Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
19/1 (2026), 63-75. DOI: http://dx.doi.org/10.21609/jiki.v19i1.1600

Continuous Sign Language Recognition for Quranic Recitation by Deaf People Using Deep Learning

Yulrio Brianorman[1,2], Rinaldi Munir[1], Nur Ulfa Maulidevi[1]

[1] School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia
[2] Faculty of Engineering and Computer Science, University of Muhammadiyah Pontianak, Pontianak, Indonesia

Email: y.brianorman@unmuhpnk.ac.id, rinaldi@staff.stei.itb.ac.id, ulfa@itb.ac.id

Abstract

This study proposes a deep learning-based system for recognizing Quranic recitation in the sign language, aimed at enhancing accessibility for the Deaf Muslim community. A central contribution is the construction of a novel dataset comprising videos from three Deaf signers performing Surah Al-Fatihah and Surah Al-Ikhlas, guided by the 2022 official Quranic sign language standard introduced by Indonesia's Ministry of Religious Affairs. The recognition task is framed as a continuous sign language recognition (CSLR) problem to handle unsegmented input sequences. Five pre-trained convolutional neural networks—EfficientNet, GoogleNet, MobileNetV2, ResNet18, and ShuffleNet—were evaluated as visual feature extractors. These were followed by a temporal encoder composed of 1D CNN and BiLSTM, with sequence alignment performed using the Connectionist Temporal Classification (CTC). The experimental results show that ResNet18 and MobileNetV2 achieved the best performance with Word Error Rates (WER) of 5.00% and 7.92% on the test set, respectively. A cross-participant evaluation was also conducted to assess model generalization, although the results revealed performance gaps likely due to signer variation and limited data. The study highlights the suitability of lightweight and residual architectures for CSLR tasks in religious contexts and provides a benchmark for future research on inclusive sign language technologies. In cross-participant evaluation, the model achieved a validation WER of 8.44% on seen signers and 50.46% on an unseen signer, reflecting generalization challenges commonly observed in low-resource CSLR settings. The proposed system lays the groundwork for AI-assisted Quranic education tools tailored to the Deaf Muslim population.

Keywords: *Al-Quran Sign Language, CNN, BiLSTM, Continuous Sign Language Recognition (CSLR), Connectionist Temporal Classification (CTC), MobileNetV2, ResNet18*

1. Introduction

Access to spiritual knowledge is a fundamental aspect of human life, including for the Deaf community, which relies on sign language as its primary medium of communication. However, Deaf individuals often face challenges in accessing information, particularly in spiritual contexts such as reading the Quran. In Indonesia, deaf individuals account for approximately 0.11% of the total population [1], indicating that around 297,000 out of 270 million people experience hearing loss. Assuming that 80% of them are Muslims, approximately 237,000 Deaf Muslims require better access to understand and recite the Quran. Nevertheless, access to the Quran among the Deaf community remains highly limited. Therefore, the development of the Quranic sign language recognition technology is crucial for improving the quality of life and spiritual participation in society.

Similar to spoken languages, sign languages naturally develop among Deaf individuals. Each country, region, and even specific Deaf community may have its own independently developed sign language.

Each sign language has its own grammar and rules, with distinct forms and gestures that can be visually interpreted. In Indonesia, there are two main sign languages: *Sistem Isyarat Bahasa Indonesia* (SIBI) and *Bahasa Isyarat Indonesia* (BISINDO). The official sign language used in schools is SIBI [2], whereas BISINDO is more commonly used by the Deaf community in their daily communication [3]. Additionally, the Indonesian government has developed a specific sign language for Quranic recitation, which plays a crucial role in conveying spiritual values to Deaf Muslims. The Ministry of Religious Affairs of the Republic of Indonesia officially introduced the Quranic sign language in 2022.

Quranic sign language helps Deaf Muslims understand and recite sacred texts more effectively. As part of an inclusivity initiative, the Ministry of Religious Affairs has developed methods to expand Quranic accessibility for the Deaf community. The Quranic sign language guidelines can be accessed online at https://bit.ly/4k693MZ. However, the implementation of technology to support the learning and recognition of Quranic sign language remains very limited. Therefore, this study aims to address these accessibility challenges by developing a sustainable Quranic sign language recognition system using a deep learning approach.

One possible way to break down communication barriers between the Deaf and the general public is through sign language recognition technology. Studies in this field have been ongoing since 1983, with multiple approaches being continually refined. Presently, Deep Learning (DL) methods are employed to tackle the intricacies of visual analysis of hand movements and the multimodal properties of sign language [4]. DL, with its ability to process spatial and temporal data, has become the preferred method for Continuous Sign Language Recognition (CSLR).

Research on sign and hand gesture recognition generally falls into two categories: Isolated Sign Language Recognition (ISLR) and Continuous Sign Language Recognition (CSLR) [5]. In ISLR, the system recognizes individual hand gestures, typically representing single words. In contrast, CSLR focuses on sequences of gestures to form complete sentences. In the context of Quranic sign language recognition, CSLR is more relevant because Quranic recitation involves complex and continuous sequences of gestures.

CSLR models are typically categorized into three approaches [6]: classical models, Hidden Markov Model (HMM)-based models, and Deep Learning (DL)-based models. Before 2015, most CSLR models relied on HMMs [7] and sensor-based devices for data collection. However, DL-based models have since demonstrated greater effectiveness in handling the spatial and temporal complexities of sign language, particularly in continuous settings.

DL-based CSLR systems generally comprise three core components: a visual feature extractor, a sequence modeling module, and an alignment mechanism. Convolutional Neural Networks (CNNs) are used to extract spatial representations from video frames. The sequence modeling module—such as BiLSTM, 1D CNNs, or Transformers—is employed to capture temporal dependencies across frames [8]. Lastly, the alignment module synchronizes the temporal visual sequences with the corresponding textual labels.

One of the main challenges in CSLR is the accurate alignment of time-series image sequences with text labels. A common method used to resolve this issue is Connectionist Temporal Classification (CTC), which matches the visual input with target labels without needing detailed frame-level annotation [9]. CTC effectively deals with problems like overfitting and the scarcity of labels in continuous data.

