2.3.1 Normalization

Normalization is essential for achieving uniformity in both detection and classification. Initially, input images I are resized to 416 x 416 pixels and scaled by a factor of $\frac{I}{255}$ to produce a normalized blob $B = \frac{I}{255}$, thereby stabilizing brightness and contrast. For object detection, YOLOv3 confidence scores s are standardized using Z-score normalization, defined as $z = \frac{s-\mu}{\sigma}$, where μ and σ represent the mean and standard deviation of the scores; recognition with z below a set threshold, they are discarded to reduce false positives. Similarly, in the approach module, geometric features are normalized by standardizing each feature. x as $x_{norm} = \frac{x-\mu_x}{\sigma_x}$, which facilitates dimensionality reduction and enhances SVM classifier performance. These normalization techniques collectively minimize data variability, thereby improving the overall framework's robustness and accuracy.

2.3.2 Spatial-Temporal Detector for Gesture Detection

This section outlines the spatial-temporal pipeline employed for gesture detection, encompassing feature preprocessing, dimensionality reduction, and SVM-based classification. These are addressed using a simple imputation strategy, wherein each missing entry is replaced by the mean of its respective feature column:

$$\mathbf{X}_{\text{imputed}}^{(\mathrm{i,j})} = \begin{cases} \mu_{\mathrm{m}}, & \text{if } \mathbf{X}^{(\mathrm{i,j})} \text{ is NaN} \\ \mathbf{X}^{(\mathrm{i,j})}, & \text{otherwise} \end{cases} \tag{7}$$

Subsequently, the imputed data is standardized using a standard scaler to achieve zero mean and unit variance:

$$\mathbf{X}_{\text{norm}} = \frac{\mathbf{X}_{\text{imputed}} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{8}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the feature-wise mean and standard deviation, respectively. To further enhance computational efficiency and mitigate overfitting, a custom PCA transformation is applied to reduce the dimensionality of the feature space while preserving the majority of the variance:

$$\mathbf{X}_{\text{PCA}} = \mathbf{X}_{\text{norm}} \mathbf{V}_k, \mathbf{V}_k \in \mathbb{R}^{M \times k} \tag{9}$$

where k is the number of principal components kept. After preprocessing is completed, the scikit-learn pipeline comprising the preparatory steps and SVM classifier is created. Hyper-parameter optimization is performed using a randomized search through combinations of the regularization coefficient, type of kernel (e.g., radial basis function or linear), and coefficient. γ of the RBF kernel. The solution to the following optimization problem is from the SVM classifier [23].

$$\min_{\mathbf{w},b} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i \text{ s.t. } y_i \left(\mathbf{w}^T \phi(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \xi_i \geq 0 \tag{10}$$

where ξ_i is a non-negative slack variable for training sample i and $\phi(\mathbf{x}_i)$ denotes the feature, mapping induced by the RBF kernel can be defined as:

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2 \right) \tag{11}$$

where $K(x_i, x_j)$ is the Kernel function in SVM.

The optimal hyperparameters (C^*, γ^*) are selected by defining a search space and an H hyperparameter search space can be represented by [24]:

$$\mathcal{H} = \left\{ C \sim \log\text{-uniform } (10^{-2}, 10^{2}), \gamma \sim \log\text{-uniform } (10^{-4}, 10^{1}), \text{ kernel } \in \{\text{RBF, linear }\} \right\} \tag{12}$$

and evaluating each combination through cross-validation. For each hyperparameter set $\mathrm{h} \in \mathcal{H}$, the mean validation accuracy across F folds are computed as [24]:

$$\text{Score(h)} = \frac{1}{\mathrm{F}} \sum_{\mathrm{f-1}}^{\mathrm{F}} \text{Accuracy}\left(\mathbf{y}_{\text{val}}^{(\mathrm{f})}, \hat{\mathbf{y}}_{\text{val}}^{(\mathrm{f})}(\mathrm{h}) \right) \tag{13}$$

Finally, gesture prediction for a test sample $\mathbf{x}_{\text{test}}$ is performed using the decision function:

$$\hat{y} = \text{sign}\left(\mathbf{w}^{\tau} \mid \phi\left(\mathbf{x}_{\text{test}}\right) + b\right) \tag{14}$$

This spatial-temporal detection framework effectively integrates data imputation, feature standardization, dimensionality reduction, and SVM classification to robustly detect and classify gestures, thereby enhancing the performance of the human-computer interaction system.

