Arabic Sign Language Recognition Using Image-Based Deep Learning and Edge User-Centered Interface

Ahmad Alzu'bi[1(✉)], Amjad Albashayreh[2], and Lojin Bani Younis[1]

[1] Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan
agalzubi@just.edu.jo, lhbaniyounis19@cit.just.edu.jo

[2] Faculty of Information Technology, Applied Science Private University, Amman, Jordan
a_albashayreh@asu.edu.jo

Abstract. A growing trend in assisted living involves the use of machine learning techniques to recognize hand gesture patterns tailored for people with disabilities. However, in recognition systems, the limited availability of sign language data gives rise to both data scarcity and privacy concerns. In this paper, a federated deep learning architecture is proposed for Arabic sign language recognition, aimed at addressing the challenge of deciphering the meanings conveyed by image-based hand gestures. The distributed client-server paradigm of federated learning is based on a distributed stochastic gradient descent (SGD) optimizer with a federated averaging mechanism. In addition, an interactive user interface is designed to manage distributed learning on smartphones through a client-server model, where several edge nodes collaborate to jointly learn the discriminating features of confidential data without breaching its privacy. This interactive interface improves accessibility for people with deafness or impairment using image gestures and navigation panels. Several deep learning backbones were employed in the procedure of transfer learning to fit the optimum fine-tuning setup while maximizing recognition accuracy. The experimental results show the effectiveness of the proposed FL model, achieving an accuracy of 98.80% by the FL-VGG19 configuration. There is also a significant reduction in the number of training epochs and rounds, resulting in a decrease in the computational complexity for convergence. This demonstrates the model's capabilities in recognizing Arabic sign language and improving the communication experience for people with disabilities.

Keywords: Federated deep learning · Arabic sign recognition · Smartphone accessibility · Gesture user interface · Communication disabilities

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15886, pp. 243–259, 2026.
https://doi.org/10.1007/978-3-031-97576-9_16

1 Introduction

Sign language (SL) is a visual communication method utilizing gestures, facial expressions, and body movements used by individuals with disabilities, such as deaf or hard-of-hearing individuals, to convey meaning. Because sign language is the primary mode of communication for millions of people worldwide, there is a lot of interest in the potential applications of robust sign language recognition (SLR) tools [1]. Considering the broad range of possibilities, such assisted living solutions could go beyond only translation by facilitating accessible sign language broadcasts, fostering the development of responsive devices that operate seamlessly via sign language commands, and even pioneering sophisticated systems specifically designed to assist individuals with impairments in performing daily tasks more independently. In recent years, there has been a significant focus on the SLR domain with various approaches [2–4] proposed to address the communication challenges faced by people with disabilities. Deep neural networks (DNNs) have emerged as a promising technological advancement for researchers, poised to change the SLR [5]. These networks offer the potential for ground-breaking advances in the automated recognition of hand gestures and signals used in interactions between people who have difficulty speaking or hearing.

However, there are challenges along the way to develop SLR tools that are useful and broadly applicable. Maintaining the accuracy, dependability, and speed of the interpretation algorithms remains a top priority. Errors or delays in interpretation can prevent effective communication and usability; therefore, achieving a delicate balance between precision and real-time processing is critical. Advances in technology and research endeavors are facilitating the ongoing exploration of novel approaches, datasets, and model architectures to improve the performance of SLR systems [6–8]. Collaboration among experts in linguistics, computer vision, and machine learning is crucial to advance this field [9]. Additionally, the presence of extensive databases that encompass a diverse array of sign language gestures is vital for the accurate training and evaluation of SLR systems.

Federated learning (FL) is an emerging paradigm of machine learning associated with decentralized methods, which is shown to be an effective approach to training shared global models [10,11]. FL methods entail coordinating the training of a central model from a collection of participating devices. When training data is sourced from user interactions with mobile applications, for example, a significant application scenario for FL arises [10]. In this context, FL enables mobile phones to collectively learn a shared prediction model while retaining all training data on the device, effectively separating the machine learning capability from the requirement of storing data in the cloud. Instead of storing training data in the cloud, it is kept on users' mobile devices, which act as intelligent edge nodes, performing computations on their local data to update a global model. This approach goes beyond the use of local models for mobile device predictions by bringing model training to the device level.

FL has enormous potential to transform the development of SLR systems. In the context of SLR, this approach provides a promising solution to the challenges of privacy preservation, data diversity, and model adaptability [12]. Individuals in the deaf and hard-of-hearing communities may be hesitant to share their information due to the sensitivity of sign language data. FL addresses this issue by allowing model training to take place locally on the devices of users or decentralized servers. Furthermore, FL allows for the creation of more diverse datasets without the need to centralize data from multiple sources. SLR models trained using federated learning are expected to benefit from decentralized data from a variety of geographic locations, cultural backgrounds, and signing styles. This could make the SLR systems more reliable, allowing them to take into account variations in sign language gestures and expressions effectively.

