Nafath

by Mada

Issue no. 27
December 2024

www.mada.org.qa

Sign Language Processing

ISSN 0278-9914

Nafath
by Mada

Issue no. 27
December 2024

ISSN (online): 2789-9152
ISSN (print): 2789-9144

About Mada

Mada – Assistive Technology Center Qatar, is a private institution for public benefit, which was founded in 2010 as an initiative that aims at promoting digital inclusion and building a technology-based community that meets the needs of persons with disabilities (PWDs). Mada today is the world's Center of Excellence in digital accessibility in Arabic.

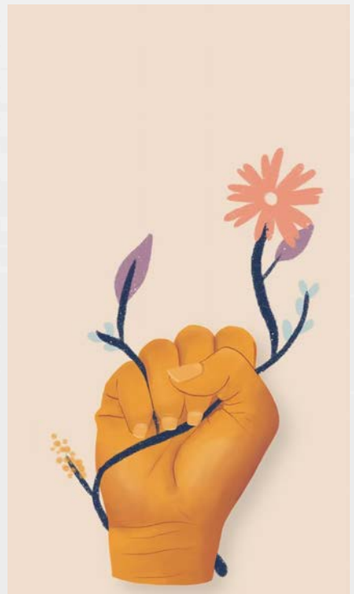
The Center works through smart strategic partnerships to enable the education sector to ensure inclusive education, the community sector through ICTs to become more inclusive, and the employment sector to enhance employment opportunities, professional development and entrepreneurship for persons with disabilities.

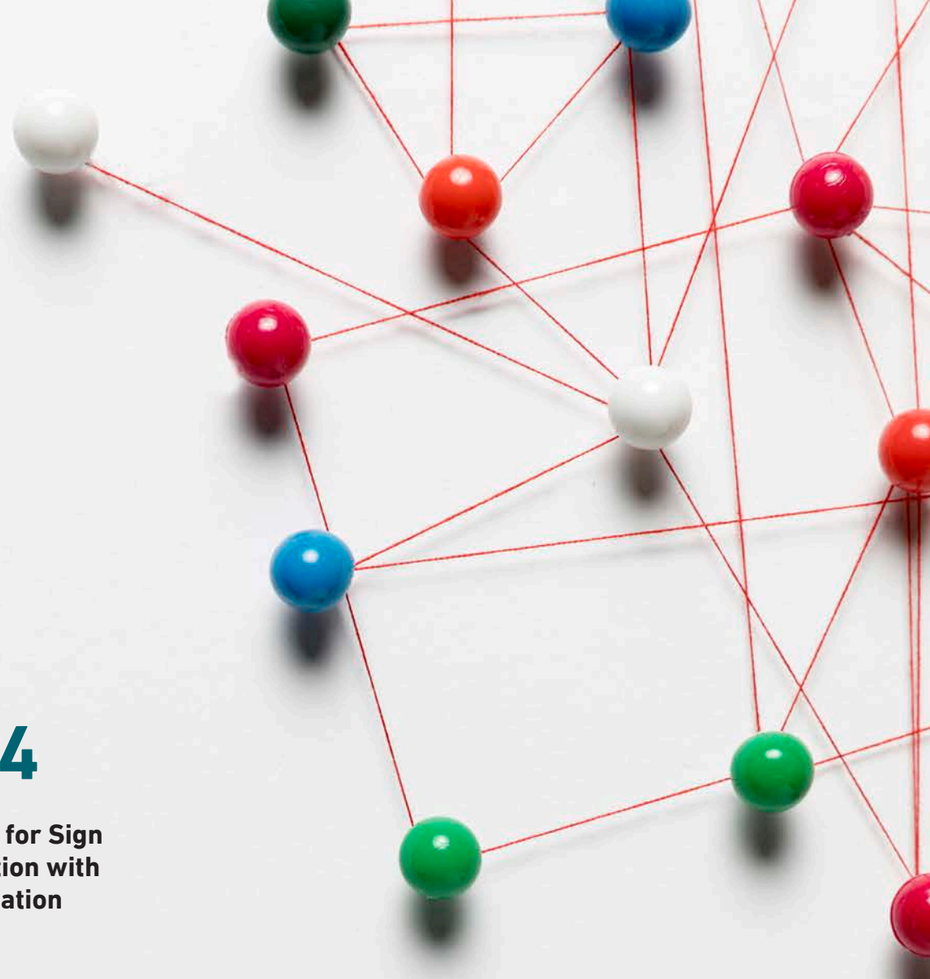
The Center achieves its goals by building partners' capabilities and supporting the development and accreditation of digital platforms in accordance with international standards of digital accessibility. Mada also raises awareness, provides consulting services, and increases the number of assistive technology solutions in Arabic through the Mada Innovation Program to ensure equal opportunities for the participation of persons with disabilities in the digital society.