Despite significant advancements in sign language recognition, research specifically focused on Quranic sign language remains very limited. This is primarily due to the absence of a publicly available, comprehensive Quranic sign language dataset. The Indonesian government has recently introduced a standard for sign language based on the Quran, necessitating further research into the most suitable deep learning framework that can efficiently process its specific semantic and temporal characteristics.

This study aims to fill this research gap by developing a new Quranic sign language dataset and evaluating several deep learning architectures for their effectiveness in Quranic sign recognition. The dataset comprises video recordings of three Deaf participants signing Surah Al-Fatihah and Surah Al-Ikhlas. Each video was processed into standard-resolution frames and annotated at the verse level.

The proposed recognition system employs the use of pre-trained CNN backbones, including EfficientNet, GoogleNet, MobileNetV2, ResNet18, and ShuffleNet, for the extraction of visual features; BiLSTM is used for temporal modeling, and CTC as the alignment loss. The findings from this study are expected to support the inclusivity of the Muslim Deaf community and enhance the role of deep learning technology in video-based Quranic sign language recognition.

2. Background

2.1. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning architecture designed to handle data with a grid-like structure, like images or videos. Convolutional neural networks use convolutional layers to extract local features from input data, such as edges, textures, and specific patterns. Following convolutional layers, pooling layers are applied, thereby decreasing data dimensionality while maintaining vital information [10]. Convolutional neural networks have been proven highly effective in visual recognition tasks, including image-based sign language recognition. A general 2D CNN process is depicted in Figure 1.

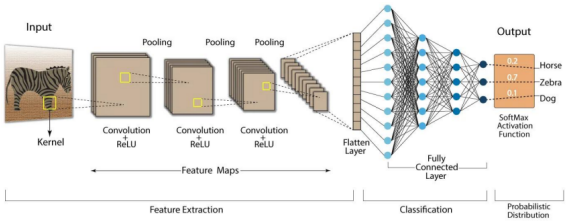


Figure 1. 2D CNN process flow [10]

2.2. Bidirectional Long Short-Term Memory (BiLSTM)

A modified version of the Recurrent Neural Network (RNN) is the Bidirectional Long Short-Term Memory (BiLSTM) type, which incorporates an LSTM architecture. A bidirectional RNN is comprised of two separate RNN layers, each of which processes input sequences in both forward and backward directions independently. Figure 2 depicts the configuration of a bidirectional RNN, with x denoting the input, y representing the output, t signifying the time step, and h describing the hidden state. $t-1$ and $t+1$ also represent the previous and next time steps, respectively [11].

2.3. Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification (CTC) is a loss function designed for sequence-to-sequence learning tasks where the alignment between input and output sequences is unknown [8]. It enables the model to learn mappings from input sequences $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ to target label sequences

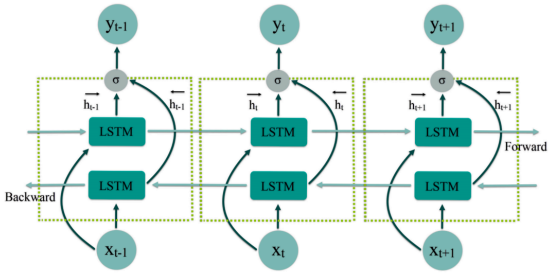


Figure 2. BiLSTM architecture across three sequential steps [11]

$\mathbf{l} = (l_1, l_2, \ldots, l_U)$ without requiring frame-level annotations, where typically $U \leq T$.

To accommodate alignment variability, CTC introduces an extended label set $L' = L \cup \{\text{blank}\}$, where L is the set of all possible labels (e.g., glosses), and 'blank' is a special symbol that represents no label output at a timestep. The softmax output layer of the model produces a probability distribution $\mathbf{y}_t = (y_t^k)_{k \in L'}$ for each time step $t \in \{1, \ldots, T\}$.

Let $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ be a path through the output space $(L')^T$. The probability of a specific path π given input $\mathbf{x}$ is:

$$P(\pi|\mathbf{x}) = \prod_{t=1}^{T} y_t^{\pi_t} \tag{1}$$

A many-to-one collapsing function $B(\pi)$ is defined, which removes repeated labels and blanks from π to form a valid output sequence. For example, $B(\text{-}a\text{-}ab\text{-}) = B(\text{-}a\text{-}abb) = \text{aab}$.

The conditional probability of the target label sequence $\mathbf{l}$ given input $\mathbf{x}$ is the sum over all valid paths π that map to $\mathbf{l}$:

$$P(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in B^{-1}(\mathbf{l})} P(\pi|\mathbf{x}) \tag{2}$$

The model is trained by maximizing this probability over the training set. This is efficiently computed using a dynamic programming approach based on the forward-backward algorithm.

2.4. Transfer Learning

A technique called transfer learning involves reusing a model initially trained on a vast dataset, for example ImageNet, as a foundation for tackling a more specific task. This study utilized pre-trained CNN models, such as EfficientNet, MobileNet, and

ResNet18, to extract visual features. These models efficiently extract visual features from images or video frames, thereby minimising the need for extensive abrasive training. The study employed a set of five pre-trained CNNs, namely EfficientNet, GoogleNet, MobileNetV2, ResNet18, and ShuffleNet, to extract spatial features from the input data. Each model has distinct architectural features and advantages in image processing.

2.4.1. EfficientNet. The EfficientNet architecture offers a compound scaling method that strikes a balance between network depth, width, and input resolution to enhance performance, while minimising the number of parameters [12].

2.4.2. GoogleNet. GoogleNet (Inception-V1) was introduced in 2015 and features the Inception module, enabling multiple parallel applications of convolution filters (1×1, 3×3, 5×5) for multi-scale feature extraction [13, 14].

2.4.3. MobileNetV2. MobileNetV2, introduced in 2018, is optimized for mobile and embedded devices. It utilises depthwise separable convolutions, inverted residuals, and linear bottlenecks to decrease computational costs while maintaining feature quality [15, 16].