Arabic sign language (ArSL) encompasses a rich vocabulary and intricate structures. Much like other languages, it involves the combination of hand shapes, orientations, motion, and facial expressions to convey various meanings. To improve the accuracy of Arabic sign language recognition, ElBadawy et al. [13] emphasized the effectiveness of a behavior-based feature extractor and 3D CNN for recognizing gestures in ArSL using depth map data. Zakariah et al. [14] tested different pre-trained models on the ArASL2018 dataset [15]. Saleh and Issa [16] also used transfer learning and fine-tuning of pretrained CNN models, including VGG16 and ResNet152. Similarly, Kamruzzaman [17] proposed a vision-based method to recognize Arabic hand sign-based letters and translate them into spoken Arabic using deep CNNs. Alawwad et al. [18] examined a faster RCNN framework with VGG-16 and ResNet-18 networks, achieving an accuracy of 93.4%. Hayani et al. [19] presented a CNN-based model to identify ArSL characters, including both the alphabet and numerical digits with 2030 images of numbers (from 0 to 10) and 5839 images of letters. Aldhahri et al. [20] developed a model to recognize Arabic letters using MobileNet. Several 3D-based SLR systems [21–23] used the depth and motion information captured in the three-dimensional space, showing efforts to improve communication accessibility for individuals who rely on sign language.

Although various deep learning algorithms have been applied to recognize Arabic sign language, prior studies did not employ federated learning architectures. Their reliance on centralized data storage and processing mechanisms may limit them in meeting the growing demand for decentralized, privacy-preserving, and scalable solutions in ARSL model development. This motivated us to address this gap by using and investigating a federated deep learning model to recognize the Arabic sign language, ensuring privacy for individuals with disabilities, and providing high performance with low time complexity. The primary contribution of this research lies in the introduction of an efficient federated deep learning pipeline for Arabic sign language recognition based on still images. In addition, an interactive use interface is designed to improve accessibility for individuals with communication disabilities and to facilitate federated deep learning of ArSL data through smartphones, serving as intelligent edge nodes. This allows model training to take place locally on the local devices of users or decentral-

ized servers. This method will also ensure that sensitive data remains on users' devices and that only model updates are transmitted to a central server, protecting the privacy of confidential information. Therefore, this approach enables for more accurate and robust recognition of gestures in diverse environments and conditions. Finally, optimization of training time complexity is achieved through comprehensive analytical experiments that utilize transfer learning with various deep learning backbones.

The remaining part of this paper is organized as follows. Section 2 introduces the methodology of the FL-based framework, Sect. 3 presents the experimental results, Sect. 4 discusses the main findings, and Sect. 5 concludes this paper.

2 Methodology

The method we adopted begins by processing the ArSL images. Then, the architecture of federated deep learning is constructed, leveraging several pretrained deep learning backbones. An interactive user interface is designed for the FL-based ArSLR model to facilitate the user experience and enhance the online learning of extracted gesture features through smartphones. The performance of the proposed FL-based model is finally evaluated using several standard metrics in various setups. These phases are detailed in the following subsections.

2.1 Dataset Preparation

The benchmarking dataset used in this work is the Arabic Alphabet Sign Language (ARASL) dataset [15] that comprises 54,049 images of hand gestures representing the Arabic alphabets, designed to aid the deaf in comprehending the language and freely expressing their thoughts and emotions. This dataset includes 32 classes corresponding to Arabic letters, each class containing a specific number of images, as detailed in Table 1. A sample of ARASL hand-gesture images is shown in Fig. 1.

In the data preparation phase, the collections of hand gesture images have been further processed to be suitable for the configuration of federated learning architecture. The image processing procedure includes the use of a custom dataset module to simplify the transfer learning process. This module initializes the data and outlines images and labels, ensuring efficient data preparation. The data undergo a transformation pipeline consisting of three stages: image resizing, tensor conversion, and normalization. Resizing ensures uniform dimensions for all images, while tensor conversion converts data into a multidimensional array suitable for deep learning frameworks. The normalization process sets pixel values to a consistent scale between 0 and 1, handling explosive gradients during training, and improving the model's resilience to fluctuations in pixel intensity and color across different images. Finally, the image collection is divided into 70% for training, 10% for validation, and 20% for testing. The three evaluation splits are further subdivided into multiple subsets for each processing node, an essential phase for emulating distinct datasets on decentralized clients within the federated learning framework.

Table 1. The classes and images of Arabic alphabet signs in the ASASL dataset.

#	Letter in English Script	Letter in Arabic Script	Images	#	Letter in English Script	Letter in Arabic Script	Images
1	Alif	(ألف) أ	1672	17	Za	(ظاء) ظ	1723
2	Ba	(باء) ب	1791	18	Ain	(عَين) ع	2114
3	Ta	(تاء) ت	1838	19	Ghayn	(غين) غ	1977
4	Tha	(ثاء) ث	1766	20	Fa	(فاء) ف	1955
5	Jim	(جيم) ج	1552	21	Qaf	(قاف) ق	1705
6	Ha	(حاء) ح	1526	22	Kaf	(كاف) ك	1774
7	Kha	(خاء) خ	1607	23	Lam	(لام) ل	1832
8	Dal	(دال) د	1634	24	Mim	(ميم) م	1765
9	Dhal	(ذال) ذ	1582	25	Nun	(نون) ن	1819
10	Ra	(راء) ر	1659	26	Ha	(هاء) ه	1592
11	Zay	(زاي) ز	1374	27	Waw	(واو) و	1371
12	Sin	(سين) س	1638	28	Ya	(يا) ئ	1722
13	Shin	(شين) ش	1507	29	Taa	(ة) ة	1791
14	Sad	(صاد) ص	1895	30	Al	(ال) ال	1343
15	Dad	(ضاد) ض	1670	31	Laa	(لا) لا	1746
16	Ta	(طاء) ط	1816	32	Yaa	(ياء) ياء	1293



Fig. 1. Sample images from the ARASL dataset with the corresponding Arabic letters.

