

Real-Time 2D Pose Recognition System Based on MoveNet and Long Short-Term Memory: Design and Implementation [†]

Li-Ci Lin  and Hang-Hong Kuo * 

Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan; rickylinrickylin@gmail.com

* Correspondence: cliffkuo@ncut.edu.tw

[†] Presented at the 6th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB 2026), New Taipei, Taiwan, 24–26 April 2026.

Abstract

With the rapid development of imaging and computing technologies, camera-based human pose analysis has become an important tool for training and performance evaluation. In applications such as fitness training, military drills, and home exercise, there is an increasing demand for real-time and objective posture assessment. This study combines 2D human pose estimation with a temporal model to develop a real-time recognition and feedback system for fixed poses and short upper-limb actions. The system uses MoveNet to extract joint coordinates and angles and a long short-term memory (LSTM) network to distinguish correct from non-compliant postures. Skeleton overlays and textual suggestions provide intuitive corrective feedback when professional instructors are unavailable. Experiments using a salute posture as the target action show that the proposed method produces confidence scores above 0.9 for clearly correct and clearly incorrect cases, demonstrating its feasibility and effectiveness.

Keywords: MoveNet; human pose analysis; LSTM; real-time posture recognition

1. Introduction

The rapid development of deep learning and computer vision has made camera-based human pose analysis an important research direction. Human pose estimation models can convert 2D images into skeletal joint coordinates for applications such as sports performance evaluation, rehabilitation monitoring, and human–computer interaction. Compared with motion-capture systems that require multiple cameras or wearable sensors, 2D pose estimation using a single RGB camera offers lower cost, greater deployment flexibility, and less user burden. Consequently, pose-estimation-based methods have been increasingly applied to motion analysis, posture correction, and real-time interactive systems [1–4].

However, posture assessment in training and practice settings still largely depends on the subjective judgment of coaches or instructors, who visually inspect movements and provide verbal comments. Although this approach is flexible, it is difficult to maintain consistent evaluation standards or generate continuous, quantitative, and traceable posture records. When users practice independently, they may not be able to determine in real time whether their posture deviates from the target. Most existing pose-estimation-based applications focus on general action classification, outcome prediction, or overall performance assessment. Relatively few systems specifically target fixed postures, such as salutes or ready positions, to provide explicit correct/incorrect decisions and concrete adjustment



Academic Editors: Teen-Hang Meen, Shu-Han Liao and Cheng-Fu Yang

Published: 15 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

suggestions. In addition, few systems integrate skeleton visualization, joint angles, and textual feedback in a single real-time interface for self-correction.

Therefore, this study develops a real-time posture recognition and feedback system by combining 2D skeleton-based human pose estimation with a temporal classification model. MoveNet 2D pose estimation is used to extract 17 human body keypoints in real time [5,6], and a long short-term memory (LSTM) network is used to model temporal dependencies in sequential skeletal features [7]. The system classifies fixed upper-limb postures, using the military salute as a demonstration case, and provides on-screen feedback by overlaying skeletons, joint angles, and textual prompts. Preliminary experiments were conducted to assess the feasibility of the proposed system and its usefulness for real-time posture feedback.

2. System Design

The proposed posture recognition and feedback system adopts a layered architecture, and the overall processing flow is shown in Figure 1. The system comprises pose data collection and annotation, data preprocessing and normalization, LSTM-based pose classification model training, and real-time pose recognition and feedback. On the training data and model side, the system extracts 2D skeletal joint sequences from pre-annotated pose image clips. After sequence segmentation, normalization, and data augmentation, these sequences are consolidated into a fixed-length joint time-series dataset, which is then used to train a classification model in an LSTM network and to obtain the model weights and class information for online recognition.