2.4.4. ResNet18. Introduced in 2016, ResNet18 is a residual network architecture. It employs skip connections (residuals) to resolve vanishing gradient issues in deep architectures, thus improving training efficiency for moderate-sized datasets [17, 18].

2.4.5. ShuffleNet. ShuffleNet is a lightweight architecture for CNNs that was introduced in 2018 for low-power devices. It uses pointwise group convolution and channel shuffle operations to maintain high accuracy at a low computational cost [19, 20].

3. Related Works

3.1. Continuous Sign Language Recognition

Continuous Sign Language Recognition (CSLR) has evolved significantly with the advancement of deep learning. Early studies typically employed CNN-RNN-CTC pipelines [21, 22], leveraging spatial and temporal features for gloss-level prediction. While effective, these approaches often suffer from limitations in visual-semantic alignment and temporal modeling.

Recent research has suggested the development of more advanced architectures to overcome difficulties in sign language recognition. A method integrates spatial, temporal, and skeletal cues within a multi-branch network to enhance feature representations [23]. The model introduces a visual-lexical alignment constraint (VLAC) to bridge the semantic gap between video frames and gloss embeddings [24]. A further refinement incorporates an auto-reinforcing mechanism that selectively amplifies temporal frames to enhance short-term context modelling [25].

A spatial-temporal enhanced network was proposed to refine feature extractors, incorporating a spatial-visual alignment module and a temporal feature difference module to enhance the distinction of hand and facial features, as well as motion dynamics [26]. An alternative approach used self-supervised pretraining based on hand pose estimation to provide domain-relevant priors for continuous sign language recognition tasks [27].

This direction involves learning joint representations for both recognition and translation tasks. A contrastive visual-textual learning method featuring variational alignment has been used to integrate gloss and sentence-level semantics [28]. A conditional variational autoencoder referred to as CV-SLT was also proposed, incorporating dual KL divergences and attention-based residual modeling to facilitate sign language translation without the need for gloss supervision [29].

Despite these innovations, CSLR remains underexplored in low-resource and domain-specific contexts. Most studies focus on benchmark datasets such as PHOENIX14 or CSL-Daily, with limited generalization to Arabic or Quranic sign domains. Our work addresses this gap by introducing a Quranic CSLR dataset and evaluating lightweight backbones under signer-dependent settings.

3.2. CSLR for Quranic Sign Language

The study titled ”Arabic Sign Language Recognition System for Letters Using Machine Learning Techniques” developed an Arabic Alphabet Sign Language Recognition System (AArSLRS) using a vision-based approach. The dataset consisted of 9,240 images representing 28 Arabic letter letters, collected from 10 signers with variations in age, hand size, angle (ranging from 60° to -60°), and background complexity. The data were categorized into three types: bare hands with a dark background, bare hands with a bright background, and gloved hands. The methodology included four main stages: image acquisition, preprocessing (grayscale conversion, segmentation using global thresholding or Sobel filtering, and noise removal), feature extraction (15 shape-based features such as centroid ratio

and perimeter), and classification using supervised learning algorithms (KNN, C4.5, Naive Bayes, and MLP), with KNN achieving the highest accuracy. The results demonstrated hand detection accuracy up to 98.64% (for gloved datasets) and an alphabet recognition accuracy of 99.5% using KNN with cityblock distance measurement, as well as 97.548% for the recognition of 14 muqatta'at letters found in Quranic surahs. While effective for educational purposes among the Deaf community, the system remains signer-dependent [30].

The follow-up research adopted a CNN-based method comprising preprocessing (grayscale conversion and normalization), feature extraction, and classification with resampling techniques (SMOTE, RMO, and RMU) to address data imbalance. The results indicated a training accuracy of 98.75% and a testing accuracy of 97.31% over 200 epochs without resampling, which improved to 97.79% with the application of SMOTE [31].

However, both studies were limited to recognizing individual Arabic sign language letters and did not address continuous sign language recognition (CSLR) for full Quranic verses or surahs.

Based on the reviewed literature, it can be concluded that CSLR for the Arabic sign language differs from CSLR for the Quranic sign language due to differences in context, vocabulary, and signing structure. CSLR in the Quranic context typically involves character-by-character representation of Quranic verses, whereas Arabic CSLR focuses more on general word or sentence recognition. Moreover, existing CSLR research on the Quranic sign language remains scarce and is mostly restricted to recognizing isolated hijaiyah letters rather than full verses represented in continuous signing videos.

Therefore, this study focuses on developing a CSLR system specifically for the Quranic sign language, employing a novel approach distinct from prior work.

4. Research Method

This study was conducted through five main stages: (1) dataset collection, (2) dataset preprocessing, (3) system architecture design, (4) model training, and (5) model evaluation. The overall workflow is illustrated in Figure 3.

4.1. Dataset Collection

The dataset was collected from three deaf individuals who had memorized the Quran using sign language. Each participant signed Surah Al-Fatihah and Surah Al-Ikhlas ten times, resulting in 361 video recordings. The decision to repeat each surah multiple times was made to capture variations in gesture execution, signing speed, and temporal alignment. Although the signers and verses remained the same, natural differences in each repetition provided valuable diversity in motion and timing, thereby enhancing model robustness.

The participants were selected on the basis of their fluency and accuracy in the Quranic sign language, in accordance with the standardized gesture guidelines issued by the Indonesian Ministry of Religious Affairs. This ensured the authenticity and consistency of the signing style.

Recordings were conducted using a Logitech webcam (720p resolution), positioned approximately 1 m from the signer. Videos were captured using OBS Studio at a fixed frame rate of 30 frames per second and exported at a standardized resolution of 480×480 pixels. Additional lighting was employed to minimize shadows and improve visual clarity.

After collection, the videos were segmented into frame sequences. Each frame was resized to 256×256 pixels to maintain consistency across the dataset. The frame sequences were organized into directories labeled by signer ID, surah name, and verse number.