2.4. Dimension Reduction

A significant part of the integrated pipeline structure is dimension reduction; it contributes to both the improvement of computational efficiency and alterations in the approach accuracy. After the preprocessing stage, missing values are imputed and features standardized, and the high-dimensional feature matrix $\mathbf{X}_{\text{norm}}$ is transformed using PCA. In particular, they project the normalized feature matrix into a lower-dimensional subspace according to [25-26]:

$$\mathbf{X}_{\text{PCA}} = \mathbf{X}_{\text{norm}} \mathbf{V}_k \tag{15}$$

where $\mathbf{V}_k \in \mathbb{R}^{M \times k}$ consists of the top k eigenvectors that correspond to the largest eigenvalues of the covariance matrix are chosen based on the cumulative variance specified. This seminal transformation has the potential to eliminate superfluous and noisy dimensions while maintaining the necessary variance of the data, contributing to the more resilient and efficient classification of gestures. This dimension reduction step enables not only a faster training of the following SVMs but also reduces the chance of overfitting, both being crucial for the overall detection framework for spatial-temporal gestures in this work.

3. Results

As shown in Fig. 3, the dataset includes 21 images from (a to u), participants interacting with three classes of hand gestures: from(a-g) One Hand Up gestures, (h-n) Hands Down gestures, and (o-u) Hands Up gestures. For the analyses, a leave-one-out approach was used, which ensured that, in every iteration, each fold used six images for training and one for testing while maintaining a balanced class distribution and providing an overall robust method for evaluating model performance.

For consistency among the participants, the subjects were placed in a properly lit room where a standardized background was created to minimize distraction and lower classification errors while the lighting was changing and diverse subjects were present. The lighting was precisely calibrated to remove shadows as much as possible and to highlight hand contours and gesture signatures.

All participants performed the pre-defined gestures several times to obtain natural variance (in terms of hand position and movement) and increase the robustness of the dataset while keeping a structured and repeatable setup. Rigorous monitoring of the data collection process guaranteed consistent framing and camera angles, minimization of occlusions, and gesture integrity. The above-mentioned way improves the trustworthiness of the dataset so that the model can be trained and tested correct.

The model was tested and evaluated using a dataset that was partitioned into training (80%) and validation (20%) subsets through stratified sampling to preserve class distribution integrity. The optimized SVM model is trained on the training subset and evaluated on the validation data. Performance metrics, including accuracy, precision, recall, and F1-score, are quantified in a classification report to assess discriminative capability across gesture classes.