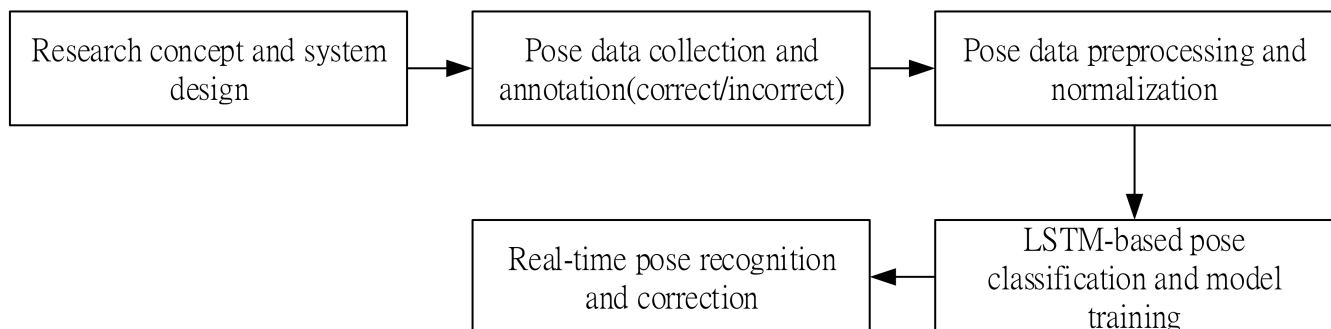


Figure 1. Flowchart of the pose recognition and evaluation system.

On the pose analysis and decision side, the system loads the MoveNet pose estimation model and the trained LSTM model at runtime, extracts skeletal joint coordinates from live camera images, maintains a fixed-length time window for pose classification, and combines rule-based features such as joint angles and head displacement to generate posture assessment and feedback results. On the user side, a human–computer interaction interface presents the overlaid skeleton, angle values, and textual prompts in real time within the Spyder execution window, helping subjects observe their own posture and perform self-correction. The overall flowchart can be divided into five steps.

1. Research concept and system design: Define the target posture evaluation scenario and functional requirements, and design the corresponding system architecture and processing flow;
2. Pose data collection and annotation: Collect images or videos of fixed poses and, according to predefined criteria, label each sample as correct or incorrect to build a supervised dataset;

3. Pose data preprocessing and normalization: Use MoveNet to extract 2D joint coordinates, segment continuous frames into fixed-length sequences, and apply left–right mirroring, scale normalization, and data augmentation;
4. LSTM-based pose classification and training: Input the processed joint sequences and their labels into an LSTM network, tune the model using training and validation results, and obtain a deployable pose classification model;
5. Real-time pose recognition and feedback: During execution, continuously acquire camera images, generate joint sequences with the same preprocessing, classify them with the trained LSTM model, and combine joint-angle and head-movement rules to provide real-time textual feedback on the user interface.

3. Methodology

We combine MoveNet 2D pose estimation with an LSTM network to develop an upper-limb motion recognition and real-time posture correction system. The system captures upper-body images with a camera, uses MoveNet SinglePose-Lightning to extract 2D coordinates and confidence scores of 17 body keypoints [6,7], and organizes them into skeletal sequences for temporal learning. Samples labeled correct and incorrect are used to train the LSTM model to learn temporal patterns of different postures [7]. In the real-time stage, continuous skeletal sequences are fed into the trained LSTM model, and a rule-based module provides immediate feedback on whether the posture matches the target motion and which body parts need adjustment.

The system is implemented on a laptop using Python. TensorFlow is used to build and train the LSTM model, and MoveNet is accessed via TensorFlow Hub for pose inference. OpenCV handles image capture, preprocessing, and overlay of skeletons and textual prompts, forming an interactive interface for self-practice and posture correction. The hardware and software specifications are listed in Table 1.

Table 1. System specifications.