2.2 Federated Deep Learning Architecture

As can be observed in Fig. 2, the generic pipeline of federated deep learning consists of two main components which are a central server with a global model and multiple clients (M clients) with multiple local models. Each client (indexed from 0 to M) possesses a subset of hand-gesture images D_i consisting of n_i labeled samples drawn from a distribution D_i over input space X and label space Y. The learning model $h_\theta : X \rightarrow Y$ is parameterized by $\theta \in \mathbb{R}^D$. The loss of the model on a sample $(x, y) \in X \times Y$ is denoted by $`(h_\theta(x), y)$, in which we used cross-entropy loss. The loss of the model parameters θ in the i-th client is the average loss of the model h_θ in the samples in D_i, calculated as $f_i(\theta) = \frac{1}{n_i} \sum_{j=1}^{n_i} l(h_\theta(x_{i,j}, y_{i,j}))$, where $(x_{i,j}, y_{i,j})$ represents the j-th sample in D_i. This server configuration aims to leverage data from all clients to identify models that minimize loss $f_i(\theta)$ for each client. To do this, the typical method entails choosing a single model θ that minimizes the average of client losses weighted by the number of samples as follows:

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j \in \hat{D}_i} `(h_\theta(x_{i,j}), y_{i,j})), \tag{1}$$

where $N = \sum_{i=1}^{M} n_i$. Due to communication and privacy constraints, clients cannot share their local data $\hat{D}_i$. Upon receiving θ_t, each client performs local computations as follows:

$$\theta_{t,i,s+1} = \theta_{t,i,s} - \alpha g_{t,i,s}(\theta_{t,i,s}) \tag{2}$$

for $s = 1, \ldots, \tau - 1$, where τ represents the number of local steps, $\theta_{t,i,0} = \theta_t$, $g_{t,i,s}(\theta_{t,i,s})$ is the stochastic gradient of f_i evaluated at $\theta_{t,i,s}$ using b samples from $\hat{D}_i$. The clients then transmit $\theta_{t,i,\tau}$ back to the server. Finally, the server computes the next global iterate as follows:

$$\theta_{t+1} = \frac{1}{N_t} \sum_{i \in I_t} n_i \theta_{t,i,\tau}, \tag{3}$$

where $N_t := \sum_{i \in I_t} n_i$.

In this paper, federated averaging (FedAvg) [24] is applied for federated learning with a network of five clients. The FedAvg approach improves communication efficiency by performing multiple ($\tau \geq 2$) local updates between communication rounds in distributed stochastic gradient descent (D-SGD). The training phase is applied in 10 epochs and 10 rounds with iterative model modifications. This learning procedure entails thorough data preparation, followed by an equal distribution of transformed data among the five clients. In each iteration t, a subset I_t of $m \leq M$ clients is uniformly sampled by the server. These selected clients receive the current global parameters θ_t and perform multiple stochastic gradient descent (SGD) steps on their local data starting from

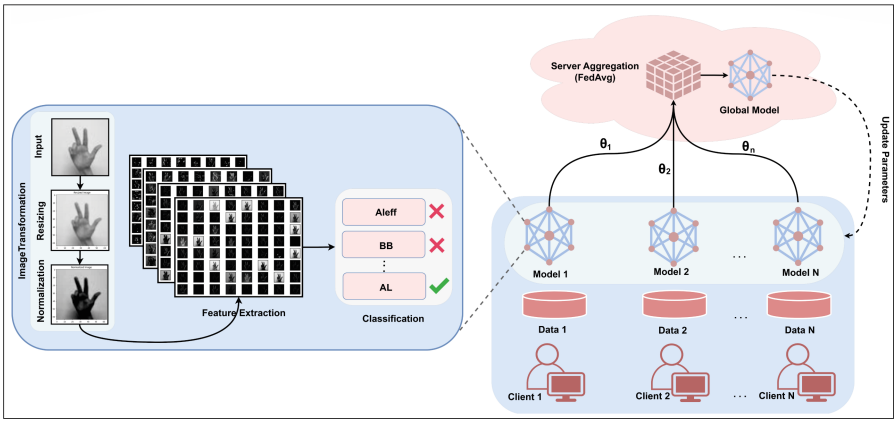

Fig. 2. The pipeline of the proposed FL-based ArSL Recognition.

θ_t. With this setup, we maintain that local training occurs on individual client nodes without the need to share raw image data of people with disabilities.

