Indonesian Journal of Data and Science

Volume 7 Issue 1
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v7i1.391

Research Article

Kodály Hand Sign Recognition from Hand Landmarks Using XGBoost

Achmad Zulfikar [1,*]; Farniwati Fattah [2]; Andi Widya Mufila Gaffar [3]
[1] Universitas Muslim Indonesia, Makassar, Indonesia, 13020220007@student.umi.ac.id
[2] Universitas Muslim Indonesia, Makassar, Indonesia, farniwati.fattah@umi.ac.id
[3] Universitas Muslim Indonesia, Makassar, Indonesia, widya.mufila@umi.ac.id

Correspondence should be addressed to Achmad Zulfikar; 13020220007@student.umi.ac.id

Received 25 November 2025; Accepted 17 Jan 2026; Published 30 March 2026

Abstract:
Introduction: Angklung is a traditional Indonesian musical instrument that continues to evolve through digital technology. However, computer vision–based gesture recognition for controlling physical angklung instruments remains limited. This study investigates landmark-based recognition of Kodály hand signs and evaluates its application for real-time angklung interaction. Method: Hand landmarks were extracted using MediaPipe Hands from RGB camera input. Each gesture was represented by 63 normalized numerical features derived from 21 landmarks. The dataset consists of 8,000 images representing eight Kodály gesture classes (Do–Do'). Gesture classification was performed using the Extreme Gradient Boosting (XGBoost) algorithm. Model evaluation applied a subject-independent two-fold scheme using accuracy, precision, recall, F1-score, and confusion matrix analysis. Real-time system trials were conducted under different lighting conditions and capture distances, and TCP communication with an ESP32 controller was evaluated. Results: The model achieved 96.63% accuracy in Fold 1 and 96.40% in Fold 2. Misclassifications were mainly observed between visually similar gestures, particularly La and Mi. Separate real-time system trials showed consistent recognition under bright lighting, while accuracy decreased under dim lighting, especially for Do (90%) and Mi (86.7%). Gesture recognition remained reliable up to approximately 1.5 m. TCP testing over 200 command events recorded 0% failed acknowledgments with a mean round-trip time of 87.36 ms. Conclusion: These indicate that landmark-based Kodály gesture classification using MediaPipe Hands and XGBoost can support real-time angklung interaction under controlled conditions, although improvements are needed for low-light environments and visually similar gestures.

Keywords: *Angklung*, Gesture Recognition, XGBoost, MediaPipe Hands, Kodály Hand Sign.

1. Introduction:

Angklung, a traditional Indonesian musical instrument, continues to be modernized, driven by developments in embedded systems technology and the Internet of Things (IoT). In the early stage, Arduino-based microcontrollers were used to implement programmed actuator control for angklung automation [1], [2]. As the need for wireless connectivity continues to increase, research has evolved to integrate the ESP32, which supports Wi-Fi communication for remote control [3], [4]. The Internet of Musical Things (IoMusT) concept expands the integration of traditional instruments with sensors, actuators, and network connectivity [5]. Advances in computer vision and machine learning indicate the potential for real-time control of musical instruments via hand gestures [6], [7], [8], where visual decisions can be translated into control commands to actuate physical devices [9].

Advances in computer vision enable musical instrument control through visual interaction, including real-time hand gesture recognition [10]. Previous work developed a Kodály gesture recognition system, in which Kodály hand signs represent musical scales through specific hand poses [11], [12], using MediaPipe Holistic to extract hand-and-arm landmarks and a Multi-Layer Perceptron classifier, achieving 99% accuracy. However, the system generated only audio output and did not interact with a physical musical instrument.

The gap between gesture recognition and physical actuation in prior studies motivates the development of a system that bridges gesture classification with direct angklung control. This study employs MediaPipe Hands, an open-source, machine-learning-based framework from Google that tracks hands by detecting 21 landmark points in real time from RGB camera input [8], in contrast to prior work that used MediaPipe Holistic. For Kodály gesture classification, Extreme Gradient Boosting (XGBoost) is selected for its effectiveness in handling numerical features, modeling nonlinear relationships among landmarks, and using regularization to mitigate overfitting. XGBoost is an ensemble method based on gradient-boosted decision trees that builds models iteratively by correcting previous prediction errors [13]. In comparative studies on image-based hand gesture recognition, XGBoost has been shown to achieve higher accuracy than alternative methods such as Random Forest, Decision Tree, and SVM [14], [15].

This study evaluates the effectiveness of MediaPipe Hands landmarks in discriminating Kodály gestures and assesses the performance of the XGBoost classifier in an eight-class multiclass classification task corresponding to musical notes. In addition, the study examines the reliability of TCP network communication for real-time control of the angklung system using an ESP32 controller. Accordingly, this study addresses the following research questions: (1) how effectively landmark features extracted by MediaPipe Hands can represent Kodály gestures, (2) how accurately the XGBoost classifier can classify eight Kodály hand-sign gestures in a multiclass recognition task, and (3) how lighting and capture distance influence the reliability of real-time gesture-based instrument control.