	Component	Specification
Hardware	Laptop computer	Intel Core i7-1165G7; 16 GB RAM of random access memory
	Webcam	Logitech C310 HD webcam (Lausanne, Switzerland)
Software	Component	Specification
	Operating system	Windows 11
	Programming language	Python 3.13.5
	Deep learning framework	TensorFlow 2.20.0
	Pose estimation model	MoveNet SinglePose-Lightning (TF Hub v4)
	Image processing library	Open Source Computer Vision (OpenCV) (https://opencv.org/)
	Development environment	Spyder 6.0.7

3.1. Data Augmentation and Normalization

In the preprocessing stage, we use the military-style salute as a case to describe the data augmentation and normalization procedure, as shown in Figure 2. Two folders of graphics interchange format (GIF) videos, corresponding to correct and incorrect salutes, are used as the data source. Each GIF is treated as a continuous motion sequence, and every frame is encoded as a 51-dimensional feature vector from 17 body keypoints.

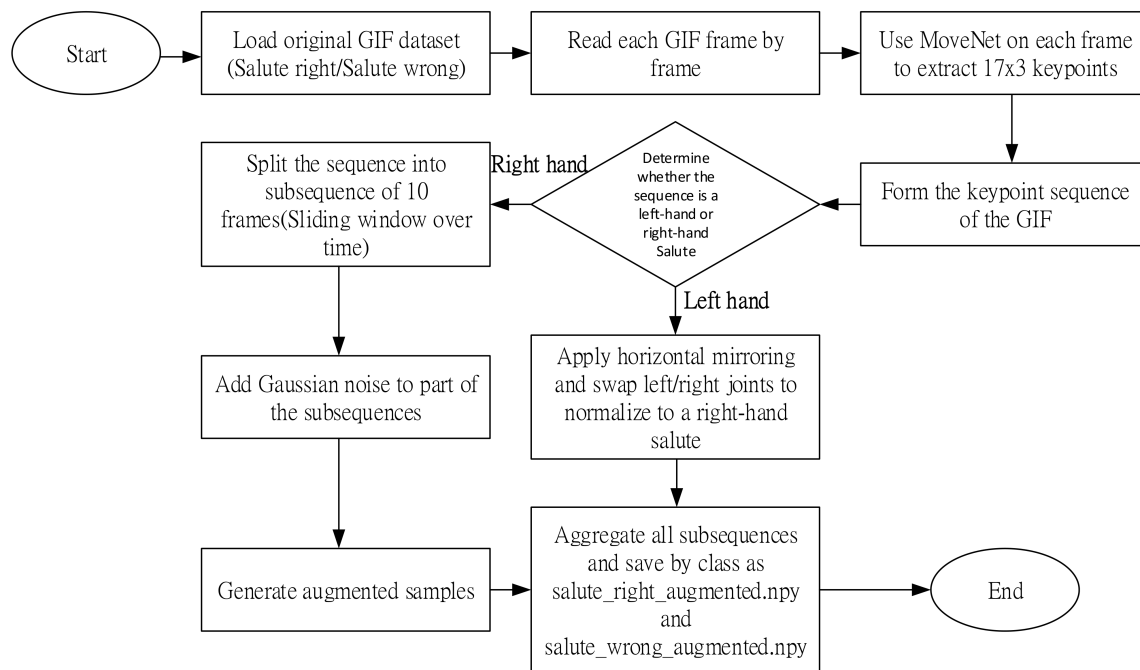


Figure 2. Flowchart of data augmentation and normalization.

MoveNet 2D pose estimation is applied to each frame to obtain the coordinates and confidence scores of the 17 keypoints and form the skeletal joint sequence. For each sequence, the system first checks whether it is a right-hand salute; if a left-hand salute is detected, the horizontal coordinates of paired joints are mirrored, and their indices swapped with respect to the image width, normalizing the pose to an equivalent right-hand salute:

$$x'_{left} = W - x_{right}, x'_{right} = W - x_{left} \quad (1)$$

Here, W is the image width in pixels.

