Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943



A Lightweight Real-Time Recognition System for Static Quranic Fingerspelling in the Malaysian Sign Language Context

Muhamad Zulhairi Mohd Nizam[1,*], ShaharilMad Saad[1], Mohamed Hussein[1], Norakyairee Mohd Raus[2], Muhamad Zulhelmi Mohd Nizam[1]

[1] Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[2] Faculty of Quranic and Sunnah Studies, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan Darul Khusus, Malaysia

ARTICLE INFO

Article history:
Received 3 September 2026
Received in revised form 15 September 2026
Accepted 17 September 2026
Available online 21 September 2026

Keywords:
Static sign language recognition; Quranic fingerspelling; Malaysian Sign Language; Hand landmark features; Real-time recognition

ABSTRACT

Accessible Quranic learning for Deaf learners in Malaysia requires recognition of Quran Rasm handshapes used within Malaysian Sign Language (MySL) practice. However, existing approaches provide limited support for real-time feedback on the handshapes performed by learners, and no dedicated public dataset is currently available for Quran Rasm in MySL. This study develops a lightweight real-time recognition application for 29 static Quran Rasm handshapes using MediaPipe hand landmarks and a compact feedforward neural network. Three geometric preprocessing configurations based on wrist normalization, hand mirroring, and scale normalization were evaluated using stratified five-fold cross-validation across multiple model initializations while maintaining the same classifier architecture. The recognition model was integrated into a multithreaded desktop application incorporating temporal stability filtering, interactive quiz generation, correctness feedback, scoring, optional answer hints, and result saving. The best preprocessing configuration achieved a macro-averaged F1-score of 0.9608, while the classifier contained 9,617 parameters with a model size of approximately 42 KB. The completed application was also used by 17 Deaf learners in the intended school environment, demonstrating practical deployment and direct learner interaction. These findings show that lightweight landmark-based recognition can be integrated into a functional Quran Rasm learning application for the Malaysian context.

1. Introduction

Sign language plays an important role in communication and education for Deaf communities. Sign languages differ across regions and communities and convey information through visual components such as handshape, movement, orientation, location, and facial expression. Despite advances in assistive technologies, access to appropriate communication and learning support remains limited in many educational environments, particularly when learners require individualized guidance and feedback [1], [2], [3].

These challenges are also present in Quranic education for Deaf learners in Malaysia. Previous studies have reported difficulties related to the consistency of teaching approaches and the need for

* *Corresponding author.*
E-mail address: mzulhairi6@graduate.utm.my

teachers to adapt learning activities according to student ability [4], [5]. Digital initiatives such as i-Sign QurANIS have been introduced to provide pre-recorded instructional materials. However, such resources mainly demonstrate the required signs and do not automatically determine whether a learner has produced the intended sign. As a result, corrective feedback during practice still depends largely on direct observation by a teacher or instructor. Computational support for Quranic sign language learning also remains limited, particularly in the development of recognition systems that can be used in practical learning environments [3].

Sign Language Processing (SLP) provides computational approaches for processing signed communication and generally includes sign language generation, recognition, and translation [1], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Among these tasks, Sign Language Recognition (SLR) focuses on identifying signs produced by a user and returning the corresponding label or representation. This makes SLR particularly relevant to interactive learning applications, where the system must determine the sign produced by a learner before providing a response. Existing Arabic and Quran-related recognition studies have demonstrated that alphabetic handshapes can be recognized using image-based machine learning and deep learning approaches [12], [18], [19]. Other studies have explored wearable sensing approaches using devices such as instrumented gloves [20], [21]. However, these studies address different recognition settings and application requirements, while the development of Quranic sign language recognition for practical learning applications remains relatively limited. Although these approaches have achieved promising recognition performance, several practical limitations remain. Image-based systems can be affected by variations in signer appearance, lighting, background, and camera conditions, while wearable approaches require additional hardware. Many recognition studies also focus primarily on classification performance using prepared datasets rather than evaluating the behavior of a complete system during continuous interaction.

Despite continued progress in sign language technologies, many systems remain at the prototype stage, with limited transition into real-world use [22]. Bragg *et al.* [2] emphasized the importance of considering real-world use when developing assistive sign language technologies, while Rodríguez-Correa *et al.* [23] highlighted the role of digital technologies in improving accessibility within sign language learning environments. These considerations are particularly important when recognition is intended to support learning. In such settings, a recognition model must operate as part of a complete application rather than only achieve high classification performance on an offline dataset. Practical use introduces additional requirements such as continuous camera processing, stable recognition during user movement, responsive feedback, and integration of recognition into an actual learning activity.

The present study therefore focuses on moving beyond offline classification by integrating a lightweight static Quran Rasm handshape recognizer into a complete real-time learning application for the Malaysian Quranic context. The target handshapes are those used for Quran Rasm in Malaysian Sign Language (MySL) practice, while a dedicated public dataset for this specific context is currently unavailable. Corresponding static handshapes are therefore selected from an existing ArSL dataset for development of the recognition system. The individual components used in the system, including hand landmark extraction, geometric preprocessing, feedforward classification, and multithreaded execution, are established methods. The novelty of the present study lies at the application and deployment level, where these established components are adapted and integrated into a lightweight real-time Quran Rasm recognition and assessment application for the Malaysian learning context. The resulting system combines real-time handshape recognition, temporal stability filtering, automatic evaluation of learner responses, immediate correctness feedback, scoring, result recording, and CPU-based operation within a single application. Three geometric preprocessing

configurations are evaluated using the same classifier architecture to examine the effects of position normalization, handedness unification, and scale normalization. The selected representation is integrated with temporal stability filtering and a multithreaded desktop application for continuous real-time recognition under CPU-based execution. The recognition output is further incorporated into a teacher-configurable quiz and assessment module that presents randomized target items, evaluates recognized responses, provides correctness feedback, records scores, and stores learner results for teacher review. The completed system is evaluated through offline classification, computational performance measurements, and practical use in the intended school environment. The present study focuses on the technical development, evaluation, and practical deployment of the application. The main contributions are as follows:

1. Development of an interactive real-time recognition application for static Quran Rasm handshapes in the Malaysian learning context. The application integrates live-camera recognition with a teacher-configurable quiz and assessment workflow that presents target signs, evaluates recognized responses, provides correctness feedback, updates scores, and records learner results.
2. Deployment-oriented implementation and evaluation of the real-time recognition pipeline. Temporal stability filtering is combined with a multithreaded architecture that separates graphical interaction, hand-landmark processing, and classification. Computational performance is evaluated under CPU-only execution using model complexity, classifier inference time, and end-to-end latency.
3. Evaluation of geometric hand-landmark preprocessing for the deployed recognizer. Wrist-based position normalization, handedness unification through mirroring, and scale normalization are compared using the same classifier architecture, allowing the effect of the landmark representation to be examined independently of changes to the recognition model.
4. Practical deployment of the completed application with Deaf learners in the intended school environment. The application was used directly by learners during a school session, demonstrating operation of the recognition, quiz, feedback, scoring, and result-recording functions under the target usage setting.

1.1 Quran Rasm in the Malaysian Learning Context

Sign language has been applied in different ways to support Quranic learning for Deaf learners. The representation of Quranic content is not uniform across these approaches. Some systems use lexical signs, where a complete Quranic word or expression is represented by a corresponding sign. For example, AbdElghfar *et al.* [24] used Arabic Sign Language (ArSL) representations for selected Quranic words and expressions. In this type of approach, the recognition target is the complete signed item rather than the individual Arabic letters that form the word.

Other approaches represent Quranic content through individual Arabic-letter signs. In these approaches, the letters contained in the written Quranic text are represented separately using their corresponding handshapes. This creates a different recognition problem because the system must identify individual letter signs rather than complete lexical signs. Figure 1 illustrates the difference between lexical representation and the fingerspelling-based representation considered in the present study.