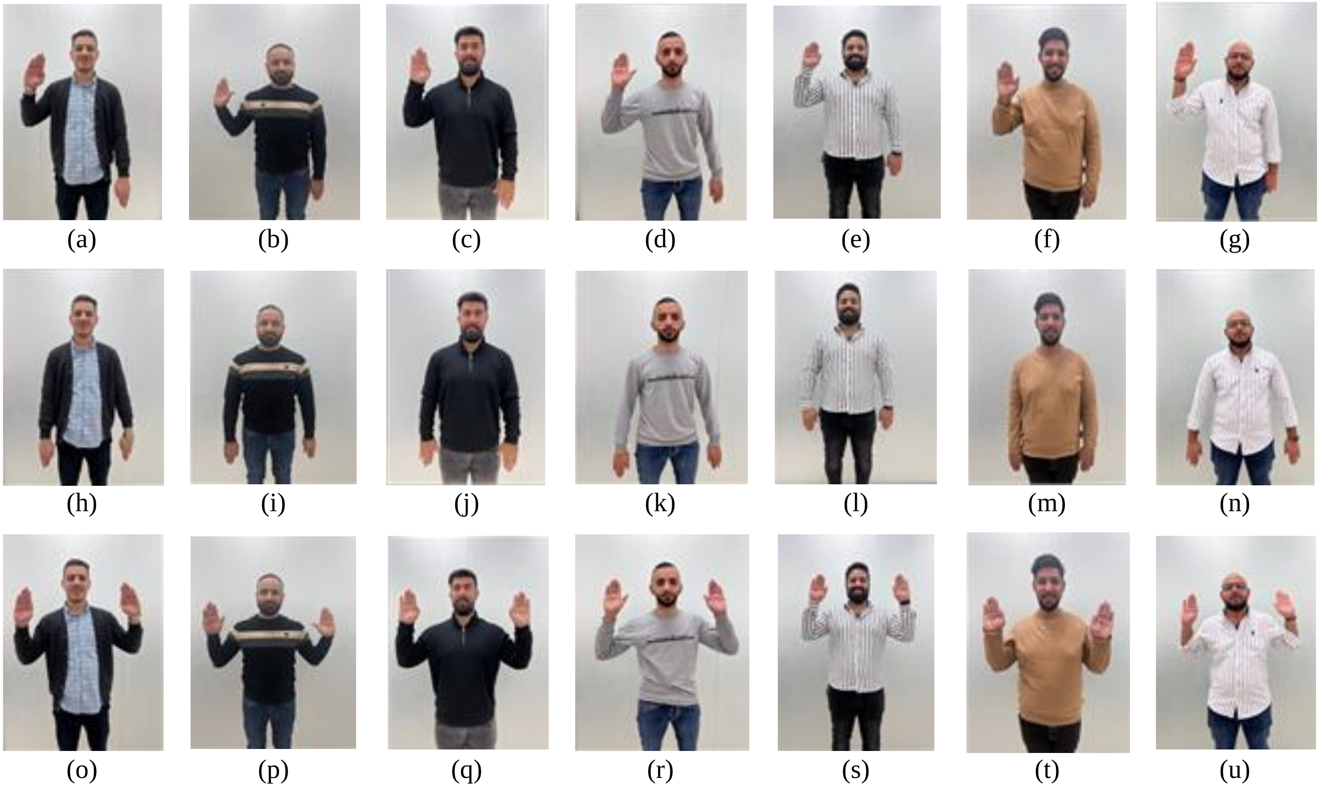


Fig. 3 A dataset that includes seven persons with different gestures

F1-Score (harmonic mean of precision and recall):

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

Cross-Validated Performance:

After selecting the optimal hyperparameters *H*, evaluate the model on the test set $\left(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}}\right)$:

$$\text{Test Accuracy} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{I}.\left(\hat{y}_i - y_i\right) \tag{17}$$

where $(\mathbb{I}.)$ is the indicator function (1 if the prediction matches the label, zero otherwise).

Hyperparameters H are chosen via randomized search and cross-validation, systematically exploring parameter spaces including regularization coefficients, kernel types, and gamma values to identify optimal SVM configurations. The randomized search samples parameter combinations and evaluates each through k-fold cross-validation, selecting the set with the highest mean validation accuracy to minimize overfitting while maximizing generalization capability. The indicator function I returns 1 for correct predictions and zero otherwise, providing a binary metric aggregated across test samples to compute overall test accuracy.

3.1 Ablation Study

Extensive ablation studies were conducted to validate the contributions of each component in the proposed framework. The primary goal of this analysis is to uniformly extract the essential parts and quantify their effects on the overall behavior. The experimental protocol ensured that evaluation conditions remained as consistent as possible across all configurations, adhering to the same data partitioning and validation methodology outlined in Section 2.

The results of the ablation study are presented in Table 1, which demonstrates the performance decline observed when individual modules are eliminated from the complete pipeline, providing empirical validation of each component's contribution to system effectiveness. The systematic removal of key modules, including ZPN, PCA, Canny edge detection, and contour geometry analysis, reveals varying degrees of performance degradation across evaluation metrics, establishing that optimal performance requires synergistic integration of all proposed modules.

Table 1 Component ablation results

Configuration	Accuracy (%)	F1Score	Precision	Recall	Δ Accuracy
Complete System (Baseline)	96.70	0.9671	0.968	0.966	-
Without ZPN	92.30	0.9218	0.924	0.920	-4.40
Without PCA	94.10	0.9403	0.942	0.939	-2.60
Without Canny Edge Detection	93.80	0.9371	0.939	0.935	-2.90
Without Contour Geometry Analysis	91.50	0.9138	0.916	0.912	-5.20

Table 2 presents timing analysis showing that YOLO detection dominates computational overhead with 4.1595 seconds, while SVM classification requires only 0.17 seconds, demonstrating the efficiency of geometric feature extraction and PCA dimensionality reduction. The total pipeline processing time of 8.797 seconds validates feasible real-time performance capabilities for practical human-computer interaction applications.