Let $S = \{x_1, x_2, \dots, x_T\}$ denote a sequence of frame-wise feature vectors. A sliding window with length 10 and stride 3 is applied to obtain subsequences $S_i = \{x_i, x_{i+1}, \dots, x_{i+9}\}$ where each S_i contains 10 consecutive frames (51 features per frame) and serves as one LSTM training sample. To enhance robustness to subtle posture changes and measurement noise, zero-mean Gaussian noise $\epsilon \sim N(0, \sigma^2)$ is added to part of the subsequences to create augmented samples:

$$\tilde{s}_i = \text{Clip}(s_i + \epsilon, 0, 1) \quad (2)$$

Here, $\tilde{s}_i$ is the noise level and $\text{Clip}(\cdot)$ constrains the feature values to a valid range. All original and augmented subsequences from the correct and incorrect classes are finally aggregated and stored as NumPy Array (NPY) training files for subsequent LSTM-based pose classification.

3.2. LSTM Training—Merging NPY and GIF Data

In the LSTM training stage, the augmented NPY data are combined with skeletal sequences extracted from the original GIF images to form the pose classification training set, as shown in Figure 3. The system first sets the training data paths and parameters, and checks whether augmented NPY files (such as `salute_right_augmented.npy` and `salute_wrong_augmented.npy`) are available. If they exist, they are loaded and used to build feature matrices and label vectors for correct and incorrect samples. Otherwise, the system checks for the original GIF data and, if needed, reruns skeleton extraction and sequence segmentation to ensure sufficient training inputs.

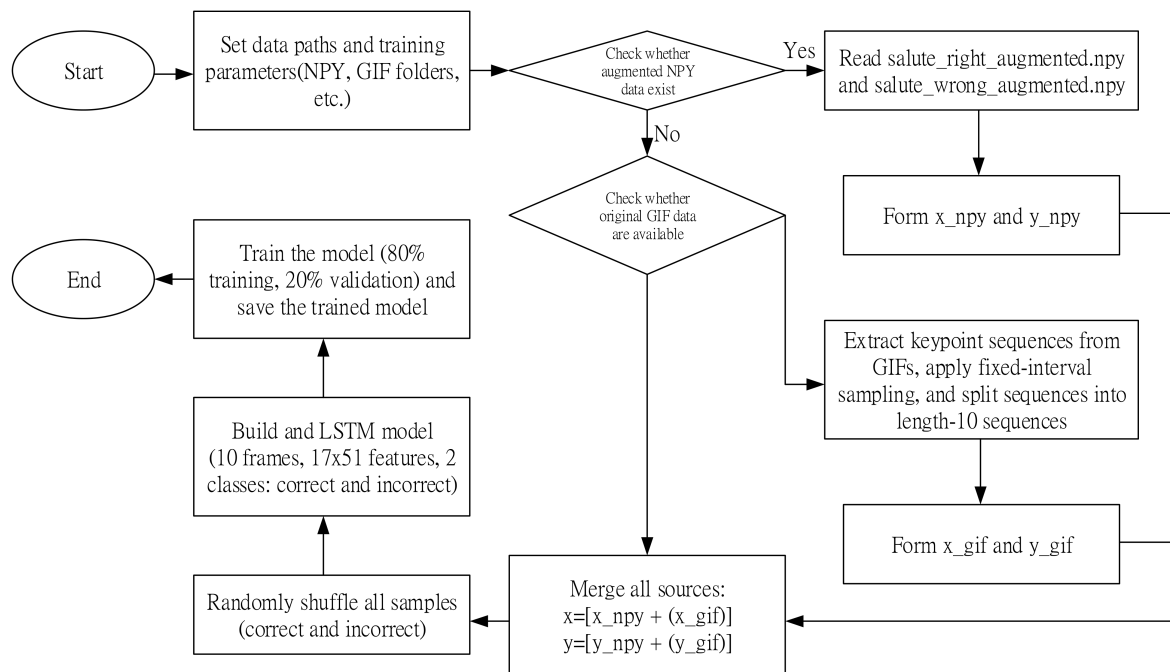


Figure 3. LSTM training flowchart.

