



Automated Conversion of Arabic Sign Language to Text Based on the Hybrid CNN-LSTM Model to Enable Communication with the Deaf and Non-Speaking Individuals

Khansaa Dheyaa Ismael¹, Elaf A. Saeed^{2*} and Ammar Ibrahim Majeed³

^{1,2,3}Department of Automation and Artificial Intelligence Engineering, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

*Corresponding Author's E-mail: elaf.ahmed@nahrainuniv.edu.iq

(Received 21 June 2025; Revised 14 October 2025; Accepted 29 December 2025; Published 1 September 2026)

<https://doi.org/10.22153/kej.2026.12.018>

Abstract

Communicating with the Deaf and non-speaking individuals and understanding sign language are complicated processes. With the continuous development in deep learning and automation of systems, however, solutions are available to overcome the difficulty of communicating with this segment of society and understanding sign language, especially in Arabic-speaking regions. This article proposes an automated sign language recognition system based on a hybrid convolutional neural network (CNN)–long short-term memory (LSTM) architecture, which represents two layers: CNN, which extracts spatial features from sign language motions, and LSTM, which records the temporal connections within gesture sequences. The model's ability to provide smooth communication is demonstrated by evaluating its performance using a specially created Arabic Sign Language (ArSL) dataset, and then the model was tested and trained by using input videos recorded for three signers, each one of whom pronounced 50 different signs. We added three other signers to pronounce the days of the week in sign language, improving the accuracy of the pretraining. These videos served as added data to the ArSL dataset. The key point was extracted by using matplotlib, the key point values for training and testing were collected, and the data were preprocessed through feature extraction and batch normalisation. The signs were recognised in the Arabic language. The proposed model detects the sign language action and recognition according to the dataset training, which was achieved building and training by using the LSTM neural network and CNN. The accuracy rating, F-1 score, recall and precision of the proposed model for ArSL were 95%, 95%, 95% and 96%, respectively, with 126 epochs and batch normalisation equal to 30. Ultimately, the proposed model was able to recognise and detect ArSL accurately.

Keywords: Deep learning; sign language; CNN; LSTM; deaf and mute

1. Introduction

Social communication between people is important because it includes the process of understanding others, conveying thoughts, feelings and asking for help when needed [1]. The ability to do this well is important. However, the Deaf and non-speaking individuals often face difficulties while communicating with others in the auditory environment [2]. Sign language is their only appropriate means of communicating with others,

but it is considered difficult to understand. This obstacle causes a gap in communication for the Deaf and non-speaking individuals, resulting in difficulty asking for help, social integration, accessing basic services and obtaining educational opportunities [3]. Arabic Sign Language (ArSL) needs to be converted into readable text to create a reliable system that enables the Deaf and non-speaking individuals to communicate smoothly with people who do not understand sign language [4]. This task ensures improved social integration, enhances the

This is an open access article under the CC BY license:



removal of barriers to communication with society, and promotes inclusion and understanding. As a result, the Deaf and non-speaking individuals can have greater educational and professional opportunities because they have access to educational resources and professional development. Thus, the proposed model will help the Deaf and non-speaking individuals achieve their educational and professional goals. Convolutional neural networks (CNNs) [5] can efficiently identify spatial characteristics in pictures and video frames [6]. They can be used to examine the body postures, facial emotions and hand movements that make up the ArSL Identification System signs in our setting. ArSL [7] is one of the sign languages in Arab nations. Its syntax and structure are different from those of spoken Arabic. The available ArSL recognition algorithms directly map the recognised sign in the ArSL phrase and its Arabic word, with the ArSL sentence's structure and grammar being maintained. The process of translating a recognised ArSL statement into an Arabic sentence that coheres to Arabic grammar and structure is known as ArSL translation [8]. This process enables algorithms to identify and decode the sign language's visual features.

However, developing models for static and dynamic sign language recognition (SLR) remains a complex challenge. Static signs consist of a single hand posture, while dynamic signs involve sequences of movements that require learning both spatial and temporal patterns. As a result, static SLR often struggles with visually similar signs, whereas dynamic SLR must address variations in speed, signer style and temporal alignment, making robust system design an ongoing research problem.

The complementarity between CNNs [9], [10], and long short-term memory (LSTM) [11], [12] ensures that sequential data can be processed [13] and that interim relationships between signs can be recorded so that the system can understand the conversation's flow and context. The proposed hybrid model that combines CNN [14] and LSTM takes advantage of both architectures [15]. After the CNN extracts visual features from the sequential data of ArSL, the LSTM [16], [17] processes this data to obtain a more thorough and accurate interpretation of ArSL. The successive data are processed by the LSTM, producing a deeper and specific interpretation of ArSL. In data acquisition, an accurate ArSL dataset must be created [18]. This step entails gathering a sizeable and varied collection of videos with a range of signers and sign variations. ArSL has individual and geographical variances, much like any other language. Accuracy must be maintained while accommodating these

variances in the system's architecture. Using advanced transfer deep learning techniques [19], this proposed model seeks to improve multiple research fields while advancing ArSL recognition, using deep neural networks to create an accurate ArSL dataset [20]–[22]. The present work also aims to differentiate ArSL by exposing a neural network to various lighting conditions and orientations related to hand gestures and translate such gestures in to written text. The objective is to efficiently handle actions of different sizes, decrease training time with fewer epochs and attain higher accuracy in comparison to current methods.

A hybrid CNN-LSTM model is trained to recognise and translate ArSL accurately by converting each hand gesture into readable text [23]. Our goal for this research is to help the Arabic-speaking community become a more accessible and inclusive place for the Deaf and non-speaking individuals. The novelty of the proposed model is that it seeks to solve an essential social issue through advanced motion and gesture recognition techniques with computer vision and deep learning techniques. However, creating static and dynamic SLR [24]–[26] delts remain a difficult research task [27]. The proposed model utilises machine learning and computer vision to translate hand gestures into written Arabic text after capturing and recognising sign language.

2. Proposed Models

Deep learning tools [28] were employed to convert the sign language used by the Deaf and non-speaking individuals into written speech through the proposed model and thus achieve seamless social communication with people who suffer from hearing problems or speech difficulties. Sign detection was performed using different models. The first model used a single LSTM, the second used two LSTMs and the third added a CNN with two LSTMs. Different models were used from the pre-trained models with two LSTMs improve the results[29].

The models are designed according to the following steps.

2.1. Dataset Preparation

• Data Collection

In the first stage of the proposed method, a collection of videos, each of which represents a different sign in sign language, are gathered. The model is trained and evaluated using these videos as

the primary input. The dataset contains many samples for every sign, capturing differences in execution across several people to guarantee that the model learns efficiently. A person is shown making a certain gesture in each film, hence ensuring that the data for categorisation are representative and varied. The ArSL database [27] contains a set of ArSL signs that synchronise manual and non-manual motions. Facial expression is a key element of the non-manual components seen in all signs in the dataset, and it is essential for communicating grammatical and emotional information. Facial expressions provide visual emphasis and communicate the signer's attitude or mood through synchronised motions of the eyes, eyebrows, lips, nose and cheeks. After a thorough examination of a large variety of sign language motions, the signs that highlight motions that necessitate the combination of manual and non-manual aspects were included in the database. In general, the signs in the ArSL database can be categorised into two main groups on the basis of the number of postures utilised in each sign. The first group includes signs with only one pose. For instance, the sign reads 'Hungry'. The second group includes multiple postural indications, as demonstrated by the sign for 'Very small'. The total collection includes 6,748 video samples covering 50 different ArSL markers. Throughout several sessions, each of the four trained signers who performed these signs produced about 30 repetitions of each sign, with some of them generating more than 30 samples. The time in each video sample varies according to the signer and the specific sign. With a total of 337,991 frames, the dataset provides a thorough and diverse resource for developing and evaluating SLR systems. The sign names that were utilised are listed in Table 1.