This study makes two main contributions. First, it presents an empirical evaluation of MediaPipe Hands landmark features combined with the XGBoost classifier for multiclass Kodály gesture recognition. Second, it evaluates the reliability of TCP-based communication between the gesture recognition module and an ESP32 controller for real-time angklung actuation, demonstrating the applicability of the proposed approach in a physical musical instrument system.

Method:

Research Design

This study methodology was designed in stages to develop a physical angklung control system based on hand gesture recognition. The stages included creating a Kodály hand-sign gesture dataset, training and evaluating the classification model, integrating the model into the physical angklung system, and testing the system's performance. The Research methodology flow is shown in **Figure 1**.

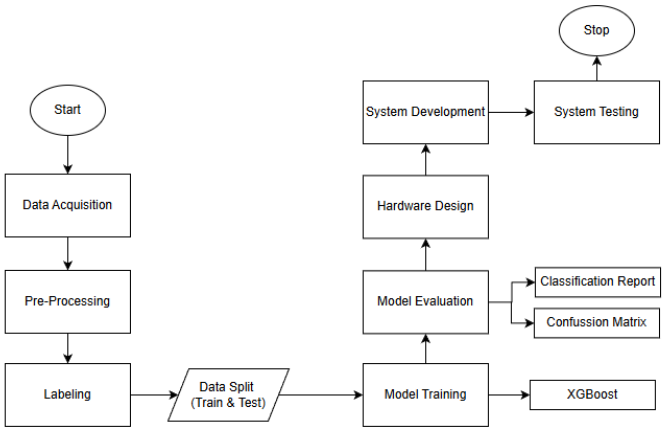

Figure 1. Research Process Diagram

A. Data Acquisition

Data acquisition was performed using an external camera with a resolution of 1920×1080 pixels. Recording was controlled by a Python script that utilized OpenCV for real-time image acquisition and MediaPipe Hands for hand detection and extraction of 21 landmarks. Capture was performed at a rate of two images per second for each Kodály gesture. The total dataset consists of 8000 images divided evenly into eight pitch classes (do–do'), corresponding to the Kodály hand sign poses shown in **Figure 2**. This ensures class balance and sufficient sample diversity, improving

model generalization. Data acquisition was performed in two sessions under controlled lighting conditions with two illumination levels: dim (<25 lux) and bright (≥25 lux), as shown in **Figure 3**. Data were recorded at three capture distances (0.5 m, 1 m, and 1.5 m), with the maximum distance limited to 1.5 m to ensure reliable motion recognition [11].

Figure 2. Handsign Kodály Gesture

B. Pre-processing

The acquired image is first converted from BGR color space to RGB to match the MediaPipe Hands pipeline input format. Next, the image is processed with MediaPipe's BlazeHand model, which applies palm detection and hand-landmark regression to detect hands and extract 21 anatomical landmark points. Each landmark has x, y, and z coordinates in a normalized coordinate system.

To reduce variability due to changes in camera distance, hand size, and shooting angle, all landmark coordinates are normalized against the bounding box of the hand generated by the detector [16]. This normalization produces 63 scale-invariant numerical features that consistently represent the finger pose configuration for each gesture.

Images that do not meet processing quality standards, such as failed hand detection, incomplete landmarks, or gesture configurations that do not match the class, are filtered and removed from the dataset to ensure that only valid data with complete landmark structures are used for model training.

Figure 3. Collecting Dataset "Do" from two subjects under six conditions (bright and dim lighting at 0.5 m, 1 m, and 1.5 m)

C. Labeling

Each sample is automatically labeled through an internal script mapping where keys 1–8 correspond to Kodály gestures (Do–Do'). The dataset contains 63 landmark features extracted from 21 MediaPipe hand landmarks, along

with a subject identifier and a label_encoded column representing numerical class indices for model training, as shown in **Figure 4**.

	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	...	f56	f57	f58	f59	f60	f61	f62	label	subject	label_encoded
0	0.000000	0.518535	-1.505119e-09	0.297393	0.575051	-0.000317	0.088860	0.662429	-0.000548	0.069182	...	-0.000682	0.839350	0.640194	-0.000660	0.803509	0.432580	-0.000617	Do	S1	0
1	0.563373	0.542934	-1.492573e-09	0.328227	0.621401	-0.000335	0.101221	0.719453	-0.000586	0.068303	...	-0.000700	0.860440	0.667460	-0.000676	0.823957	0.465514	-0.000636	Do	S1	0
2	0.561200	0.572871	-2.149227e-09	0.322149	0.632107	-0.000329	0.097060	0.705161	-0.000571	0.070902	...	-0.000716	0.812985	0.745942	-0.000689	0.756857	0.586500	-0.000648	Do	S1	0
3	0.572336	0.555096	-1.654448e-09	0.341926	0.627270	-0.000324	0.114808	0.724525	-0.000561	0.069783	...	-0.000662	0.824011	0.732167	-0.000626	0.783321	0.541138	-0.000573	Do	S1	0
4	0.611516	0.707659	-2.321237e-09	0.372128	0.731321	-0.000242	0.142307	0.771927	-0.000440	0.081062	...	-0.000631	0.855316	0.773358	-0.000582	0.823901	0.592588	-0.000521	Do	S1	0

Figure 4. Example of preprocessing feature data from subject 1 (S1)

D. Model Training

Model training was conducted to classify Kodály gestures using hand landmark features from MediaPipe. The Extreme Gradient Boosting (XGBoost) algorithm was chosen because it builds models incrementally by adding decision trees that correct previous prediction errors with strong regularization to prevent overfitting [17], [15].