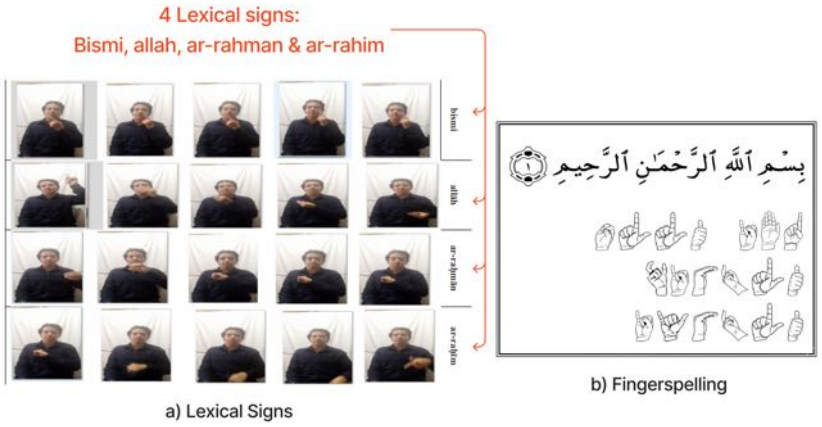


Fig. 1. Quranic representation using (a) lexical signs in ArSL [24] and (b) fingerspelling in Quran Rasm in MySL

In the Malaysian context considered in this study, Quran Rasm in MySL is used as a letter-based representation of Quranic text through fingerspelling. Quranic words are represented by producing their written Arabic letters according to their order in the text. The sign set contains basic Arabic-letter signs and additional letter forms used in Quranic text. Figure 2 presents the Quran Rasm handshapes used in the Malaysian context together with the corresponding ArSL handshapes considered in this study. A number of the static Quran Rasm handshapes show close visual correspondence with classes available in the ArSL alphabet dataset. These corresponding static classes form the recognition scope of the present work. Signs that require movement are excluded because the current study focuses specifically on static handshape recognition.



Fig. 2. Visual comparison between Quran Rasm handshapes used in the Malaysian context and ArSL alphabet handshapes. Green boxes indicate static handshapes included in the present study, while red boxes indicate signs outside the scope of the current static recognition system

A dedicated public dataset specifically developed for Quran Rasm in MySL is currently unavailable [3]. The visual correspondence shown in Figure 2 therefore provides a practical basis for selecting classes from an existing ArSL dataset for model development. The use of the ArSL dataset is based on similarity of the target handshapes and does not imply that ArSL and Quran Rasm in MySL are linguistically identical. Classes that do not correspond to the target MySL sign set are excluded during dataset preparation. The detailed class-selection procedure is described in the Methodology.

1.2 Challenges in Quranic Learning for Deaf Learners

Despite the availability of sign-based approaches for Quranic learning, Deaf learners in Malaysia continue to face several educational challenges. Quranic teaching commonly depends on direct guidance and visual demonstration from teachers. Differences in students communication abilities and learning needs also require teachers to adapt their teaching approaches according to the individual learner [4], [5], [25].

Previous studies have also reported limitations in suitable learning resources and in the availability of teachers with knowledge of both Quranic education and sign language [3], [25], [26], [27]. These conditions can limit the amount of individualized support available to learners, particularly during repeated practice that depends on direct observation and feedback from the teacher.

Digital learning materials can provide additional support by presenting instructional content and visual demonstrations. However, resources that only demonstrate the expected sign do not determine what the learner produces in response. A learner may therefore be able to observe the correct sign without receiving automatic feedback on whether the sign that they produce corresponds to the requested handshape.

Sign Language Recognition provides a means of addressing this specific limitation by identifying the sign produced by the learner and allowing the recognized response to be used within an interactive activity [2]. In the present study, this capability is applied specifically to the static Quran Rasm handshapes used in the Malaysian learning context. The recognition output is integrated into a quiz and assessment application in which a target item is presented to the learner, the produced handshape is recognized, and the response is compared with the requested target. This allows the application to provide recognition-based feedback during practice rather than functioning only as a source of instructional content.

1.3 Existing Quranic and Arabic Fingerspelling Recognition Systems

Previous studies have explored Arabic alphabet recognition and Quran-related sign recognition using different sensing and classification approaches. These studies demonstrate that Arabic handshapes can be recognized using camera-based image classification, hand landmarks, and wearable sensors. However, their recognition scope, interaction design, and intended application differ from the Quran Rasm learning context considered in the present study. Table 1 summarizes the most directly relevant systems.

Table 1
Comparison of related Arabic fingerspelling and Quran-related sign recognition systems

Study	Scope	Recognition Approach	Main Focus and distinction
Tharwat et al. (2021) [12]	Arabic alphabet with Quran-related application	Camera-based KNN	Static alphabet recognition applied to a limited set of Quran-related letters. The study is primarily recognition-oriented model offline rather than within a real-time or practical application setting.
AbdElghfar et al. (2023) [28]	ArSL alphabet	CNN using static RGB images	Evaluates static Arabic alphabet recognition using an image dataset. It provides a technically comparable alphabet-recognition task but is not designed specifically for Quran Rasm in MySL or for an integrated learner-assessment workflow.
Amor et al. (2023) [28]	ArSL alphabet	EMG sensors with CNN-LSTM	Recognizes Arabic alphabet signs using wearable sensing. The approach differs from the camera-only interaction required for the present learning application.
Elshareif et al. (2024)[21]	Static Arabic Hijaiyah hand codes for Quran recitation	Flex sensors and accelerometer with rule-based classification	Provides Quran-related hand-code recognition with a mobile application, but requires a wearable smart glove and covers a limited static sign set
El-Sayed et al. (2026) [29]	ArSL letters, numbers, words, and sentences	MediaPipe, Transformer, and GRU	Provides a more complex landmark-based temporal recognition framework for general ArSL. It is not specific to Quran Rasm learning and addresses a broader recognition task rather than the learner-assessment workflow targeted here.
AbdElghfar et al. (2026)[24]	Selected Quranic expressions from Surah Al-Ikhlas (Lexical Signs)	MediaPipe with conventional ML, ResNet50, and hybrid classification	Develops an automated Quranic education framework using static poses and images. The reported scope is limited to selected Quranic expressions (lexical signs, not alphabet signs), with on-device real-time implementation identified as a future direction.

Present study	29 static Quran Rasm handshapes in MySL	MediaPipe landmarks with a lightweight FFNN	Integrates real-time camera-based recognition with randomized target questions, correctness feedback, scoring, optional hints, result saving, and deployment with Deaf learners in the intended Malaysian school environment.

The comparison shows that previous work has already established the feasibility of Arabic alphabet recognition, Quran-related sign recognition, wearable Quranic hand-code systems, and more complex real-time ArSL recognition models. Therefore, the contribution of the present study does not rely on claiming that hand landmark extraction, feedforward classification, geometric preprocessing, or real-time sign recognition are individually new. Instead, the research gap addressed in this study concerns the application and deployment of these established recognition techniques for the specific Quran Rasm system used in the Malaysian learning context. Among the directly comparable studies considered here, limited attention has been given to integrating recognition of the target static Quran Rasm handshapes with a complete learner-facing workflow that presents target signs, evaluates the produced response, provides immediate correctness feedback and scoring, and records assessment results for teacher review. The present study addresses this application-level and deployment-oriented gap through a lightweight real-time Quran Rasm recognition and assessment application designed for the Malaysian learning context.

2. Methodology

2.1 Overall System Architecture and Application Workflow

The developed application integrates real-time static handshape recognition with an interactive learning and assessment workflow. The recognition component follows a multithreaded architecture to support continuous camera processing while maintaining a responsive graphical user interface. The system consists of three main processing components, namely the graphical user interface, hand landmark extraction with temporal stability filtering, and the classification module. The recognition output is subsequently used by the application to provide feedback during sign practice and to evaluate learner responses during quiz activities. The overall system architecture is illustrated in Figure 3.