When valid GIF files are found, MoveNet is used to extract keypoint sequences, which are then segmented at fixed intervals into 10-frame subsequences. Each subsequence is assigned a correct or incorrect label, producing additional feature matrices and label vectors from the GIF source. The NPY- and GIF-based samples are concatenated along the sample dimension to obtain the final training feature set X and label set Y , increasing sample diversity and enhancing model generalization. All samples are randomly shuffled and split into 80% training and 20% validation data:

$$X = \begin{bmatrix} X_{npy} \\ X_{gif} \end{bmatrix}, \quad Y = \begin{bmatrix} Y_{npy} \\ Y_{gif} \end{bmatrix}, \quad (3)$$

The classification model is an LSTM network that takes 10-frame skeleton sequences as input, with each frame represented by a 51-dimensional feature vector. The model is trained using the cross-entropy loss and the Adam optimizer, and its performance is monitored by training and validation accuracies. After training, the model weights and class name file are saved to a specified directory for subsequent real-time pose recognition.

3.3. Real-Time Recognition

The system defines the salute posture classes, loads the MoveNet 2D pose estimation model and the trained LSTM pose classification model, and allocates a fixed-length buffer for skeletal sequences as the model input. A camera captures live video, and optional recording and logging functions can be enabled, while the user decides after each practice round whether to continue or stop.

For each incoming frame, the image is horizontally flipped and passed to MoveNet to obtain the 2D coordinates and confidence scores of 17 keypoints. These are converted into a 51-dimensional feature vector and appended in temporal order to a buffer of 10 frames [8]. Once the buffer is full, it is treated as a complete skeletal time series and fed into the LSTM model to produce a probability vector over all classes. The final decision at time t is given as follows.

$$C_t^* = \left(\operatorname{argmax}_c \hat{y} \right) \ddot{t}_{ic} \quad (4)$$

Here, $\text{argmax}_c \hat{y}$ selects the class index with the highest predicted probability. If this maximum probability exceeds a predefined threshold, the screen displays the result as OK or NG in different colors, together with the confidence score.

When the LSTM judges the posture as incorrect, a rule-based module analyzes features such as the right upper arm angle, the active hand, whether both arms are raised, and the extent of head motion to determine the specific error type, and overlays the corresponding text message on the video. At the same time, skeletal lines, a visual scale, and angle or position information near key joints are drawn on the screen to indicate which body parts and directions must be adjusted, assisting users in self-practice and posture correction (Figure 4).

