A Real-Time Libyan Sign Language Recognition using Deep Learning Method with Vocal Feedback

Alhaam Alariyibi
Computer Science Department, Information Technology Faculty, Benghazi University, Benghazi, Libya

Rana Faraj Amsaad
Computer Science Department, Information Technology Faculty, Benghazi University, Benghazi, Libya

Abdelsalam Maatuk
Information System Department, Information Technology Faculty, Benghazi University, Benghazi, Libya

ABSTRACT
There are over 12,000 deaf and hearing-impaired individuals in Libya, according to 2018 statistics from the Social Solidarity Fund. Despite this significant population, access to effective communication tools remains limited. Deep learning has revolutionized various domains, and its impact on the recognition and translation of sign languages is no exception. This paper explores the application of deep learning, particularly Long Short-Term Memory (LSTM) networks, in the context of Libyan Sign Language (LSL) recognition and translation, aiming to bridge communication barriers for the hearing-impaired community in Libya. The paper presents a novel dataset and a robust LSL recognition model based on LSTM architecture and key point extraction using MediaPipe Holistic. Furthermore, the real-time testing showcases the practicality of the proposed LSL recognition model, offering the potential for real-world applications to empower the deaf community. The proposed LSTM model achieves an impressive testing accuracy of 84% in recognizing LSL gestures and translating them into Spoken Arabic. This work is a critical milestone in enhancing accessibility and empowering the deaf community in Libya.

Keywords
Deep Learning; Libyan Sign Language; LSTM, MediaPipe; Real-Time Translation

1. INTRODUCTION
Communication is key to understanding, connection, and empathy. According to the World Health Organization (WHO) [1], more than 430 million people worldwide, including 432 million adults and 34 million children, require rehabilitation to address their hearing loss, accounting for over 5% of the global population. This number is expected to rise to 700 million people, or approximately one in ten individuals, by 2050. The impacts of hearing and speech impairments are broad and can be profound, affecting social interaction, education, and employment opportunities. Additionally, nearly 80% of people with disabling hearing loss live in low- and middle-income countries. These disparities highlight the urgent need for global efforts to address hearing loss and ensure equal access to education and employment opportunities for all.

Sign Language communicates through physical movements rather than spoken words, using visible cues from hands, eyes, facial expressions, and movements. This method of communication is used by over 70 million deaf or hard-of-hearing individuals worldwide [2]. Like spoken languages, there is no "universal" sign language, and different countries generally have their unique version of sign language that reflects their culture and region. Instead of dialects or accents as in oral language, the differences in sign language are expressed through various signs and gestures [3]; hence communication barriers still exist. Moreover, accessing professional interpreters can be challenging [4]. Therefore, it emphasizes the importance of an accurate and efficient sign language recognition system that facilitates communication not only between deaf and hearing individuals but also among those who use varied sign styles.

Sign Language Recognition (SLR) aims to bridge the communication gap between deaf or hard- of- hearing and the general population [5]. SLR has gained significant attention recently, due to its potential to facilitate inclusive communication and improve the quality of life for individuals with hearing impairments [6]. Challenges in SLR include the large variability in sign language across different regions, the dynamic and continuous nature of sig language, and the limited availability of annotated sign language datasets. Recent advancements in SLR have shown promising results, with improved accuracy and real-time performance [7]. However, there is still room for further research to enhance the robustness and adaptability of SLR systems, making them more accessible and effective for individuals with hearing impairments.

Various approaches have been explored in SLR, including computer vision-based methods and sensor-based methods. Deep learning has emerged as a powerful technique for SLR, leveraging the capabilities of artificial neural networks to automatically learn hierarchical representations from raw input data. Deep learning approaches for sign language recognition typically involve Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs excel in extracting spatial features from images or video frames, while RNNs are effective in modelling sequential information in sign language gestures. These deep learning models have shown remarkable performance improvements in SLR, achieving state-of-the-art results on various benchmark datasets. However, the success of deep learning approaches heavily relies on the availability of large and diverse annotated sign language datasets, which are often limited in size and scope [7].

In Libya, there are over 12,000 deaf and hearing-impaired individuals, according to 2018 statistics from the Social Solidarity Fund. Despite this significant population, access to effective communication tools remains limited. Furthermore, hearing-impaired people face significant communication challenges due to the lack of a standardized sign language system. This impedes their social interaction, limiting educational opportunities and social inclusion [5]. This research is motivated by the absence of a standardized system for Libyan Sign Language (LSL) and the lack of prior work in this field due to a lack of available data for LSL. Therefore, this work intends to address the communication barriers experienced by individuals with hearing impairments in Libya.

The fundamental goal of this research is to develop a computer vision-based system that can accurately recognize LSL gestures through deep learning techniques, improving communication between SL users and non-users. By leveraging technology, this research has the potential to enhance the quality of life for individuals with hearing impairments in Libya, enabling greater participation in society. This study introduces a novel dataset from the Hope Centre for the Deaf and Hard of Hearing and employs deep learning algorithms to overcome traditional limitations in sign language recognition. The system aims to improve sign language interpretation, address the need for low-resource Libyan Sign Language (LSL) recognition, and enhance communication and social inclusion for individuals with hearing impairments. This paper proposes an integrated system for sign language recognition and translation, focusing on the collection of the LSL gesture dataset, developing a preprocessing and feature extraction framework, and designing a deep learning model using MediaPipe and LSTM networks. It will be trained and validated on the dataset, integrated with a text-to-audio translation system, and optimized for real-time performance. These objectives, when achieved, will allow sign language users to communicate in real-time with greater ease, thus making communication more accessible and encouraging greater engagement in daily activities.

The remainder of this paper is organized as follows: In section 2, related works are described. In section 3, the proposed model for sign language recognition and the methodology employed are described, along with the process of data collection and preparation. In section 4, the experimental results are reported. In Section 5, further experimental results are discussed. Finally, conclusions are highlighted in section 6.

2. RELATED WORKS

In recent years, there has been a growing interest in the development of computer vision systems for sign language recognition. Several studies have focused on using deep learning techniques, particularly CNNs, for recognizing static sign gestures in various sign languages.

In [8], the work addresses the recognition problem of the static alphabet in Indian sign language using a vision-based approach. The authors proposed a CNN architecture called Signet that consists of a total of nine layers, including six hidden layers, one input layer, one dropout layer, and one output layer. The dataset of 24 static alphabet letters in Indian Sign Language used in their study consists of 2,500 images, which were augmented to 5,157 images. The researchers' work focused on extracting only hand features from the images. Hence, the Viola-Jones face detection and skin color segmentation algorithms were employed to detect the faces of the signers. The pixels in the region of the signers' faces were then replaced with black pixels, and the remaining image was processed to extract the hand regions. Following this step, the images were used for training and testing. The developed model achieved a training accuracy of 99.93% when using all 24 ISL static alphabet images. Additionally, it achieved testing and validation accuracies of 98.64%.