The number of frames per video varies depending on the signer and the timing of each repetition, typically ranging from 150 to 633 frames per verse (5 to 22 seconds per video at 30 fps). Hence, the input length is variable across samples. No frame truncation or padding was applied during preprocessing; instead, the model operates on variable-length input sequences, which are handled naturally by the CTC-based architecture.

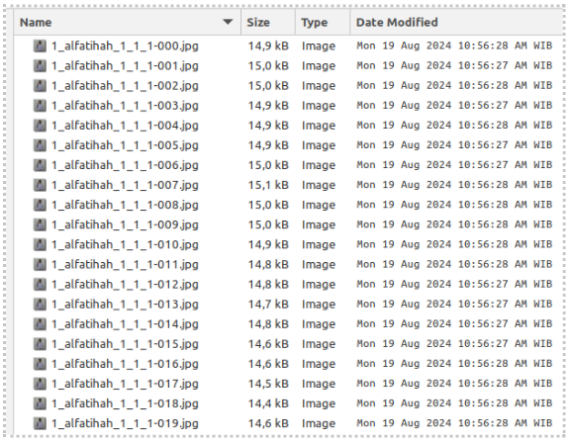
A weak-labeling approach was adopted for the annotation. Each video corresponds to a single Quranic verse and was labeled using the complete transcription of that verse. Frame-level segmentation or gloss alignment was not performed; the entire video sequence was treated as a temporal representation of the verse-level gloss. This strategy aligns with prior work in CSLR that relies on weak supervision to avoid costly frame-level annotation [22].

All 361 videos were used in this study. For model training and evaluation, the dataset was split into training, validation, and test subsets. These splits were constructed to assess the model's ability to generalize across repetitions and potentially across participants. The details of the data split strategy are described in the section evaluation.

The Quranic sign language dataset is publicly accessible at https://quranisyarat.org, and a sample is shown in Figure 4.

Figure 3. Flowchart of the research methodology

Figure 4. Example of a Quranic sign language video on the quranisyarat.org website



Name	Size	Type	Date Modified
1_alfatihah_1_1_1-000.jpg	14,9 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-001.jpg	15,0 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-002.jpg	15,0 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-003.jpg	14,9 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-004.jpg	14,9 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-005.jpg	14,9 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-006.jpg	15,0 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-007.jpg	15,1 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-008.jpg	15,0 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-009.jpg	15,0 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-010.jpg	14,9 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-011.jpg	14,8 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-012.jpg	14,8 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-013.jpg	14,7 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-014.jpg	14,8 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-015.jpg	14,6 kB	Image	Mon 19 Aug 2024 10:56:27 AM WIB
1_alfatihah_1_1_1-016.jpg	14,6 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-017.jpg	14,5 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-018.jpg	14,4 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB
1_alfatihah_1_1_1-019.jpg	14,6 kB	Image	Mon 19 Aug 2024 10:56:28 AM WIB

Figure 5. Sample frames from the preprocessing stage.

4.2. Dataset Preprocessing

Each video segment was converted into a frame sequence using a Python script. The frames were stored in structured folders and named sequentially for ease of identification. An example of the preprocessing output is shown in Figure Figure 5.

Table 1 summarizes the frame count distribution for each verse recorded by all three signers in the dataset. The table includes surah names (Al-Fatihah and Al-Ikhlas), specific verse numbers, transliterated text labels, and the number of frames per sample.

As seen in the table, the number of frames varies significantly across different verses and signers. For example, Surah Al-Fatihah verse 7b has the longest average duration, especially for signer-1 who recorded up to 633 frames, whereas shorter verses like Al-Ikhlas verse 2 contain as few as 150–161 frames.

This variability confirms the need for sequence models capable of handling variable-length inputs. It also reflects the natural differences in the signing speed and gesture complexity among the signers. These insights are essential for developing robust CSLR systems and justify the use of CTC-based architectures that do not require fixed input lengths.

Additionally, the average number of frames per video across the dataset was 285 (see Table 2), with frame lengths varying between 150 and 633. This reinforces the importance of choosing a model architecture that can robustly process variable-length input sequences.

4.3. CSLR Design

The proposed CSLR architecture comprises three main components: (1) a visual feature extractor (2D CNN), (2) a temporal encoder (1D CNN + BiLSTM), and (3) a sequence alignment module (CTC). Figure 6 illustrates the complete architecture.

Five pre-trained CNN backbones were evaluated: EfficientNet, GoogleNet, MobileNetV2, ResNet18, and ShuffleNet. Each CNN was used to extract spatial features from individual RGB frames, resulting in a sequence of *frame features* with shape (T, D_f), where T is the number of frames and D_f is the CNN output dimension (e.g., 512 or 1024, depending on the backbone).

To capture the local temporal context between adjacent frames, a 1D convolutional layer was applied across the time axis. This produces a sequence of *visual features* with shape (T', D_v), where $T' < T$ due to temporal downsampling and D_v denotes the visual embedding dimension (e.g., 256). The 1D CNN helps encode micro-gestures and smooth short-term variations that may occur between consecutive frames.

These visual features are then passed to a two-layer Bidirectional Long Short-Term Memory (BiLSTM) network with 512 hidden units in each direction, resulting in a contextualized temporal representation of shape $(T', 1024)$. This BiLSTM captures long-range dependencies across the entire sequence,

Table 1. Detailed frame count for each verse and signer in the dataset.