**Table 1,
ArSL dataset-1**

SignID	Sign (Arabic)	Translation
1	كثير	Very
2	قليل	Little
3	سعيد	Happy
4	سعيد جدا	Very happy
5	سعيد قليل	Slightly happy
6	حزين	Sad
7	حزين جدا	Very sad
8	حزين قليل	Slightly sad
9	Angry	Angry
10	Angry جدا	Very angry
11	غاضب قليل	Slightly happy
12	خجول	Shy
13	خجول كثير	Very shy
14	خجول قليل	Slightly shy

15	حساس	Sensitive
16	حساس كثير	Very sensitive
17	حساس قليل	Slightly sensitive
18	خائف	Afraid
19	خائف كثير	Very afraid
20	سعيد قليل	Slightly happy
21	مريح	Comfortable
22	مريح كثير	Very comfortable
23	مريح قليل	Slightly comfortable
24	برد	Cold
25	برد كثير	Very cold
26	برد قليل	Slightly cold
27	جوعان	Hungry
28	جوعان كثير	Very hungry
29	جوعان قليل	Slightly hungry
30	شبعان	Full
31	شبعان كثير	Very full
32	شبعان قليل	Slightly full
33	صغير	Small
34	صغير كثير	Very small
35	صغير قليل	Slightly small
36	كبير	Big
37	كبير كثير	Very big
38	كبير قليل	Slightly big
39	سمين	Fat
40	سمين كثير	Very fat
41	سمين قليل	Slightly fat
42	سهل	Easy
43	سهل كثير	Very easy
44	سهل قليل	Slightly easy
45	شرير	Evil
46	شرير كثير	Very evil
47	شرير قليل	Slightly evil
48	صلب	Solid
49	صلب كثير	Very solid
50	صلب قليل	Slightly solid

As a contribution to the dataset, a fresh set of videos describing the days of the week in Arabic was added to the initial dataset, which included 50 ArSL signs. These videos were taken from recordings that were made publicly available online to increase the dataset's variation and expand its contents. Each video was duplicated 25 times to guarantee similarity and expand the dataset's size. Each video sample was also divided into 30 frames for consistency with the current data structure and ensure that it can be integrated easily with the feature extraction and training procedures. Table 2 shows the days' sign names.

Table 2,
ArSL dataset-2

Sign-ID	Sign (Arabic)	Translation
1	السبت	Saturday
2	الاحد	Sunday
3	الاثنين	Monday
4	الثلاثاء	Tuesday
5	الاربعاء	Wednesday
6	الخميس	Thursday
7	الجمعة	Friday

Fig. 1 shows the examples of images that were used.



(a) Angry



(b) Happy

Fig. 1. Examples of images that were used

Holistic landmark [30]–[32] identification is essential for preprocessing and feature extraction when used in tandem with dataset gathering. Key landmarks may be simultaneously detected throughout the complete human body, including the face, hands and full-body stance, by this extensive computer vision approach. Holistic techniques greatly improve the model's comprehension of intricate human movements by capturing interconnected landmarks in a single framework, in contrast to standard models that identify discrete pieces. One prominent use of this strategy is the MediaPipe Holistic model, which tracks and detects hand, face and position markers in real time using lightweight neural networks. The system receives rich spatiotemporal information required for precise gesture identification when a well-structured dataset and comprehensive landmark detection are combined. Thirty videos were collected for each

sign, for example, hello, thanks, etc. Each video was divided into 30 frames, and each frame contains 1,669 landmark values (3×30 sequences, 30 frames). These values are saved as a matrix to input to the model.

• Converting Videos to RGB Frames

Every recorded video was transformed into a series of frames in RGB format during the video processing stage. Each frame contains three colour channels: red, green and blue. This transformation allows the deep learning model to interpret sign language motions as a series of discrete images rather than as continuous motion, allowing for a more thorough analysis of spatial and temporal information. The model is able to efficiently extract important visual information required for precise classification when the film is split up into separate RGB frames. This stage is important for preprocessing because it sets the conditions necessary for feature extraction, enabling the system to collect and analyse the colour data required to differentiate between different hand movements and letters.

The data were then systematically separated into a test set and a training set during the data preparation stage. Typically, 80% of the data is allocated for training, allowing the proposed deep learning model to learn and recognise patterns. The remaining 20% of the data is used for testing, where the model's performance is evaluated using samples it has never seen before. These data were partitioned before the repetitions for the dataset, creating days of the week. For the other datasets used, the 50 signs were filmed more than once, and the same video was not repeated more than once. Eighty percent of the signs for the days of the week were repeated for training only, while other videos were used for testing. This separation is essential to ensure the proposed model's ability to generalise when dealing with new inputs while retaining the prior training data. This stage is vital in the data preparation process and directly impacts the accuracy and reliability of the proposed model.

2.2. Preprocessing

• Feature Extraction

Feature extraction is a key step because it involves understanding the specific features of each object entering the model. The first layers of CNNs [33], [34] are for feature extraction, including pooling layers for dimensionality reduction. After the important features have been obtained, the fully

connected layers facilitate the classification of the output. In addition, the CNN is connected to the LSTM [35], [36] to capture the sequence of gestures.

With the use of various cutting-edge pretrained deep learning architectures, the feature extraction phase of the proposed SLR system extracted spatial and semantic representations from input video frames. Several models, such as the Swin transformer, VGG16, VGG19, ResNet-152, EfficientNet-B0, DenseNet-201 and a specially designed CNN, have unique architectural benefits that support reliable and varied feature learning. ResNet-152 uses residual blocks with shortcut connections to qualify training of very deep networks; DenseNet-201 connects each layer to all subsequent layers within dense blocks to promote feature reuse; and EfficientNet-B0 applies a combination coefficient to uniformly scale network resolution, width and depth, resulting in a high-performance and parameter-efficient architecture ideal for resource-constrained environments. Conventional CNNs VGG16 [37]–[39] and VGG19 [40]–[42] are known to be deep and straightforward, able to capture low-level and mid-level visual characteristics by using a stack of tiny 3×3 convolution filters. With three extra convolutional layers, VGG19 has a slightly higher representational capacity than VGG16, which has 16 layers. These models work well when architectural clarity and computational simplicity are needed, especially when high-level semantic abstraction is not the main objective.

With the introduction of residual connections by ResNet-152 [43]–[45], deep networks are less likely to have the vanishing gradient issue. ResNet-152's 152 layers allow intricate patterns to be learned while preserving training stability. It is especially well suited for complicated major visual tasks, such as precise gesture identification in video sequences, because of its capacity to maintain performance even as depth rises.

A contemporary family of models called EfficientNet-B0 [46]–[48] balances network depth, breadth and resolution by employing a compound scaling technique. EfficientNet-B0 maintains competitive accuracy while being highly computationally efficient. This approach works well in situations when resources are few, such as edge deployment or real-time inference on constrained devices.

In DenseNet-201, each layer receives input from all previous layers due to its extensive cross-level connections [49]. This method strengthens gradient flow and promotes feature reuse, increasing accuracy with fewer parameters. DenseNet-201 is

helpful when solid performance, economy and compactness are needed.

A simple CNN is used as a standard to evaluate how well more intricate networks operate. Even though it lacks the depth and representational power of larger models, a basic CNN is ideal for small-scale datasets or preliminary testing because it offers interpretability and faster training. The vision transformer model Swin transformer offers a hierarchical architecture with moving windows for self-attention computation. Swin transformer outperforms conventional CNNs in simulating contextual linkages and long-range dependencies, which are crucial for identifying temporal and spatial subtleties in sign language motions. When working with intricate sequential patterns spanning video frames, its incorporation offers a strong and contemporary substitute for convolution-based methods.

For the following classification layers to acquire the high-level representations necessary for precise sign language identification, each of these models was used to extract significant features from RGB frames. Aside from allowing ensemble techniques to improve overall model accuracy and resilience, having a variety of model topologies ensures that their efficacy in collecting pertinent information can be investigated thoroughly.