Another paper used a similar architecture [9], where the system aimed to automatically detect hand-sign letters and translate them into spoken Arabic. The system's architecture is based on CNN, which consists of feature extraction and classification components. The authors used RGB images of hands to represent the 31 letters in Arabic sign language, along with data augmentation techniques to increase the training data. Their model achieves 90% accuracy in recognizing 31 Arabic hand signs. The model is also connected to the Google Text-to-Speech (GTTS) API for converting hand signs into Arabic speech.

Another study [10] discusses the development of a video-based Egyptian Sign Language (ESL) recognition system, highlighting the challenges posed by variations in ESL across different regions and the lack of officially documented resources for ESL vocabulary. The researchers employed supervised deep learning, exploring two network architectures: CNNs and CNN-LSTM. To extract features and perform classification, they used the Inception-v3 model, pre-trained on the ImageNet dataset. To overcome the lack of reliable datasets, the authors collected their own by visiting a school for the deaf and recording videos of a volunteer deaf student performing nine Egyptian Sign Language gestures. The experiments and results showed that using CNNs alone achieved 90% accuracy. However, when the predicted labels from the CNN were passed to the LSTM, the accuracy dropped to 72%, suggesting that the CNN-LSTM architecture would be a better fit for continuous word sign recognition.

Another group of researchers from Saudi Arabia [11] used a deep learning model called Convolutional Long Short-Term Memory (ConvLSTM) to recognize dynamic Saudi sign language based on real-time videos. The model architecture combines convolutional layers with LSTM, making it an extension of the LSTM RNN. Their model consists of two ConvLSTM layers, where convolutional gates replace the fully connected gates in the LSTM. ConvLSTM also uses the convolution operation instead of matrix multiplication at each gate within the LSTM cell, enabling it to capture both spatial and temporal features effectively. The dataset used in this research focuses on health and disease signs, containing a total of 3,454 videos covering 35 different sign gestures. However, due to limitations of the computer device used for implementation, the model's training was constrained to 6 out of the 35 classes. It achieved 70% accuracy in recognizing the signs.

The study [12] aimed to develop a lightweight approach for real-time dynamic sign language recognition (DSLR) by integrating deep learning techniques with the MediaPipe framework. Researchers utilized two models: a Gated Recurrent Unit (GRU), known for its efficiency and low memory usage, and a 1D Convolutional Neural Network (CNN). A custom video dataset, DSL46, containing 2,910 videos of 46 commonly used American Sign Language (ASL) signs, was developed. MediaPipe was used to extract key points for hand and body movements, providing crucial details for gesture recognition. The dataset underwent preprocessing to address depth variation and ensure alignment between training and testing data. Experiments on the DSL46, LSA64, and LIBRAS-BSL datasets demonstrated high accuracy, with the CNN model achieving 98.8%, 99.84%, and 88.40%, and the GRU model achieving 97.08%, 97.96%, and 87.86%, respectively, for each dataset. These results highlight the effectiveness of the proposed approach for accurate and efficient sign language recognition.

The study in [13] also employed the MediaPipe framework to estimate the pose, hand, and face landmarks and extract features. OpenCV was used to capture videos for the dataset via webcam, from which key points were extracted and saved as an array instead of as video data. The authors also manually extracted some frames to create another dataset for static signs. They used a CNN model with three convolutional layers and an LSTM model with three layers. The models were applied to both static and dynamic datasets. The experiment showed that

the LSTM model was more effective at recognizing the dynamic gestures of sign language, whereas the CNN model was more efficient with static sign language.

An end-to-end model using MediaPipe and RNN models has been introduced in [14]. The authors created a custom dataset called DSL10-Dataset, which consists of 750 videos recorded in an indoor natural environment. The authors used the same feature extraction method as in the previous paper, extracting hand and face key points from the dynamic dataset. The RNN models (GRU, LSTM, and Bi-LSTM) were trained on the DSL10-Dataset. Two experiments were conducted: one without including face key points and one with them. The results show comparable accuracy in both cases, with GRU achieving the highest accuracy of 100%, while the other two models achieved around 99% on low-complexity sequences.

One of the main points gathered from the reviewed literature is that the availability of datasets and the lack of trusted resources has been the core challenge in sign language recognition research, leading researchers to create their own datasets and employ data augmentation techniques [9], [10], [12], [14]. The literature highlights that existing recognition models have primarily focused on Indian, Saudi, Egyptian Arabic, and American sign languages, among others. Three of the reviewed papers focused on the recognition of Arabic Sign Language (ArSL). However, just like the dialects in spoken Arabic, ArSL also varies regionally. ArSL (Arabic Sign Language) is used across approximately 22 Arab countries, each with its own distinct set of gestures. The variations in word gestures can be attributed to the cultural diversity among these countries. Nevertheless, despite the lack of consistency in ArSL across all 22 countries, there is commonality in the gestures representing Arabic letters and numbers [15]. In our research, the primary objective is to construct a comprehensive dataset for LSL, as there has been no prior research due to the absence of available data. This dataset, which is personally collected, will serve as a foundational resource for training and developing models for LSL recognition.

3. THE PROPOSED MODEL

Our model bridges the communication gap, empowering deaf individuals to communicate effectively with Arabic speakers, fostering inclusivity and accessibility. In this study, we employ a multi-phase framework. First, after collecting the dataset, it is augmented and annotated to create a valuable resource for training and evaluation. The dataset is then pre-processed using various techniques to enhance and prepare it for further use. Next, the dataset is split and fed into a real-time model for recognizing LSL and translating it into spoken Arabic using MediaPipe, LSTM networks, and GTTS. After training the model, it is evaluated on a set of unseen data to obtain classifications. Figure 1 illustrates an abstract view of the proposed model.

3.1 Dataset Construction and Collection

Recognizing and understanding sign language requires a substantial amount of labeled data to train accurate and robust models. The absence of a readily available LSL dataset presents a significant challenge. Therefore, we initiated the collection and curation of a dataset specifically for this research. The data collection involved collaboration with deaf students from the Hope Centre for Deaf and Hard of Hearing in Benghazi. Five students, who have a strong command of LSL, performed the recordings, ensuring that the gestures and expressions captured are authentic and representative of the language. The students received clear instructions to perform the selected signs naturally and accurately.

The dataset consists of 50 videos recorded with an iPhone 14 Pro Max, providing high-quality footage for further analysis and processing. The recording parameters were set to 30 frames per second (fps) and a resolution of 1080 x 1920 pixels, ensuring smooth and clear capture of the hand, face, and pose movements. Five students participated in the recordings, performing ten LSL signs in 1-second videos. The selected signs, which include words and phrases, are shown in Table 1.

Table 1. The selected Libyan signs.