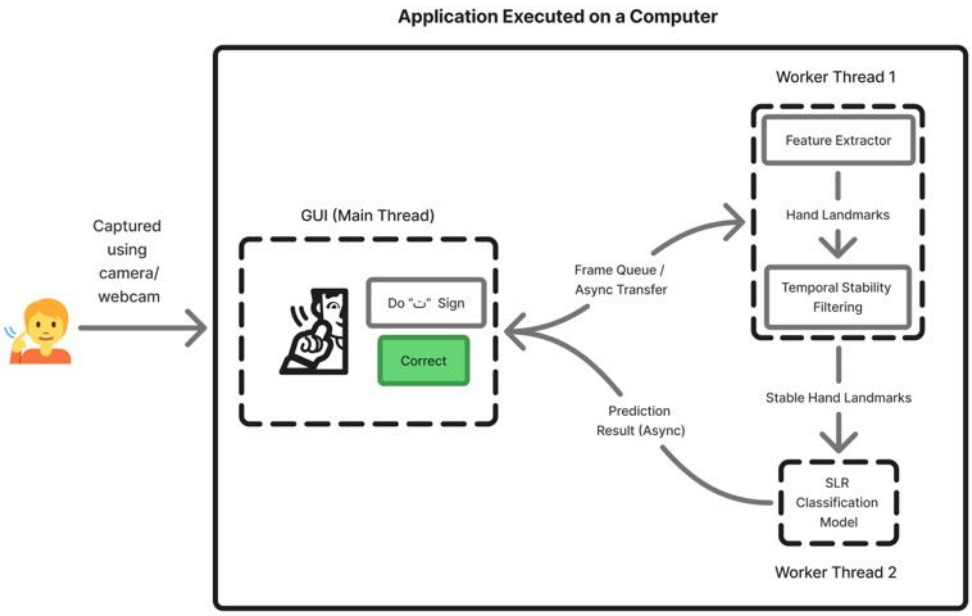


Fig. 3. System architecture of the proposed real-time sign language recognition application, showing the separation between the GUI main thread and worker threads for feature extraction, stability filtering, and classification

The graphical user interface is executed on the main thread and is responsible for receiving camera input and displaying real-time feedback to the user. Hand landmark extraction and motion-based temporal stability filtering are performed in a separate worker thread. Incoming video frames are processed using MediaPipe Hands to obtain 21 hand landmarks. The detected landmark coordinates are stored in a fixed-length buffer containing N consecutive frames. Once the buffer is filled, the temporal stability of the hand configuration is evaluated before classification is performed. The stability condition is based on the amount of landmark movement across the buffered frames. For each pair of consecutive frames, the Euclidean displacement of each of the 21 landmarks is calculated and averaged across the landmarks. These frame-to-frame motion values are then averaged across the complete buffer to obtain the overall motion score:

$$M = \frac{1}{N-1}\sum_{i=2}^{N}\left[\frac{1}{21}\sum_{j=1}^{21}\left\|\boldsymbol{p}_{i,j} - \boldsymbol{p}_{i-1,j}\right\|_2\right], \tag{1}$$

where $\boldsymbol{p}_{i,j}$ represents the raw MediaPipe landmark coordinate vector of landmark j in frame i, N is the number of frames contained in the buffer, and M represents the average motion across the buffered sequence. Since the stability calculation is performed before classifier preprocessing, the motion score is calculated directly from the landmark coordinates returned by MediaPipe. A hand configuration is considered sufficiently stable when:

$$M \leq \theta_s, \tag{2}$$

where θ_s is the predefined motion threshold. When the stability condition is satisfied, the most recent landmark frame in the buffer is selected and passed through the geometric preprocessing procedure before being forwarded to the FFNN classifier. If the motion score exceeds the threshold, classification is not performed for the current buffer. This prevents predictions from being generated while the learner is still positioning or moving the hand.

In the implemented application, the buffer length was set to $N = 30$ frames and the motion threshold to $\theta_s = 0.01$. These values were selected empirically during preliminary application development to reduce predictions during hand movement while maintaining responsive recognition. Both parameters remain configurable within the application to allow adjustment for different interaction conditions. However, $N = 30$ and $\theta_s = 0.01$ were kept fixed throughout all evaluations reported in this study.

The classification module is executed in another worker thread and receives the pre-processed landmark representation only after the temporal stability condition has been satisfied. The FFNN then predicts one of the 29 static Quran Rasm handshape classes. The predicted class is returned to the application interface, where it can either be displayed as the recognized output or compared with the requested target during a quiz activity. Separating the graphical interface, hand landmark processing, and classification helps prevent computational operations from blocking the user interface while allowing continuous processing during live recognition.

2.2 Dataset and Target Class Selection

Currently, there is no dedicated public dataset available specifically for Quran Rasm in MySL [3]. Therefore, the ArSL dataset introduced by Al-Barham *et al.* [9] was used as the source dataset for developing the static handshape recognizer. The dataset consists of RGB images of individual Arabic alphabet signs. Several of the handshapes represented in this dataset visually correspond to the static Quran Rasm handshapes used in the Malaysian learning context, allowing the corresponding classes to be selected for the present recognition task.

The source dataset contains several Arabic alphabet classes that correspond to the static Quran Rasm handshapes used in the present application. It also includes the Al and Laa classes. These classes were not used in the target recognition set because Quran Rasm in MySL represents the corresponding letters using separate individual signs rather than the combined sign forms provided in the source ArSL dataset, as illustrated in Figure 4.

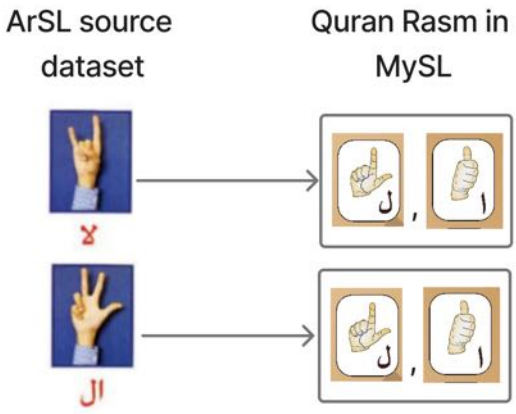


Fig. 4. Difference in the representation of Al and Laa between the ArSL source dataset and Quran Rasm in MySL. In the source dataset, each is represented by a single combined sign form, whereas Quran Rasm in MySL represents the corresponding letters using separate individual signs.

The retained recognition set consisted of 29 static classes: Alif (ا), Ba (ب), Ta (ت), Tha (ث), Jeem (ج), Hah (ح), Kha (خ), Dal (د), Dhal (ذ), Ra (ر), Zay (ز), Sin (س), Shin (ش), Sad (ص), Dad (ض), Ta (ط), Za (ظ), Ain (ع), Ghain (غ), Fa (ف), Qaf (ق), Kaf (ك), Lam (ل), Meem (م), Noon (ن), Haa (ه), Waw (و), Ya (ي), and Ta Marbuta (ة). Furthermore, the present study is limited to isolated static handshape recognition because the source dataset contains only static alphabet images. As a result, Quran Rasm signs that require movement are not included in the current recognition model. The source images corresponding to the retained classes are subsequently filtered and converted into hand landmark representations during dataset preprocessing.

2.3 Machine Learning Pipeline Overview

The developed application uses a lightweight landmark-based pipeline for real-time static handshape recognition. Image or video frames captured from the camera are first processed using MediaPipe Hands to extract the hand landmark coordinates. The extracted landmarks are then evaluated using the motion-based temporal stability criterion described in Section 2.1. Only hand configurations that satisfy the stability condition are accepted for classification.