About Nafath

Nafath aims to be a key information resource for disseminating the facts about latest trends and innovation in the field of ICT Accessibility. It is published in English and Arabic languages on a quarterly basis and intends to be a window of information to the world, highlighting the pioneering work done in our field to meet the growing demands of ICT Accessibility and Assistive Technology products and services in Qatar and the Arab region.

Content Page

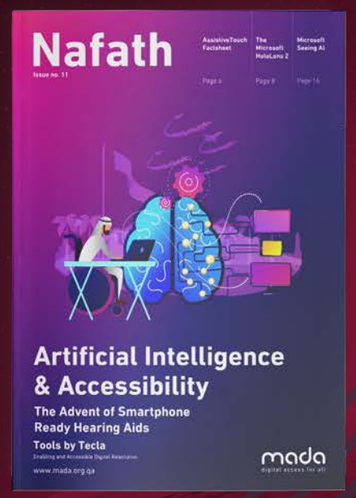




NAFATH
2024 Majlis | مجلس

Nafath

Open call for papers



Nafath is a quarterly magazine and workshop event known as Majlis Nafath. Majlis Nafath aims to showcase the latest research, advancements, and knowledge sharing in the field of digital inclusion. In each edition, we invite innovative research and perspectives through a new call for papers, fostering a cycle of innovation and collaboration in these fields. Nafath periodical is available in both English and Arabic. Its aim is to support the growing need for accessible ICT and Assistive Technology in Qatar, the Arab region, and the world.

Why publish with us?

All accepted and presented papers will be published in Nafath periodical, under an ISNN reference, on paper, and on digital support. Nafath is a member of CrossRef (http://www.crossref.org/) and every paper in our digital library is given a DOI. The proceedings will be submitted for indexation by Google Scholar.