In addition, the performance of deep convolutional models is evaluated during training using validation datasets split equally between clients to assess accuracy and generalization capability. Each client then transmits its model updates in the form of gradients to the central server. These updates represent the gradients of the local model's parameters concerning the loss function, illustrating how the model's parameters should be adjusted to minimize the steepest increase in the loss function. The server computes the subsequent global parameters θ_{t+1} as a weighted average of the client updates. This procedure harmonizes the local contributions of each client, resulting in an improved global hand-gesture image classification. The global model re-transmits its updates to all local models on the client side. The ten-round federated learning cycle includes local training, model modifications, aggregation, and distribution. This iterative approach enables the global image classification model of Arabic signs to converge on optimal performance, incorporating knowledge from various client datasets.

2.3 Deep Learning Backbones

The procedure of transfer learning adopted for the FL-based Arabic sign recognition model is initialized with seven distinct deep learning backbones, which are: EfficientNet-B, EfficientNet-B1, EfficientNet-B2, ResNet32, ResNet50, VGG16, and VGG19. These models were incorporated into the federated framework to facilitate the training and evaluation process on a federated network of participating client devices. Each model variation was distributed among the client devices to take advantage of their local subset of hand-gesture images for fine-tuning and updating the global model. The main characteristics of the pre-trained deep convolutional architectures are as follows:

- **EfficientNet** [25]**:** EfficientNet is a family of convolutional neural network architectures known for their efficiency and accuracy. These models are based on compound scaling, balancing network depth, width, and resolution to optimize performance. EfficientNet B0, B1, and B2 represent variants within this family, with varying depths and computational efficiency.
- **ResNet** [26]**:** ResNet (Residual Network) is a widely recognized and influential deep neural network architecture. ResNet32 and ResNet50 are variants of the ResNet architecture that differ in their depth and number of layers. ResNet32 is a shallower version, while ResNet50 is deeper, leveraging residual blocks that facilitate the training of very deep networks.
- **VGG** [27]**:** VGG (Visual Geometry Group) is a convolutional neural network architecture known for its simplicity and effectiveness. VGG16 and VGG19 refer to variants of the VGG architecture, differing in the number of layers. VGG16 has 16 layers, and VGG19 has 19 layers, both employing small 3×3 convolutional filters and max-pooling layers.

2.4 Edge User-Centered Interface

Within the federated learning paradigm designed for ArSL recognition, there is a demand to ensure optimal accessibility on edge nodes, or clients. This objective aims to ensure that applications are accessible to a wide range of individuals, regardless of their expertise, skills, or technical specifications of the devices they use. The user-centered approach needs to be followed to design a mainstream solution that performs its intended function; otherwise, there is a chance that a solution will need to be re-engineered to accommodate the unique demands of a certain set of individuals [28], such as people with disabilities.

When discussing the incorporation of a decentralized learning model and assistive technologies, it is important to note the potential benefits of integrating accessibility features into everyday applications. Some solutions proposed by researchers include the use of IoT ([29] to facilitate assistance and technology integration, thereby enhancing quality of life and connectivity. However, in this work, we aim to provide a user-centered interface design for facilitating the procedure of the new data collection and processing on the client side, i.e., an online learning of Arabic sign language. Therefore, improving accessibility in mobile devices will make ArSL recognition applications accessible to as many people as possible. Accessibility is a sub-characteristic of usability [30], and this requirement will be met by the interactive FL-based interface that we build, which enables users with communication impairments to communicate with the recognition system through edge devices such as smartphones.

Figure 3 shows the design of the processing interface offered to the clients involved in the proposed FL-based ArSL recognition system. It demonstrates the generic communication procedure initiated from the server side by broadcasting the global model to all participating edge devices. Each edge device, e.g., smartphone, confirms joining the training phase and therefore permits the application of the recognition model on the local data, i.e., sign images. The progress