The dataset was processed by separating features and labels, followed by label encoding for multi-class representation. A 50:50 subject-independent split was applied, where data from one subject were used for training, and the other subject for testing, and the roles were reversed in the second fold to evaluate cross-subject generalization. All features were then normalized using StandardScaler to ensure numerical consistency among hand landmark coordinates [18], [19].

The XGBoost model is configured using several core parameters, namely the number of decision trees (n_estimators) to control model complexity, the maximum tree depth (max_depth) to balance representation and generalization capabilities, and the learning rate to maintain the stability of the learning process [17], [20], [21]. In addition, the subsample parameter improves model generalization by training each tree on a subset of the training data. The multi:softprob objective function is applied to generate probabilities for each gesture class, enabling accurate mapping of gestures to angklung tones [15].

E. Model Evaluation

XGBoost performance was evaluated using accuracy, precision, recall, F1-score, and the confusion matrix to assess the model's performance in classifying gestures, following standard practices for multi-class classification. [22], [23], [24], [25]. A 2-fold subject-independent scheme was applied, where data from one subject were used for training, and the other for testing, and the roles were reversed in the second fold. The model with the highest testing accuracy was selected as the final model and serialized together with the scaler and label encoder for real-time inference without retraining. [17]. This stage ensures an accurate model that is ready to be integrated with the angklung actuation system.

F. Hardware Design

The electronic circuit was designed in Fritzing, as shown in **Figure 5**, which depicts the circuit configuration.

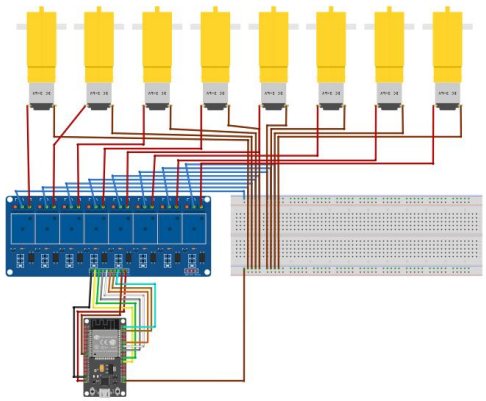

Figure 5. Hardware Design

Several components of the electronic design in this study can be described as follows:

1. Eight DC motors function as mechanical actuators that move the striking rods to sound each angklung tube according to the notes commanded by the system.
2. The relay module, consisting of 8 channels, as an electronic switch that connects and disconnects the current to the DC motor, allowing the system to control when the motor is activated or stopped.
3. ESP32 as a control center that receives tone commands from the Machine Learning-based gesture detection system, then forwards the control signal to the relay module to produce angklung sounds in real time.

G. System Development

The system development in this study comprises the development environment, hardware, software, and an end-to-end workflow for controlling physical angklung in real time. The system is implemented in Python (Visual Studio Code) to capture camera images, detect hands, extract landmarks, normalize and standardize features, and perform multi-class XGBoost inference to generate tone labels and their confidence values. The prediction results are then converted into control signals and sent to the ESP32 over a TCP/IP Wi-Fi connection (Python as the client, ESP32 as the server). The microcontroller then receives the commands via the network and executes the angklung actuator control in real-time [26], [27], [28]. The architecture is divided into a gesture-recognition module and a communication-actuation module to ensure greater stability and lower latency.

H. System Testing

System testing was conducted to evaluate the performance of the physical angklung in real-time conditions. The system consists of eight bamboo tubes mounted on a wooden support structure with integrated webcams for gesture detection. System evaluation included testing the entire system under both bright and dim lighting conditions, software testing, hardware testing, and TCP network testing, which comprehensively validated the system's functionality and reliability. In addition, the system was tested at several user-to-camera distances (0.5 m, 1 m, and 1.5 m) to assess the robustness of gesture recognition under different interaction ranges. The entire angklung instrument system used in this study is shown in **Figure 6**.

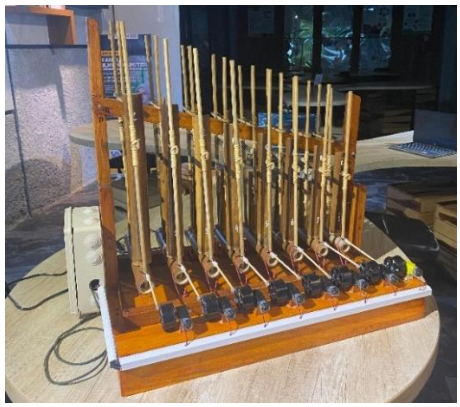

Figure 6. Physical Angklung System

Model testing directly integrated with the angklung system is conducted using a real-time gesture-recognition scheme on a physical angklung, where hand gestures are processed to directly control the angklung's actuation. Each scale is tested repeatedly under randomized sequences and lighting conditions, and predictions are made only when the confidence exceeds the 60% threshold. This scheme ensures that the test results represent the model's performance in real-time conditions.