Once the stability condition is satisfied, the most recent landmark frame in the buffer is processed using the selected geometric preprocessing procedure. The resulting landmark representation is then passed to the classifier to obtain the predicted static Quran Rasm class. The predicted class is returned to the application interface, where it is displayed as the recognized output or compared with the requested target during a quiz activity. Figure 5 illustrates the overall recognition pipeline implemented in the application.

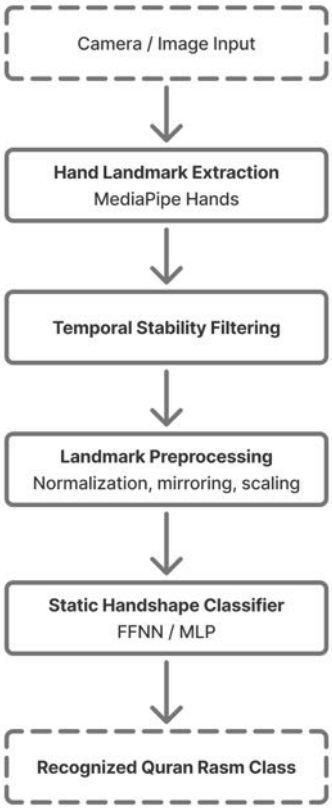


Fig. 5. Recognition pipeline for real-time static Quran Rasm handshape recognition.

2.4 Feature Extraction Model

The development of sign language recognition systems depends not only on dataset size but also on the diversity and independence of the data used for training and evaluation. When the same signers appear in both the training and testing sets, reported performance may be overly optimistic because the model can learn signer-specific characteristics rather than the underlying sign representation [30]. This issue is particularly relevant for RGB-based approaches, where visual information such as skin appearance, clothing, illumination, and background can influence the extracted features. Previous sign language recognition systems have therefore explored different strategies to reduce dependence on raw visual appearance, including to rely on gloves to simplify hand detection and recognition [31], [32], [33].

In the present study, handshape information is represented using the MediaPipe Hand Landmark model. MediaPipe Hands is designed for real-time hand tracking and estimates 21 landmarks distributed across the wrist, palm, and finger joints [34]. Converting the RGB hand image into a landmark representation allows the subsequent classifier to operate on the geometric configuration of the detected hand rather than directly on raw pixel appearance. This reduces the classifier's direct dependence on visual characteristics of the input image, although the reliability of the representation remains dependent on the accuracy of the landmark detector. Figure 6 illustrates the 21 hand landmarks used in the recognition pipeline.

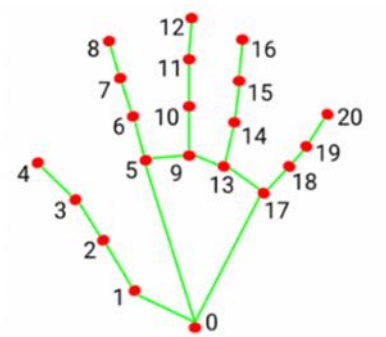


Fig. 6. Extracted skeletal landmarks with 21 nodes corresponding to the finger joints.

For each detected hand, MediaPipe provides three coordinates, x, y, and z, for every landmark. The x and y coordinates describe the landmark position in the image plane, while the z value represents an estimated relative depth from the monocular RGB input rather than a direct physical depth measurement. Since the present study focuses on static handshape recognition, only the $x$and $y$coordinates are retained. The resulting feature representation contains 42 values for each detected hand, corresponding to 21 landmarks with two coordinates per landmark.

2.5 Preprocessing Dataset

The dataset consists of RGB images and includes several Arabic alphabet classes, including the combined Al and Laa sign forms. As described in Section 2.2, these classes were excluded because they do not correspond to the separate letter representations used in Quran Rasm in MySL. Their removal reduced the dataset from 7,773 to 7,299 images. Since the proposed classifier operates on skeletal hand landmark features rather than raw RGB images, the remaining images were processed using the MediaPipe Hand Landmark model as a feature extractor. Valid hand landmarks were detected and retained for 5,462 images, corresponding to approximately 74.9% of the retained

source images. Various methods have been proposed for processing and normalizing hand landmark data [35], [36]. In this study, three preprocessing configurations were investigated to study their impact on model performance. All configurations use the same classifier architecture and differ only in the applied preprocessing strategy.

In the first configuration, wrist-based normalization is applied by aligning all hand landmarks relative to the wrist joint (node index 0), as defined in (3):

$$P'_i = P_i - P_0 \tag{3}$$

where $\mathbf{P}_i$denotes the x–y coordinates of the i-th hand landmark and $\mathbf{P}_0$represents the wrist landmark. This step removes global translation and centers the hand representation.

In the second configuration, wrist normalization is combined with horizontal mirroring. As the MediaPipe model may detect either left or right hands, the same gesture can produce mirrored landmark patterns. To ensure consistency, all detected left-hand landmarks are horizontally flipped to match a canonical right-hand representation. This reduces variation in the input data and improves model learning.

In the third configuration, wrist normalization and mirroring are further extended with scale normalization to account for variations in hand size and distance from the camera. Two reference vectors are computed to estimate palm size:

$$S_1 = P_9 - P_0 \tag{4}$$

$$S_2 = P_{17} - P_5 \tag{5}$$

where $S_1$and S_2approximate palm length and palm width, respectively. As these measurements may vary due to changes in hand posture, their average is used to obtain a more stable scale factor:

$$S' = \frac{S_1 + S_2}{2} \tag{6}$$

The final scaled landmark representation is then computed as:

$$P'_i = \frac{P_i}{S'} \tag{7}$$

To evaluate the effectiveness of these preprocessing configurations, stratified k-fold cross-validation is employed with $k = 5$, preserving the class distribution across all folds. All configurations

are trained and evaluated under identical conditions, enabling a fair comparison of their impact on classification performance.

2.6 Classifier Model

A Feed-Forward Neural Network (FFNN) was used to classify the preprocessed hand landmark features. The classifier receives a compact landmark vector rather than raw RGB images. Since the spatial information of the hand has already been represented through the extracted landmark coordinates, the classifier operates directly on this structured geometric representation without requiring additional image feature extraction through convolutional layers.

Each input sample contains 42 features corresponding to the $x$and $y$coordinates of 21 hand landmarks. The network consists of two fully connected hidden layers, each containing 68 units. The first hidden layer is followed by a Rectified Linear Unit (ReLU) activation function and a dropout layer with a probability of 0.3. The second hidden layer is followed by another ReLU activation. The final linear layer maps the learned feature representation to 29 output classes corresponding to the static Quran Rasm handshapes included in the recognition set.

The same FFNN architecture was used across all preprocessing configurations so that the influence of the preprocessing strategy could be evaluated without introducing changes in the classifier structure. The compact architecture was also selected to support low-cost inference during real-time application use. In this study, the classifier therefore serves as a lightweight recognition component within the complete learning application rather than as a new neural network architecture.

2.7 Interactive Quiz and Assessment Module

The recognition system was integrated into an interactive quiz and assessment module designed for Quran Rasm handshape practice. Before the session begins, the teacher can specify the number of quiz questions to be presented. For each question, the application randomly selects a target sign from the available static recognition classes and displays it to the learner. Figure 7 shows the quiz interface of the developed application.