Surah	Verse	Label (Transliteration)	Signer-1	Signer-2	Signer-3
Al-Fatihah	1	bi s mi allā hi rra ḥ mā ni rra ḥī m	326	280	304
Al-Fatihah	2	al ḥa m du li llā hi ra bbi l ’ā la mī n	422	322	311
Al-Fatihah	3	a rra ḥ mā ni rra ḥī m	209	178	203
Al-Fatihah	4	mā li ki ya u mi ddī n	214	184	187
Al-Fatihah	5	i yyā ka na ’ bu du wa i yyā ka na s ta ’ī n	471	358	316
Al-Fatihah	6	i h di na ṣṣi rā ṭa l mu s ta qī m	355	299	261
Al-Fatihah	7a	ṣi rā ṭa lla ẓī na a n ’a m ta ’a la i hi m	425	376	280
Al-Fatihah	7b	ga i ri l ma g ḍū bi ’a la i hi m wa la ḍḍā llī n	633	521	544
Al-Ikhlas	1	qu l hu wa allā hu a ḥa d	240	241	201
Al-Ikhlas	2	a llā hu ṣṣa ma d	161	154	150
Al-Ikhlas	3	la m ya li d wa la m yū la d	242	270	215
Al-Ikhlas	4	wa la m ya ku l la hū ku fu wa n a ḥa d	332	336	268

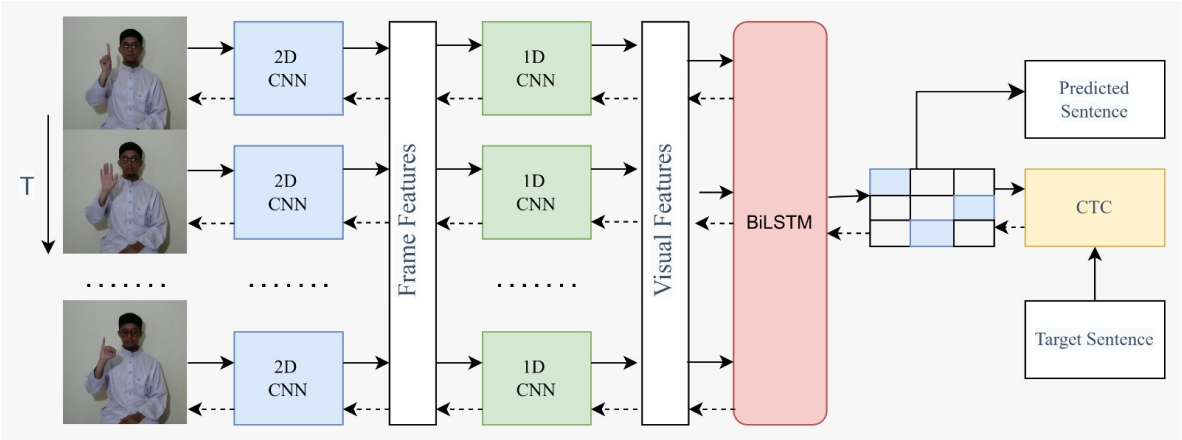


Figure 6. Proposed CSLR architecture for Quranic sign language

Table 2. Summary of frame counts across all verse-level videos.

Statistic	Value
Total Videos	108
Minimum Frames	140
Maximum Frames	633
Average Frames	285
Frame Rate	30 fps
Input Length Type	Variable

enabling the model to interpret sign transitions and temporal dynamics effectively.

The output of the BiLSTM is passed through a linear projection layer, resulting in logits with shape (T', V), where V is the vocabulary size. These logits were optimized using the connectionist temporal classification (CTC) loss function, which enables sequence-to-sequence alignment without requiring frame-level annotations.

Throughout the architecture, the input and output dimensions evolve as follows: the model receives an input sequence of T RGB frames of size $256 \times 256 \times 3$, which are encoded into frame-level features of shape (T, D_f). After the temporal convolution layer, the sequence becomes (T', D_v), which is then transformed by the BiLSTM into $(T', 1024)$. Finally, the linear projection yields output logits of shape (T', V), which are decoded into the final predicted sequence.

Each input video corresponds to one Quranic verse and is annotated with a tokenized transliteration of that verse. This sequence serves as the target during training. The number of input frames and output tokens are both variable, and the CTC loss allows for flexible alignment between them.

The choice of employing a combination of 1D CNN and BiLSTM is driven by its effectiveness in recent CSLR architectures [23, 24]. A 1D CNN acts as a short-term temporal encoder, capturing local motion patterns and reducing frame-level noise before forwarding the processed information to the BiLSTM. This lessens the load on the BiLSTM to learn both local and global temporal dependencies at the same time. Experimen-

tally, this hierarchical method stabilizes the training process and enhances convergence rates, especially in low-resource environments. Preliminary trials showed that BiLSTM-only models achieved less stable WER performance. Our study centered on assessing lightweight recurrent encoder models capable of handling small datasets and low-latency applications, despite the potential of transformer-based models. Among the evaluated architectures, the ResNet18 backbone combined with 1D CNN + BiLSTM was empirically selected as the final model due to its consistent convergence behavior and superior WER performance during within-participant training. ResNet18 was chosen due to its stability during training and ability to model temporal patterns when combined with a 1D CNN and BiLSTM.

4.4. Training Process

The dataset was divided into three subsets: 80% for training, 10% for validation, and 10% for testing. Each input comprised a variable-length sequence of RGB frames corresponding to a Quranic verse, paired with a transliterated sentence-level label.

To mitigate the limited size and signer diversity of the dataset, we initially experimented with simple data augmentation techniques such as horizontal flipping, random cropping, and brightness jittering. However, these did not yield consistent improvements in the WER and were thus excluded from the final training pipeline. More systematic augmentation strategies (e.g., signer simulation and temporal warping) are proposed for future work.

All experiments were conducted on a high-performance computing server running Ubuntu 20.04, equipped with dual Intel(R) Xeon(R) CPUs (80 threads total), 503 GB of RAM, and eight NVIDIA Tesla V100-SXM2 GPUs, each with 32 GB VRAM. The training was executed on a single GPU using CUDA 11.8, cuDNN 8.x, and PyTorch 2.4.0 with CUDA.

Hyperparameter selection followed common practices in the continuous sign language recognition (CSLR) literature [32]. Due to resource constraints, no extensive tuning was performed. The learning rate and batch size were determined empirically from the early convergence behavior. We employed a step decay learning rate schedule and utilised a batch size of 1 to maintain frame sequence consistency. Training was conducted under supervision, utilising the Connectionist Temporal Classification (CTC) loss, with the decoder implementing a beam search. No frame-level normalisation or temporal padding was applied during the training process.

Training hyperparameters are summarized in Table 3.

Table 3. Training hyperparameters.