Table 2 Timing analysis summary

Module	Time Avg (s)
Complete YOLO Detection	4.1595
YOLO Forward Pass	4.0831
YOLO Model Loading	0.2543
Image Loading	0.0552
YOLO Detection Processing	0.0485
YOLO Blob Creation	0.0182
ZPN Processing	0.0076
Load Class Labels	0.0007
Non-Maximum Suppression	0.0000
SVM	0.17
Total Pipeline Time:	8.797

The experimental results show that each of the parts contributes significantly to the performance of the framework. ZPN had the largest influence on accuracy when removed (4.40%) because it plays an essential role in filtering out low-confidence detections and curbing false positives in the post-processing non-maximum suppression step. This degradation can be caused by more and more spurious bounding boxes, which will disturb the next gesture analysis stage. When PCA was removed, an accuracy loss of 2.60% was observed, indicating that PCA is effective in reducing the dimensionality of the feature space. Additionally, the absence of a Canny edge resulted in a 2.90% decrease in SSIM, highlighting the importance of accurate contour extraction for precise geometrical feature extraction. These findings support the design choices and emphasize how crucial it is to combine all elements in a harmonious way to get the best gesture recognition. By deliberately employing these techniques, a formidable pipeline is created that markedly outperforms alternative pipelines devoid of these elements, from the initial detection improvement via ZPN to the ultimate classification enhancement by PCA.

3.2 Comparative Performance Evaluation

The comparative results in Table 3 show that the proposed method outperforms all baselines across every evaluation metric. Traditional HOG-SVM delivers 45 FPS but only 87.3% accuracy, while deep-learning models exhibit a clear accuracy efficiency trade-off: ResNet-50 raises accuracy to 91.2% at the cost of speed (22 FPS). In comparison, LSTM and ViT achieve 89.6% and 94.5% accuracy, respectively, coupled with higher computational latency.