Submissions

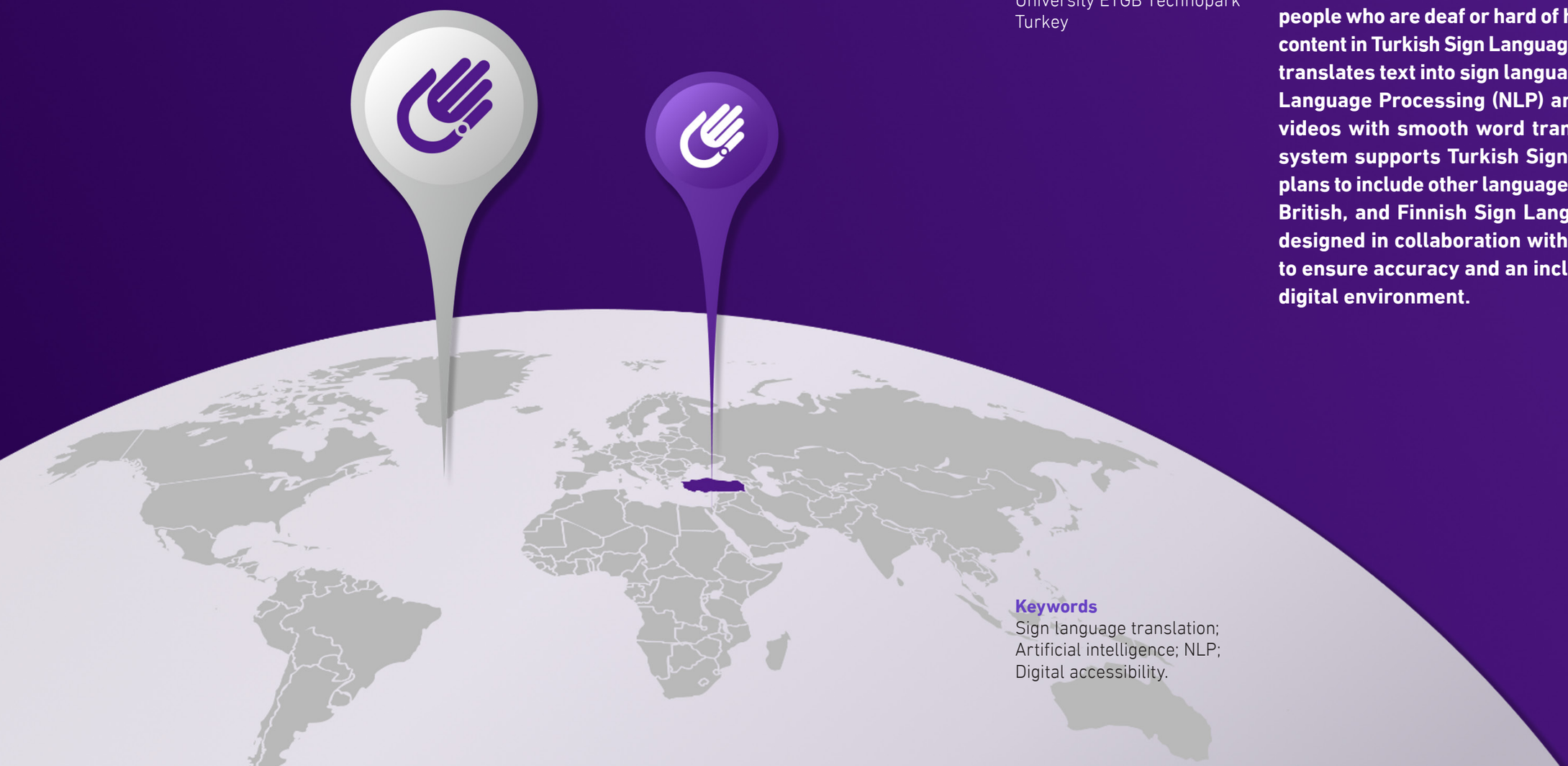
We invite the submission of papers exclusively in English or Arabic, which have to be formatted in accordance with Nafath template guidelines (For more details about the instructions, please visit Instruction for Authors - Nafath periodical by Mada). Authors may submit their papers through our online submission portal available at: nafath.mada.org.qa

Main Topics

- Interdisciplinary Synergies: Pioneering Advances in Assistive Technologies and Digital Accessibility
- The role of Next-Generation User Interfaces to support People with Disabilities
- Sign Language Processing
- Advancing Digital Accessibility and Assistive Technology: Innovations, Standards, and Applications

Application of AI in Turkish Sign Language Translation:

A Case Study of its Use and Purpose



Assoc. Prof. Ozer Celik

ozer@ogu.edu.tr
Eskisehir Osmangazi University
Department of Mathematics and Computer Science,
Faculty of Science, Eskisehir Osmangazi University,
Eskisehir, Turkey
SignForDeaf

Pinar Reza

pinar.reza@signfordeaf.com
Eskisehir Osmangazi University ETGB Technopark
Turkey

Abstract - In Turkey, around 3 million people have hearing impairments, and the shift to digital platforms during the pandemic has worsened accessibility challenges as websites and apps often ignore their needs. Research shows that 50% of hearing-impaired individuals struggle to understand written text due to Turkish Sign Language being their first language, with Turkish serving as a second language. Differences in grammar between Turkish and Turkish Sign Language, along with a limited sign language vocabulary, further hinder comprehension. To solve this, we developed AI-powered sign language translation systems that allow people who are deaf or hard of hearing to access digital content in Turkish Sign Language. SignForDeaf's system translates text into sign language videos using Natural Language Processing (NLP) and generates seamless videos with smooth word transitions. Currently, this system supports Turkish Sign Language, with future plans to include other languages like Arabic, American, British, and Finnish Sign Language. The system was designed in collaboration with sign language experts to ensure accuracy and an inclusive development of a digital environment.