No	Hyperparameter	Value
1.	Optimizer	Adam
2.	Learning Rate	0.0001
3.	Learning Rate Schedule	StepLR (20, 35)
4.	Weight Decay	0.0001
5.	Epochs	25
6.	Batch Size (Train/Test)	1 / 4
7.	Loss Function	CTC
8.	Decoder	Beam Search
9.	Temporal Encoder	1D CNN + BiLSTM
10.	BiLSTM Hidden Units	512 × 2
11.	Normalization	BatchNorm (enabled)
12.	Frame Transformations	None

4.5. Evaluation Metrics

Model performance was assessed using the Word Error Rate (WER), a widely used metric in sequence recognition tasks such as speech and sign language [8, 22]. WER evaluates the similarity between a predicted sequence and its reference by computing the minimum number of substitutions (S), deletions (D), and insertions (I) required to transform one into the other:

$$\text{WER} = \frac{S + D + I}{N} \times 100\% \tag{3}$$

where N is the total number of words in the reference sequence. A lower WER indicates higher accuracy and better alignment with the intended label.

Unlike standard classification accuracy, WER captures both structural and sequential errors, which are critical in continuous sign language recognition (CSLR). For instance, omitting a token (deletion), repeating a word incorrectly (insertion), or replacing one phrase with another (substitution) will affect the score differently.

In this work, all evaluations were performed at the word level. The metric is particularly well-suited to the Quranic sign language task, where each video corresponds to a complete transliterated verse. In addition to reporting the overall WER, we include the breakdown of S, D, and I to provide deeper insight into the model's typical failure modes—such as visual ambiguity, temporal misalignment, or over-segmentation.

5. Results and Discussion

This section presents the results of model training and testing using five pretrained CNN architectures for feature extraction, followed by BiLSTM

and CTC for sequence modeling and alignment. The discussion is structured into two subsections: training/validation performance and testing outcomes.

5.1. Training and Validation Results

Table 4 and Figure 7 summarize the training performance of the five architectures across 25 epochs, evaluated on validation data using the WER. Among the models, ResNet18 and MobileNetV2 consistently demonstrated superior performance with the lowest WER values of 2.73% and 4.09%, respectively.

ResNet18 benefits from residual connections that improve the gradient flow and stabilize learning across epochs. MobileNetV2 also showed promising results because of its use of depthwise separable convolutions, which enhance feature extraction while maintaining computational efficiency.

In contrast, EfficientNet, GoogleNet, and ShuffleNet exhibited less stable training and higher WER values. Although EfficientNet is designed for parameter optimization, it underperformed in this context—possibly due to overfitting or incompatibility with the dataset's temporal complexity.

A notable spike in WER is often seen in specific epochs, including epoch 17. This occurrence is frequently seen in CTC-based models and has been examined in earlier research [33]. During training, the Connectionist Temporal Classification (CTC) loss frequently promotes the model to produce high-confidence blank frames where gloss boundaries are unclear. Misalignments in the predicted path can cause significant jumps in word error rate, particularly when the model switches between different alignment techniques. These spikes typically settle down in later stages and don't necessarily indicate long-term deterioration.

These findings guided the selection of ResNet18 and MobileNetV2 as candidates for final testing on unseen data.

5.2. Testing Results

The final models were assessed on previously unexamined test data. The table, as shown in reference 5, reports the WER for each CNN backbone, with ResNet18 producing the best results at a WER of 5.00%, and MobileNetV2 achieving a WER of 7.92%. The greatest error was recorded by ShuffleNet at 13.64%.

The superior performance of ResNet18 reaffirms its robustness in modeling visual-temporal patterns, while the lightweight MobileNetV2 demonstrates promising trade-offs for deployment in resource-constrained environments such as mobile-based religious learning tools.

To complement the quantitative WER, Table 6 presents examples of the ground truth versus the predicted outputs. These samples illustrate typical model errors, such as substitutions and insertions, which are particularly frequent in mid-sequence transitions.

The qualitative analysis shows that even the best-performing models can misclassify signs that are visually similar or poorly segmented in time. Signer-specific variations and limited training data are likely the cause of these challenges. In certain instances, duplicated hand forms and concurrent movements caused substitution or omission errors. A future direction to mitigate these issues involves incorporating attention mechanisms, explicit temporal modeling, or multi-view inputs.

To better understand the trade-off between model accuracy and efficiency, we provide a comparison of the latency and model complexity for each CNN backbone in Table 7. Metrics include the number of parameters, estimated model size, and approximate inference speed measured in frames per second (fps) on a single NVIDIA Tesla V100 GPU.

The latency and size values are based on the model specifications and typical benchmark results on the V100 GPU using 256×256 input resolution and batch size 1.

From this comparison, we observe that MobileNetV2 and ShuffleNet are significantly more efficient in terms of parameter count and inference speed, making them strong candidates for real-time applications on low-resource devices. However, this comes at the cost of a higher WER compared to heavier architectures such as ResNet18. According to the results, ResNet18 strikes the optimal balance between accuracy and computational cost, alongside MobileNetV2 providing a lightweight option with satisfactory performance. All latency and size values are based on published specifications and empirical benchmarks using the V100 GPU, 256×256 input, and batch size 1.

In summary, ResNet18 and MobileNetV2 emerged as the most effective architectures for the Quranic CSLR, balancing accuracy and computational efficiency. These results support the feasibility of deploying CSLR models in inclusive educational tools for the Deaf Muslim community.