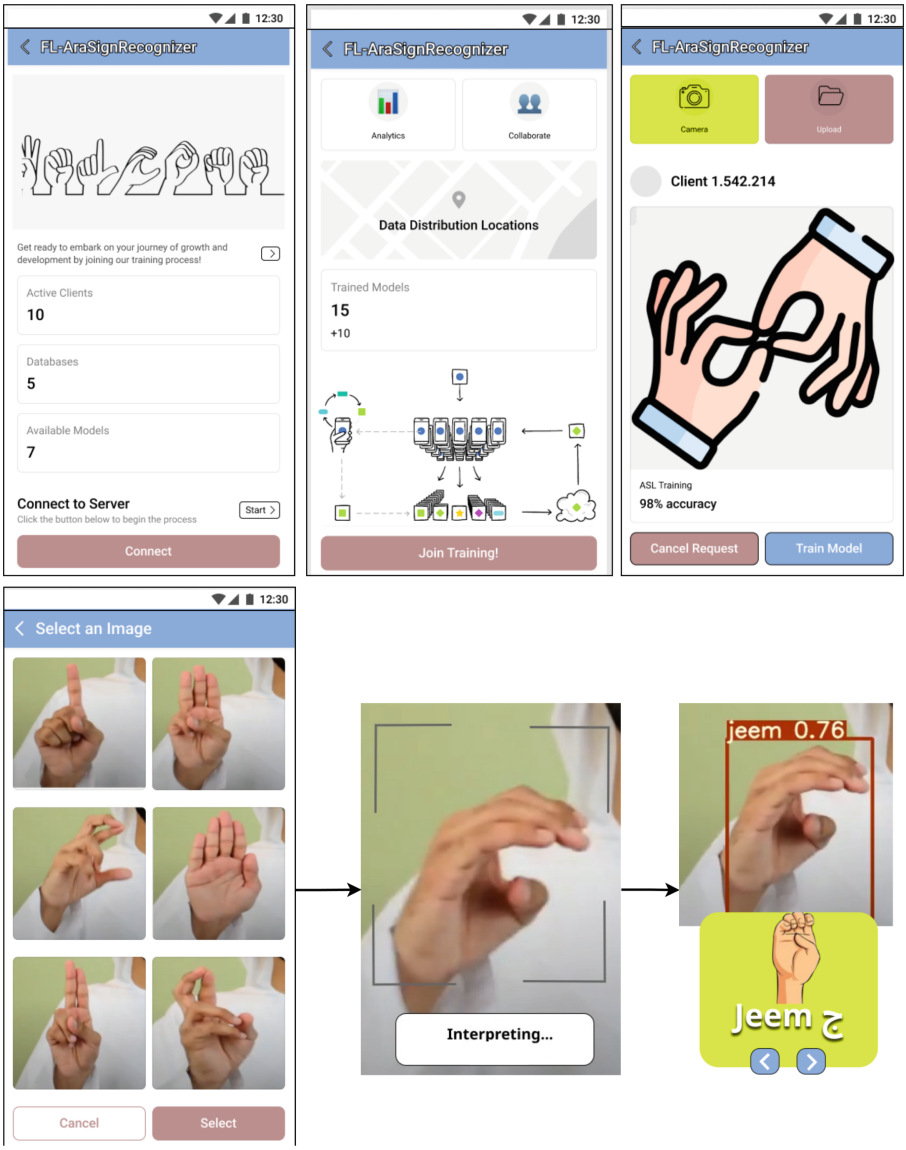

Fig. 3. The user-centered interface proposed for the FL-based ArSL Recognition.

of training is shown on the client's device, and the central server aggregates the results by which it updates the global model and broadcasts it again in the next training round. On the client side, there is the ability to submit and recognize a sign image with the final trained model. Finally, the interactive interface shows the recognition result of any Arabic sign image submitted to the system.

2.5 Evaluation Metrics

A range of performance metrics are used to assess the proposed FL-based deep model to recognize Arabic sign gestures. This comprehensive evaluation offers insights into the model's performance across diverse classification characteristics, ensuring its accuracy and capability to handle various scenarios and classes. The primary metric obtained in this work is the classification accuracy of ArSL images. A high level of accuracy suggests that the model performs well in identifying various hand movements and generating fewer classification errors. True Positive (TP) in ASL recognition occurs when the model accurately predicts the letter represented by a sign gesture. For example, if the sign gesture represents the letter "A" and the model correctly classified it as "A", it is considered a true positive. A False Positive (FP) occurs when a model predicts a letter that does not represent the sign gesture. For example, if the sign gesture represents the letter "A" but the model predicts "B", it is classified as a False Positive. True Negative (TN) occurs when the model accurately predicts that a sign gesture does not represent a specific letter. For instance, if the sign gesture does not correlate with any letter, the classification result will be a True Negative. Finally, a False Negative (FN) occurs when a model fails to accurately predict the correct letter represented by a sign gesture, such as predicting "A" instead of "K" or predicting a different letter, resulting in a false negative. Based on these essential metrics, the accuracy, precision, recall, and F1-Score are calculated.

Accuracy: this measures the number of correct predictions divided by the total number of predictions, and it is calculated as follows:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

Precision: it is the percentage of true positive predictions among all positive occurrences predicted by the model, which determines how many detected positive cases are truly positive. It can be calculated as follows:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

Recall: it is the percentage of true positive predictions, calculated by determining how many positive instances the model correctly predicted. It can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1-Score: this metric balances precision and recall with the best value at 1 and the worst value at 0, and it is calculated as follows:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

3 Results

3.1 Experiments Configuration

All experiments were carried out in a cloud-based environment using a PyTorch implementation. Seven different deep learning models were used as part of the federated deep learning framework. Several experiments were performed to identify the optimal hyperparameters for the models, ensuring a fair and consistent environment to evaluate their performance based on average time, accuracy, precision, recall, and F1-score. The models were compared under identical conditions and using identical hyperparameters. Each experiment uses 64 data samples, involving five clients trained over ten epochs. The federated averaging method aggregates gradients on the server side, while categorical cross entropy is used as a loss function. Softmax is used as an activation function, and stochastic gradient descent is used as a primary optimizer with a learning rate of 0.01. Table 2 provides a summary of the hyperparameters used for the deep FL models.

Table 2. The experimental hyperparameters.

Hyperparameter	Value
Aggregation Method	FedAvg
No. of Clients	5
No. of Epochs	10
No. of Rounds	10
Entropy	Categorical Cross-Entropy
Activation Function	Softmax
Optimizer	SGD
Learning Rate	0.01
Batch Size	64

3.2 Experimental Results

Seven experimental setups have been carried out using three versions of EfficientNet (b0, b1, and b2), two versions of ResNet (ResNet32 and ResNet50), and two versions of VGG (VGG16 and VGG19). The performance of each model has been evaluated under a consistent configuration with identical hyperparameters. Figures 4 and 5 depict the validation and testing accuracies for each model over ten rounds. The VGG models show consistent improvement from the initial to the final round, whereas the EfficientNet models demonstrate slower progress initially but show significant improvement after the fourth round.