No.	Arabic Gesture	English Meaning
1	مرحبًا	Hello
2	كيف حالك؟	How are you?
3	أين تسكن؟	Where do you live?
4	السلام عليكم	Peace be upon you
5	بنغازي	Benghazi
6	طرابلس	Tripoli
7	البيضاء	Albayda
8	طبرق	Toubrok
9	لا	No
10	نعم	Yes

3.2 Data Augmentation and Annotation

In our study, we employed two techniques—data augmentation and annotation—to enhance the quality of the dataset. Data augmentation techniques were applied to enrich the dataset and increase its variability. These techniques involved manipulating the original 50 videos using OpenCV to generate additional samples. As a result, the augmented dataset comprised a total of 800 videos after applying 16 different augmentation techniques, ensuring a more robust and diverse training set for the recognition and translation model.

The augmentation techniques were applied to the videos to introduce variations in lighting, contrast, blur, flipping, rotation, grayscale conversion, and hue adjustments. In some cases, a combination of these techniques was used, such as applying flipping followed by rotation. These techniques aimed to simulate real-world variations in lighting conditions and capture different perspectives of the LSL gestures. By augmenting the dataset, we sought to enhance the model's ability to generalize across various real-life scenarios, ensuring accurate analysis and recognition of LSL gestures. Additionally, the annotation process was carried out using a supervised platform, which offers efficient and intuitive annotation capabilities. By precisely annotating the face and hands in the LSL videos, we ensured that the model could learn the relevant spatial information and make accurate predictions during real-time recognition.

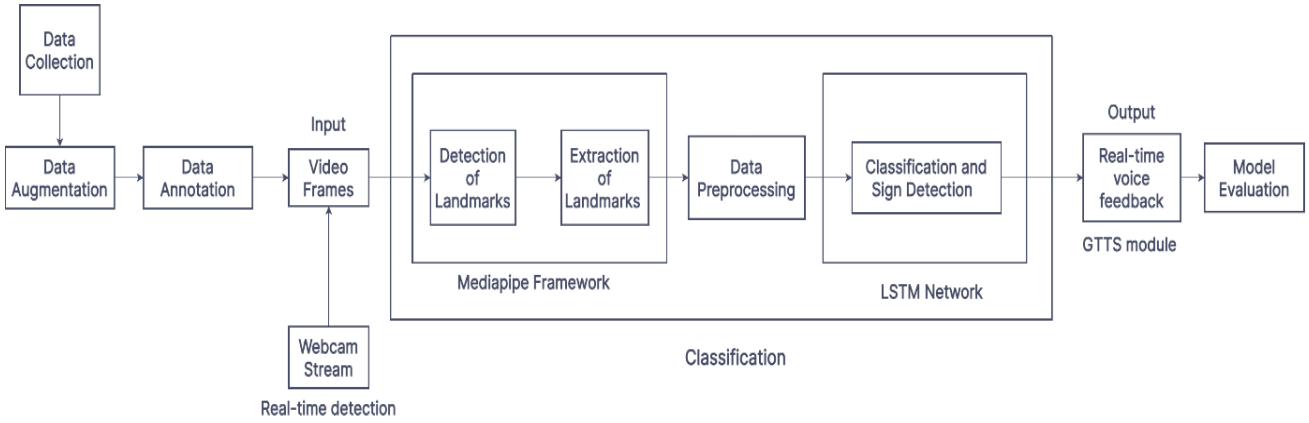


Figure 1. The components of the proposed model.

3.3 The Structure of the Proposed Model

Recognizing and translating sign language in real-time presents unique challenges due to the intricate nature of gestures and the requirement for precise interpretation. To overcome these challenges, it is essential to select a model capable of capturing the temporal dynamics of LSL gestures and effectively translating them into spoken Arabic. The following section describes the structure of the proposed classification model.

3.3.1 MediaPipe Framework

MediaPipe is an open-source framework that enables real-time perception of human pose, face landmarks, and hand tracking on mobile devices. It offers separate, fast, and accurate solutions for these tasks, but combining them into a unified solution is challenging. MediaPipe Holistic is a novel, state-of-the-art solution that addresses this challenge. It consists of a new pipeline with optimized components for pose, face, and hand tracking, which can run in real-time with minimal memory transfer.

MediaPipe Holistic provides a unified topology with over 540 key points in three dimensions, including pose, hand, and facial landmarks [16]. Figure 2 illustrates the 21 landmark key points detected on a hand. The framework serves as the foundation of the proposed approach, providing a robust and efficient platform for real-time perceptual computing tasks. By leveraging MediaPipe's hand-tracking capabilities, we can accurately capture the movements and positions of the hands, which are crucial for LSL recognition.

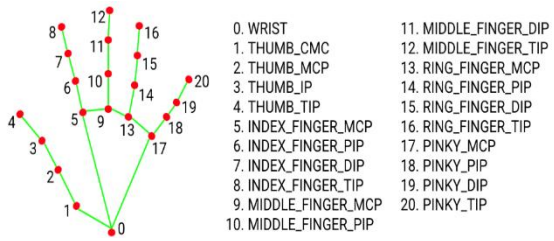


Figure 2. Hand landmarks - Mediapipe 21 key points [16].

3.3.2 Dataset Preprocessing

The dataset pre-processing stage is crucial for enhancing the quality, diversity, and usability of the collected dataset used for recognizing LSL. Moreover, it encompasses several pivotal steps to ensure that the videos are ready for training and testing the LSTM-based LSL recognition model. In this study, the preprocessing stages include label encoding using one-hot encoding and data preparation for the LSTM model.

3.3.2.1 Label Encoding Using One-Hot Encoding

Recognizing and interpreting LSL gestures require the model to understand and classify each sign accurately. To facilitate this, we employed label encoding using one-hot encoding. Each LSL sign was assigned a unique label, which was then transformed into a one-hot encoded vector. The following are the LSL signs and their corresponding labels:

"Hello" - Label: 1, "How are you?" - Label: 2, "Where do you live?" - Label: 3 "Peace be upon you" - Label: 4, "Benghazi" - Label: 5, "Tripoli" - Label: 6, "Albayda" - Label: 7, "Toubrok" - Label: 8, "No" - Label: 9, "Yes" - Label: 10.

Using one-hot encoding, these labels were converted into binary vectors with the categorical function from keras, where each vector had a length equal to the total number of unique labels (in this case, 10). For example, the one-hot encoded vector for "Hello" (Label: 1) would be [1, 0, 0, 0, 0, 0, 0, 0, 0, 0], where the "1" indicates the presence of the corresponding sign label.

3.3.2.2 Data Preparation for the LSTM Model

To effectively train the LSTM-based LSL recognition model, the preprocessed key point data extracted with MediaPipe must be organized into sequences and prepared for model input. In this work, the data preparation process consists of two steps: sequence formation and padding.

(1) Sequence Formation: The key points extracted from each video frame were organized into sequences of NumPy arrays. Each sequence represented a continuous flow of key points over time, effectively encoding the motion and shape of the hands during the LSL gestures. These sequences enabled the LSTM model to capture temporal dynamics.

(2) Padding: Sequences of LSL key points varied in length due to differences in gesture duration. To ensure uniformity and facilitate batch processing during model training, padding was applied to the sequences. This involved appending zeros to shorter sequences, extending them to match the length of the longest sequence in the dataset. Standardizing sequence length ensured efficient processing by the LSTM model.