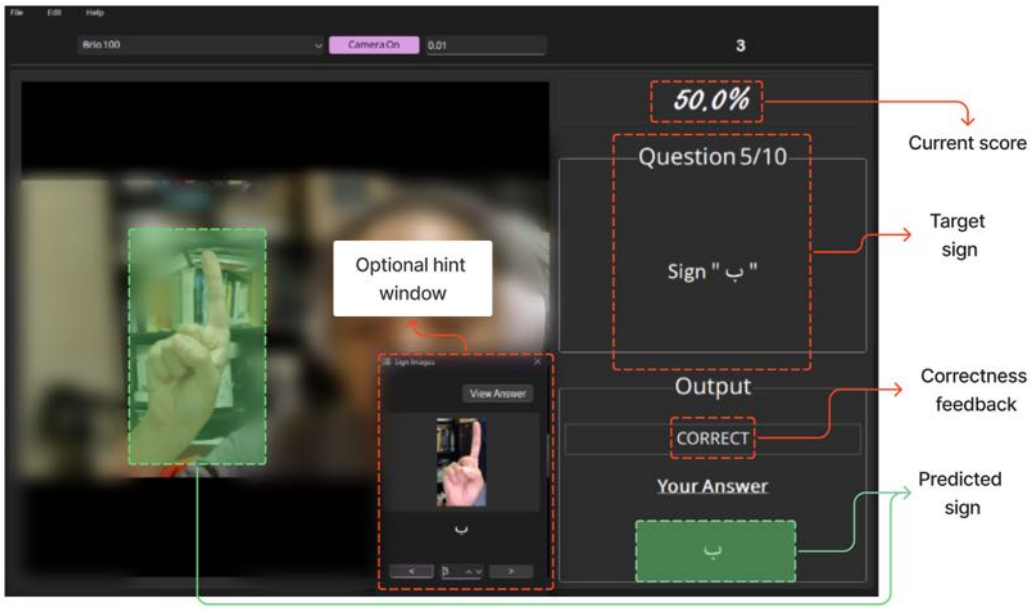


Fig. 7. Quiz interface of the developed Quran Rasm learning application

The learner then performs the requested handshape in front of the camera. The recognized output from the classifier is compared with the target sign selected by the application. When the recognized class matches the target, the response is recorded as correct and the learner's score is updated. The interface also provides a hint function that allows the learner to view the expected answer during practice. At the end of the session, the application displays the learner's final quiz result and allows learner information to be entered before saving the assessment outcome. The recorded information includes the learner's name, class, and final score. Figure 8 shows the result dialog used for this purpose.



Fig. 8. Result dialog showing learner information and final quiz score

In the present implementation, the quiz result is stored in text format for later teacher review. By incorporating recognition into this quiz and assessment workflow, the application uses the predicted sign as part of an interactive feedback process rather than presenting it only as an isolated classification output.

2.8 System Evaluation/Experimental Setup

Two technical evaluation stages were conducted in this study. The first evaluated the classification performance of the FFNN under the different preprocessing configurations, while the second assessed the computational performance of the deployed system under CPU-only execution. The same classifier architecture and training settings were used for all preprocessing configurations, with only the preprocessing strategy being changed.

For each preprocessing configuration, the FFNN was trained using a batch size of 64 for 200 epochs. The Adam optimizer was employed with a learning rate of 0.001, and cross-entropy loss was used during training. Classification performance was evaluated using stratified 5-fold cross-validation, with class distributions preserved across the folds. Multiple random seed initializations were also used to account for variation in model training. The final performance was reported as the mean and standard deviation across the folds and seeds. Macro-averaged and micro-averaged F1-scores were used to compare the preprocessing configurations. Macro F1 provides equal consideration to each class, while micro F1 reflects overall classification performance across all samples.

Because signer identity information was not available in the source dataset, the cross-validation procedure was stratified according to class labels rather than signer identity. Consequently, the reported classification results should not be interpreted as a signer-independent evaluation.

For the real-time performance evaluation, the deployed system was tested under CPU-only execution. Model inference time was measured from the input feature vector to the output of the FFNN classifier. End-to-end latency was measured during live execution from frame acquisition through hand landmark extraction, preprocessing, model inference, and display of the predicted label on the graphical user interface. The evaluation was conducted over 200 completed sign-recognition trials, where each trial corresponded to one completed sign prediction. The experiments were conducted on a desktop workstation equipped with an Intel Core i5-12400 CPU and 32 GB of RAM using a Logitech Brio 100 webcam with a nominal frame rate of 30 FPS. Although an NVIDIA RTX 3070 GPU was available on the workstation, GPU acceleration was not used during testing.

In addition to the technical evaluation, the completed application was used with Deaf learners in the intended school environment. The session was conducted in the school hall, where the learners interacted with the application under the target usage setting. This activity was intended to demonstrate the operation of the developed application in practice and was not conducted as a formal usability or educational-effectiveness study. No standardized measures of learner experience, teacher acceptance, accessibility, or pedagogical outcomes were collected

3. Results and Discussion

3.1 Effect of Preprocessing Strategies on Model Performance

The effect of the three preprocessing configurations was evaluated using the same FFNN architecture and training settings. Performance was assessed using stratified 5-fold cross-validation with multiple random seed initializations, and the results are reported as the mean and standard deviation across all folds and seeds. Table 2 summarizes the macro-averaged and micro-averaged F1-scores obtained for each preprocessing configuration.

Table 2
Classification Performance Across Preprocessing Configurations

Preprocessing Configuration	Macro F1 (mean ± std)	Micro F1 (mean ± std)
Wrist normalization	0.9490 ± 0.0094	0.9523 ± 0.0086
Wrist + mirroring	0.9582 ± 0.0067	0.9614 ± 0.0061
Wrist + mirroring + scaling	0.9608 ± 0.0057	0.9639 ± 0.0052

Using wrist normalization alone, the classifier achieved a macro F1-score of 0.9490 ± 0.0094 and a micro F1-score of 0.9523 ± 0.0086. Adding horizontal mirroring improved the scores to 0.9582 ± 0.0067 and 0.9614 ± 0.0061, respectively. The highest performance was obtained when scale normalization was also included, with a macro F1-score of 0.9608 ± 0.0057 and a micro F1-score of 0.9639 ± 0.0052.

The results show a progressive improvement as additional geometric normalization steps were introduced. Wrist normalization provides a baseline by removing global hand translation and centering the landmarks relative to the wrist. Mirroring produced a larger improvement by converting equivalent left- and right-hand representations into a consistent orientation, thereby reducing variation in the input representation. This approach is appropriate for the present application, which considers single-hand static fingerspelling and does not require preservation of left-right hand distinction. The addition of scale normalization produced a smaller further

improvement, increasing the macro F1-score from 0.9582 to 0.9608 and the micro F1-score from 0.9614 to 0.9639. The standard deviation was also reduced for both metrics. This indicates that reducing variation associated with hand size and apparent hand scale provided a more consistent landmark representation, although the improvement over mirroring alone was modest.

Overall, the comparison indicates that the preprocessing configuration affected classification performance while the underlying classifier architecture remained unchanged. Among the evaluated configurations, wrist normalization combined with mirroring and scaling produced the highest overall performance and was therefore selected for the deployed application.

3.2 Confusion Matrix Analysis

To further examine the classification performance of the best-performing preprocessing configuration, the confusion matrix obtained using wrist normalization, mirroring, and scaling is presented in Figure 9. Most predictions are concentrated along the main diagonal, indicating that the majority of static handshape classes were correctly distinguished by the classifier.