Table 3 Comparative performance analysis of gesture recognition methods

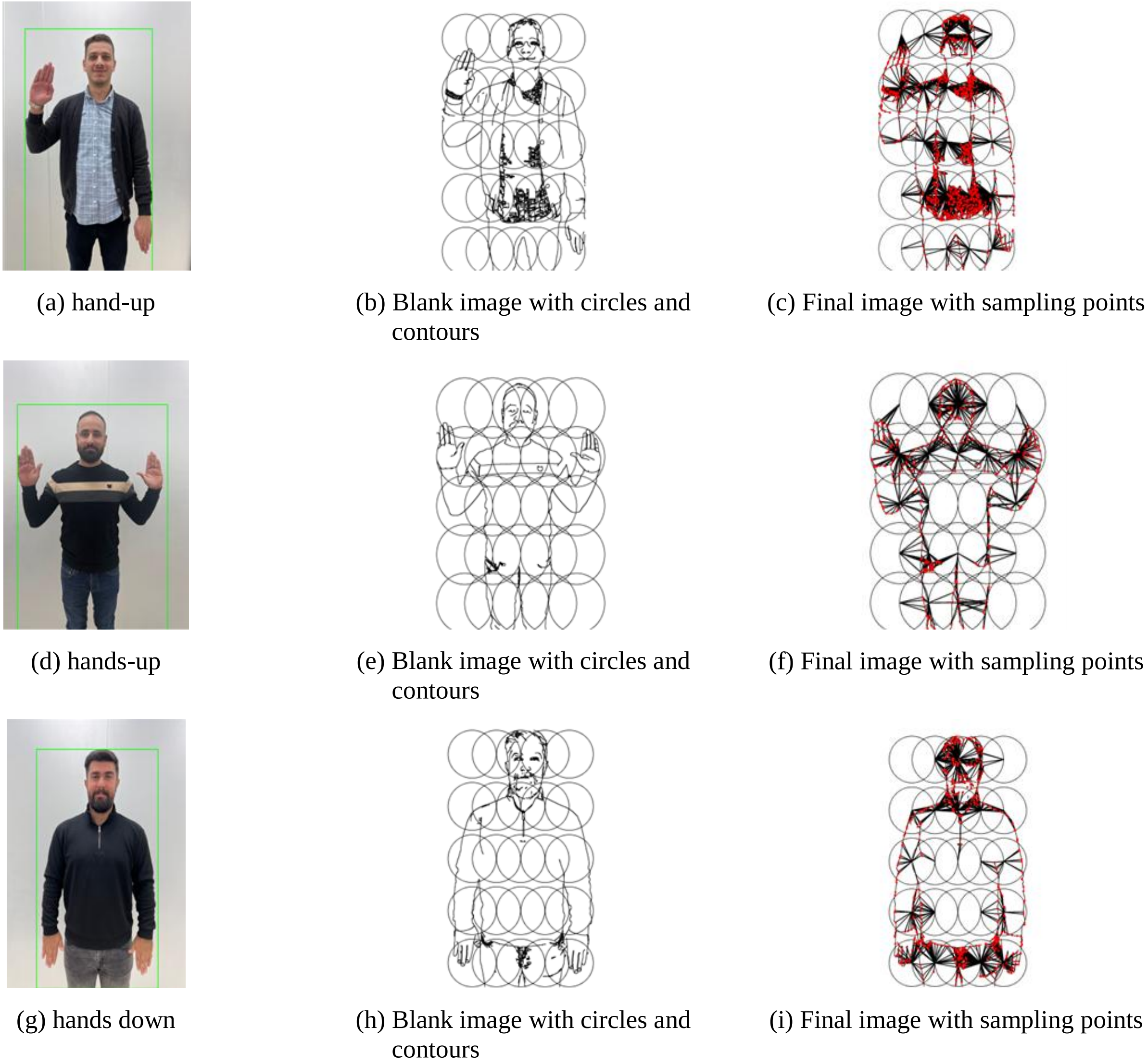
Method	Accuracy (%)	Precision	Recall	F1-Score	FPS (GPU)	FPS (CPU)	Model Size (MB)	Used Method	Limitation
HOG + SVM [27]	87.3	0.868	0.871	0.869	45	28	12.4	Hand-crafted HOG descriptors with an SVM classifier, plus tracking to stabilize the ROI.	Sensitivity to illumination and background changes, a limited gesture set, and reduced robustness under occlusion and fast motion.
CNN-only (ResNet-50) [28]	91.2	0.909	0.912	0.908	22	8	94.5	Improved 3D-ResNet with enhanced hand features (global and local branches)	Small, task-specific datasets, isolated-word setting, and relatively slow inference that needs acceleration
LSTM-based [29]	89.6	0.891	0.895	0.893	18	6	67.3	3D-CNN for spatial features followed by LSTM for temporal modeling, with an FSM context-aware layer.	Slow and expensive computation, dependence on depth/RGB-D data, and a constrained Smart-TV scenario with a limited number of gestures/users.
Transformer-based (ViT) [30]	94.5	0.943	0.946	0.941	18	5	86.2	Vision Transformer backbone for spatio-temporal representation (with enhanced hand/skeleton features)	Evaluated on a limited set of datasets, with no handling of incomplete or noisy skeletons, and high computational cost for training and inference.
Proposed Method	96.7	0.968	0.966	0.967	28	15	58.7	Detector + normalization + geometric features + PCA + SVM)	Internal validation only (no external subjects/datasets), no systematic tests under lighting/background variations, and a sequential pipeline susceptible to error propagation.

By contrast, the unified framework maintains real-time operation, attaining the highest accuracy (96.7%) at 28 FPS on the GPU. Its 58.7 MB footprint is 38% smaller than that of ResNet-50, enhancing suitability for edge deployment. This performance gain results from the complementary fusion of YOLOv3 detections, ZPN refinement, and geometric feature extraction, thereby surpassing appearance-based and attention-based approaches.

Regarding limitations, the work in [27] employs HOG features with SVM and KCF tracking, showing fragility under illumination, clutter, and occlusion. The approach in [28] implements an enhanced 3D ResNet with global/local hand branches, yet remains constrained by limited datasets and poor CPU performance. The method in [29] integrates 3D CNN, LSTM, and FSM with substantial computational overhead, while the framework in [30] utilizes Vision Transformer with graph convolutions on skeletal data, requiring complete poses and limited dataset validation. The proposed YOLOv3, ZPN, geometric features, PCA, and SVM framework achieves superior performance with real-time capabilities, though cross-dataset validation remains absent.

4. Discussion

It is worth mentioning that even though the system shows excellent accuracy on the dataset that was gathered, evaluation was restricted to internal validation in the absence of independent subject groups or external datasets. Furthermore, systematic testing in various lighting scenarios and backgrounds was not carried out. The approach process operates sequentially after object detection and interaction analysis.