By applying these preprocessing steps, the dataset was transformed into a suitable format for training the LSTM model. The sequences of one-hot encoded labels and padded key points served as input, allowing the model to learn the temporal patterns and spatial relationships essential for accurately recognizing and translating LSL gestures.

3.3.3 LSTM Networks

LSTM networks, a specialized type of Recurrent Neural Network (RNN), are designed to address the challenges of traditional RNNs, such as the vanishing gradient problem, by incorporating memory cells and gates that regulate the flow of information. This enables LSTMs to capture long-term dependencies in sequential data, making them highly effective for tasks such as Sign Language Recognition (SLR). LSTM networks excel at modeling the temporal dynamics of gestures, retaining and recalling relevant information about hand movements over time for accurate real-time recognition. Their architecture, which includes input, forget, and output gates, allows for the efficient storage and manipulation of information, making them particularly suitable for understanding the nuanced temporal variations inherent in sign language gestures [17].

The proposed model is constructed as a sequential neural network, emphasizing the temporal dependencies prevalent in sign language gestures. The architecture consists of multiple layers, with a summary of the sequential model shown in Table 2, each contributing to the extraction of relevant features and information.

Table 2. A summary of the sequential model.

Layer Type	Output Shape	Parameters	Activation	Regulariza -tion
LSTM	(None, 30, 64)	442112	ReLU	L2(0.01)
LSTM	(None, 30, 128)	98816	-	-
LSTM	(None, 64)	49408	-	-
Dense	(None, 64)	4160	ReLU	L2(0.01)
Dense	(None, 32)	2080	-	-
Dense	(None, 10)	330	-	-

3.3.3.1 The Architecture of Proposed LSTM

Three consecutive LSTM layers are embedded within the architecture. These layers facilitate the encoding of sequential dependencies present in sign language gestures. The first LSTM layer consists of 64 units, followed by the second layer with 128 units, and the final layer with 64 units. These layers enable the model to extract and comprehend intricate temporal patterns within the input sequence. The number of layers and units selected in our model reflects the classification complexity and the size of the data. Three fully connected dense layers are employed to further process the information learned from the LSTM layers. The first dense layer consists of 64 units, followed by a layer with 32 units, and finally, a layer with 10 units. These layers are responsible for higher-level feature extraction and eventual classification. The LSTM layers and the first two dense layers leverage the Rectified Linear Unit (ReLU) activation function. Noteworthy is its computational efficiency, where this simple yet effective function helps neural networks learn complex patterns by introducing non-linearity while avoiding vanishing gradient issues [19]. Nonetheless, in the final dense layer, the SoftMax function is employed to generate a probability distribution across the 10 output classes, aligning with the requirements of multi-class classification tasks.

3.4 Real-Time Voice Feedback Processing

The translated output of the recognized sign language gestures will be converted into spoken Arabic using GTTS (Google Text-to-Speech). GTTS is a powerful text-to-speech synthesis system developed by Google, known for generating high-quality speech output in multiple languages, including Arabic [20]. Integrating GTTS into our approach will enable the conversion of recognized gestures into spoken words, facilitating effective communication between sign language users and Arabic speakers.

3.5 Model Evaluation

Model evaluation is the process of assessing the model's performance and effectiveness by using metrics and techniques to measure its ability to make accurate predictions or produce desired outcomes on new, unseen data [21]. In our work, classification measures such as Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), accuracy, precision, recall, and F1-score were used to assess the model's accuracy in the analysis of sign gesture predictions. The final four metrics are calculated using the outcomes of the confusion matrix. The confusion matrix displays the frequency of correct and incorrect predictions. The outcomes of a confusion matrix are four key values: True Positive, correct prediction of positive instances (TP), False Positive, incorrect prediction of positive instances (FP), True Negative, correct prediction of negative instances (TN), and False Negative, incorrect prediction of negative instances (FN). Mathematically, accuracy, precision, recall, and F1-score are calculated from equations 1, 2, 3, and 4 respectively [21][22].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Percision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \times \left(\frac{Percision \times Recall}{Percision + Recall}\right) \quad (4)$$

4. EXPERIMENTS AND RESULTS

In this section, we detail the experimental study. We examine the effects of key factors, such as data augmentation, data segmentation, and comparison studies, to validate and enhance the performance of our proposed model. The outcomes collectively offer valuable insights into the effectiveness and robustness of our approach.

4.1 Experimental Environment Setup

This study presented a comprehensive experimental method and analysis of the proposed LSL recognition model. The experiments were conducted using Python, Keras library, TensorFlow as a back-end, OpenCV, Matplotlib, Sci-kit Learn, and gTTS. It's important to note that all these experiments were conducted with a batch size of 32 and 40 epochs. Moreover, the ADAM (Adaptive Moment Estimation) optimizer is used as the optimization algorithm for training the proposed model. It dynamically adjusts the learning rates for different parameters, leading to faster convergence and improved training performance [23]. The learning rate of 0.001 is used as a starting point, which is the optimal value when we want to monitor the model learning process and also helps prevent the model from fitting the noise in the data too quickly, therefore, preventing overfitting to some extent [12]. The categorical cross-entropy loss function is commonly used in tasks involving multi-class classification. It quantifies how well the predicted probabilities match the actual class labels, encouraging the model to make accurate predictions [23]. For the first experiment, a consistent 20% testing and 80% training data split was maintained. On the other hand, the dataset was divided into 3 sets: 60% for training, 20% for validation, and

20% for testing for the last experiment. Each set contained samples from the 10 classes. The hyperparameters used for the models are outlined in Table 3.

Table 3. Hyperparameters for classification.

Hyperparameters	Value
Optimizer	ADAM
Learning Rate	0.001
Loss	Categorical Cross entropy
Metrics	Categorical Accuracy
L2 Regularization	0.002
Epochs	40
Batch Size	32
Validation Split	20%

We conducted a series of experiments to investigate the impact of data augmentation on the proposed model's performance using both the basic dataset and augmented data, as well as to explore the effect of different data segmentation ratios on testing, training, and validation. Lastly, we conducted a comparative study between the GRU and LSTM architectures.

4.2 Effect of Data Augmentation on Proposed Model

In this experiment, we examined the influence of data augmentation on the performance of the proposed model. We compared the model's performance using the basic dataset and augmented data to assess the impact of increased data variability on recognition accuracy.

First, we evaluated the performance of our proposed model using the initial dataset, which comprised 50 videos. The evaluation of model performance is presented in Figure 3, which shows the training accuracy and loss of the proposed model. The model achieved an overall accuracy of 40% on the training data at epoch 34 and a testing accuracy of 30%. Figure 4 shows the confusion matrix of the model using the basic dataset.