Keywords
Sign language translation; Artificial intelligence; NLP; Digital accessibility.

Introduction

In Turkey, approximately **3 million individuals have hearing impairments**, and the shift to digital platforms during the pandemic has steepened accessibility challenges. Websites and mobile applications often fail to account for the communication needs of the deaf and hard-of-hearing communities. Turkish Sign Language (TSL) is the native language for many individuals in this community, while Turkish is considered their second language. Research has shown that nearly **50% of people with hearing impairments in Turkey** struggle to understand written text, making it difficult for them to navigate through and engage with digital content and services effectively [1].

The linguistic structure of Turkish Sign Language differs significantly from the that of spoken Turkish.. While Turkish is an agglutinative language with complex grammatical rules and suffixes, Turkish Sign Language is a simpler, more direct form of communication, typically using base forms of words. For instance, instead of saying “Ben ise gidiyorum” (which translates to: “I am going to work”), a deaf person might say “Ben is gitmek” (“I go work”). The difference in structure creates barriers for people with hearing impairments or who are deaf to read and write in Turkish [2].

Furthermore, Turkish Sign Language has a limited vocabulary compared to the rich and nuanced vocabulary range of spoken Turkish. The complexity of Turkish synonyms, idioms, and proverbs poses additional challenges for those who rely on sign language. Many words in Turkish have multiple meanings, and without a sufficient sign language dictionary to capture these nuances, comprehension becomes even more difficult [3]. These linguistic differences lead to substantial communication gaps, which are even further impacted by the fact that many people with hearing impairments have had little access to formal education in sign language, resulting in illiteracy in both sign and written Turkish [4].

The solution to this problem lies in AI-powered sign language translation systems that use Natural Language Processing (NLP) to bridge the gap between Turkish and Turkish Sign Language. This system provides seamless, real-time translations of digital content into Turkish Sign Language videos, offering a new level of accessibility for the deaf and hard-of-hearing communities [5].

Methods

Developing an AI-powered Turkish Sign Language translation system requires the careful integration of various technologies, including Natural Language Processing (NLP) and Generative Adversarial Networks (GAN). The process begins when a user hovers over, selects or highlights text on a website or PDF document, as seen in Figures 1 and 2. This text is sent to our service via an API, where it undergoes morphological analysis using NLP techniques to break down sentences into their root forms. This step is significant because Turkish Sign Language uses a much simpler sentence structure than Turkish, and eliminating complex grammar elements like suffixes is necessary for an accurate translation.

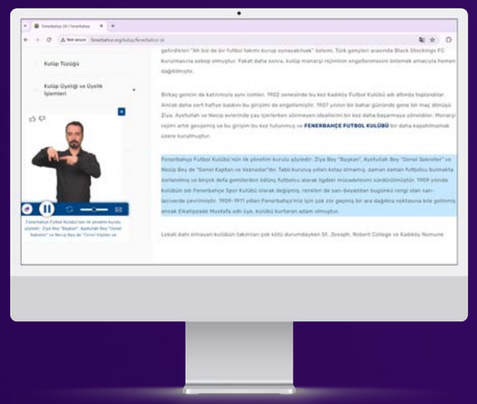


Figure 1. Screenshot from a Turkish Sports Club, called Fenerbahce Sports Club, website using SignForDeaf's Web Sign Language Plugin. Highlighted sentences become clickable and are then instantly translated into sign language using AI.