TCP network testing was conducted to evaluate the reliability and performance of data communication between a Python program acting as the client and an ESP32 as the server in the physical angklung control system. The test was conducted by sending 200 tone commands sequentially via the Wi-Fi network and recording the data transmission and reception times [29]. The network performance parameters measured include the average round-trip time (RTT) value, standard deviation, and 95th percentile as indicators of data communication stability and reliability during testing [30], [31].

3. Result and Discussion

Model Evaluation:

The gesture classification model was evaluated using a subject-independent two-fold scheme to assess its ability to generalize to unseen users. The dataset collected from two participants was divided into two folds. In Fold 1, the model was trained on data from subject S1 and tested on subject S2, while in Fold 2, the configuration was reversed.

Each gesture sample was represented as a feature vector derived from normalized hand landmark coordinates extracted using MediaPipe. The classifier was implemented using XGBoost configured for eight-class gesture recognition (Do, Do', Fa, La, Mi, Re, Si, and So) with parameters n_estimators = 500, max_depth = 6, and learning_rate = 0.05 using the multi:softprob objective function.

Model evaluation was performed on test data using the accuracy, precision, recall, and F1-score metrics, supported by a confusion matrix to show the correspondence between actual labels and predicted results for each class.

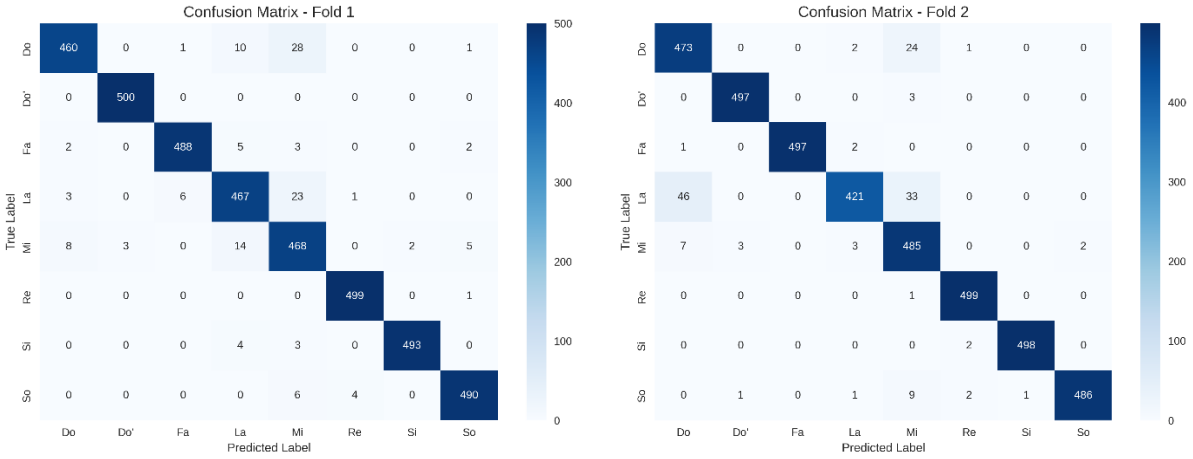

Figure 7. Confusion Matrix of Fold 1 (Train: S1, Test: S2) and Fold 2 (Train: S2, Test: S1)

The confusion matrix shows that most predictions are concentrated along the main diagonal, indicating high classification accuracy across almost all gesture classes. In Fold 1, a small number of misclassifications mainly occur between the La–Mi classes. Meanwhile, in Fold 2, several errors appear between the Do–La and La–Mi classes. Despite these minor errors, the majority of gesture samples in both folds are correctly classified.

Table 1. Classification Report Fold 1

Label	precision	recall	F1-score	support
Do	0.9725	0.9200	0.9455	500
Do'	0.9940	1.0000	0.9970	500
Fa	0.9859	0.9760	0.9809	500
La	0.9340	0.9340	0.9340	500
Mi	0.8814	0.9360	0.9079	500
Re	0.9901	0.9980	0.9940	500
Si	0.9960	0.9860	0.9910	500
So	0.9820	0.9800	0.9810	500
Accuracy	-	-	0.9663	4000
Macro avg	0.9670	0.9663	0.9664	4000
Weighted avg	0.9670	0.9663	0.9664	4000

Table 1 shows the classification report for Fold 1. The model achieved an accuracy of 96.63%, with macro- and weighted-average precision, recall, and F1-score values around 96%. Most gesture classes obtain precision and recall values above 97%, while the La class records 93.40% for precision, recall, and F1-score. The Mi class shows

comparatively lower precision (88.14%) and an F1-score of 90.79%, indicating that more samples from this class were misclassified than from the other classes.

Table 2. Classification Report Fold 2

Label	precision	recall	F1-score	support
Do	0.8975	0.9460	0.9211	500
Do’	0.9920	0.9940	0.9930	500
Fa	1.0000	0.9940	0.9970	500
La	0.9814	0.8420	0.9064	500
Mi	0.8739	0.9700	0.9194	500
Re	0.9901	0.9980	0.9940	500
Si	0.9980	0.9960	0.9970	500
So	0.9959	0.9720	0.9838	500
Accuracy	-	-	0.9640	4000
Macro avg	0.9661	0.9640	0.9640	4000
Weighted avg	0.9661	0.9640	0.9640	4000

Table 2 shows the classification report for Fold 2. The model achieved an accuracy of 96.40%, slightly lower than fold 1. Most motion classes show classification scores above 96%, while the La and Mi classes record relatively lower values compared to the other classes.