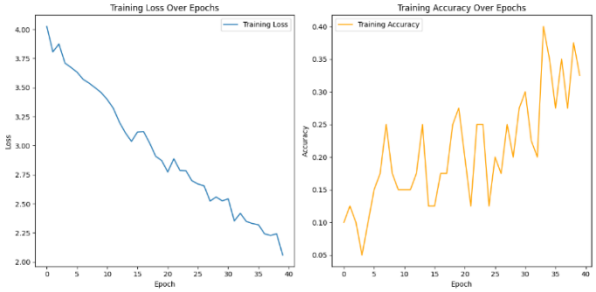


Figure 3. Training and loss accuracy of the proposed model using the basic dataset over epochs

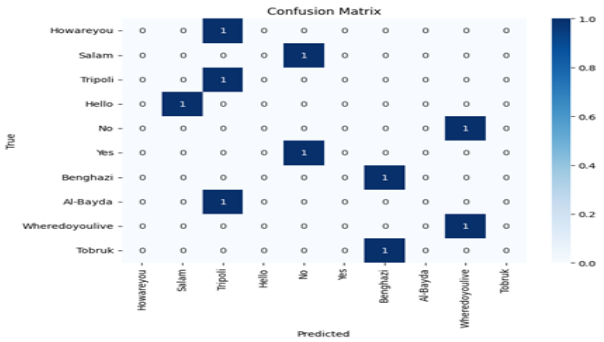


Figure 4. The confusion matrix of the model using the basic dataset

After that, we assessed the model's performance using the augmented dataset comprising 800 videos. Figure 5 shows training accuracy and loss analysis. We examined how data augmentation enhances the model's recognition accuracy and overall classification performance. The model was trained for 40 epochs, during which it attained a maximum training accuracy of 87.97% at epoch 38 and a testing accuracy of 80.6%. Table 4 presents the classification evaluation metrics of the proposed model using both the basic dataset and the augmented dataset. The confusion matrix of the model using the augmented dataset is illustrated in Figure 6.

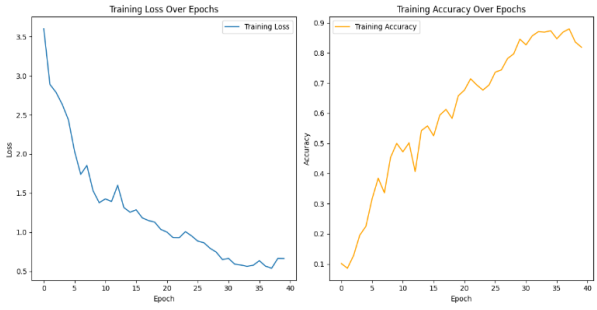


Figure 5. Training and loss accuracy of the proposed model using the augmented dataset over epochs

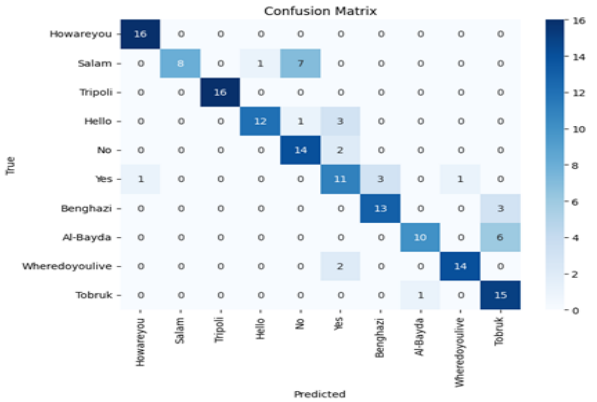


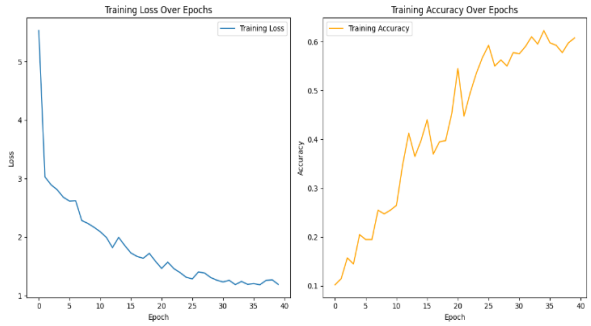
Figure 6. The confusion matrix of the model using the augmented dataset

Table 4. The proposed model performance with and without augmentation

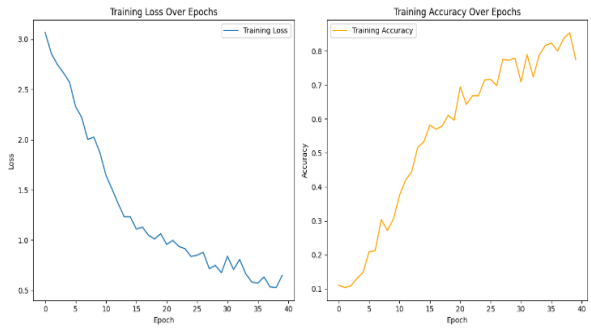
Proposed Model	MSE	MAE	R-squared	Precision	Recall	F1-score	Testing Accuracy	Training Accuracy
without augmentation	6.80	2.00	0.1758	13%	30%	18%	30%	40%
with augmentation	1.20	4.50	0.8545	84%	81%	81%	80.62%	87.97%

4.3 Effect of Data Segmentation

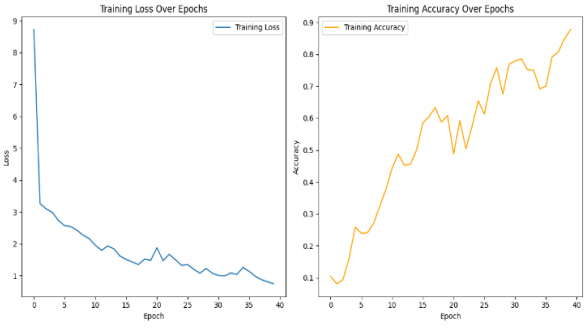
In this experiment, we investigated the impact of different data segmentations on the performance of our model. By varying the proportions of testing, training, and validation data, we assessed how the allocation of data influences the recognition accuracy and generalization of the model. We evaluated the model's performance four times with different segmentations (50%-50%, 30%-70%, 40%-60%, and 20%-60%-20%) using the augmented dataset. In Figure 7, the training accuracy and loss of each segmentation were presented. The model was trained for 40 epochs. Table 5 presents the classification evaluation matrices for each segmentation.