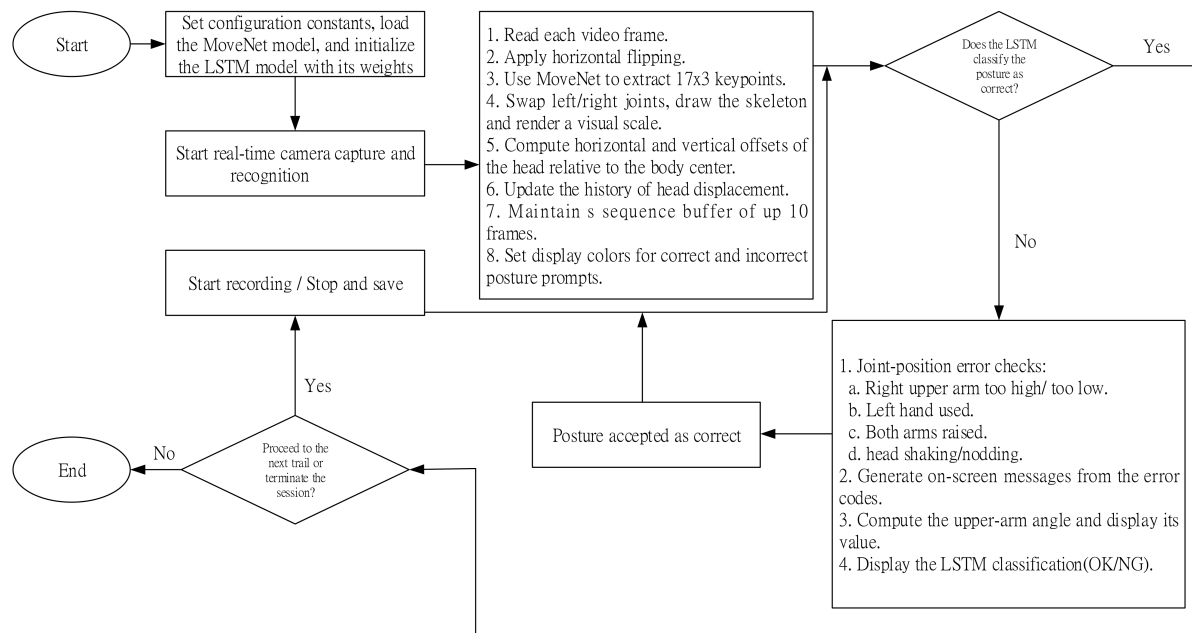


Figure 4. Real-time recognition flowchart of the main program.

4. Result and Discussion

In this study, a military-style salute is used as a demonstration case to evaluate the proposed real-time recognition system. The system runs on a personal computer, and multiple test videos are recorded to observe both the model decisions and the on-screen feedback. The hardware and software environment of the prototype is summarized in Table 1. In front of the camera, the subject repeatedly performs the standard salute posture as well as several deliberately designed incorrect postures, including head shaking, head nodding, excessively high or low right upper-arm angles, using the left hand to salute, and simultaneously raising both arms. The system uses MoveNet to extract the 2D coordinates and confidence scores of 17 body joints and organizes them into 10-frame skeletal sequences as input to the LSTM model. After each sequence is classified, the upper-left corner of the screen displays “salute OK” or “salute NG” together with the corresponding confidence value.

From multiple test videos, it is observed that when the subject maintains a stable and standard salute posture, most segments yield confidence values in the range of approximately 0.95–1.00 and are displayed as “salute OK”, indicating that the model is highly confident in recognizing correct motions. When clearly incorrect postures, such as using the left hand or raising both arms, are performed, the confidence values are usually above 0.9 and are displayed as “salute NG”. These cases are summarized in the “Correct motion (OK)” and “Incorrect motion (NG, high confidence)” rows of Table 2. For borderline cases,

the confidence values fall around 0.5–0.7, typically corresponding to slight angular deviations or mild head motion, as listed in the “Incorrect motion (NG, borderline)” row of Table 2. This result indicates that the model can roughly reflect variations in posture quality near the decision boundary.

Beyond the OK/NG decision, the system also provides textual prompts for multiple error types to help users understand which body parts and directions require adjustment. When the overall posture matches the target salute, the screen overlays a green skeleton and auxiliary lines to indicate joint positions and angles, and messages such as “Good posture, keep it” are shown, as illustrated in Figure 5. When excessive head shaking or nodding is detected, the system displays “Head shaking, keep it still” or “Head nodding, look straight ahead”, as shown in Figure 6. When the left hand or both hands are raised near the head, the prompts “Left arm up, let it hang down” and “Both arms up, keep only right arm up” are displayed, as illustrated in Figure 7. These motion states and representative on-screen texts are summarized in Table 3. The feedback mechanism refines the single OK/NG decision into several intuitive error categories, allowing users to relate the feedback directly to head stability, arm height, and active-hand selection.

Combining the confidence distributions in Table 2 with the error types and feedback examples in Table 3 and Figures 5–7, the proposed MoveNet+LSTM system can distinguish clearly correct and clearly incorrect postures for a single military salute motion in a stable manner. It also provides auxiliary textual hints for borderline postures between the two extremes. These findings demonstrate that integrating 2D pose estimation, temporal classification, and simple rule-based checks can realize a practical real-time posture-checking function under ordinary hardware conditions, supporting posture assessment and self-practice.

Table 2. Description of model decisions and confidence levels.