Table 3 summarizes the performance results of all FL-based models. This evaluation procedure provides a detailed understanding of the model's performance across various classification characteristics, ensuring its correctness and

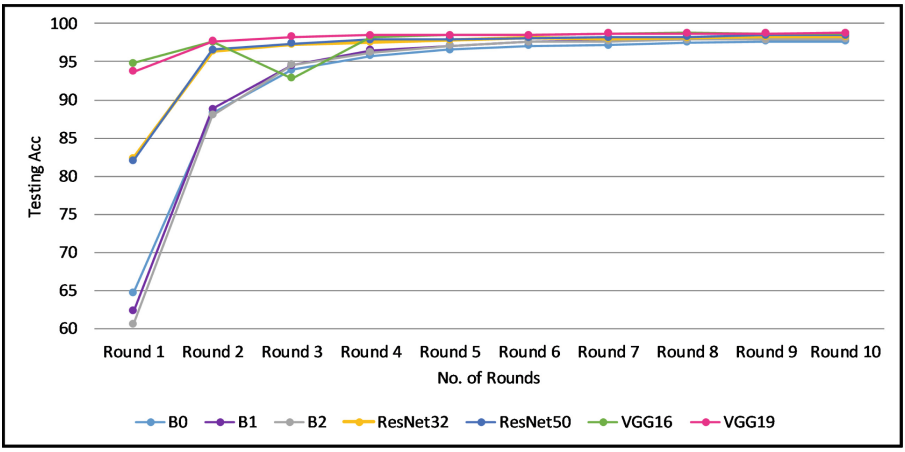

Fig. 4. The testing accuracy per round achieved by the FL-based ArSL classifiers.

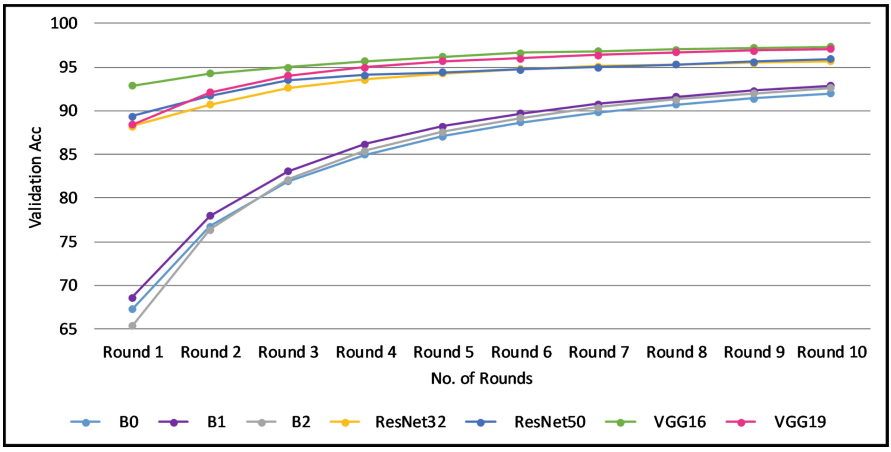

Fig. 5. The validation accuracy per round achieved by the FL-based ArSL classifiers.

ability to handle various scenarios and classes in the dataset. However, due to the imbalanced nature of the Arabic sign language dataset, the model's performance is assessed using accuracy and macro-averaging metrics. In imbalanced circumstances, macro-averaging is very useful since it treats all classes equally without being affected by the majority class. The FL-VGG19 has the highest testing performance, achieving an accuracy of 98.8%, a precision of 98.79%, a recall of 98.78%, and an F1-score of 98.78%. However, FL-VGG16 and FL-VGG19 showed comparable performance in all metrics. FL-VGG16 achieved slightly higher validation accuracy, with a marginal difference of 0.2%, underscoring the minor discrepancies between the performance of the two models.

Table 3. The Performance results of the proposed deep FL Models in 10 rounds.

Model	Val Acc	Test Acc	Precision	Recall	F1-Score	Client Time[1]	Round Time[2]
FL-B0	92.00	97.70	97.74	97.72	97.72	44	3.67
FL-B1	92.90	98.10	98.16	98.14	98.14	55	4.58
FL-B2	92.60	98.00	98.02	98.00	98.00	57	4.75
FL-ResNet32	95.70	98.30	98.28	98.26	98.27	**33**	**2.75**
FL-ResNet50	95.90	98.50	98.46	98.46	98.46	60	5
FL-VGG16	**97.30**	98.70	98.72	98.71	98.71	60	5
FL-VGG19	97.10	**98.80**	**98.79**	**98.78**	**98.78**	65	5.41

[1] Training time per client in *seconds*.
[2] Training time per round in *minutes*.