• Batch Normalisation

Batch normalisation [50], which normalises the inputs to each layer, is essential to the training process because it stabilises the learning process and significantly speeds up convergence. Batch normalisation ensures that the activity of the layers is distributed consistently and stably during training, preventing variables from drifting and the possibility of disappearing or inflating gradients. Reducing overfitting also helps improve smoothing performance. Dropout is sometimes also used to randomly disable a portion of the cells during training.

Combining these strategies can further improve regularisation and the performance and efficiency of deep learning models [51].

2.3. LSTM-Based Temporal Dependency Modelling

• One LSTM

The core component of gesture recognition is the LSTM architecture, which is designed to recognise temporal correlations between upper-body gestures. Videos are segmented into 30 frames

to gradually understand body movement patterns part by part. Overfitting is prevented by adding dropout and batch normalisation to increase training stability. At the head layer, a softmax activation function is used to transform the temporal gestures into 50 classes. This model accurately predicts the signs and understands the sequential nature of the sign. An example of the LSTM model is shown in Figure 2.

• Two LSTMs

A second LSTM is added to achieve more accurate results. The first one understands low-level features, and the second one understands higher-level features. Thus, the improved design consists of two LSTMs, the same as the previous design when only one LSTM was used.

After each video is segmented into 30 frames and fed into the model, the first layer of the LSTM, consisting of 256 layers, handles the understanding of low-level patterns. Dropout was used to increase coverage and batch normalization and to reduce overfitting. The second layer, which is the same LSTM as the first layer and consists of 256 units, was used to give the model a deeper understanding of temporal gestures. Batch normalisation and dropout were also added to increase coverage and regularisation. Thus, sequential stacking increases the model's representational power. Batch normalisation and dropout reduce overfitting and unknowns and enhance generalisation to unknown samples. LSTMs are particularly suitable for SLR tasks because they preserve the temporal sequence of gestures and represent them across time. Figure 2 shows the two-LSTM models.

2.4. Prediction Model

By converting the network's raw outputs into normalised probability scores, the softmax activation function is used in the model's last stage for classification [52], [53]. Each possible sign class is given a probability value using this mathematical function, which guarantees that the sum of all probabilities equals one. Consequently, the projected sign is chosen from the class with the highest likelihood. In addition to making multiclass classification easier, softmax's application [54], [55] offers interpretability by displaying the degree of confidence attached to each prediction. This level is known as the prediction stage, during which the system uses the learnt information from previous layers to choose the most likely indication.

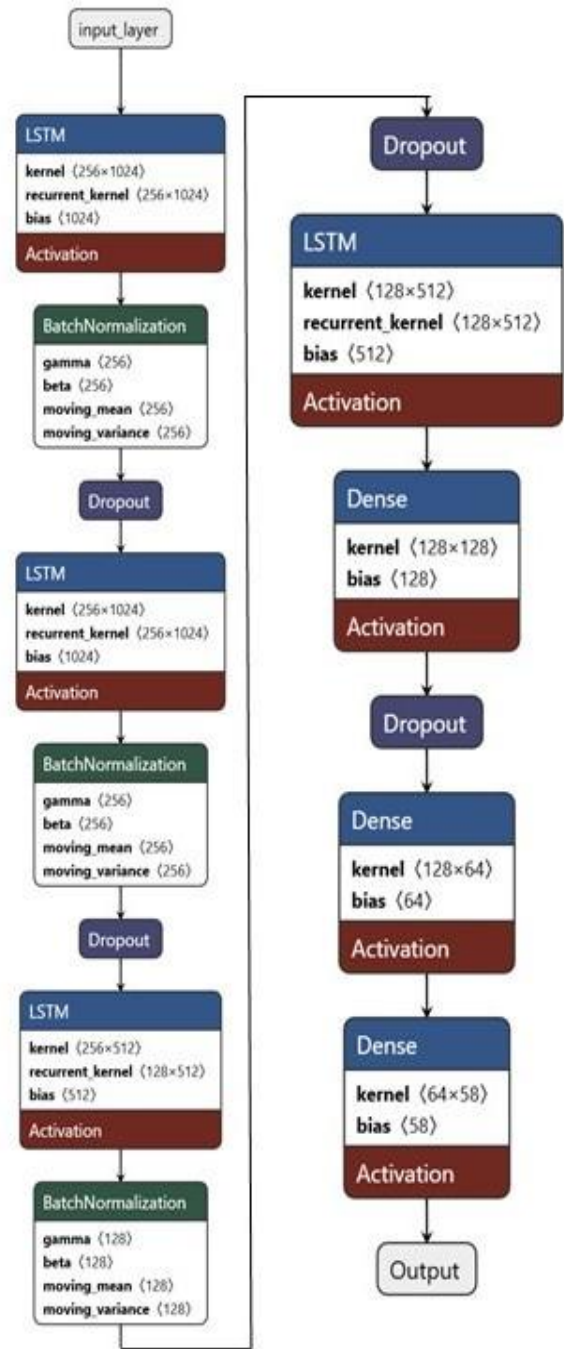


Fig. 2. Two-LSTM models

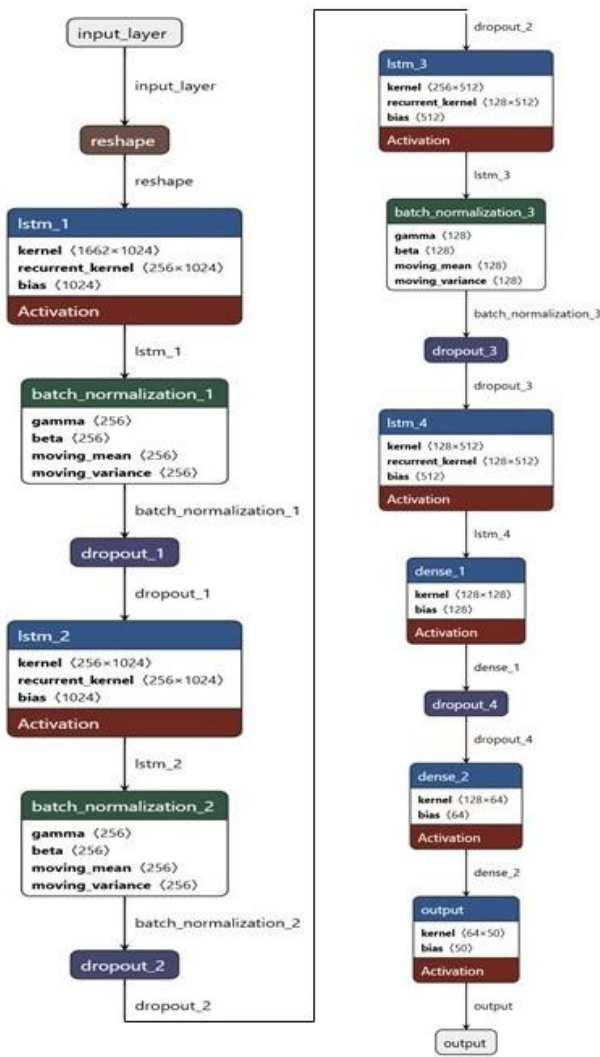
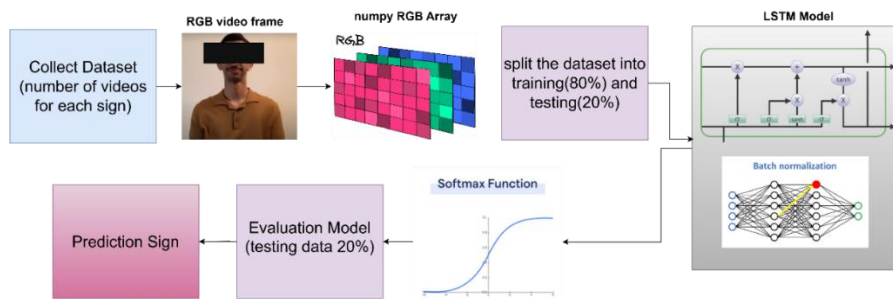