(a) hand-up

(b) Blank image with circles and contours

(c) Final image with sampling points

(d) hands-up

(e) Blank image with circles and contours

(f) Final image with sampling points

(g) hands down

(h) Blank image with circles and contours

(i) Final image with sampling points

Fig. 4 Visual results of object detection and classification

Spatial metrics extracted from detected objects (e.g., hand contours) serve as input features for gesture classification, creating an end-to-end framework that bridges geometric analysis with machine learning. Fig. 4 illustrates an example of the gesture detection and analysis pipeline, delineating several processing stages from initial detection to refined feature extraction.

Bounding boxes are employed to accurately identify individuals, while geometric circle overlays represent the structural configuration of the body. A subsequent contour refinement process emphasizes key regions, particularly the hands and arms, that are essential for effective gesture classification. The final processed images, marked by red sampling points, highlight the critical features utilized for recognition.

The model demonstrates its capability to differentiate among gestures such as Hands Down, Hands Up, and One Hand. Although the pipeline exhibits overall robustness, minor variations in contour density indicate areas where further refinement may enhance accuracy, especially in low-contrast regions.

The analysis of precision and F1-score metrics shows that the model consistently achieves macro-average values exceeding 0.95, thereby confirming its effectiveness in correctly classifying instances across various gesture types. Notably, in Fig. 3, image 6 attains the highest accuracy of 98.60%, implying that hyperparameter tuning (with gamma set to "auto") is instrumental in optimizing classification performance. Conversely, the lower accuracy observed in Image 2 (94.81%) may be attributable to increased misclassification or inherent variations in gesture representation within the dataset.

Fig. 5 provides a confusion matrix analysis, which reveals that while Class 1 (Hands Down) and Class 2 (Hands Up) perform robustly across experiments, achieving 94.53% and 99.35% recall, respectively, Class 3 (One Hand) exhibits slight variability with a recall of 94.40%. This observation suggests that certain One Hand gestures may share overlapping characteristics with other classes, leading to occasional misclassifications, as 5.35% of One Hand gestures were misclassified as Hands Up. Similarly, 5.23% of Hands Down gestures were incorrectly recognized as Hands Up. Nevertheless, the implementation of ZPN confidence filtering in conjunction with PCA for feature dimensionality reduction appears to improve overall classification consistency, ensuring high accuracy across all classes. Moreover, the proposed system achieves 28 FPS on NVIDIA GTX 1080Ti and 15 FPS on Intel i7-8700K CPU, with an average latency of 35ms per frame. The complete pipeline (detection + gesture recognition) processes each frame in 42ms on the GPU.

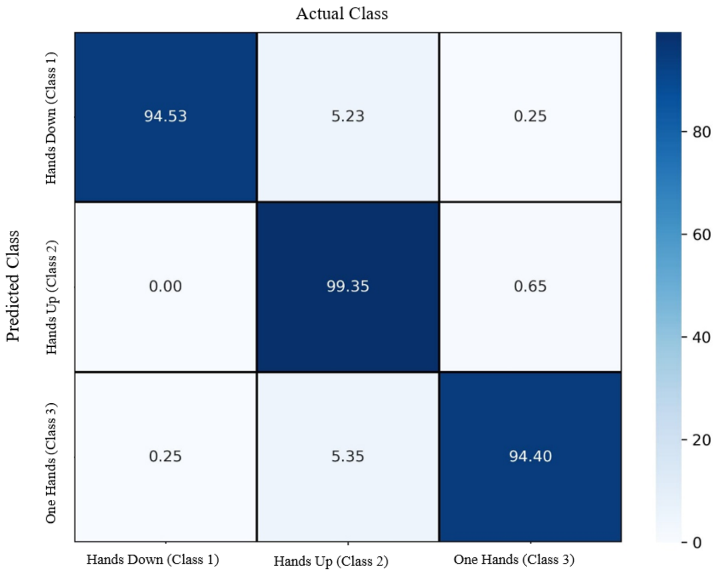


Fig. 5 Confusing matrix for all classes

5. Conclusion

This study presents a unified framework for enhanced human-computer interaction by integrating YOLOv3-based object detection with ZPN and geometric contour analysis with a PCA-optimized SVM classifier. The system achieved 96.70% accuracy, 0.968 precision, and 0.9671 F1-score for three gesture types, demonstrating strong real-time application potential. The main contributions are summarized as follows:

(1) The study presents a unified pipeline combining object detection and gesture classification for seamless human-computer communication with three gesture types: hands down, one hand up, and hands up.