5.3. Cross-Participant Evaluation

To further evaluate the model's generalization capability across different signers, we conducted

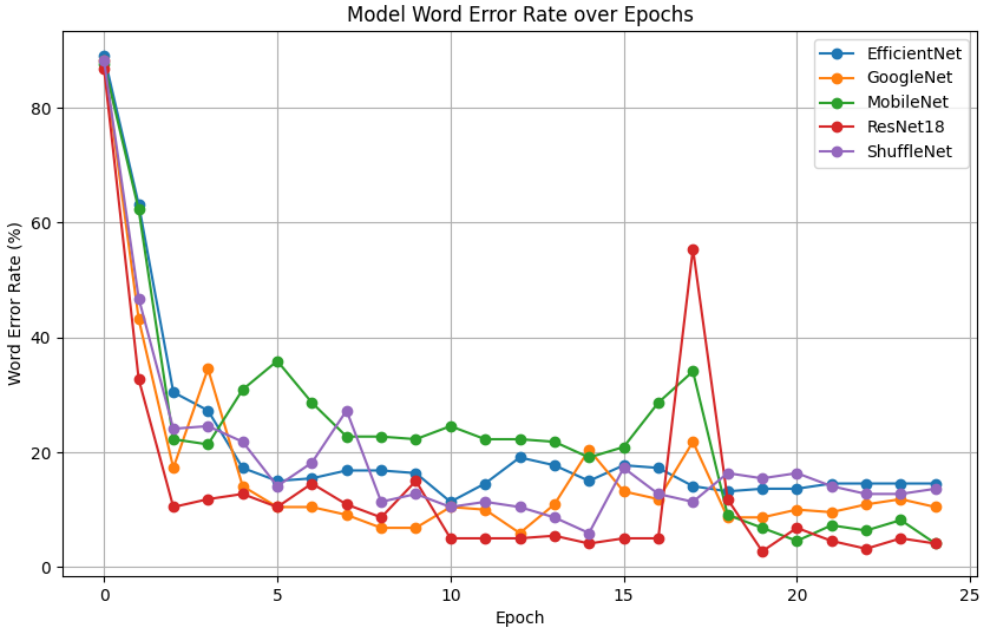


Figure 7. Word Error Rate (WER) for each epoch across five CNN models

Table 4. Word Error Rate (%) during training for each model

Epoch	EfficientNet	GoogleNet	MobileNetV2	ResNet18	ShuffleNet
1	89.09	88.18	87.73	86.82	88.18
2	63.18	43.18	62.27	32.73	46.82
3	30.45	17.27	22.27	10.45	24.09
⋮	⋮	⋮	⋮	⋮	⋮
20	13.64	8.64	6.82	**2.73**	15.45
24	14.55	11.82	**4.09**	5.00	12.73
25	14.55	10.45	4.09	4.09	13.64

Table 5. Word Error Rate (%) on test set.

EfficientNet	GoogleNet	MobileNetV2	ResNet18	ShuffleNet
8.33	9.17	7.92	**5.00**	13.64

Table 6. Sample Ground Truth vs. Prediction Outputs.

Video	Ground Truth	Prediction
1_alfatihah_2_6_7	i h di na ṣṣi rā ṭa l mu s ta qī m	i h di na ṭa ṣṣi rā ṭa l mu s ta qī m
112_alikhlas_1_1_4	qu l hu wa allā hu a ḥa d	ṣi qu l hu hi du wa allā hu a ḥa d

a cross-participant evaluation using the ResNet18 backbone, which had previously demonstrated the best performance in the within-participant setting.

In this setup, all data from Signer-1 and Signer-3 were split into 80% for training and 20% for validation. The entire dataset from Signer-2—who was not seen during training—was reserved for testing. This simulates a real-world scenario in which the system must accurately recognize signs from new users.

Figure 8 shows the validation WER across 25

Table 7. WER vs. Model Complexity and Inference Speed (estimated values).

Model	WER (%)	Parameters (M)	Model Size (MB)	Inference Speed (fps)
ResNet18	**5.00**	11.7	45	95
MobileNetV2	7.92	3.4	14	120
EfficientNet	8.33	5.3	20	80
GoogleNet	9.17	6.6	27	70
ShuffleNet	13.64	1.0	5	130

training epochs. The model achieved a validation WER of as low as 8.44%, signifying a robust performance on seen participants. Results on the unseen test signer (Signer-3) showed a WER of 50.46%, indicating that the model continues to face difficulties in effectively generalizing across signers.

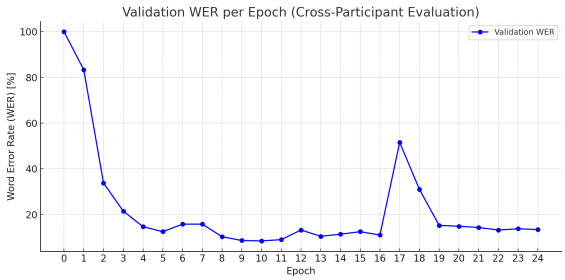


Figure 8. WER on the validation set across epochs during cross-participant training using ResNet18.

These results indicate that while the model can effectively learn signer-specific patterns, its performance degrades when encountering unseen signers. This limitation is expected due to the relatively small number of participants and the high inter-signer variation in signing speed, style, and articulation.

While the WER of 50.46% on unseen signers appears high compared to within-participant results, this is consistent with prior findings in signer-independent CSLR tasks with limited training data [21, 22]. Previous studies have also reported similar challenges in generalization when using only a few signers due to substantial inter-signer variation. Thus, this result serves not as a failure of the model but as a realistic benchmark for future work on signer-invariant learning under low-resource conditions.

Future work will explore signer-invariant modeling approaches and more robust temporal encoders to improve generalization. Additionally, expanding the dataset with more signers or using signer simulation techniques could help the model adapt better to diverse visual-linguistic patterns.

6. Conclusion

This study proposed a deep learning-based system for recognizing Quranic recitation using sign language, designed to enhance inclusivity for the Deaf Muslim community. This work addressed a critical gap in Quranic accessibility by constructing a new dataset based on the recently standardized Quranic Sign Language in Indonesia. The dataset comprises recordings of three Deaf participants performing Surah Al-Fatihah and Surah Al-Ikhlas, segmented at the verse level for training and evaluation.

The recognition task was framed as a continuous sign language recognition (CSLR) problem. Five pretrained CNN architectures—EfficientNet, GoogleNet, MobileNetV2, ResNet18, and ShuffleNet—were evaluated as visual feature extractors, combined with a BiLSTM-based temporal encoder and a CTC loss function for sequence alignment. Among these, ResNet18 achieved the best performance with a test set Word Error Rate (WER) of 5.00%, followed by MobileNetV2 with 7.92%. Their superior performance reflects the effectiveness of the residual and mobile-friendly architectures in modeling the spatiotemporal dynamics of sign language sequences.