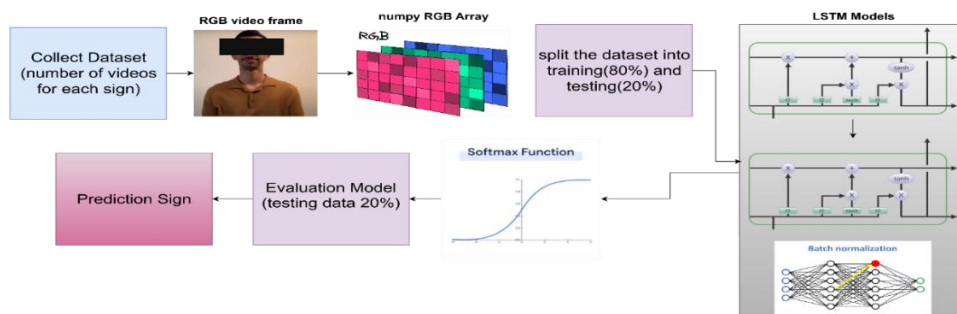
Fig. 3. Two-LSTM + CNN models

2.5. Evaluation Model

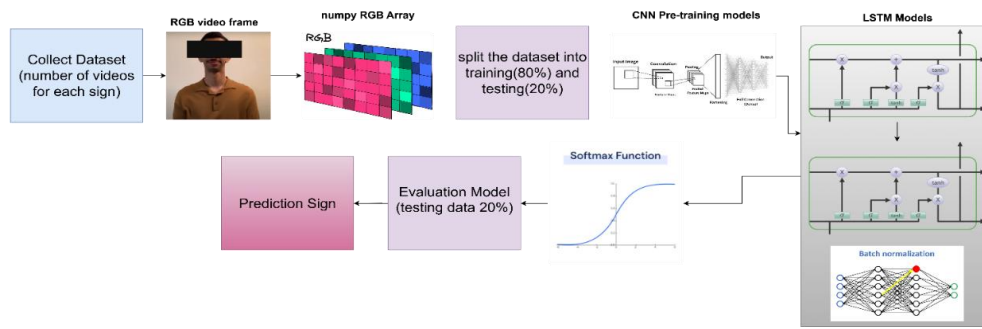
Model evaluation is a critical phase in assessing the performance and generalisation capability of the trained deep learning model. The 20% of the dataset that was set aside for testing — which includes unseen samples that were not used during the training phase — is used to test the model in this step. The goal is to verify the model's ability to identify sign language motions in response to fresh data. Evaluation metrics, including accuracy, precision, recall and loss, are calculated to provide a numerical assessment of performance. Other methods, such as the confusion matrix, may be used to examine categorisation errors and strengths. This phase is essential for confirming the model's prediction power and making sure it has not overfitted to the training set. This procedure makes it possible to use deep learning methods for automatic SLR. The process of a deep learning system for SLR is depicted in the block diagrams in Figure 4.



(a) One-LSTM model



(b) Two-LSTM model



(c) Pretraining models with LSTMs

Fig. 4. Proposed models

3. Proposed Models' Results

The results of the proposed model can be illustrated in two phases. The first one is represented by the training and testing of the proposed model on the ArSL dataset, and the second phase is the results of training and testing the proposed model on the

dataset after the days of the week have been added. The performance of the 10 hybrid models for ArSL recognition is shown in Table 3. The training was conducted on a computer with an NVIDIA GPU RTX-4090 CPU and an Intel Core i7 processor.

Table 3,
Evaluation results-1

NO.	Model description	Precision	Recall	F1-score	Accuracy	Epoch	Total params	Training time
1	One-LSTM	0.82	0.79	0.79	0.79	83	4,612,525 (17.60 MB)	≈ 30 minutes
2	Two-LSTM	0.90	0.89	0.89	0.89	127	8,546,392 (32.60 MB)	≈ 29 minutes
3	Two-LSTM + CNN - 80 20	0.91	0.91	0.91	0.91	104	8,546,392 (32.60 MB)	≈ 22 minutes
4	Two-LSTM + VGG16 - 80 20	0.94	0.92	0.92	0.92	140	8,546,392 (32.60 MB)	≈ 27 minutes
5	Two-LSTM + VGG19 - 80 20	0.72	0.68	0.68	0.68	135	8,546,392 (32.60 MB)	≈ 4 hours
6	Two-LSTM + ResNet152 - 80 20	0.92	0.91	0.91	0.91	119	8,546,392 (32.60 MB)	≈ 4:52
7	Two-LSTM + EfficientNetB7 - 80 20	0.96	0.95	0.95	0.95	180	8,546,392 (32.60 MB)	≈ 9:25
8	Two-LSTM + DenseNet201 - 80 20	0.90	0.90	0.90	0.90	180	8,546,392 (32.60 MB)	≈ 11:1
9	Two-LSTM + SwimTrans - 80 20	0.79	0.78	0.77	0.79	189	8,546,392 (32.60 MB)	≈ 3.66
10	Two-LSTM + ConvNext - 80 20	0.91	0.90	0.90	0.90	125	8,546,392 (32.60 MB)	≈ 25.85 minutes

In addition to total and trainable parameters, the assessment metrics include precision, recall, F1-score, accuracy and training epochs. The benefit of

stacking LSTM layers was demonstrated by the two-LSTM model's enhanced performance, achieving 89% accuracy compared with the single-

LSTM model's 79% accuracy. Adding two LSTMs to CNN topologies improved performance even further. With an accuracy of 92%, the two-LSTM + VGG16 model was the most accurate, closely followed by the two-LSTM + ResNet152 model at 91%. With the greatest overall scores across measures and 95% accuracy, EfficientNetB7 also demonstrated impressive performance. While models such as DenseNet201[54], [55] and ConvNeXt demonstrated consistent outcomes, the SwimTrans [52], [53] integration performed

somewhat worse at 77%. The total and trainable parameters were the same for all models, suggesting that architectural decisions rather than model size were the primary cause of performance variations. These findings show that SLR may be successfully enhanced by integrating LSTM with powerful CNN architectures. The two-LSTM + EfficientNetB7 models will detect and recognise each hand gesture motion and translate them into the Arabic language with high accuracy, reaching 100%, as shown in Figure 5.

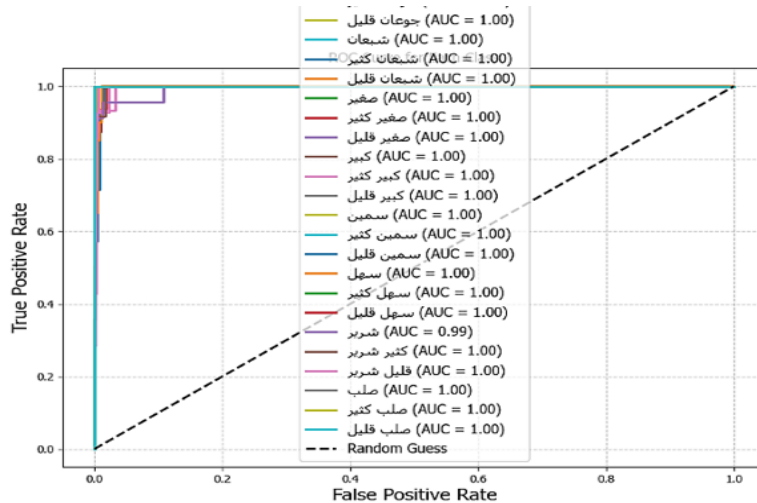


Fig. 5. Two-LSTM models

Among all the tested models, the two-LSTM + EfficientNetB7 model performed the best, having

the highest accuracy (95%), precision (96%), recall (95%) and F1-score 95%, with training and validation accuracy and loss shown in Figure 6.