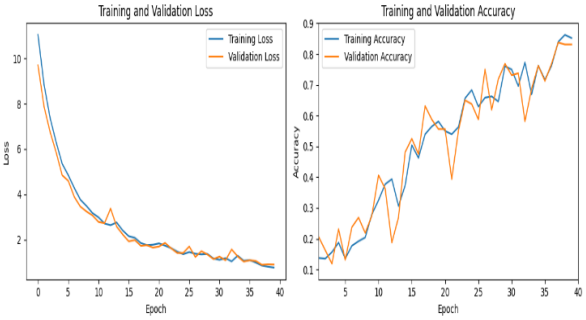
(a)



(b)



(c)



(d)

Figure7. (a) Training loss and accuracy with 50% training data over epochs. (b) Training loss and accuracy with 70% training data over epochs. (c) Training loss and accuracy with 60% training data over epochs. (d) Training and validation loss and accuracy over epochs

Table 5. The model classification evaluation matrices for each segmentation

Segmentation			MSE	MAE	R-squared	Precision	Recall	F1-score	Testing Accuracy	Training Accuracy	Epochs
Training	Testing	Validation									
50%	50%	-	2.135	0.88	0.7412	64%	57%	52%	56.75%	62.25%	36
70%	30%	-	1.2625	0.5125	0.8470	81%	77%	76%	76.7%	85%	39
60%	40%	-	0.8031	0.3281	0.9027	85%	84%	84%	83.75%	87.71%	40
60%	20%	20%	0.8187	0.3187	0.9008	87%	84%	84%	84.38%	87%	37

Testing multiple splits of data in deep learning reduces overfitting by improving the performance of such models on unseen data while avoiding issues related to data imbalance. All these aspects enhance the real-world applicability of the model since it ensures consistency of performance without bias in any distribution.

4.4 Comparative Study

We conducted a comparative study by implementing the Gated Recurrent Unit (GRU) architecture from a previous research paper on our dataset. More details about GRU are described in [14]. The aim was to evaluate the performance of the GRU model on our dataset in contrast to our proposed model. In the experiment with the GRU model, we adopted a 20% testing, 20% validation, and 60% training data split, as this configuration consistently yielded the best results. Figure 8 provides insights into training accuracy and loss. Throughout the 40 training epochs, our GRU model reached its peak training accuracy of 89.53% during epoch 39 and achieved a testing accuracy of 91.25%. The confusion matrix analysis is shown in Figure 9.

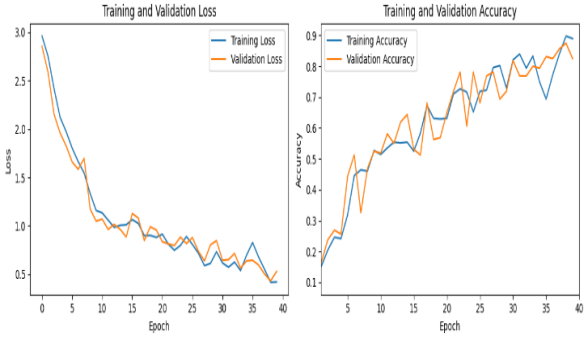


Figure 8. Training loss and accuracy of the GRU model using our dataset over epochs

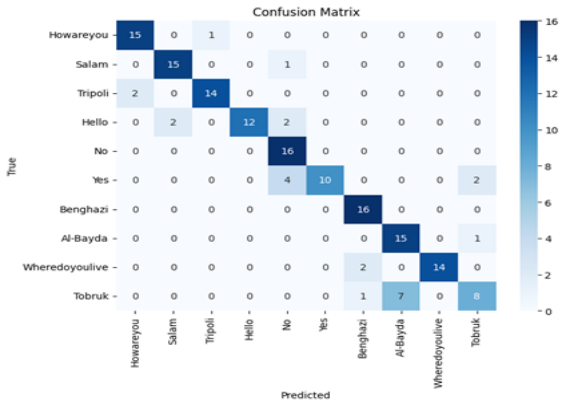


Figure 9. The confusion matrix of the GRU model using our dataset

We conducted a comparative analysis between the proposed LSTM model and the GRU architecture to assess their respective performances in LSL recognition. Tables (6 and 7) show a comparison between the models in terms of model performance and classification metrics.

Table 6. A comparison of the model performance between the proposed model and GRU model

Evaluation metric	Proposed model (LSTM)	GRU
MSE	0.8187	0.4562
MAE	0.3187	0.1688
R^2	0.9008	0.9447

Table 7. A comparison of the model classification matrices between the proposed model and GRU model

Class	Proposed model (LSTM)			GRU		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Howareyou	94%	100%	97%	89%	100%	94%
Salam	94%	100%	97%	89%	100%	94%
Tripoli	100%	100%	100%	100%	94%	97%
Hello	100%	56%	72%	79%	94%	86%
No	70%	100%	82%	79%	69%	73%
Yes	100%	69%	81%	100%	69%	81%
Benghazi	92%	69%	79%	94%	100%	97%
Al-Bayda	70%	100%	82%	0.94%	100%	97%
Wheredoyoulive	79%	94%	86%	100%	94%	97%
Tobruk	69%	56%	62%	94%	94%	94%
Accuracy	84%			91%		
Macro Avg	87%	84%	84%	92%	91%	91%
Weighted Avg	87%	84%	84%	92%	91%	91%

4.5 Real-Time Model Testing

The real-time testing of the proposed LSL recognition model was executed using a combination of OpenCV and MediaPipe Holistic. This experiment demonstrates the integration of these tools to process live frames from a webcam, capturing crucial key points, pre-processing them, and subsequently feeding them into the model for sign prediction. Figures (10, 11, 12, and 13) are screenshots from our real-time model testing. This real-time testing framework provides a glimpse into the usability and effectiveness of our sign language recognition system, bridging the communication gap for the hearing-impaired community.

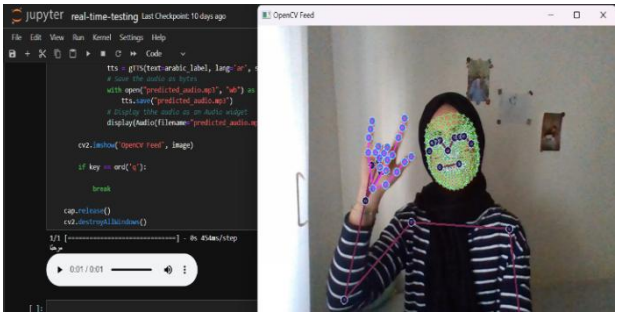


Figure 10. The detection of "Hello" gesture in LSL in Real-Time model testing.

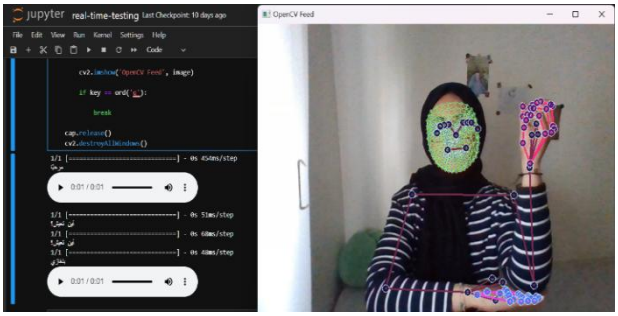


Figure 11. The detection of "Benghazi" gesture in LSL in Real-Time model testing.