In contrast, EfficientNet, GoogleNet, and ShuffleNet yielded higher WERs, highlighting architectural mismatches with the task's temporal complexity and the limited training data. A latency and model-size analysis further demonstrated that MobileNetV2 offers a strong trade-off between performance and efficiency, making it a viable choice for deployment on low-resource platforms.

A cross-participant experiment was also conducted using ResNet18 to evaluate the model's generalization. The model was trained on data provided by two signers and evaluated on a signer who had not been seen before. Achieving a low WER of 8.62% on the development set resulted in a test WER of 50.46%, indicating that variations dependent on the signer had a substantial impact on performance. This discovery highlights the requirement for further research on representations that are invariant to signers, more diverse training data, and robust evaluation methods.

In summary, this study contributes to the following: (1) a novel dataset for Quranic sign language recognition, (2) a comparative analysis of lightweight CNN backbones for CSLR, and (3) empirical evidence that residual and efficient architectures such as ResNet18 and MobileNetV2 are well-suited for Quranic sign language recognition tasks.

Future work will explore advanced temporal modeling, such as attention mechanisms or transformers, and strategies to overcome data scarcity, including signer simulation and transfer learning. This research lays the groundwork for inclusive AI-powered learning tools that promote spiritual engagement and equity for the Deaf Muslim community.

References

[1] Badan Penelitian dan Pengembangan Kesehatan, "Riskesdas 2018: Riset kesehatan dasar," 2018, accessed: 3 June 2025.

[2] E. Rakun, I. G. B. H. Widhinugraha, and N. F. P. Setyono, "Word recognition and automated epenthesis removal for indonesian sign system sentence gestures," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, pp. 1402–1414, 6 2022.

[3] D. M. Adimas, E. Rakun, and D. Hardianto, "Recognizing indonesian sign language gestures using features generated by elliptical model tracking and angular projection," in *2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS)*. IEEE, 2 2019, pp. 25–31.

[4] M. Jebali, A. Dakhli, and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion," *Evolving Systems*, vol. 12, pp. 1031–1044, 12 2021.

[5] T. Tao, Y. Zhao, T. Liu, and J. Zhu, "Sign language recognition: A comprehensive review of traditional and deep learning approaches, datasets, and challenges," *IEEE Access*, vol. 12, pp. 75 034–75 060, 2024.

[6] E.-S. M. El-Alfy and H. Luqman, "A comprehensive survey and taxonomy of sign language research," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105198, 9 2022.

[7] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October. Institute of Electrical and Electronics Engineers Inc., 12 2017, pp. 3075–3084.

[8] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*. ACM Press, 2006, pp. 369–376.

[9] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid cnn-hmm for continuous sign language recognition," in *Procedings of the British Machine Vision Conference 2016*. British Machine Vision Association, 2016, pp. 136.1–136.12.

[10] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

[11] M. Koklu, I. Cinar, and Y. S. Taspinar, "Cnn-based bi-directional and directional long-short term memory network for determination of face mask," *Biomedical Signal Processing and Control*, vol. 71, p. 103216, 2022.

[12] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[14] M. S. AL-Huseiny and A. S. Sajit, "Transfer learning with googlenet for detection of lung cancer," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, p. 1078, 5 2021.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] C.-Y. Tsai and Y.-K. Su, "Mobilenet-jde: a lightweight multi-object tracking model for embedded systems," *Multimedia Tools and Applications*, vol. 81, pp. 1–23, 03 2022.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] Z. Chen, H. Wang, C.-H. Yeh, and X. Liu, "Classify respiratory abnormality in lung sounds using stft and a fine-tuned resnet18 network," in *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 10 2022, pp. 233–237.

[19] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] M. Potnuru and B. S. Naick, "Semantic segmentation of mri images for brain tumour detection with shufflenet-based unet," *SN Computer Science*, vol. 4, p. 445, 6 2023.

[21] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7784–7793.

[22] O. Koller, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 282–301, 2019.

[23] H. Zhou, X. Huang, Y. Zhou, G. Wu, and L. Wang, "Spatial-temporal multi-cue network for sign language recognition and translation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 804–12 813.

[24] Y. Jiang, L. Wang, and G. Wu, "Visual-lexical alignment constraint for continuous sign language recognition," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 581–597.

[25] W. Yin, H. Hu, G. Wu, and L. Wang, "Self-emphasizing network for continuous sign language recognition," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4347–4355.

[26] W. Yin, H. Hu, L. Wang, and G. Wu, "Spatial-temporal enhanced network for continuous sign language recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1869–1877.

[27] H. Hu, H. Zhou, W. Yin, G. Wu, and L. Wang, "Hand-model-aware self-supervised pre-training for sign language understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 20 426–20 435.

[28] J. Zheng, H. Zhou, G. Wu, and L. Wang, "Contrastive visual-textual transformation for sign language recognition with variational alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 20 215–20 224.

[29] J. Zhao, H. Zhou, G. Wu, and L. Wang, "Conditional variational autoencoder for sign language translation with cross-modal alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024, early Access.

[30] G. Tharwat, A. M. Ahmed, and B. Bouallegue, "Arabic sign language recognition system for alphabets using machine learning techniques," *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–17, 2021.

[31] H. A. AbdElghfar, A. M. Ahmed, A. A. Alani, H. M. AbdElaal, B. Bouallegue, M. M. Khattab, G. Tharwat, and H. A. Youness, "A model for qur'anic sign language recognition based on deep learning algorithms," *Journal of Sensors*, vol. 2023, pp. 1–13, 2023.

[32] Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 10 2021, pp. 11 522–11 531.

[33] A. Hao, Y. Min, and X. Chen, "Self-mutual distillation learning for continuous sign language recognition," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 10 2021, pp. 11 283–11 292.