Based on the evaluation results, Fold 1 was selected as the reference model because it achieved slightly higher accuracy (96.63%) than Fold 2 (96.40%). Although the difference is relatively small, Fold 1 shows marginally better classification performance and is therefore used as the representative model configuration in this study.

System Evaluation:

System testing was conducted using a randomized gesture-testing scheme under two lighting conditions, with image capture distances varying up to 1.5 m. Each gesture class was tested in 30 trials in total: 15 under bright lighting and 15 under low-light conditions. Predictions were recorded only when the confidence score exceeded 60%. A summary of the test results is presented in Table 3.

Table 3. Performance of Real-Time Gesture Recognition under Different Lighting Conditions

Gesture	Lighting Condition	Total Trials	Correct Prediction	Accuracy (%)
Do	Bright Lighting	30	30	100%
Do	Dim Lighting	30	27	90%
Do'	Bright Lighting	30	30	100%
Do'	Dim Lighting	30	30	100%
Fa	Bright Lighting	30	30	100%
Fa	Dim Lighting	30	30	100%
La	Bright Lighting	30	30	100%
La	Dim Lighting	30	28	93.3%
Mi	Bright Lighting	30	30	100%
Mi	Dim Lighting	30	26	86.7%
Re	Bright Lighting	30	30	100%
Re	Dim Lighting	30	30	100%
Si	Bright Lighting	30	30	100%
Si	Dim Lighting	30	30	100%
So	Bright Lighting	30	30	100%

Gesture	Lighting Condition	Total Trials	Correct Prediction	Accuracy (%)
So	Dim Lighting	30	29	96.7%

Based on the results presented in **Table 3**, the gesture recognition system achieved perfect performance under bright lighting conditions. All gestures were correctly classified, yielding 100% accuracy across 30 trials for each gesture. This result indicates that the proposed model is highly reliable for gesture recognition when operating under optimal lighting conditions.

Figure 8. (A) "La" testing at 0.5-meter range and dim lighting, (B) "La" testing at 1-meter range and dim lighting, (C) "La" testing at 1.5-meter range and dim lighting

Next, testing focused on dim lighting conditions, as the bright lighting scenario had already achieved 100% accuracy. This evaluation was conducted to assess the system's performance under more challenging lighting conditions. As shown in **Figure 8**, the model was able to classify gestures under dim lighting conditions at distances of up to 1.5 meters.

Figure 9. (D) gesture "Do" incorrectly predicted as "Mi", and (E) gesture "Mi" incorrectly predicted as "La"

However, at a distance of 1.5 meters under dim lighting conditions, the model exhibited several misclassifications, as shown in **Figure 9**. In some cases, gestures expected to be recognized as "Do" or "Mi" were incorrectly predicted as "Mi" and "La" This misclassification may be influenced by the similarity in the hand-back curvature patterns among the gestures "Do," "Mi," and "La." In addition, reduced illumination in dim lighting conditions may affect the clarity of visual features captured by the camera, thereby influencing the model's ability to distinguish subtle differences between similar gesture shapes.

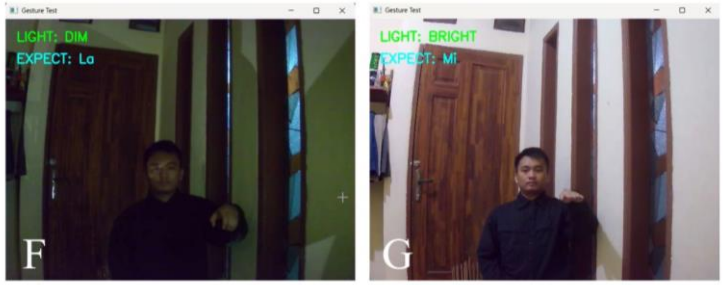

Figure 10. Gesture recognition testing at a distance of 2.5 meters: (F) dim lighting condition and (G) bright lighting condition

Next, to evaluate the effect of distance on the model's performance, additional testing was conducted at a distance of 2 meters. The experiment was performed under both bright and dim lighting conditions. Based on the results shown in **Figure 10**, the model was unable to correctly recognize the hand gestures at this distance. This observation suggests that greater distance may impair the system's ability to detect and classify hand gestures.

Software Testing Result

Software testing in this study was conducted to ensure that the hand-gesture processing system operated reliably and in accordance with the specified design. As shown in **Table 4**, all tested software functions performed as expected. These results confirm the software's readiness for integration with the physical angklung actuator.