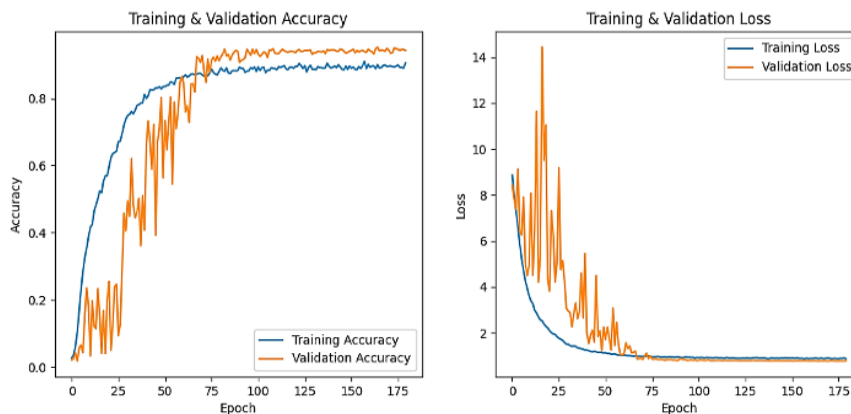


Fig. 6. Accuracy and loss in the two-LSTM + EfficientNetB7 models

The confusion matrix shows how the model's predictions match the real sign labels. In the context of sign language detection, the confusion matrix is a useful approach for determining the kinds of mistakes the model makes. The columns represent the signs predicted by the proposed model, and the

rows represent the actual (true) signs performed by the person. The diagonal values compared with the off-diagonal values provide a general sense of how well the model is performing; higher values indicate better performance, as shown in Fig. 7.

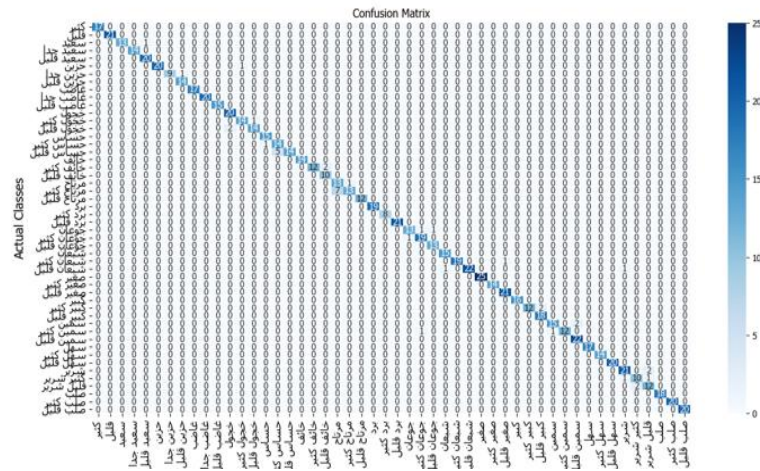


Fig. 7. Confusion matrix

All the results in the second phase after the days of the week were added to the ArSL dataset show the training of the proposed model in different steps, each of which illustrates the precision, recall, F1-score and accuracy. Updated assessment results for the ArSL recognition models are shown in Table 4, together with information on accuracy, precision, recall, F1-score and training parameters. With an accuracy of 94% as opposed to 89% earlier, the two-LSTM models demonstrated a substantial improvement over the prior findings (Table 3). Their stability was confirmed by the excellent accuracy of 92%–94% in combination with CNN and VGG16. Meanwhile, the accuracy of the two LSTMs that used VGG19 and EfficientNetB7 declined considerably, with values of 49% and 61%,

respectively, indicating sensitivity to changes in the dataset. Robustness was demonstrated by models such as DenseNet201 and ConvNeXt, which remained consistently strong throughout both studies. SwimTrans exhibited mediocre performance that was somewhat worse than that of other models. The significance of data features on model stability was highlighted by the fact that while most models maintained or enhanced performance during dataset enlargement, certain designs demonstrated decreased efficacy. The proposed model detects and predicts the days of the week from hand gestures, which are then displayed on the PyCharm console. The translation from hand gestures to the Arabic language achieved 100% accuracy, as shown in Figure 8.

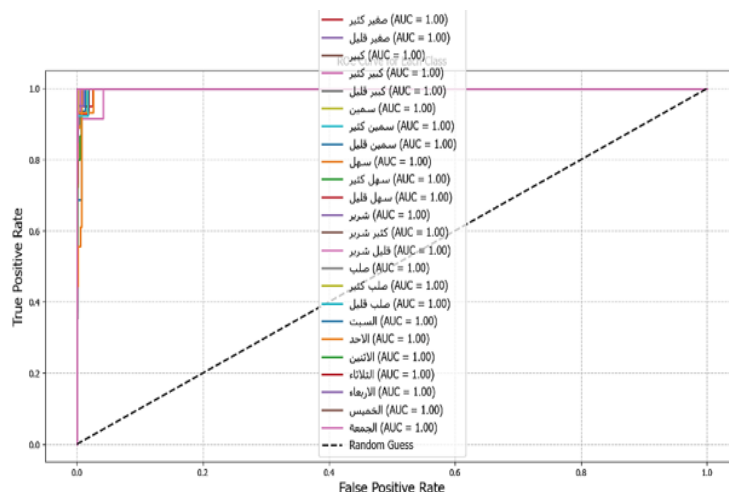


Fig. 8. Two-LSTM models

Among all the tested models after the days of the week were added, the two-LSTM + CNN model performed the best, having the highest accuracy

(94%), precision (95%), recall (94%) and F1-score (94%). The training and validation accuracy and loss are shown in Fig. 9.

Finally, the confusion matrix shows the model's predictions versus real sign labels, as shown in Fig. 10. The confusion matrix is an essential tool for assessing and comprehending the sign language

detection model's performance and accuracy score. It offers insightful information on the model's advantages and disadvantages in identifying various indicators.

Table 4,
Evaluation results

NO.	Model description	Precision	Recall	F1-score	Accuracy	Epoch	Total params	Trainable params
1	One-LSTM	0.78	0.74	0.74	0.74	81	4,612,525 (17.60 MB)	≈ 12.48 minutes
2	Two-LSTM - 80 20	0.95	0.94	0.94	0.94	156	8,547,757 (32.61 MB)	≈ 40.1 minutes
3	Two-LSTM + CNN - 80 20	0.95	0.94	0.94	0.94	156	8,547,757 (32.61 MB)	≈ 49 minutes
4	Two-LSTM + VGG16 - 80 20	0.92	0.92	0.92	0.91	146	8,547,757 (32.61 MB)	≈ 1 hour
5	Two-LSTM + VGG19 - 80 20	0.49	0.49	0.46	0.49	110	8,547,757 (32.61 MB)	≈ 38 minutes
6	LSTM + ResNet152 - 80 20	0.89	0.88	0.88	0.88	154	8,547,757 (32.61 MB)	≈ 37 minutes
7	Two-LSTM + EfficientNetB7 - 80 20	0.65	0.61	0.59	0.61	126	8,547,757 (32.61 MB)	≈ 37.82 minutes
8	Two-LSTM + DenseNet201 - 80 20	0.93	0.93	0.93	0.93		8,547,757 (32.61 MB)	≈ 26 minutes
9	Two-LSTM + SwimTrans - 80 20	0.75	0.74	0.73	0.74	176	8,547,757 (32.61 MB)	≈ 43.43 minutes
10	Two-LSTM + ConvNext - 80 20	0.93	0.92	0.92	0.92	130	8,547,757 (32.61 MB)	≈ 50.40 minutes