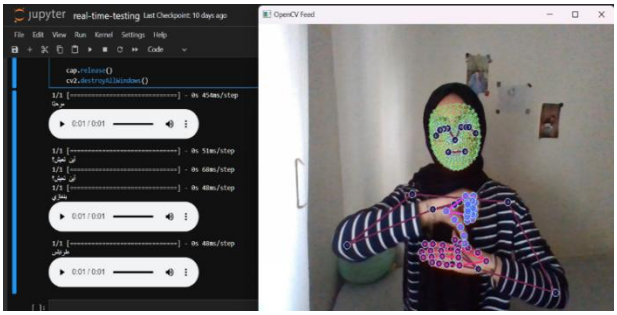


Figure 12. The detection of "Tripoli" gesture in LSL in Real-Time model testing

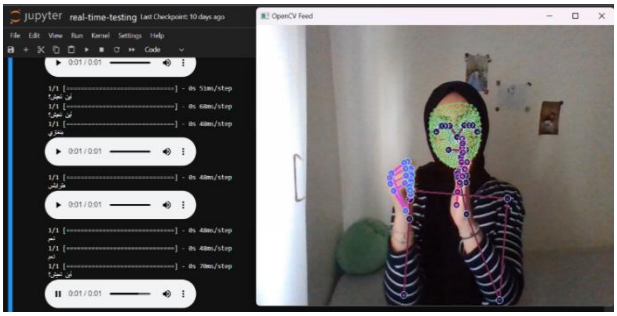


Figure 13. The detection of "Wheredoyoulive" gesture in LSL in Real-Time model testing

5. DISCUSSION

The experiments identified optimal hyperparameters through extensive testing, where the number of epochs played a critical role. Although increasing the number of epochs improved accuracy, it also led to significant overfitting. The selected parameters struck a balance between accuracy and generalization. This section presents a detailed interpretation of the results, emphasizing key findings and their implications for the study.

The first experiment aimed to assess the impact of data augmentation on the performance of our proposed model for recognizing LSL gestures. An analysis of the basic dataset revealed a gradual decrease in training loss and a corresponding increase in training accuracy, as shown in Figure 3. These trends indicated that the model was learning to minimize errors as it iterated through the training data and became more proficient at classifying LSL gestures. However, fluctuations in both metrics suggested potential challenges in capturing certain nuances and raised concerns about overfitting due to the limited dataset size.

The study highlights the contrasting performance of a model trained on a basic dataset versus an augmented dataset. As shown in Table 4, for the basic dataset, the training loss and accuracy graphs showed limitations in learning, with an MSE of 6.8000, MAE of 2.0000, and a low R-squared value of 0.1758, reflecting poor predictive performance. The classification metrics also indicated low precision, recall, and F1-scores, leading to an overall accuracy of just 30%.

In contrast, training on the augmented dataset demonstrated significant improvements. As presented in Figure 5, the training loss consistently decreased, and accuracy steadily improved, reaching higher values. The evaluation report in Table 4 showed substantial gains, with MSE reduced to 1.2000, MAE to 0.4500, and an R-squared value of 0.8545, indicating a strong correlation between predictions and actual values. Classification metrics also improved markedly, achieving an overall accuracy of 80.6%, showcasing the model's enhanced ability to recognize LSL gestures effectively. In a nutshell, the first experiment demonstrated the substantial positive impact of data augmentation on our model's performance. The augmented dataset significantly reduced errors, boosted accuracy, and improved generalization across all sign classes.

The second experiment demonstrated that data segmentation significantly impacts the performance of the LSL recognition model. With a 50% training and 50% testing split, the model achieved a moderate accuracy of 56.75%, with an MSE of 2.1350, indicating room for improvement. Increasing the training data to 70% improved accuracy to 76.6% and reduced the MSE to 1.2625, highlighting the benefits of a larger training dataset. When 60% of the data was used for training and 40% for testing, the model achieved an accuracy of 83.75% with a significantly reduced MSE of 0.8031. The best results were obtained with a 60/20/20 split for training, validation, and testing, yielding an accuracy of 84.37% and an MSE of 0.8187. These findings emphasize that increasing training data and including a validation set enhance the model's robustness and generalization for LSL gesture recognition.

In the context of the model learning process with different data segmentation, illustrated in Figure 7, it was demonstrated that as the training data percentage increased from 50% to 60%, 70%, and to the balanced 60% training, 20% validation approach, both loss and accuracy graphs showed a consistent improvement. The balanced segmentation approach resulted in the most favorable learning curves, with minimal loss, peak accuracy, and effective learning, avoiding significant overfitting. These demonstrate the model's sensitivity to data distribution, emphasizing that higher proportions of training data contribute to improved prediction accuracy and correlation. However, excessively large training sets can lead to overfitting, while smaller testing sets might limit generalization.

The last experiment aimed to compare GRU and LSTM architectures on an augmented dataset to evaluate their effectiveness in sign language recognition. This comparison sheds light on the model's sensitivity to different architectural choices and their impact on sign language recognition. The training loss and accuracy curves illustrated in Figure 8 revealed that The GRU's training loss curve demonstrated a consistent decline, indicating stable convergence with few fluctuations. Similarly, the training accuracy curve showed a gradual upward trend, suggesting steady learning. While both models eventually achieved comparable accuracy, the smoother curves of the LSTM model in Figure 7(c) implied a more stable learning process compared to the GRU model in Figure 8, which might have experienced slightly more variability. Table 6 presents a comparison of model performance metrics. GRU demonstrated superior performance with lower MSE (0.4562) and MAE (0.1688) compared to LSTM (MSE of 0.8187 and MAE of 0.3187), alongside a higher R-squared value (0.9447 vs. 0.9008), indicating better predictive accuracy. Table 7 further examines classification metrics. Both models displayed strong precision, recall, and F1-scores for several LSL signs. However, notable differences emerged. The GRU model achieved slightly higher precision and recall for some signs, such as "Benghazi" and "Al-Bayda". Despite this, the LSTM model excelled in terms of precision and recall for signs like "Howareyou," "Salam," and "Tripoli." Overall accuracy favoured the GRU model with 91% compared to 84% for the LSTM model. These findings suggest that GRU's architecture is more effective for this task, though LSTM demonstrated smoother and more stable learning curves, reflecting consistent training dynamics.

In short, the comparative study highlighted the strengths and weaknesses of LSTM and GRU architectures in LSL gesture recognition. While the GRU model showed promise in terms of predictive accuracy, the LSTM model exhibited strengths in recognizing certain LSL signs. The decision between these models should consider the unique requirements and characteristics of the sign language dataset and the specific gestures of interest. Further research may explore hybrid models or other neural network architectures to harness the combined strengths of both LSTM and GRU for improved LSL recognition.

In the real-time testing, our model effectively detected and translated sign gestures into spoken Arabic using gTTS. While it succeeded in most cases, challenges arose due to the complexity of sign language. This highlights the need for ongoing improvements, including the collection of a diverse dataset and model refinement, to enhance accuracy and usability.