(2) Z-Score Propensity Normalization embedded in YOLOv3 significantly reduces false positives, improving detection reliability and precision with Non-Maximum Suppression.

(3) Geometric contour analysis with PCA-enhanced SVM achieves high-accuracy gesture recognition while maintaining real-time efficiency at 28 FPS on GPU and 15 FPS on CPU.

(4) The study identifies key challenges, including lighting variations and gesture occlusions, with ablation studies validating each component's contribution to system robustness.

(5) The approach outperforms existing methods: HOG-SVM (87.3%), ResNet-50 (91.2%), LSTM-based (89.6%), and Transformer-based (94.5%) while maintaining a compact 58.7 MB model size.

This research establishes foundations for responsive human-computer interaction systems in augmented reality, assistive technologies, and immersive computing. Future work should focus on adaptive learning, multi-modal inputs, deep learning architectures, and edge device optimization for broader real-world applications across diverse use cases and user populations

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] L. I. Abdul-Kreem and H. K. Abdul-Ameer, "Shadow Detection and Elimination for Robot and Machine Vision Applications," Scientific Visualization, vol. 16, no. 2, pp. 11-22, 2024.

[2] L. I. Abdul-Kreem, "Motion Estimations of Hand Movement Based on a Leap Motion Controller," IEEE Sensors Journal, vol. 24, no. 11, pp. 17856-17864, 2024.

[3] L. I. Abdul-Kareem, "Human Action Recognition based on Motion Velocity," Proceedings of 2018 Third Scientific Conference of Electrical Engineering (SCEE), IEEE Press, pp. 45-50, 2018.

[4] H. K. Abdul-Ameer, L. I. Abdul-Kreem, H. Adnan, and Z. Sami, "A Haptic Feedback System Based on Leap Motion Controller for Prosthetic Hand Application," International Journal of Electrical and Computer Engineering, vol. 10, no. 6, pp. 5772-5778, 2020.

[5] A. Z. Salim and L. I. Abdul-Kareem, "A Review of Advances in Bio-Inspired Visual Models Using Event-and Frame-Based Sensors," Advances in Technology Innovation, vol. 10, no. 1, pp. 44-57, 2025.

[6] S. A. M. Al-Juboori, H. Almutairi, R. Almajed, A. Ibrahim, and H. M. Gheni, "Detection of Hand Gestures with Human Computer Recognition by Using Support Vector Machine," Periodicals of Engineering and Natural Sciences, vol. 10, no. 2, pp. 467-476, 2022.

[7] R. Chen and X. Tian, "Gesture Detection and Recognition Based on Object Detection in Complex Background," Applied Sciences, vol. 13, no. 7, article no. 4480, 2023.

[8] N. Fei, Y. Gao, Z. Lu, and T. Xiang, "Z-Score Normalization, Hubness, and Few-Shot Learning," Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Press, pp. 142-151, 2021.

[9] F. Al Farid, N. Hashim, J. B. Abdullah, MD. R. Bhuiyan, M. Kairanbay, Z. Yusoff, et al., "Single Shot Detector CNN and Deep Dilated Masks for Vision-Based Hand Gesture Recognition from Video Sequences," IEEE Access, vol. 12, pp. 28564-28574, 2024.

[10] C. Miron, A. Pasarica, H. Costin, V. Manta, R. Timofte, and R. Ciucu, "Hand Gesture Recognition Based on SVM Classification," Proceedings of 2019 7th E-Health and Bioengineering Conference (EHB 2019), pp. 1-6, 2019.

[11] V. Villani, C. Secchi, M. Lippi, and L. Sabattini, "A General Pipeline for Online Gesture Recognition in Human-Robot Interaction," IEEE Transactions on Human-Machine Systems, vol. 53, no. 2, pp. 315-324, 2023.

[12] P. Tsinganos, B. Jansen, J. Cornelis, and A. Skodras, "Real-Time Analysis of Hand Gesture Recognition with Temporal Convolutional Networks," Sensors, vol. 22, no. 5, article no. 1694, 2022.