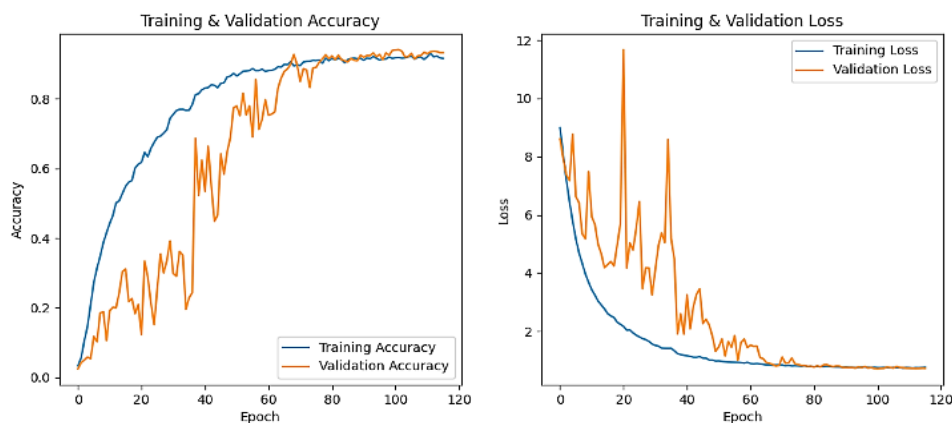


Fig. 9. ROC for the proposed model after adding the days of the week

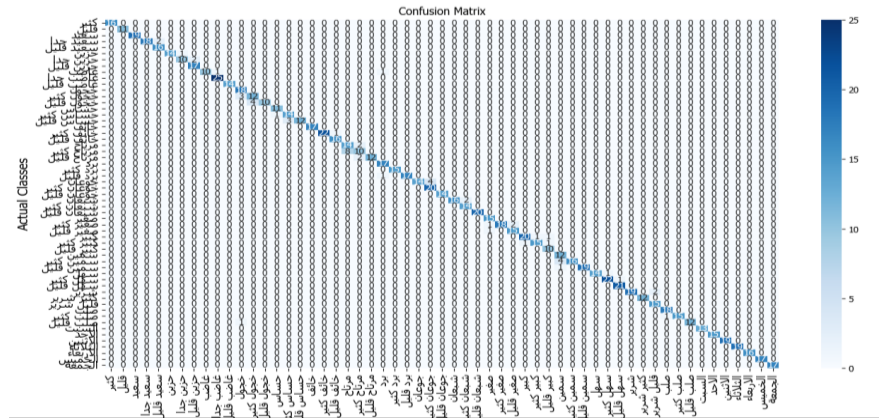


Fig. 10. Confusion matrix after adding the days of the week

Figs. 11 and 12 show the statistical calculation for the results.

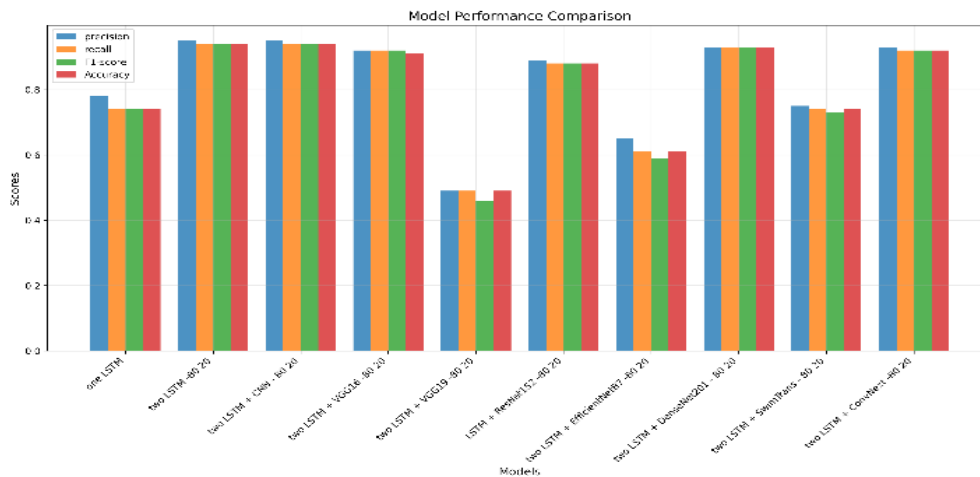


Fig. 11. Model performance comparison

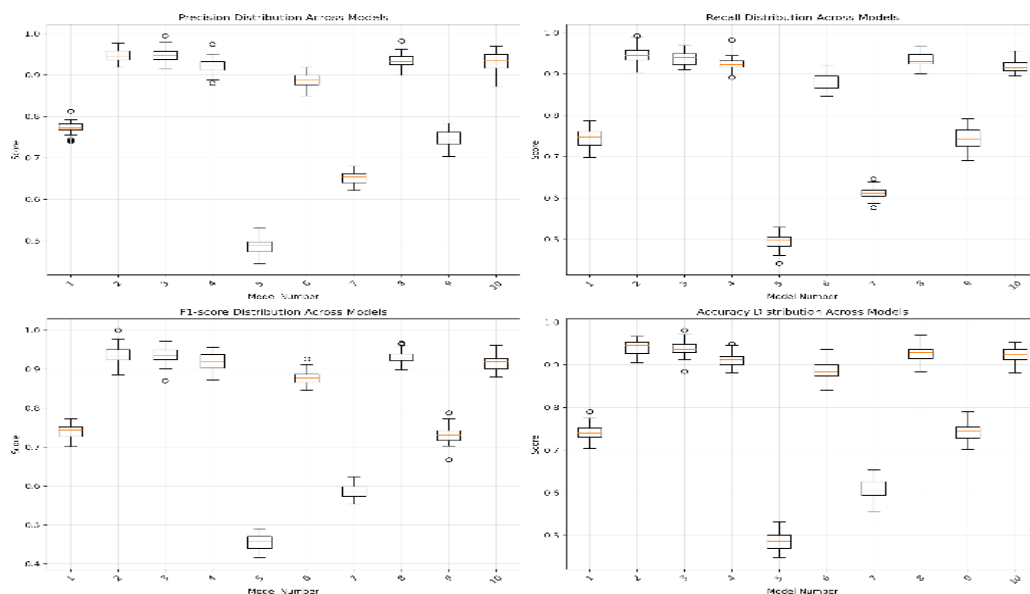


Fig. 12. Model performance comparison

4. Conclusion

The proposed model is a suitable solution to facilitate communication between the Deaf and non-speaking individuals and the speaking community. The novelty introduced by the hybrid CNN model is that it uses a classifier for obtaining spatial characteristics from input in sign language, and it also uses an LSTM classifier to record the temporal and physical characteristics needed for sequential data processing.

The proposed model obtained impressive results after being trained on the massive ArSL dataset, which contains 6,748 samples that correspond to 50 ArSL signs. In training the proposed model on the ArSL dataset, the hybrid CNN-LSTM model predicts and translates sign language to text successfully with an accuracy of 95%. The CNN and LSTM were combined in the proposed model, and their individual performance was improved through optimisation techniques.

Our work added three other signers who were recorded gesturing the days of the week in sign language, and the video was added to the ArSL dataset. The updated dataset was trained by the proposed hybrid CNN-LSTM model, and results show that the model achieved 94% accuracy in translation and a confusion matrix with 100% detection.

Future improvements could involve modifying the proposed model for use in real time and involve a computer vision algorithm. By meeting the demands of the Deaf and non-speaking individuals, the proposed model has the potential to increase accessible translation services, which would improve accessibility, encourage effective communication and strengthen ties with this significant community.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

References

- [1] A. M. Mahmoud Ibrahim and H. H. Kamel, "Social welfare services as a mechanism to reduce social exclusion of the deaf and mute," *Egyptian Journal of Social Work*, vol. 14, no. 1, pp. 35–56, 2022, doi: <https://doi.org/10.21608/ejsw.2022.129263.1158>.
- [2] L. Sirch, L. Salvador, and A. Palese, "Communication difficulties experienced by deaf male patients during their in-hospital stay: findings from a qualitative descriptive study," *Scand. J. Caring Sci.*, vol. 31, no. 2, pp. 368–377, 2017, doi: <https://doi.org/10.1111/scs.12356>.
- [3] S. Bae, "A Study on the Difficulty of Communication through Sign Language in Non-English Speaking Countries," *Open J. Soc. Sci.*, vol. 11, no. 10, pp. 494–506, 2023, doi: <https://doi.org/10.4236/jss.2023.1110028>.
- [4] T. Jamil, "Design and implementation of an intelligent system to translate arabic text into arabic sign language," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, 2020, pp. 1–4. doi: <https://doi.org/10.1109/CCECE47787.2020.9255774>.
- [5] S. K. Mahato and R. Jeya, "Convert sign language to text with CNN," in *AIP Conference Proceedings*, AIP Publishing, 2024. doi: <https://doi.org/10.1063/5.0217230>.
- [6] A. Baihan, A. I. Alutaibi, M. Alshehri, and S. K. Sharma, "Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach," *Sci. Rep.*, vol. 14, no. 1, p. 26111, 2024, doi: <https://doi.org/10.1038/s41598-024-76174-7>.
- [7] A. A. I. Sidig, H. Luqman, S. Mahmoud, and M. Mohandes, "KArSL: Arabic sign language database," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 20, no. 1, pp. 1–19, 2021, doi: <https://doi.org/10.1145/3423420>.
- [8] H. Luqman and S. A. Mahmoud, "A machine translation system from Arabic sign language to Arabic," *Univers. Access Inf. Soc.*, vol. 19, no. 4, pp. 891–904, 2020, doi: <https://doi.org/10.1007/s10209-019-00695-6>.
- [9] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: <https://doi.org/10.1007/s10462-020-09825-6>.
- [10] V. R. Allugunti, "A machine learning model for skin disease classification using convolution neural network," *International Journal of Computing, Programming and Database Management*, vol. 3, no. 1, pp. 141–147, 2022, doi: <https://doi.org/10.33545/27076636.2022.v3.i1b.53>.