In addition to the accuracy rates reported in the training phase, we show the time taken on average to train each client and train the whole model in every round. In the validation and testing stages, all FL-EfficientNet models demonstrate high accuracy rates; however, FL-B0 performs marginally worse than FL-B1 and FL-B2. This could be due to architectural variations and hyperparameter optimization. B1 and B2 have greater depth and width, allowing them to capture more intricate patterns in Arabic sign images. However, FL-B0 achieved the classification results in 44 s and 3.67 min on average for the epochs training time and the rounds training time, respectively. Likewise, FL-ResNet23 demonstrated the shortest training times compared to all other FL-based models under the same experimental setups.

4 Discussion

An important finding concluded from the experimental results is that when deploying real-time applications across distributed execution environments, it is essential to weigh the trade-off between the computational complexity and the accuracy of the model.

We also discuss the performance results in the context of previous related work. Table 4 presents a comparison of the performance of our proposed federated deep learning model with related Arabic Sign Language (ArSL) approaches evaluated on the ARASL2018 dataset. It outlines the key features and performance results documented during the testing phase. As can be observed, the proposed FL-VGG19 model outperforms the other approaches in both the accuracy rates and the training time. In this regard, FL-ResNet32 also demonstrates the ability to accurately recognize Arabic sign language, achieving an accuracy of 98.3% with an average training time of 33 s over 10 epochs. Additionally, it requires approximately 28 min on average to train the entire model over 10 rounds with 5 distributed clients, i.e., edge nodes.

It should be noted that some other related studies have evaluated and reported the accuracy of ArSL tasks; however, specifically in the training and

Table 4. A comparison of test accuracy with the related works on ArASL2018.

Work	Method	Model	Test Acc.	Prec.	Recall	F1-score	Epochs
[17]	Transfer Learning	CNN	0.900	N/A	N/A	N/A	100
[20]	Transfer Learning	Mobile-Net	0.945	0.984	0.983	0.984	15
[14]	Transfer learning	EfficientNetB4	0.950	0.956	0.962	0.950	30
ours	Federated Learning	FL-VGG19	0.988	0.987	0.987	0.987	10
ours	Federated Learning	FL-ResNet32	0.983	0.982	0.982	0.982	10

validation phases. Bani Baker et al. [31] suggested using deep transfer learning using ResNet50V2 and InceptionV3 to classify Arabic signs and reported an accuracy of 100% during the training and validation phases. Saleh and Issa [16] also demonstrated the training and validation accuracy rates obtained by transfer learning, achieving a validation accuracy of 99.57% for ResNet152. However, no accuracy results were provided for the testing phase, which is the appropriate performance indicator for unseen Arabic sign language images.

As artificial intelligence developers continue advancing sign language recognition technologies, human-computer interaction (HCI)researchers need to initiate exploration of future interaction possibilities of this technology. Specifically, there is a necessity to comprehend users' desires regarding this technology and to devise optimal interaction designs. Therefore, the sophisticated design of an interactive interface should align with the HCI standards. For example, while native mobile applications offer numerous advantages, not all are accessible, which means that they do not facilitate easy navigation for users [32], particularly those with disabilities or diminished visual capabilities over the years. Further accessibility improvements for people with disabilities could be achieved by facilitating mobility assistance systems in public environments [33], and establishing guidelines to mitigate the emergence of accessibility barriers when designing FL-based recognition systems on edge devices [34].

As a result, the proposed federated deep learning solution stands out as being very successful among the methods discussed. It achieved the best accuracy while maintaining the crucial practice of preserving patient data privacy, a feature that is absent in other techniques. This collaborative distributed learning approach enables efficient model training across remote devices, addressing current concerns about data privacy in data-driven applications. When initialized with several pre-trained deep neural architectures, the federated learning model demonstrated its capability to recognize Arabic sign images, surpassing the state-of-the-art in terms of recognition accuracy and computational complexity.

5 Conclusion

This paper introduces a federated deep learning model aimed at improving communication for people with disabilities by recognizing Arabic sign language

and interpreting the meanings of image-based hand gestures. The experimental results revealed that the proposed FL-based framework achieves robust performance with diverse deep learning models, demonstrating efficiency in terms of training time and recognition accuracy. FL-VGG19 achieved the highest test accuracy of 98.8%, with an average training time of 5 min per round across 5 clients, and approximately 65 s per 10 epochs. However, FL-ResNet32 achieved the best time complexity by minimizing the average time for each client per epoch and the total time per round. The further investigation aims to improve recognition and interpretation of Arabic sign language gestures by integrating multimodal data sources, which could improve collaborative learning of sign expressions.