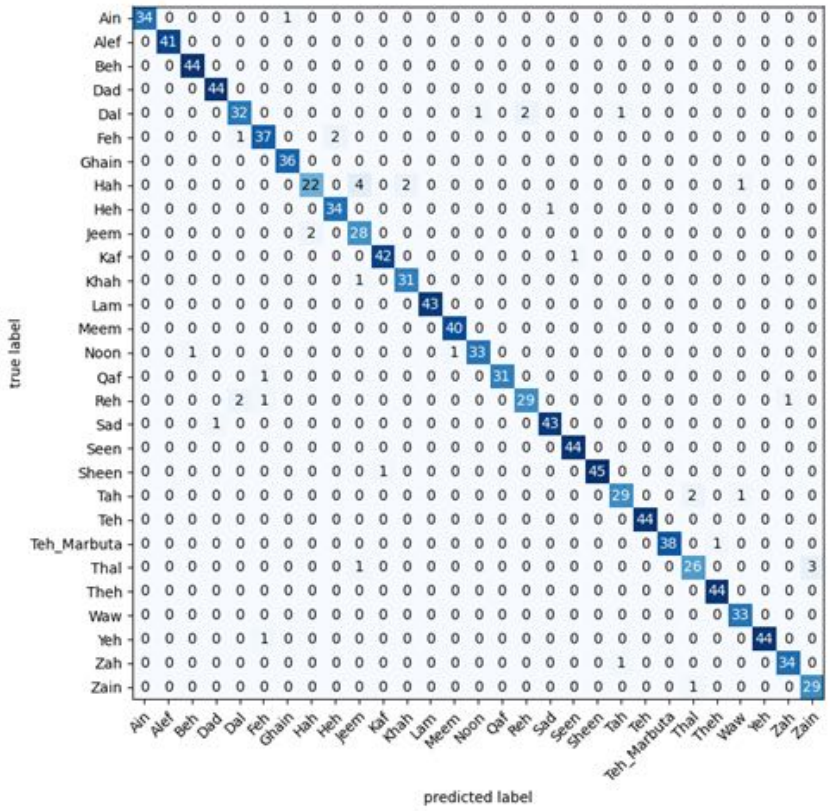


Fig. 9. Confusion matrix of the proposed model using the best-performing preprocessing configuration (wrist normalization with mirroring and scaling)

The remaining errors were examined to identify possible causes of misclassification. One recurring example involved the visually similar Jeem and Hah handshapes. Figure 10(A) shows two samples representing the intended Jeem sign. In one case, the estimated landmarks did not capture the thumb configuration accurately, resulting in a landmark pattern that more closely resembled Hah. Figure 10(B) shows reference samples for Jeem and Hah, where the difference between the two classes is largely associated with the thumb configuration.

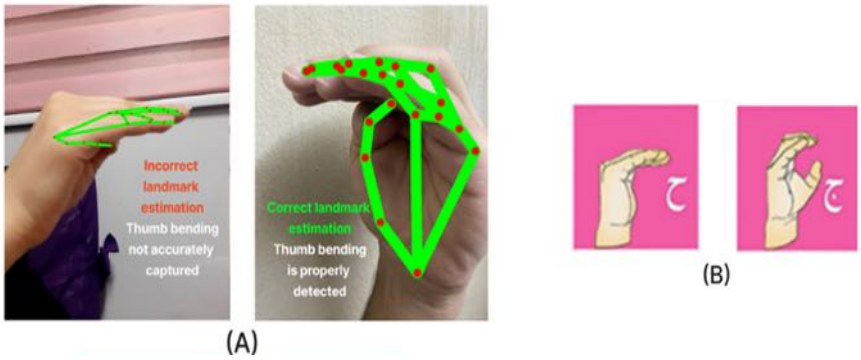


Fig. 10. Example of misclassification between Jeem and Hah. (A) Incorrect landmark estimation for Jeem where thumb bending is not captured, leading to misclassification as Hah. (B) Reference samples of Jeem and Hah showing differences in thumb configuration)

This example indicates that classification performance is affected by the quality of the hand landmark representation. When important structural details are not captured accurately, particularly for classes that differ through relatively small finger or thumb configurations, the resulting feature vectors may become difficult to distinguish. The error therefore highlights a limitation of using detected landmark coordinates as the sole input representation, since classification depends on the accuracy with which the landmark extractor preserves discriminative handshape information.

3.3 Real-Time System Performance

To evaluate the computational performance of the deployed recognition system, model inference time and system processing latency were measured during live execution under CPU-only conditions. Figure 11 shows the application during real-time sign recognition.

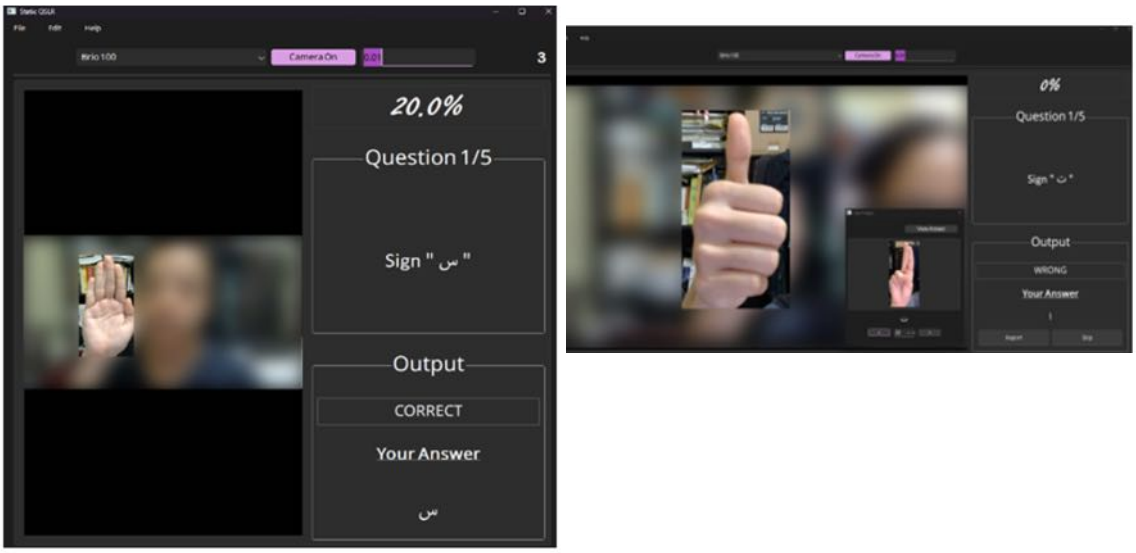


Fig. 11. Screenshot of the proposed SLR application during real-time sign recognition

The system was evaluated over 200 completed sign-recognition trials. The isolated FFNN inference time, measured from feature-vector input to model output, was 0.49 ± 0.81 ms. The mean

end-to-end latency, measured from frame acquisition through hand landmark extraction, preprocessing, model inference, and display of the predicted label on the graphical user interface, was 0.95 ± 0.21 ms. The classifier contains 9,617 trainable parameters and has a model size of approximately 42 KB, excluding the MediaPipe hand landmark model used for feature extraction. Table 3 summarizes the model complexity and real-time performance measurements.

Table 3
Model Complexity and Real-Time Performance Metrics

Metric	Value
Total Parameters	9,617
Model Size (KB)	42
Model Inference Time (ms)	0.49 ± 0.81
End-to-end latency (ms)	0.95 ± 0.21

The compact FFNN architecture and 42-dimensional landmark representation resulted in low isolated inference time during CPU-only execution. The measured end-to-end latency of 0.95 ± 0.21 ms was also well below the 33.33 ms frame interval associated with a 30 FPS input stream [38], indicating that the recognition pipeline could complete processing within the interval between consecutive frames.

3.4 Target-Environment Use

The completed application was deployed at the school as part of its intended use. At the request of the school principal, Deaf learners were gathered in the school hall and were given the opportunity to use the application individually. A total of 17 learners interacted with the system, with each learner completing five quiz attempts.

During the activity, the application presented target signs, processed the learner's hand input, displayed the recognized sign, provided correctness feedback, and updated the quiz score. The learners also entered and saved their information and quiz results using the application's built-in result-saving function. Figure 12 shows the application being used during the school session.



Fig. 12. Use of the developed application with Deaf learners in the school hall

The session demonstrated practical operation of the completed application in the intended school environment. The recognition, quiz, feedback, scoring, and result-saving functions were used during actual learner interaction. The present study focuses on the technical development and deployment of the application, while structured evaluation of learner experience and educational outcomes is outside the scope of the current work.