- [11] K. Zarzycki and M. Ławryńczuk, "Advanced predictive control for GRU and LSTM networks," *Inf. Sci. (N. Y.)*, vol. 616, pp. 229–254, 2022, doi: <https://doi.org/10.1016/j.ins.2022.10.078>.
- [12] B. C. Mateus, M. Mendes, J. T. Farinha, and A. M. Cardoso, "Anticipating future behavior of an industrial press using LSTM networks," *Applied Sciences*, vol. 11, no. 13, p. 6101, 2021, doi: <https://doi.org/10.3390/app11136101>.
- [13] S. M. S. Abdullah and A. M. Abdulazeez, "Facial expression recognition based on deep learning convolution neural network: A review," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 53–65, 2021, doi: <https://doi.org/10.30880/jscdm.2021.02.01.006>.
- [14] V. Passricha and R. K. Aggarwal, "A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1261–1274, 2019, doi: <https://doi.org/10.1515/jisys-2018-0372>.
- [15] T. Lees *et al.*, "Hydrological concept formation inside long short-term memory (LSTM) networks," *Hydrology and Earth System Sciences Discussions*, vol. 2021, pp. 1–37, 2021, doi: <https://doi.org/10.5194/hess-26-3079-2022.2022>.
- [16] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Comput. Ind.*, vol. 131, p. 103498, 2021, doi: <https://doi.org/10.1016/j.compind.2021.103498>.
- [17] T. H. Noor *et al.*, "Real-time arabic sign language recognition using a hybrid deep learning model," *Sensors*, vol. 24, no. 11, p. 3683, 2024, doi: <https://doi.org/10.3390/s24113683>.
- [18] K. D. Ismael and E. A. Saeed, "Controlling Mobile Robot Navigation Equipped with Computer Vision Perception for Text using OCR Technique," in *2024 21st International Multi-Conference on Systems, Signals & Devices (SSD)*, IEEE, 2024, pp. 1–7. doi: <https://doi.org/10.1109/SSD61670.2024.10549002>.
- [19] A. M. A. Moustafa *et al.*, "Arabic sign language recognition systems: A systematic review," *Indian Journal of Computer Science and Engineering*, vol. 15, pp. 1–18, 2024, doi: <https://doi.org/10.21817/indjcse/2024/v15i1/241501008>.
- [20] M. A. Bencherif *et al.*, "Arabic sign language recognition system using 2D hands and body skeleton data," *IEEE Access*, vol. 9, pp. 59612–59627, 2021, doi: <https://doi.org/10.1109/ACCESS.2021.3069714>.
- [21] Q. Bani Baker, N. Alqudah, T. Alsmadi, and R. Awawdeh, "Image-Based Arabic Sign Language Recognition System Using Transfer Deep Learning Models," *Applied Computational Intelligence and Soft Computing*, vol. 2023, no. 1, p. 5195007, 2023, doi: <https://doi.org/10.1155/2023/5195007>.
- [22] S. Das, M. Chakraborty, and B. Purkayastha, "A review on sign language recognition (slr) system: ML and dl for slr," in *2021 IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT)*, IEEE, 2021, pp. 177–182. doi: <https://doi.org/10.1109/ICISSGT52025.2021.00045>.
- [23] K. Amrutha and P. Prabu, "ML based sign language recognition system," in *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, IEEE, 2021, pp. 1–6. doi: <https://doi.org/10.1109/ICITIIT51526.2021.9399594>.
- [24] S. Das, S. K. Biswas, and B. Purkayastha, "A deep sign language recognition system for Indian sign language," *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1469–1481, 2023, doi: <https://doi.org/10.1007/s00521-022-07840-y>.
- [25] A. Wali, R. Shariq, S. Shoaib, S. Amir, and A. A. Farhan, "Recent progress in sign language recognition: a review," *Mach. Vis. Appl.*, vol. 34, no. 6, p. 127, 2023, doi: <https://doi.org/10.1007/s00138-023-01479-y>.
- [26] N. Ganatra and A. Patel, "A comprehensive study of deep learning architectures, applications and tools," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 12, pp. 701–705, 2018, doi: <https://doi.org/10.26438/ijcse/v6i12.701705>.
- [27] "mArSL-A multimodal manual and non-Manual Arabic sign language dataset." Accessed: Apr. 27, 2025. [Online]. Available: <https://hamzah-luqman.github.io/marsl/#>
- [28] B. Chen, T.-J. Chin, and M. Klimavicius, "Occlusion-robust object pose estimation with holistic representation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2929–2939. doi: <https://doi.org/10.1109/WACV51458.2022.00228>.

- [29] A. K. Singh, V. A. Kumbhare, and K. Arthi, "Real-time human pose detection and recognition using mediapipe," in *International conference on soft computing and signal processing*, Springer, 2021, pp. 145–154. doi: https://doi.org/10.1007/978-981-16-7088-6_12.
- [30] A. Chaikaew, "An applied holistic landmark with deep learning for Thai sign language recognition," in *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, IEEE, 2022, pp. 1046–1049. doi: <https://doi.org/10.1109/ITC-CSCC55581.2022.9895052>.
- [31] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sens. (Basel)*, vol. 13, no. 22, p. 4712, 2021, doi: <https://doi.org/10.3390/rs13224712>.
- [32] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS journal of photogrammetry and remote sensing*, vol. 173, pp. 24–49, 2021, doi: <https://doi.org/10.1016/j.isprsjprs.2020.12.010>.
- [33] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2021, doi: <https://doi.org/10.1109/TNNLS.2021.3084827>.
- [34] A. Shah *et al.*, "A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN)," *Clinical eHealth*, vol. 6, pp. 76–84, 2023, doi: <https://doi.org/10.1016/j.ceh.2023.08.002>.
- [35] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, IEEE, 2021, pp. 96–99. doi: <https://doi.org/10.1109/CENTCON52345.2021.9687944>.
- [36] H. Yang, J. Ni, J. Gao, Z. Han, and T. Luan, "A novel method for peanut variety identification and classification by Improved VGG16," *Sci. Rep.*, vol. 11, no. 1, p. 15756, 2021, doi: <https://doi.org/10.1038/s41598-021-95240-y>.
- [37] D. Albashish, R. Al-Sayyed, A. Abdullah, M. H. Ryalat, and N. A. Almansour, "Deep CNN model based on VGG16 for breast cancer classification," in *2021 International conference on information technology (ICIT)*, IEEE, 2021, pp. 805–810. doi: <https://doi.org/10.1109/ICIT52682.2021.9491631>.
- [38] S. Sharma and K. Guleria, "A deep learning model for early prediction of pneumonia using VGG19 and neural networks," in *Mobile Radio Communications and 5G Networks: Proceedings of Third MRCN 2022*, Springer, 2023, pp. 597–612. doi: https://doi.org/10.1007/978-981-19-7982-8_50.
- [39] A. Karacı, "VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm," *Neural Comput. Appl.*, vol. 34, no. 10, pp. 8253–8274, 2022, doi: <https://doi.org/10.1007/s00521-022-06918-x>.
- [40] O. C. Do, C. M. Luong, P.-H. Dinh, and G. S. Tran, "An efficient approach to medical image fusion based on optimization and transfer learning with VGG19," *Biomed. Signal Process. Control*, vol. 87, p. 105370, 2024, doi: <https://doi.org/10.1016/j.bspc.2023.105370>.
- [41] S. A. Hasanah, A. A. Pravitasari, A. S. Abdullah, I. N. Yulita, and M. H. Asnawi, "A deep learning review of resnet architecture for lung disease Identification in CXR Image," *Applied Sciences*, vol. 13, no. 24, p. 13111, 2023, doi: <https://doi.org/10.3390/app132413111>.
- [42] N. Jain and P. Peddi, "Gender Classification Model based on the Resnet 152 Architecture," in *2023 IEEE International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2023, pp. 1–7. doi: <https://doi.org/10.1109/ICCST59048.2023.10474266>.
- [43] M. K. Panda, B. N. Subudhi, T. Veerakumar, and V. Jakhetiya, "Modified ResNet-152 network with hybrid pyramidal pooling for local change detection," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 4, pp. 1599–1612, 2023, doi: <https://doi.org/10.1109/TAI.2023.3299903>.
- [44] M. Mahasin and I. A. Dewi, "Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 backbones on YOLO v4 as object detector," *International journal of engineering, science and information*