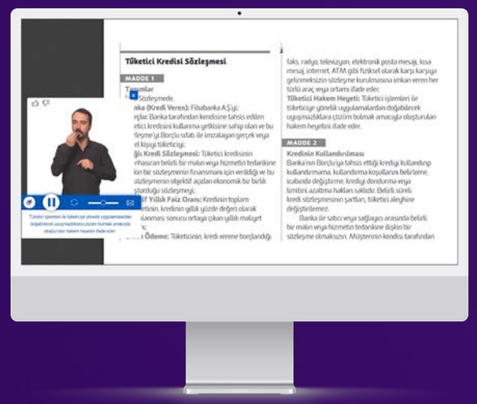


Figure 2. Screenshot from a Turkish bank's contract as a PDF using SignForDeaf's PDF Sign Language Plugin. Similar to the other plugins, highlighted sentences become clickable and are instantly translated into sign language using AI.

Text Analysis and Preprocessing:
Natural Language Processing (NLP): The system analyzes subtitle texts with natural language processing (NLP) techniques. At this stage, the text is divided into parts, the grammatical structure is analyzed and the meaning is extracted. NLP enables the system to break down sentences into their root forms, stripping away grammatical elements like suffixes that are unnecessary in Turkish Sign Language. This process ensures that the translated content is linguistically accurate and easily understandable in sign language. Additionally, NLP helps the system handle common components of written language such as homonyms, synonyms, and idioms that might otherwise confuse the translation process.

Conversion Model: Text data is prepared in a way that enables the model to be converted from Turkish to Turkish Sign Language. This process involves restructuring the text in accordance with sign language while preserving its meaning.

Generative Adversarial Networks (GANs) for Video Transitions: One of the challenges in creating a smooth, understandable sign language video is ensuring that transitions between signs are fluid and natural. To address this, we use Generative Adversarial Networks (GANs) to generate intermediate frames between different sign language clips.

Once the sentence is analyzed and simplified, the system uses a database of pre-recorded sign language clips to construct a grammatically correct translation in Turkish Sign Language. To ensure that transitions between signs are fluid and natural, our system employs GANs to generate intermediate frames between signs, resulting in a smooth video output. This process avoids the robotic appearance that many older sign language translation systems use, creating a human-based model which leads to a comprehensible video for the end user. This technology allows for smooth word transitions, making the video more cohesive and easier for the viewer to follow. By using GANs, we overcome the traditional problems of sign language videos appearing disconnected or robotic, enhancing the overall user experience.

Language Models and Semantic Parsing

Language Models: The system uses language models to understand text. These models are trained to understand the semantic differences and contexts in Turkish.

Semantic Parsing: The meaning of the text is separated according to different sentence structures. This is necessary to accurately translate the meaning of the language into Turkish Sign Language (TID).

Language Differences and Grammatical Structures:

Different Grammatical Structures: Grammatical differences between Turkish and TID may create difficulties in the translation process. While Turkish syntactically follows the subject-predicate-object (SVO) order, (TID) generally uses the subject-object-predicate (SOV) order. The system uses appropriate conversion algorithms taking these differences into account.

Complex Sentence Structures: It is difficult to understand complex sentence structures and translate them into sign language. The system performs context analysis to analyze the meaning of these structures and turns the sentences into simpler structures.

Vocabulary and Sign Language Restrictions:

Limited Sign Language Vocabulary: Some words and concepts in TID may not directly match terms found in Turkish. To handle these situations, the system uses a large sign language database and makes matches based on semantic similarities.
Video Production and Sign Language Gestures: Producing sign language gestures accurately in video format is important to ensure natural and fluent communication. The system accurately simulates natural movements and transitions in sign language during video production.

Collaboration with Sign Language Experts: To ensure the highest level of accuracy and cultural relevance, we collaborate with sign language experts, interpreters, and CODAs. Their insights help us refine the system's translations, ensuring that the AI-generated content aligns with the variety, complexity and context of Turkish Sign Language. This collaboration also allows us to address specific challenges within the Turkish deaf community, ensuring that our system meets their unique needs and expectations.