Table 4. Results of Software Testing

No.	Condition	Expected Response	Observed Response	Output Voltage	Remarks
1	Camera And Application Are Running	The System Detects The Hand In The Camera Frame	The Hand is Detected in The Camera Frame	–	Successful
2	Performing Kodály Gesture	The Model Captures Hand Gesture Features	The Model Correctly Recognizes The Hand Gesture	–	Successful
3	Confidence Below The Threshold	The System Does Not Send a Note Command	No Command is Transmitted	–	Successful
4	Hand Leaves The Camera View	The System Does Not Perform Prediction	No Actuation is Triggered	–	Successful
5	A Valid Prediction Is Sent To The ESP32	Note Data Are Transmitted Via TCP Over Wi-Fi	Data Are Transmitted Via TCP Over Wi-Fi	–	Successful

Hardware Testing Result

Hardware testing in this study aims to ensure that all physical angklung actuation components function properly and respond accurately to control commands. As shown in **Table 5**, all hardware components, including the ESP32 module, relay system, DC motor, and angklung actuator, respond as expected. This confirms that the hardware system is functioning correctly and is ready to support the overall system operation.

Table 5. Results of Hardware Testing

No.	Condition	Expected Response	Observed Response	Output Voltage	Remarks
1	ESP32 Is Powered On And Connected To Wi-Fi	ESP32 Is Ready To Receive Data	ESP32 Connected And In Standby Mode	3.3 V	Successful
2	Note Data Are Received By The ESP32	The Relay Corresponding To The Note Is Activated	The Relay Is Activated According To The Received Note	3.3 V	Successful
3	Relay Is Activated	DC Motor Rotates	DC Motor Rotates	12 V	Successful
4	DC Motor Rotates	Angklung Produces Sound	The Angklung Produces The Commanded Note	5 V	Successful
5	All Channels Are Tested	All Notes Do To Do' Function Properly	All Channels Operate Normally	5 V	Successful

Connection Testing Result

TCP network testing was performed to evaluate the communication reliability between the Python client and the ESP32 server by transmitting 200 command events and measuring the round-trip time (RTT). The experiment was conducted over a local 2.4 GHz Wi-Fi network within a ±2-meter range without additional traffic to minimize interference.

Table 6. TCP Communication Testing Results

TCP Testing Metrics	
Total Event	200
Successful ACK (%)	100%
Failed ACK (%)	0%
Mean RTT	87.36 ms
Std RTT	35.68 ms
P95 RTT	126.05 ms
P99 RTT	187.18 ms
Max RTT	209.00 ms

As shown in **Table 6**, the results indicate stable TCP communication with a 100% ACK success rate and low latency, demonstrating reliable data exchange between the Python client and the ESP32 server. The table summarizes network performance parameters, including average round-trip time (RTT), standard deviation, and the 95th percentile, as indicators of data communication stability and reliability during testing.

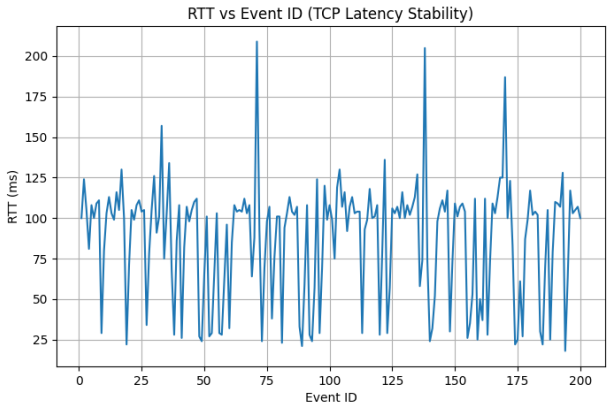

Figure 11. RTT Function of Event ID During TCP Communication Testing of The Angklung System

Based on **Table 6**, as shown in **Figure 11**, the RTT vs. event ID plot indicates that most RTT values fall within the range of tens to around one hundred milliseconds, with an average of approximately 87.36 ms. Although several latency spikes occur in some events, these conditions are temporary and do not affect overall communication stability. This demonstrates that the TCP network used provides sufficient reliability to support the physical angklung control system.

The experimental results address the research questions by evaluating the effectiveness of MediaPipe Hands landmarks and the XGBoost classifier for Kodály gesture recognition. Using 63 normalized landmark features, the model achieved an accuracy of 96.63% in Fold 1 and 96.40% in Fold 2. The confusion matrix shows that most predictions fall along the main diagonal, indicating that the landmark representation is generally effective for distinguishing Kodály gestures. However, several misclassifications were observed between visually similar gestures, particularly the La and Mi classes, suggesting that similar hand configurations can produce overlapping landmark patterns that reduce class separability.

Real-time testing further shows that environmental conditions affect the reliability of recognition. Under bright lighting conditions, all gestures were correctly recognized during the trials, whereas under dim lighting conditions, accuracy decreased for several gestures, particularly Do (90%) and Mi (86.6%). These results indicate that reduced illumination affects the clarity of visual features captured by the camera and can make it more difficult for the model

to distinguish subtle differences between gestures. Distance evaluation also shows that gestures can be reliably detected up to approximately 1.5 meters, beyond which recognition performance degrades.

From the system perspective, TCP communication between the gesture recognition module and the ESP32 controller successfully transmitted all command events during testing. The measured mean round-trip time was 87.36 ms with no failed acknowledgments, indicating that the network latency is sufficient to support real-time triggering of angklung tones. Overall, the findings demonstrate that landmark-based gesture representation combined with XGBoost can support real-time gesture recognition for physical angklung interaction under controlled conditions, although improvements in low-light robustness and discrimination of similar gestures remain important directions for future work.