- technology, vol. 2, no. 3, pp. 64–72, 2022, doi: <https://doi.org/10.52088/ijesty.v1i4.291>.
- [45] J. Frederich, J. Himawan, and M. Rizkinia, “Skin lesion classification using EfficientNet B0 and B1 via transfer learning for computer aided diagnosis,” in *AIP Conference Proceedings*, AIP Publishing, 2024. doi: <https://doi.org/10.1063/5.0200741>.
- [46] S. Abd El-Ghany, M. A. Mahmood, and A. A. Abd El-Aziz, “Adaptive Dynamic Learning Rate Optimization Technique for Colorectal Cancer Diagnosis Based on Histopathological Image Using EfficientNet-B0 Deep Learning Model,” *Electronics (Basel)*, vol. 13, no. 16, p. 3126, 2024, doi: <https://doi.org/10.3390/electronics13163126>.
- [47] H. A. Sanghvi, R. H. Patel, A. Agarwal, S. Gupta, V. Sawhney, and A. S. Pandya, “A deep learning approach for classification of COVID and pneumonia using DenseNet-201,” *Int. J. Imaging Syst. Technol.*, vol. 33, no. 1, pp. 18–38, 2023, doi: <https://doi.org/10.1002/ima.22812>.
- [48] F. Salim, F. Saeed, S. Basurra, S. N. Qasem, and T. Al-Hadhrani, “DenseNet-201 and Xception pre-trained deep learning models for fruit recognition,” *Electronics (Basel)*, vol. 12, no. 14, p. 3132, 2023, doi: <https://doi.org/10.3390/electronics12143132>.
- [49] T. Lu, B. Han, L. Chen, F. Yu, and C. Xue, “A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning,” *Sci. Rep.*, vol. 11, no. 1, p. 15824, 2021, doi: <https://doi.org/10.1038/s41598-021-95218-w>.
- [50] S. Liu *et al.*, “Consmax: Hardware-friendly alternative softmax with learnable parameters,” in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9. doi: <https://doi.org/10.1145/3676536.3676766>.
- [51] M. Franke and J. Degen, “The softmax function: Properties, motivation, and interpretation,” 2023, doi: <https://doi.org/10.31234/osf.io/vsw47>.
- [52] H. Çetiner and İ. Çetiner, “Classification of cataract disease with a DenseNet201 based deep learning model,” *Journal of the Institute of Science and Technology*, vol. 12, no. 3, pp. 1264–1276, 2022, doi: <https://doi.org/10.21597/jist.1098718>.
- [53] G. Wang, “RL-CWtrans Net: multimodal swimming coaching driven via robot vision,” *Front. Neurobot.*, vol. 18, p. 1439188, 2024, doi: <http://dx.doi.org/10.3389/fnbot.2024.1439188>.
- [54] H. Chen and X. Yue, “Swimtrans Net: a multimodal robotic system for swimming action recognition driven via Swin-Transformer,” *Front. Neurobot.*, vol. 18, p. 1452019, 2024, doi: <http://dx.doi.org/10.3389/fnbot.2024.1452019>.
- [55] X. Ke, X. Zhang, T. Zhang, J. Shi, and S. Wei, “Sar ship detection based on swin transformer and feature enhancement feature pyramid network,” in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2022, pp. 2163–2166. doi: <https://doi.org/10.1109/IGARSS46834.2022.9883800>.

التحويل الآلي للغة الإشارة العربية إلى نص مكتوب بناءً على نموذج CNN-LSTM الهجين للتواصل مع الصم والبكم

خمساء ضياء اسماعيل^١، ايلاف احمد سعيد^{٢*}، عمار ابراهيم مجيد^٣
^{١,٢,٣} قسم هندسة الأتمتة والذكاء الاصطناعي، كلية هندسة المعلومات، جامعة النهرين، بغداد، العراق
 * البريد الإلكتروني: elaf.ahmed@nahrainuniv.edu.iq

المستخلص

يُعد التواصل مع الصم والبكم وفهم لغة الإشارة عملية بالغة الصعوبة. ومع ذلك، في ظل التطور المستمر في التعلم العميق وأتمتة الأنظمة، يُمكن تقديم مجموعة من الحلول للتغلب على صعوبة التواصل مع هذه الفئة من المجتمع وفهم لغة الإشارة، وخاصة في المناطق التي تُحدث فيها اللغة العربية. تقترح هذه المقالة نظاماً آلياً للتعرف على لغة الإشارة يعتمد على بنية CNN-LSTM الهجينة، والتي تمثل طريقتين: الأولى هي CNN (الشبكة العصبية التلافيفية)، والثانية تتكون من LSTM (الذاكرة طويلة المدى). بينما تُسجل LSTM الارتباطات الزمنية ضمن تسلسلات الإيماءات، تستخلص CNN السمات المكانية من حركات لغة الإشارة. تم إثبات قدرة الوحدة على توفير تواصل سلس من خلال تقييم أدائها باستخدام مجموعة بيانات ArSL تم إنشاؤها خصيصاً في المرحلة الأولى من المقالة، وتم تمثيل المرحلة الثانية من خلال تدريب الوحدة واختبارها من خلال إدخال مقاطع فيديو تم تسجيلها لثلاثة من مستخدمي الإشارة، كل واحد منهم ينطق ٥٠ إشارة مختلفة، وتمت إضافة ثلاثة مستخدمين آخرين لنطق أيام الأسبوع بلغة الإشارة. تمثل مقاطع الفيديو المسجلة هذه البيانات المضافة إلى مجموعة بيانات ArSI، عندما يتم استخراج النقطة الرئيسية باستخدام matplotlib، متبوعاً بجمع قيم النقاط الرئيسية للتدريب والاختبار، وأخيراً، سيتم معالجة البيانات مسبقاً ووضع علامات على الميزات لكل لغة إشارة منطوقة، وستكون هذه العلامات باللغة العربية، وتكتشف الوحدة المقترحة عمل لغة الإشارة والتعرف وفقاً لتدريب مجموعة البيانات، وتمت الخطوتان من خلال بناء وتدريب الشبكة العصبية LSTM و CNN، وكان معدل الدقة للغة الإشارة العربية في مكون النظام المقترح الذي يترجمها إلى نص عربي ٩٥٪، ودرجة f1 95٪، والتذكر (٩٥٪)، والدقة (٩٦٪)، مع العصر (١٢٦)، وتطبيع الدفعة يساوي ٣٠، والنتيجة النهائية سيتم التعرف على لغة الإشارة العربية والكشف عنها.