3.5 Limitations

The results of this study should be interpreted within the scope of the implemented system and evaluation conditions. The classifier was developed using a public ArSL dataset because no dedicated public dataset is currently available specifically for Quran Rasm in MySL. Signer identity information was not available in the source dataset, and the reported stratified cross-validation results therefore do not represent a signer-independent evaluation. Model development was based on 5,462 retained landmark samples from the available public source dataset. Future evaluation using a dedicated Quran Rasm in MySL dataset with broader participant coverage would allow the recognition system to be examined across a wider range of users.

The technical experiments were conducted using a consistent camera and workstation configuration in a well-lit indoor environment with a relatively plain background. The application was subsequently used by Deaf learners in the school hall, providing additional evidence of operation in the intended environment. The school deployment involved learners from a single school and focused on practical use rather than comparison across different learner backgrounds or levels of signing experience. Variations in lighting, camera angle, background complexity, camera device, and computing hardware were not systematically compared. Recognition performance also depends on the quality of the hand landmarks extracted by MediaPipe. The error analysis showed that inaccurate estimation of small thumb or finger configurations can affect visually similar signs, particularly when the hand is partially occluded or discriminative finger configurations are not captured accurately.

The current application focuses on 29 static Quran Rasm handshapes, while signs that require movement are outside the scope of the present classifier because the available source dataset contains only static alphabet images [9] and no dedicated public dataset is currently available for Quran Rasm in MySL [3] . The selected handshapes reflect the Quran Rasm signing system used in the target Malaysian setting, while future work may examine their use across additional schools and practitioner settings. The school session demonstrated direct use of the application by Deaf learners in its intended environment. The present work focused on system development, technical performance, and practical deployment, while structured investigation of learner experience, teacher perspectives, and learning outcomes remains a direction for future user-centered evaluation.

4. Conclusions

This study developed a lightweight real-time recognition application for static Quran Rasm handshapes in the Malaysian learning context. Rather than introducing a new recognition architecture, the work focused on integrating landmark-based handshape recognition, geometric preprocessing, temporal stability filtering, and a lightweight classifier into a complete interactive learning application. The resulting system supports real-time sign recognition together with quiz generation, correctness feedback, scoring, optional answer hints, and result saving for teacher review.

The technical evaluation showed that the preprocessing strategy influenced classification performance, with the combination of wrist normalization, mirroring, and scaling producing the

strongest overall results. The compact FFNN supported low-cost CPU-based inference, while the error analysis showed that recognition performance remains dependent on the quality of the extracted hand landmarks, particularly for visually similar signs that differ through small finger or thumb configurations.

Beyond classifier evaluation, the completed application was deployed in the intended school environment and used directly by 17 Deaf learners during a school-hall session. This demonstrated that the recognition, quizzes, feedback, scoring, and result-saving functions could be used as an integrated application during actual learner interaction. The contribution of the present work therefore lies not only in the recognition performance of the classifier, but also in the implementation and practical deployment of a lightweight Quran Rasm learning application for the Malaysian context.

The present system focuses on 29 static Quran Rasm handshapes used in the Malaysian learning context. The selected handshapes were based on the Quran Rasm signing system adopted in the target setting. Quran Rasm signs that require movement are outside the scope of the current classifier because the available source dataset contains static alphabet images and no dedicated public dataset is currently available for Quran Rasm in MySL. Extending the system to dynamic signs would therefore require the collection of dedicated temporal signing data. Future work can also extend the evaluation to additional signers, varied lighting and background conditions, different camera angles, alternative camera and computing devices, and other school settings. Structured investigation of learner experience, teacher perspectives, and learning outcomes remains a direction for future user-centered evaluation.

Acknowledgement
This research was not funded by any grant

References

[1] De Coster, Mathieu, Dimitar Shterionov, Mieke Van Herreweghe, and Joni Dambre. "Machine translation from signed to spoken languages: state of the art and challenges." *Universal Access in the Information Society* 23 (2024): 1305-1331. https://doi.org/10.1007/s10209-023-00992-1

[2] Bragg, Danielle, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, et al. "Sign language recognition, generation, and translation: An interdisciplinary perspective." In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*, 16-31. 2019. https://doi.org/10.1145/3308561.3353774

[3] Abdulrahim, Khairi, Norakyairee Mohd Raus, Khairul Nabilah Zainul Ariffin, Hafizal Mohamad, Juliza Jamaludin, and N. H. Z. Anuar. "Quranic Sign Language Recognition: Challenges and Future Directions for Edge-Aware Systems." In *2025 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 42-46. 2025. https://doi.org/10.1109/ISITIA66279.2025.11137544

[4] Dzulkifli, Izuli, Asmawati Suhid, Fathiyah Mohd Fakhruddin, and Nor Aniza Ahmad. "Teacher communication in teaching al-Quran to special needs pupils with hearing disabilities." *Universal Journal of Educational Research* 8, no. 1A (2020): 36-43. https://doi.org/10.13189/ujer.2020.081306

[5] Dzulkifli, Izuli, Asmawati Suhid, Fathiyah Mohd Fakhruddin, and Nor Aniza Ahmad. "Activity-based teaching of Quran for deaf students in the special education integration program." *Pertanika Journal of Social Sciences and Humanities* 29, no. 1 (2021): 91-106. https://doi.org/10.47836/PJSSH.29.1.05

[6] Mehta, Nayan, Suraj Pai, and Sanjay Singh. "Automated 3D sign language caption generation for video." *Universal Access in the Information Society* 19, no. 4 (2020): 725-738. https://doi.org/10.1007/s10209-019-00668-9

[7] Inan, Mert, Yang Zhong, Sabit Hassan, Lorna Quandt, and Malihe Alikhani. "Modeling Intensification for Sign Language Generation: A Computational Approach." In *Findings of the Association for Computational Linguistics: ACL 2022*, 2897-2911. 2022. https://doi.org/10.18653/v1/2022.findings-acl.228

[8] Kapoor, Parul, Rudrabha Mukhopadhyay, Sindhu B. Hegde, Vinay P. Namboodiri, and C. V. Jawahar. "Towards Automatic Speech to Sign Language Generation." In *Interspeech 2021*, 3700-3704. 2021. https://doi.org/10.21437/Interspeech.2021-1094

[9] Al-Barham, Muhammad, Adham Alsharkawi, Musa Al-Yaman, Mohammad Al-Fetyani, Ashraf Elnagar, Ahmad Abu SaAleek, and Mohammad Al-Odat. "RGB Arabic Alphabets Sign Language Dataset." *arXiv preprint arXiv:2301.11932* (2023). https://doi.org/10.48550/arXiv.2301.11932

[10] Tao, Wenjin, Ming C. Leu, and Zhaozheng Yin. "American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion." *Engineering Applications of Artificial Intelligence* 76 (2018): 202-213. https://doi.org/10.1016/j.engappai.2018.09.006

[11] Hasanov, Jamaladdin, Nigar Alishzade, Aykhan Nazimzade, Samir Dadashzade, and Toghrul Tahirov. "Development of a hybrid word recognition system and dataset for the Azerbaijani Sign Language dactyl alphabet." *Speech Communication* 153 (2023): 102960. https://doi.org/10.1016/j.specom.2023.102960

[12] Tharwat, Gamal, Abdelmoty M. Ahmed, and Belgacem Bouallegue. "Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques." *Journal of Electrical and Computer Engineering* 2021 (2021): 2995851. https://doi.org/10.1155/2021/2995851

[13] Camgoz, Necati Cihan, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. "Neural Sign Language Translation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7784-7793. 2018. https://doi.org/10.1109/CVPR.2018.00812

[14] Luqman, Hamzah, and Sabri A. Mahmoud. "A machine translation system from Arabic sign language to Arabic." *Universal Access in the Information Society* 19, no. 4 (2020): 891-904. https://doi.org/10.1007/s10209-019-00695-6