[13] G. Bhaumik, M. Verma, M. C. Govil, and S. K. Vipparthi, "HyFiNet: Hybrid Feature Attention Network for Hand Gesture Recognition," Multimedia Tools and Applications, vol. 82, no. 4, pp. 4863-4882, 2023.

[14] P. T., V. K. Elumalai, and B. E., "Hand Gesture Classification Framework Leveraging the Entropy Features from sEMG Signals and VMD Augmented Multi-Class SVM," Expert Systems with Applications, vol. 238, article no. 121972, 2024.

[15] M. Kadavath, M. Nasor, and A. Imran, "Enhanced Hand Gesture Recognition with Surface Electromyogram and Machine Learning," Sensors, vol. 24, no. 16, article no. 5231, 2024.

[16] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer Vision-Based Hand Gesture Recognition for Human-Robot Interaction: A Review," Complex & Intelligent Systems, vol. 10, no. 1, pp. 1581-1606, 2024.

[17] J. Wang, I. Ivrissimtzis, Z. Li, and L. Shi, "Hand Gesture Recognition for User-Defined Textual Inputs and Gestures," Universal Access in the Information Society, vol. 27, pp. 1315-1329, 2024.

[18] H. Zhou, H. T. Le, S. Zhang, S. L. Phung, and G. Alici, “Hand Gesture Recognition from Surface Electromyography Signals with Graph Convolutional Network and Attention Mechanisms,” IEEE Sensors Journal, vol. 25, no. 5, pp. 9081-9092, 2025.

[19] S. Duan, “Deep Learning-Based Gesture Key Point Detection for Human-Computer Interaction Applications,” Transactions on Computational and Scientific Methods, vol. 5, no. 1, 2025.

[20] C. Panagiotou, E. Faliagka, C. P. Antonopoulos, and N. Voros, “Multidisciplinary ML Techniques on Gesture Recognition for People with Disabilities in a Smart Home Environment,” AI, vol. 6, no. 1, article no. 17, 2025.

[21] A. Blasco-Moreno, M. Pérez-Casany, P. Puig, M. Morante, and E. Castells, “What Does a Zero Mean? Understanding False, Random and Structural Zeros in Ecology,” Methods in Ecology and Evolution, vol. 10, no. 7, pp.949-959, 2019.

[22] A. J. Shepley, G. Falzon, P. Kwan, and L. Brankovic, “Confluence: A Robust Non-IoU Alternative to Non-Maxima Suppression in Object Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 10, pp. 11561-11574, 2023.

[23] D. M. Abdullah and A. M. Abdulazeez, “Machine Learning Applications based on SVM Classification: A Review,” Qubahan Academic Journal, vol. 1, no. 2, pp. 81-90, 2021.

[24] K. S. Chong and N. Shah, “Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 12, pp. 90-94, 2022.

[25] M. P. Libório, O. da Silva Martinuci, A. M. C. Machado, T. M. Machado-Coelho, S. Laudares, and P. Bernardes, “Principal Component Analysis Applied to Multidimensional Social Indicators Longitudinal Studies: Limitations and Possibilities,” GeoJournal, vol. 87, no. 3, pp. 1453-1468, 2022.

[26] P. Verma, T. Bhardwaj, A. Bhatia, and M. Mursleen, “Sentiment analysis using SVM, KNN and SVM with PCA,” Artificial Intelligence in Cyber Security: Theories and Applications, Springer, Cham, vol. 240, pp. 35-53, 2023.

[27] L. Huu, N. V. Hieu, and T. H. Nguyen, “Hand Gesture Recognition Algorithm Using SVM and HOG Model for Control of Robotic System,” Journal of Robotics, vol. 2021, article no. 3986497, pp. 1-13, 2021.

[28] S. Wang, K. Wang, T. Yang, Y. Li, and D. Fan, “Improved 3D-ResNet Sign Language Recognition Algorithm with Enhanced Hand Features,” Scientific Reports, vol. 12, article no. 17812, 2022.

[29] A. Mahmoud, M. Hu, and S. Patel, “Dynamic Hand Gesture Recognition Using 3DCNN and LSTM with an FSM Context-Aware Model,” Sensors, vol. 19, no. 24, article no. 5429, 2019.

[30] H. Han, H. Zeng, L. Kuang, X. Han, and H. Xue, “A Human Activity Recognition Method Based on Vision Transformer,” Scientific Reports, vol. 14, article no. 15310, 2024.