6. CONCLUSION

Throughout this study, we conducted a comprehensive exploration of LSL recognition, aiming to bridge the communication gap for the deaf and hard-of-hearing community in Libya. Our investigation provided valuable insights into sign language recognition, particularly LSL, while addressing key research objectives. This work represents a significant step forward, contributing both to academic understanding and to practical advancements that enhance the lives of the deaf and hard-of-hearing in Libya.

Our culturally sensitive dataset and LSTM model, equipped with real-time translation capabilities, have the potential to revolutionize accessibility and communication, fostering inclusivity and empowerment. This study focused on three key aspects: creating a culturally sensitive LSL dataset, developing an LSTM-based model for LSL recognition, and conducting real-time model testing. Our most significant contribution is the creation of a bespoke dataset, meticulously curated to capture the unique cultural and linguistic nuances of LSL. This diverse collection of gestures reflects the richness of LSL, forming the cornerstone of our research.

We conducted a series of experiments using Python as our primary programming language, examining the impact of data augmentation on our proposed LSTM model's performance. The results clearly demonstrated that data augmentation significantly improved the model's accuracy and robustness. This technique notably reduced prediction errors, leading to greater precision and enhancing the effectiveness of LSL communication. Furthermore, we explored the impact of varying data segmentation ratios, shedding light on the model's sensitivity to data distribution. Moreover, our proposed LSTM model for LSL recognition yielded remarkable results, achieving an accuracy rate of 84%. A comparative study with the GRU model highlighted the superiority of the GRU model in terms of accuracy, yet our model demonstrated strengths in recognizing particular LSL signs. Additionally, our model's ability to effectively convert LSL signs into spoken Arabic using Google Text-to-Speech (gTTS) technology in real-time marks a significant achievement in our research efforts.

7. REFERENCES

[1] World Health Organization, "Deafness and hearing loss," WHO, Feb. 02, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] "Human Rights of the Deaf," WFD. https://wfdeaf.org/our-work/human-rights-of-the-deaf.

[3] T. Johnston and A. Schembri, "Australian Sign Language (Auslan)," Cambridge University Press,2007, doi: https://doi.org/10.1017/cbo9780511607479.

[4] K. Kozik, "Without Sign Language, Deaf People Are Not Equal," Human Rights Watch, Sep. 23, 2019. [Online]. Available: https://www.hrw.org/news/2019/09/23/without-sign-language-deaf-people-are-not-equal.

[5] S. Dhulipala, F. F. Adedoyin, and A. Bruno, "Sign and Human Action Detection Using Deep Learning," Journal of Imaging, vol. 8, no. 7, p. 192, Jul. 2022, doi: https://doi.org/10.3390/jimaging8070192.

[6] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues," IEEE Access, vol. 9, pp. 126917–126951, 2021, doi: https://doi.org/10.1109/access.2021.3110912.

[7] M. Madhiarasan, and P. P. Roy, "A comprehensive review of sign language recognition: Different types, modalities, and datasets, " arXiv preprint, Apr. 2022, Accessed: May 15, 2023. [Online]. Available: http://arxiv.org/abs/2204.03328.

[8] C. J. Sruthi and A. Lijiya, "SigNet: A deep learning based indian sign language recognition system," Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019, pp. 596–600, Apr. 2019, doi: 10.1109/ICCSP.2019.8698006.

[9] M. M. Kamruzzaman, "Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network," Wirel Commun Mob Comput, vol. 2020, pp. 1–9, May 2020, doi: 10.1155/2020/3685614.

[10] A. Elhagry and Elrayes, Rawan Glalal, "Egyptian Sign Language Recognition Using CNN and LSTM," arXiv (Cornell University), Jul. 2021, doi: https://doi.org/10.48550/arxiv.2107.13647.

[11] B. A. Al-Mohimeed, H. O. Al-Harbi, G. S. Al-Dubayan, and A. A. Al-Shargabi, "Dynamic Sign Language Recognition Based on Real-Time Videos," International Journal of Online and Biomedical Engineering (iJOE), vol. 18, no. 01, pp. 4–14, Jan. 2022, doi: 10.3991/ijoe.v18i01.27581.

[12] M. S. Abdallah, G. H. Samaan, A. R. Wadie, F. Makhmudov, and Y.-I. Cho, "Light-Weight Deep Learning Techniques with Advanced Processing for Real-Time Hand Gesture Recognition," Sensors, vol. 23, no. 1, p. 2, Dec. 2022, doi: 10.3390/s23010002.Nn

[13] V. G and K. Goyal, "Indian Sign Language Recognition Using Mediapipe Holistic," Apr. 2023, Accessed: May 29, 2023. [Online]. Available: https://arxiv.org/abs/2304.10256v1.

[14] G. H. Samaan et al., "MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition," Electronics (Basel), vol. 11, no. 19, p. 3228, Oct. 2022, doi: 10.3390/electronics11193228.

[15] B. Y. AlKhuraym, M. M. Ben Ismail, and O. Bchir, "Arabic Sign Language Recognition using Lightweight CNN-based Architecture," International Journal of

Advanced Computer Science and Applications, vol. 13, no. 4, 2022, doi: 10.14569/IJACSA.2022.0130438.

[16] Grishchenko Ivan and Bazarevsky Valentin, "MediaPipe Holistic — Simultaneous Face, Hand and Pose Prediction, on Device," Dec. 10, 2020. https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html.

[17] J. D. Kelleher, "Deep learning," Cambridge, Ma: The Mit Press, 2019, p. 299.

[18] J. Long, A. Khaliq, and K. M. Furati, "Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach," International Journal of Computer Mathematics, vol. 98, no. 8, pp. 1617–1632, May 2021, doi: https://doi.org/10.1080/00207160.2021.1929942.

[19] S. Subburaj and S. Murugavalli, "Survey on sign language recognition in context of vision-based and deep learning," Measurement: Sensors, vol. 23, p. 100385, Oct. 2022, doi: 10.1016/j.measen.2022.100385.

[20] "gTTS — gTTS documentation," gtts.readthedocs.io, https://gtts.readthedocs.io/en/latest/.

[21] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K.-H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition," Sci Rep, vol. 12, no. 1, p. 11964, Jul. 2022, doi: 10.1038/s41598-022-15998-7.

[22] K. A. Rahim, Md. Khaliluzzaman, S. I. Khan, and M. S. Kabisha, "Face and Hand Gesture Recognition Based Person Identification System using Convolutional Neural Network," International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1, pp. 105–115, Mar. 2022, doi: 10.18201/ijisae.2022.273.

[23] B. Sundar and T. Bagyammal, "American Sign Language Recognition for Alphabets Using MediaPipe and LSTM," Procedia Comput Sci, vol. 215, pp. 642–651, 2022, doi: 10.1016/j.procs.2022.12.066.

[24] S. Afaq and S. Rao, "Significance of Epochs on Training a Neural Network", A Neural Network. International Journal of Scientific & Technology Research, vol. 9, pp. 485-488, 2020, [Online]. Available: www.ijstr.org.