Conclusion:

This study investigated the feasibility of controlling a physical angklung instrument using Kodály hand-sign recognition based on MediaPipe Hands landmarks and an XGBoost multiclass classifier. Using 63 normalized landmark features, the model achieved an accuracy of 96.63% in Fold 1 and 96.40% in Fold 2, with F1-scores around 0.96 across eight gesture classes. Most predictions were correctly classified, although several errors occurred between visually similar gestures, particularly La and Mi, indicating overlapping landmark patterns.

Real-time testing shows that environmental conditions affect the reliability of recognition. Under bright lighting, all gestures were correctly recognized during the trials, whereas under dim lighting, accuracy decreased for several gestures, particularly Do (85%) and Mi (86.6%). Gesture recognition remained reliable up to approximately 1.5 m, while performance degraded at greater distances.

Communication between the recognition system and the ESP32 controller was error-free across 200 command events, with a mean TCP round-trip time of 87.36 ms. Overall, the results indicate that landmark-based multiclass recognition can support real-time angklung actuation under controlled conditions. However, improvements in low-light robustness, discrimination between visually similar gestures, and broader dataset diversity remain important directions for future work.

References:

[1] A. R. Wicaksono, J. Subur, and M. Taufiqurrohman, “Design and Development of an Automatic Angklung Robot Based on Microcontroller,” *JEEE-U (Journal of Electrical and Electronic Engineering-UMSIDA)*, vol. 7, no. 2, pp. 107–128, Oct. 2023, doi: 10.21070/jeeeu.v7i2.1669.

[2] I. Nur Hanafi, S. Supriyono, and H. Susanti, “Rancang Bangun Angklung Elektrik dengan Mode Otomatis dan Manual Menggunakan Teknologi Mikrokontroler dan Smartphone,” *Infotekmesin*, vol. 16, no. 1, pp. 113–119, Jan. 2025, doi: 10.35970/infotekmesin.v16i1.2558.

[3] E. Murpratama, U. Sunarya, and A. Novianti, “ANGKLUNG ROBOT CONTROL SYSTEM BASED ON MICROCONTROLLER,” *Jurnal Elektro dan Telekomunikasi Terapan*, vol. 6, no. 1, p. 734, Jan. 2020, doi: 10.25124/jett.v6i1.1876.

[4] R. P. Wardana, E. M. Budi, and A. S. Sudarsono, “Development of a Wireless Distributed Real-Time Angklung Robot System,” in *2023 8th International Conference on Instrumentation, Control, and Automation (ICA)*, IEEE, Aug. 2023, pp. 253–257. doi: 10.1109/ICA58538.2023.10273119.

[5] L. Turchet and P. Casari, “Latency and Reliability Analysis of a 5G-Enabled Internet of Musical Things System,” *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1228–1240, Jan. 2024, doi: 10.1109/JIOT.2023.3288818.

[6] C. Cui, M. S. Sunar, and G. Eg Su, “Deep vision-based real-time hand gesture recognition: a review,” *PeerJ Comput. Sci.*, vol. 11, p. e2921, Jun. 2025, doi: 10.7717/peerj-cs.2921.

[7] H.-H. Li and C.-C. Hsieh, “Dynamic Hand Gesture Recognition Using MediaPipe and Transformer,” in *IEEE ICEIB 2025*, Basel Switzerland: MDPI, Sep. 2025, p. 22. doi: 10.3390/engproc2025108022.

[8] Y. Meng, H. Jiang, N. Duan, and H. Wen, “Real-Time Hand Gesture Monitoring Model Based on MediaPipe’s Registerable System,” *Sensors*, vol. 24, no. 19, p. 6262, Sep. 2024, doi: 10.3390/s24196262.

[9] R. S. Anwar, T. Hasanuddin, and S. M. Abdullah, "Sistem Keamanan Pintu Asrama Berbasis Pengenalan Wajah dengan Algoritma Haar Cascade," *Buletin Sistem Informasi dan Teknologi Islam*, vol. 3, no. 3, pp. 213–218, Aug. 2022, doi: 10.33096/busiti.v3i3.1197.

[10] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *J. Imaging*, vol. 6, no. 8, p. 73, Jul. 2020, doi: 10.3390/jimaging6080073.

[11] M. Z. Fauzi and R. Sarno, "Recognition of Real-Time Angklung Kodály Hand Gesture using Mediapipe and Machine Learning Method," in *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 980–985. doi: 10.1109/ICCoSITE57641.2023.10127808.

[12] A. Gülle, C. Akay, and N. B. Uzun, "Zoltán Kodály gives a hand to secondary school students in recorder performance and attitudes toward music in Turkey," *International Journal of Music Education*, vol. 39, no. 4, pp. 477–491, Nov. 2021, doi: 10.1177/02557614211005904.

[13] M. Noorunnahar, A. H. Chowdhury, and F. A. Mila, "A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh," *PLoS One*, vol. 18, no. 3, p. e0283452, Mar. 2023, doi: 10.1371/journal.pone.0283452.