Status Category	Sample on-Screen Text	Description
Correct motion (OK)	salute OK (confidence: approximately 0.95–1.00)	The LSTM selects the class with the highest output probability as the correct motion. When the confidence value is approximately 0.95 or higher, the model is considered highly confident in this classification.
Incorrect motion (NG, high confidence)	salute NG (confidence: approximately above 0.9)	The LSTM classifies the motion as an incorrect class, with a confidence value generally above 0.9. This indicates that the model can stably recognize clearly incorrect postures that do not meet the standard.
Incorrect motion (NG, borderline)	salute NG (confidence: approximately 0.5–0.7)	The LSTM classifies the motion as an incorrect class, but the confidence value is approximately 0.5–0.7, indicating that the posture is close to the boundary between correct and incorrect. In this situation, the rule-based on-screen prompts can assist the user in making adjustments.

Table 3. Description of system output motion states and error types.

Status Category	Sample On-Screen Text	Description
Correct motion	Good posture, keep it.	The LSTM classifies the motion as correct, and the joint angles and skeleton posture remain within the predefined ranges. The screen displays green skeleton lines and auxiliary lines to indicate the posture.
Head shaking	Head shaking	The horizontal displacement of the head exceeds the predefined threshold, indicating that the head is moving laterally and the posture is unstable.
Head nodding	Head nodding	The vertical displacement of the head exceeds the predefined threshold, indicating excessive up-and-down head movement and an unstable posture.
Right arm too high	Right arm high	The right upper-arm angle exceeds the predefined range. The system prompts the user to lower the right arm so that it returns to the target angle range.
Wrong arm used	Left arm up	The left hand is detected near the head while the right hand is not in the target position. The system identifies the use of the wrong hand and prompts the user to lower the left arm and perform the salute with the right hand.
Both arms raised	Both arms up	Both hands are detected near the head at the same time. The system identifies that both arms are raised and prompts the user to keep only the right arm raised.

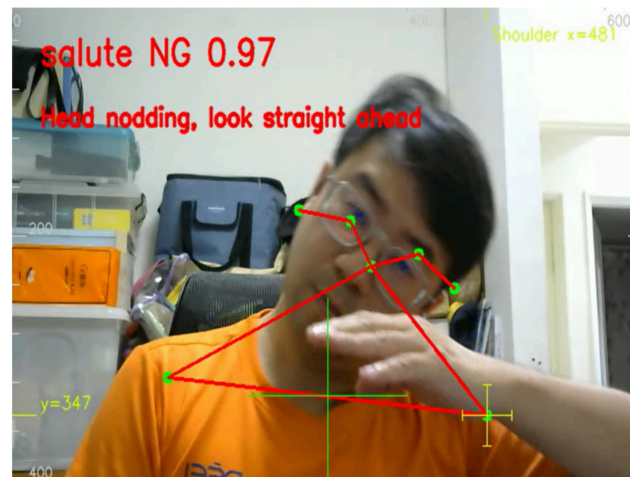


Figure 5. Top-left: Example of a correct salute, with a confidence score of approximately 1.00. The system displays “Good posture, keep it” and provides angle annotations.

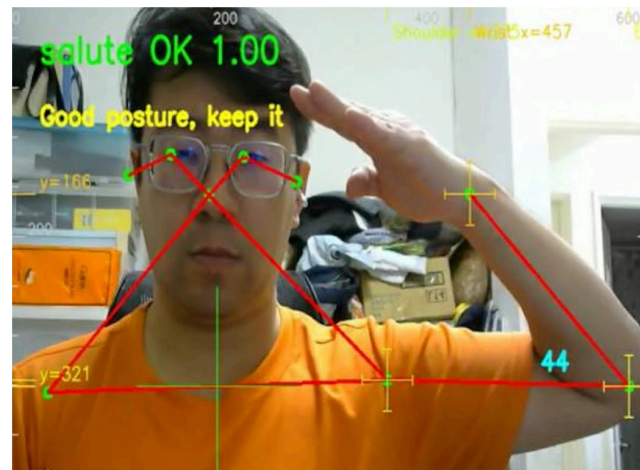


Figure 6. Top-right: Example of a head-nodding error, with a confidence score of approximately 0.97. The system displays “Head nodding, look straight ahead” to indicate unstable head movement.