[15] Camgoz, Necati Cihan, Oscar Koller, Simon Hadfield, and Richard Bowden. "Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10023-10033. 2020. https://doi.org/10.1109/CVPR42600.2020.01004

[16] Park, Chan-Il, and Chae-Bong Sohn. "Data augmentation for human keypoint estimation deep learning based sign language translation." *Electronics* 9, no. 8 (2020): 1257. https://doi.org/10.3390/electronics9081257

[17] Natarajan, B., Rajalakshmi Elangovan, R. Elakkiya, Ketan Kotecha, Ajith Abraham, Lubna Abdel Kareim Gabralla, and V. Subramaniyaswamy. "Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation." *IEEE Access* 10 (2022): 104358-104374. https://doi.org/10.1109/ACCESS.2022.3210543

[18] AbdElghfar, Hany A., Abdelmoty M. Ahmed, Ali A. Alani, Hammam M. AbdElaal, Belgacem Bouallegue, Mahmoud M. Khattab, Gamal Tharwat, and Hassan A. Youness. "A Model for Qur'anic Sign Language Recognition Based on Deep Learning Algorithms." *Journal of Sensors* 2023 (2023): 9926245. https://doi.org/10.1155/2023/9926245

[19] Latif, Ghazanfar, Nazeeruddin Mohammad, Jaafar Alghazo, Rami AlKhalaf, and Rawan AlKhalaf. "ArASL: Arabic Alphabets Sign Language Dataset." *Data in Brief* 23 (2019): 103777. https://doi.org/10.1016/j.dib.2019.103777

[20] Al-Qaisy, Noor E., Bilal R. Al-Kaseem, and Yousif Al-Dunainawi. "AI-Based Portable Gesture Recognition System for Hearing Impaired People Using Wearable Sensors." In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, 33-38. 2023. https://doi.org/10.1109/DeSE58274.2023.10099999

[21] Elshareif, Mohamed Elgilani Eltahir, Nurul Aswani Ahmad Alias, Nazean Jomhari, and Hazrina Sofian. "Smart Glove with Mobile Application to Detect Static Arabic Hijaiyah Hand Code for Quran Recitation." In *2024 21st International Learning and Technology Conference (L&T)*, 64-69. 2024. https://doi.org/10.1109/LT60077.2024.10469054

[22] Adeyanju, Ibrahim A., Opeyemi O. Bello, and Muhammed A. Adegboye. "Machine learning methods for sign language recognition: A critical review and analysis." *Intelligent Systems with Applications* 12 (2021): 200056. https://doi.org/10.1016/j.iswa.2021.200056

[23] Rodriguez-Correa, Paula Andrea, Alejandro Valencia-Arias, Oscar N. Patino-Toro, Yris Oblitas Diaz, and Roberto Teodori De la Puente. "Benefits and development of assistive technologies for Deaf people's communication: A systematic review." *Frontiers in Education* 8 (2023): 1121597. https://doi.org/10.3389/feduc.2023.1121597

[24] AbdElghfar, Hany, Hassan A. Youness, Mohamed Wahba, and Hammam M. Abdelaal. "An automated framework for Qur'anic education of the hearing-impaired using body pose classification and Arabic sign language integration." *Scientific Reports* 16 (2026): 5939. https://doi.org/10.1038/s41598-026-36578-z

[25] Dzulkifli, Izuli. "Quranic Education for Deaf Students in Malaysia: Implementation and Challenges." *Nadwa: Jurnal Pendidikan Islam* 16, no. 1 (2022): 1-16. https://doi.org/10.21580/nw.2022.16.1.10823

[26] Seman, Nik Hassan, Muhammad Ridzuam Idris, Mohd Iskandar Daud, Nik Azhar Nik Ab. Rahman, and Mohd Nazri Abdul Rahman. "Teaching Al-Quran to Deaf Students: Challenges for Islamic Education Teachers." *Online Journal of Islamic Education* 7, no. 1 (2019): 46-53.

[27] Ghadim, Nafiseh Alaghehband, Nazean Jomhari, Norlidah Alias, Syar Meeze Mohd Rashid, and Mohd Yakub Zulkifli Bin Mohd Yusoff. "Mother's Perspective Toward al-Quran Education for Hearing Impaired Children in Malaysia."

Malaysian Online Journal of Educational Technology 1, no. 4 (2013): 26-30. https://doi.org/10.52380/mojet.2013.1.4.23

[28] AbdElghfar, Hany A., Abdelmoty M. Ahmed, Ali A. Alani, Hammam M. AbdElaal, Belgacem Bouallegue, Mahmoud M. Khattab, and Hassan A. Youness. "QSLRS-CNN: Qur'anic sign language recognition system based on convolutional neural networks." *The Imaging Science Journal* 72, no. 2 (2024): 254-266. https://doi.org/10.1080/13682199.2023.2202576

[29] Ben Haj Amor, Amina, Oussama El Ghoul, and Mohamed Jemni. "Deep learning approach for sign language's handshapes recognition from EMG signals." In *2022 IEEE Information Technologies & Smart Industrial Systems (ITSIS)*. 2022. https://doi.org/10.1109/ITSIS56166.2022.10118359

[30] Nasef, Mohammed M., Aya El-Sayed, and Eman M. AbouNassar. "ArSL-TGRU: A Hybrid Model for Signer-Independent Arabic Sign Language Recognition from Videos Based on Transformer and GRU." *International Journal of Computational Intelligence Systems* 19 (2026): 40. https://doi.org/10.1007/s44196-025-01120-y

[31] Pal, Abhilash, Stephan Huber, Cyrine Chaabani, Alessandro Manzotti, and Oscar Koller. "On the Importance of Signer Overlap for Sign Language Detection." *arXiv preprint arXiv:2303.10782* (2023). https://doi.org/10.48550/arXiv.2303.10782

[32] Bantupalli, Kshitij, and Ying Xie. "American Sign Language Recognition using Deep Learning and Computer Vision." In *2018 IEEE International Conference on Big Data (Big Data)*, 4896-4899. 2018. https://doi.org/10.1109/BigData.2018.8622141

[33] Ibrahim, Nada B., Mazen M. Selim, and Hala H. Zayed. "An Automatic Arabic Sign Language Recognition System (ArSLRS)." *Journal of King Saud University - Computer and Information Sciences* 30, no. 4 (2018): 470-477. https://doi.org/10.1016/j.jksuci.2017.09.007

[34] Rastgoo, Razieh, Kourosh Kiani, and Sergio Escalera. "Sign Language Recognition: A Deep Survey." *Expert Systems with Applications* 164 (2021): 113794. https://doi.org/10.1016/j.eswa.2020.113794

[35] Zhang, Fan, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. "MediaPipe Hands: On-device Real-time Hand Tracking." *arXiv preprint arXiv:2006.10214* (2020). https://doi.org/10.48550/arXiv.2006.10214

[36] Meng, Yuting, Haibo Jiang, Nengquan Duan, and Haijun Wen. "Real-Time Hand Gesture Monitoring Model Based on MediaPipe's Registerable System." *Sensors* 24, no. 19 (2024): 6262. https://doi.org/10.3390/s24196262

[37] Gil-Martin, Manuel, Marco R. Marini, Ivan Martin-Fernandez, Sergio Esteban-Romero, and Luigi Cinque. "Hand Gesture Recognition Using MediaPipe Landmarks and Deep Learning Networks." In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART*, 24-30. 2025. https://doi.org/10.5220/0013053500003890

[38] Schweihoff, Jens F., Matvey Loshakov, Irina Pavlova, Laura Kuck, Laura A. Ewell, and Martin K. Schwarz. "DeepLabStream enables closed-loop behavioral experiments using deep learning-based markerless, real-time posture detection." *Communications Biology* 4 (2021): 130. https://doi.org/10.1038/s42003-021-01654-9