[14] M. Imani, A. Beikmohammadi, and H. R. Arabnia, "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels," *Technologies (Basel).*, vol. 13, no. 3, Mar. 2025, doi: 10.3390/technologies13030088.

[15] N. Andriyanov and S. Mikhailova, "Improving Gesture Recognition Efficiency with MediaPipe and YOLO-Pose," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-2/W9-2025, pp. 13–18, Sep. 2025, doi: 10.5194/isprs-archives-XLVIII-2-W9-2025-13-2025.

[16] T. L. Dang, S. D. Tran, T. H. Nguyen, S. Kim, and N. Monet, "An improved hand gesture recognition system using keypoints and hand bounding boxes," *Array*, vol. 16, Dec. 2022, doi: 10.1016/j.array.2022.100251.

[17] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[18] S. Bhushan, M. Alshehri, I. Keshta, A. K. Chakraverti, J. Rajpurohit, and A. Abugabah, "An Experimental Analysis of Various Machine Learning Algorithms for Hand Gesture Recognition," *Electronics (Basel).*, vol. 11, no. 6, p. 968, Mar. 2022, doi: 10.3390/electronics11060968.

[19] N. Zheng, Y. Li, W. Zhang, and M. Du, "User-Independent EMG Gesture Recognition Method Based on Adaptive Learning," *Front. Neurosci.*, vol. 16, Mar. 2022, doi: 10.3389/fnins.2022.847180.

[20] P. Zhang, Y. Jia, and Y. Shang, "Research and application of XGBoost in imbalanced data," *Int. J. Distrib. Sens. Netw.*, vol. 18, no. 6, p. 155013292211069, Jun. 2022, doi: 10.1177/15501329221106935.

[21] P. Sarker, J.-J. Tiang, and A.-A. Nahid, "Metaheuristic-Driven Feature Selection for Human Activity Recognition on KU-HAR Dataset Using XGBoost Classifier," *Sensors*, vol. 25, no. 17, p. 5303, Aug. 2025, doi: 10.3390/s25175303.

[22] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens. (Basel).*, vol. 16, no. 3, p. 533, Jan. 2024, doi: 10.3390/rs16030533.

[23] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 6086, Mar. 2024, doi: 10.1038/s41598-024-56706-x.

[24] G. Zeng, "Invariance Properties and Evaluation Metrics Derived from the Confusion Matrix in Multiclass Classification," *Mathematics*, vol. 13, no. 16, p. 2609, Aug. 2025, doi: 10.3390/math13162609.

[25] H. Azis, P. Purnawansyah, and N. Alfiyyah, "Multiclass Classification on Nominal Value of Rupiah Banknotes Based on Image Processing," *ILKOM Jurnal Ilmiah*, vol. 16, no. 1, pp. 87–99, Apr. 2024, doi: 10.33096/ilkom.v16i1.1784.87-99.

[26] M. A. Mubarak, F. Fattah, and H. Azis, "Prototipe Smart Home Berbasis ESP32 dengan Fitur Keamanan pintu, Lampu, dan AC Otomatis Berbasis IoT," *LINIER: Literatur Informatika dan Komputer*, vol. 2, no. 3, pp. 421–437, Oct. 2025, doi: 10.33096/linier.v2i3.3152.

[27] A. Muh. F. Dzikrulkhair, R. Satra, and A. W. Mufila Gafar, "Rancang Bangun Jemuran Pintar Otomatis Berbasis Internet of things (Iot)," *LINIER: Literatur Informatika dan Komputer*, vol. 2, no. 3, pp. 438–446, Oct. 2025, doi: 10.33096/linier.v2i3.3153.

[28] Y. A. Yunus, R. Satra, and F. Fattah, "Sistem Monitoring Suhu Pada Inkubator Penetas Telur Berbasis IoT," *LINIER: Literatur Informatika dan Komputer*, vol. 1, no. 4, pp. 389–394, Dec. 2024, doi: 10.33096/linier.v1i4.2538.

[29] M. Hlayel, H. Mahdin, and H. A. Mohd Adam, "Latency Analysis of WebSocket and Industrial Protocols in Real-Time Digital Twin Integration," *International Journal of Engineering Trends and Technology*, vol. 73, no. 1, pp. 120–135, Jan. 2025, doi: 10.14445/22315381/IJETT-V73I1P110.

[30] B. Amirkhanov, G. Amirkhanova, M. Kunelbayev, S. Adilzhanova, and M. Tokhtassyn, "Evaluating HTTP, MQTT over TCP and MQTT over WEBSOCKET for digital twin applications: A comparative analysis on latency, stability, and integration," *International Journal of Innovative Research and Scientific Studies*, vol. 8, no. 1, pp. 679–694, Jan. 2025, doi: 10.53894/ijirss.v8i1.4414.

[31] X. Chen, H. Kim, J. M. Aman, W. Chang, M. Lee, and J. Rexford, "Measuring TCP Round-Trip Time in the Data Plane," in *Proceedings of the Workshop on Secure Programmable Network Infrastructure*, New York, NY, USA: ACM, Aug. 2020, pp. 35–41. doi: 10.1145/3405669.3405823.