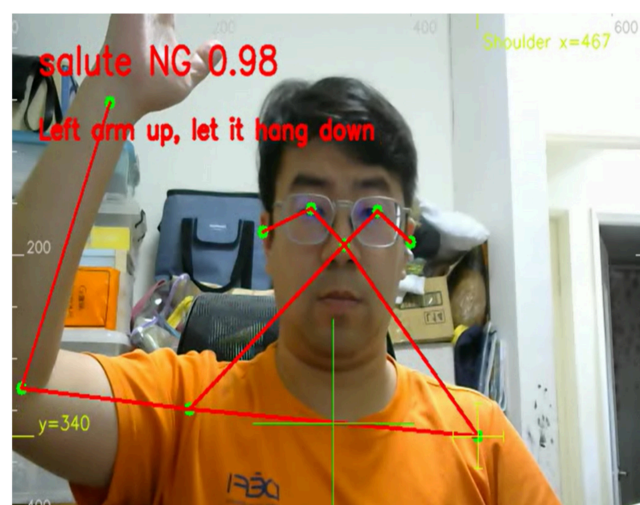


Figure 7. Bottom-left: Example of a wrong-arm error, with a confidence score of approximately 0.98. The system displays “Left arm up, let it hang down” and prompts the user to perform the salute with the right hand.

5. Conclusions

We integrated MoveNet 2D pose estimation with an LSTM network to develop a human motion recognition and real-time posture-checking system, using the military salute as a demonstration case. The system runs on a standard personal computer with a camera and uses temporal skeleton features from 17 body keypoints to classify salute postures as OK or NG in real time. It overlays the detected skeleton, joint-angle information, and textual error messages on the live video to support intuitive posture correction. Under the tested conditions, clearly correct and clearly incorrect postures typically produced confidence scores above 0.9. Future work should expand the dataset and motion categories, establish quantitative evaluation metrics, and explore deployment on mobile or wearable devices to improve the system's robustness and practical applicability.

Author Contributions: Conceptualization, H.-H.K.; methodology, H.-H.K. and L.-C.L.; software, L.-C.L.; validation, L.-C.L. and H.-H.K.; formal analysis, L.-C.L. and H.-H.K.; investigation, L.-C.L.; resources, H.-H.K.; data curation, L.-C.L.; writing—original draft preparation, L.-C.L.; writing—review and editing, H.-H.K.; visualization, L.-C.L.; supervision, H.-H.K.; project administration, H.-H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education Teaching Practice Research Project, Taiwan, grant number MOE-114-TPRSK-0043-011Y1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, H.-Y. Design and Implementation of Remote Control for Drones Based on Human Pose Recognition Technology. Master's Thesis, National Chin-Yi University of Technology, Taichung, Taiwan, 2024.
2. Ali, M. Basketball Jump Shot Prediction Using MediaPipe Pose Estimation and LSTM Networks. Master's Thesis, Kaohsiung Medical University, Kaohsiung, Taiwan, 2024.
3. Zheng, Y.-X. Lightweight OpenPose and MoveNet AI Models for Real-Time Fall Detection and Alert Systems: A Comparative Analysis. Master's Thesis, National Chin-Yi University of Technology, Taichung, Taiwan, 2024.
4. MoveNet: Ultra Fast and Accurate Pose Detection Model. Available online: <https://www.tensorflow.org/hub/tutorials/movenet> (accessed on 14 August 2026).
5. Votel, R.; Li, N. Next-Generation Pose Detection with MoveNet and TensorFlow.js. Available online: <https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html> (accessed on 14 August 2026).
6. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
7. Cabahug, S.M. Badminton Strokes Classification Based on Human Pose Estimation and Machine Learning. Master's Thesis, National Taiwan University of Science and Technology, Taipei, Taiwan, 2022.
8. Human Pose Classification with MoveNet and TensorFlow Lite. Available online: https://www.tensorflow.org/lite/tutorials/pose_classification (accessed on 14 August 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.