References

1. Al-Qurishi, M., Khalid, T., Souissi, R.: Deep learning for sign language recognition: current techniques, benchmarks, and open issues. IEEE Access **9**, 126917–126951 (2021)
2. Aly, S., Aly, W.: DeepArSLR: a novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition. IEEE Access **8**, 83199–83212 (2020)
3. Shin, J., et al.: Korean sign language recognition using transformer-based deep neural network. Appl. Sci. **13**, 3029 (2023)
4. Al-Hammadi, M., Muhammad, G., Abdul, W., Alsulaiman, M., Bencherif, M., Mekhtiche, M.: Hand gesture recognition for sign language using 3DCNN. IEEE Access **8**, 79491–79509 (2020)
5. Cui, R., Liu, H., Zhang, C.: A deep neural framework for continuous sign language recognition by iterative training. IEEE Trans. Multimedia **21**, 1880–1891 (2019)
6. Rastgoo, R., Kiani, K., Escalera, S.: Sign language recognition: a deep survey. Expert Syst. Appl. **164**, 113794 (2021)
7. Alaftekin, M., Pacal, I., Cicek, K.: Real-time sign language recognition based on YOLO algorithm. Neural Comput. Appl. **36**, 7609–7624 (2024)
8. Cheok, M., Omar, Z., Jaward, M.: A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cybern. **10**, 131–153 (2019)
9. Bragg, D., et al.: Sign language recognition, generation, and translation: an interdisciplinary perspective. In: Proceedings of The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 16–31 (2019)
10. Pei, J., Liu, W., Li, J., Wang, L., Liu, C.: A review of federated learning methods in heterogeneous scenarios. IEEE Trans. Consum. Electron. (2024)
11. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends® Mach. Learn. **14**, 1–210 (2021)
12. Li, T., Sahu, A., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. **37**, 50–60 (2020)
13. ElBadawy, M., Elons, A., Shedeed, H., Tolba, M.: Arabic sign language recognition with 3D convolutional neural networks. In: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 66–71 (2017)
14. Zakariah, M., Alotaibi, Y., Koundal, D., Guo, Y., Mamun Elahi, M., et al.: Sign language recognition for Arabic alphabets using transfer learning technique. Comput. Intell. Neurosci. **2022** (2022)

15. Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., AlKhalaf, R.: ArASL: Arabic alphabets sign language dataset. Data Brief **23**, 103777 (2019)
16. Saleh, Y., Issa, G.: Arabic sign language recognition through deep neural networks fine-tuning. International Association of Online Engineering (2020)
17. Kamruzzaman, M., et al.: Arabic sign language recognition and generating Arabic speech using convolutional neural network. Wirel. Commun. Mob. Comput. **2020** (2020)
18. Alawwad, R., Bchir, O., Ismail, M.: Arabic sign language recognition using faster R-CNN. Int. J. Adv. Comput. Sci. Appl. **12** (2021)
19. Hayani, S., Benaddy, M., El Meslouhi, O., Kardouchi, M.: Arab sign language recognition with convolutional neural networks. In: 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), pp. 1–4 (2019)
20. Aldhahri, E., et al.: Arabic sign language recognition using convolutional neural network and MobileNet. Arab. J. Sci. Eng. **48**, 2147–2154 (2023)
21. Boukdir, A., Benaddy, M., Ellahyani, A., Meslouhi, O., Kardouchi, M.: 3D gesture segmentation for word-level Arabic sign language using large-scale RGB video sequences and autoencoder convolutional networks. Signal Image Video Process. **16**, 2055–2062 (2022)
22. Ahmed, M., Zaidan, B., Zaidan, A., Salih, M., Al-Qaysi, Z., Alamoodi, A.: Based on wearable sensory device in 3D-printed humanoid: a new real-time sign language recognition system. Measurement **168**, 108431 (2021)
23. Sharma, S., Kumar, K.: ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. Multimedia Tools Appl. **80**(17), 26319–26331 (2021). https://doi.org/10.1007/s11042-021-10768-5
24. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.: Communication-efficient learning of deep networks from decentralized data. Artif. Intell. Stat. 1273–1282 (2017)
25. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. Int. Conf. Mach. Learn. 6105–6114 (2019)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ArXiv Preprint arXiv:1409.1556 (2014)
28. Nierling, L., João-Maia, M., Bratan, T., Kukk, P., Cas, J., Capari, L., et al.: Assistive Technologies For People With Disabilities–Part III: Perspectives, Needs and Opportunities: European Parliamentary Research Service (EPRS) (2018). [cited 2021 March 4]. (2018)
29. Žilak, M., Car, Ž, Čuljak, I.: A systematic literature review of handheld augmented reality solutions for people with disabilities. Sensors **22**, 7719 (2022)
30. Iso, I.: IEC 25010: 2011 systems and software engineering–systems and software quality requirements and evaluation (square)–system and software quality models. Int. Organ. Stand. **34**, 208 (2011)
31. Bani Baker, Q., Alqudah, N., Alsmadi, T., Awawdeh, R., et al.: Image-based Arabic sign language recognition system using transfer deep learning models. Appl. Comput. Intell. Soft Comput. **2023** (2023)
32. Acosta-Vargas, P., Salvador-Acosta, B., Salvador-Ullauri, L., Villegas-Ch, W., Gonzalez, M.: Accessibility in native mobile applications for users with disabilities: a scoping review. Appl. Sci. **11**, 5707 (2021)

33. Chang, I., Castillo, J., Montes, H.: Technology-based social innovation: smart city inclusive system for hearing impairment and visual disability citizens. Sensors. **22**, 848 (2022)
34. Zaina, L., Fortes, R., Casadei, V., Nozaki, L., Paiva, D.: Preventing accessibility barriers: guidelines for using user interface design patterns in mobile applications. J. Syst. Softw. **186**, 111213 (2022)