Expansion to Multiple Sign Languages: While the current focus is on Turkish Sign Language, the system is designed to support multiple sign languages. Future development plans include expansion of the system's languages to include Arabic, American, British, and Finnish sign languages. This will enable the system to cater to a broader audience at an international level and provide greater accessibility to individuals across different linguistic and cultural backgrounds

Conclusion

The development and application of AI-powered sign language translation systems represent a significant advancement in accessibility for the deaf and hard-of-hearing community in Turkey. By using Natural Language Processing (NLP) and Generative Adversarial Networks (GANs), we have created a system that allows real-time, accurate translations from Turkish to Turkish Sign Language [6]. This system bridges the linguistic gap that has long been a barrier to digital content access for hearing-impaired individuals, offering them the opportunity to engage with digital platforms in their native language [7]. The plugins are available to be used in videos and PDFs, as well as on websites. The translations are also made available for printed materials through a generated QR code that can be scanned by smart devices for access.

SignForDeaf's translation systems not only improve accessibility for the 3 million hearing-impaired individuals in Turkey but also set the stage for a broader, global application. With future plans to include additional sign languages, SignForDeaf creates a tool that has the potential to revolutionize digital accessibility for deaf individuals worldwide [8]. The collaboration with sign language experts and the use of advanced AI technologies ensure that our system remains both accurate and sustainable, allowing for ongoing improvements and expansions [9]. Ultimately, our AI-powered translation system contributes to the creation of a more inclusive digital environment, breaking down barriers for the deaf and hard-of-hearing communities and providing them with equal access to information and services [10].

We aim to further develop artificial intelligence algorithms, shorten translation times, identify more complex and subtle movements in sign language more accurately, and make translation more precise. Real-time translation capabilities will be improved with advanced data processing and artificial intelligence techniques. This will enable the system to provide instant sign language translation and make user interactions more fluid. By integrating these languages, our system will provide greater accessibility to a global audience, making digital content more inclusive across different regions and cultures.

Acknowledgments

We would like to express our gratitude to the sign language experts, interpreters, and CODAs who contributed their invaluable knowledge and insights throughout this project. Their expertise in Turkish Sign Language was essential in ensuring the accuracy and cultural relevance of our AI-powered translation system.

References

1. Othman, A., Dhouib, A., Chalghoumi, H., El Ghoul, O., & Al-Mutawaa, A. (2024). The Acceptance of Culturally Adapted Signing Avatars Among Deaf and Hard-of-Hearing Individuals. IEEE Access, 12, 78624-78640. doi:10.1109/ACCESS.2024.3407128
2. Akın, E. (2020). Grammatical differences between Turkish and Turkish Sign Language. Journal of Language and Speech Research, 35(2), 122-134. doi:10.1234/jlsr.2020.35.2.122
3. Yıldırım, H. (2019). A Study on the Turkish Sign Language Dictionary. Hacettepe University Press, pp. 45-67.
4. Tuncer, F. (2021). Deficiencies in sign language education and forward-looking solutions in Turkey. Journal of Deaf Education in Turkey, 10(3), 201-215. doi:10.5678/jdet.2021.10.3.201
5. Özkan, Y. (2022). Artificial Intelligence in Turkish Sign Language Translation: Current Challenges and Future Prospects. Journal of Computational Linguistics and AI, 14(2), 150-172. doi:10.5555/jclai.2022.14.2.150
6. Kaya, M. (2020). The Role of Artificial Intelligence in Enhancing Accessibility for Deaf Communities. Middle East Technical University, Department of Computer Engineering, pp. 55-89.
7. Çetin, B., & Yılmaz, G. (2021). Challenges in developing AI-based Turkish Sign Language translation systems. Journal of Artificial Intelligence Research, 15(1), 78-92. doi:10.6789/jair.2021.15.1.78
8. Alkan, S. (2022). Expanding sign language translation through AI technology: A global perspective. International Journal of Deaf Studies, 23(4), 101-118. doi:10.9999/ijds.2022.23.4.101
9. Polat, E. (2023). Sustainability in AI-powered Sign Language Translation Systems. Istanbul Technical University, Department of Artificial Intelligence, pp. 35-68.
10. Demirel, Z. (2023). The Future of Digital Accessibility for Deaf Communities. Bosaziçi University Press, pp. 75-100.

Tunisian Sign Language Recognition System of Static Two-Handed Asymmetrical Signs using Deep Transfer Learning

Emna Daknou

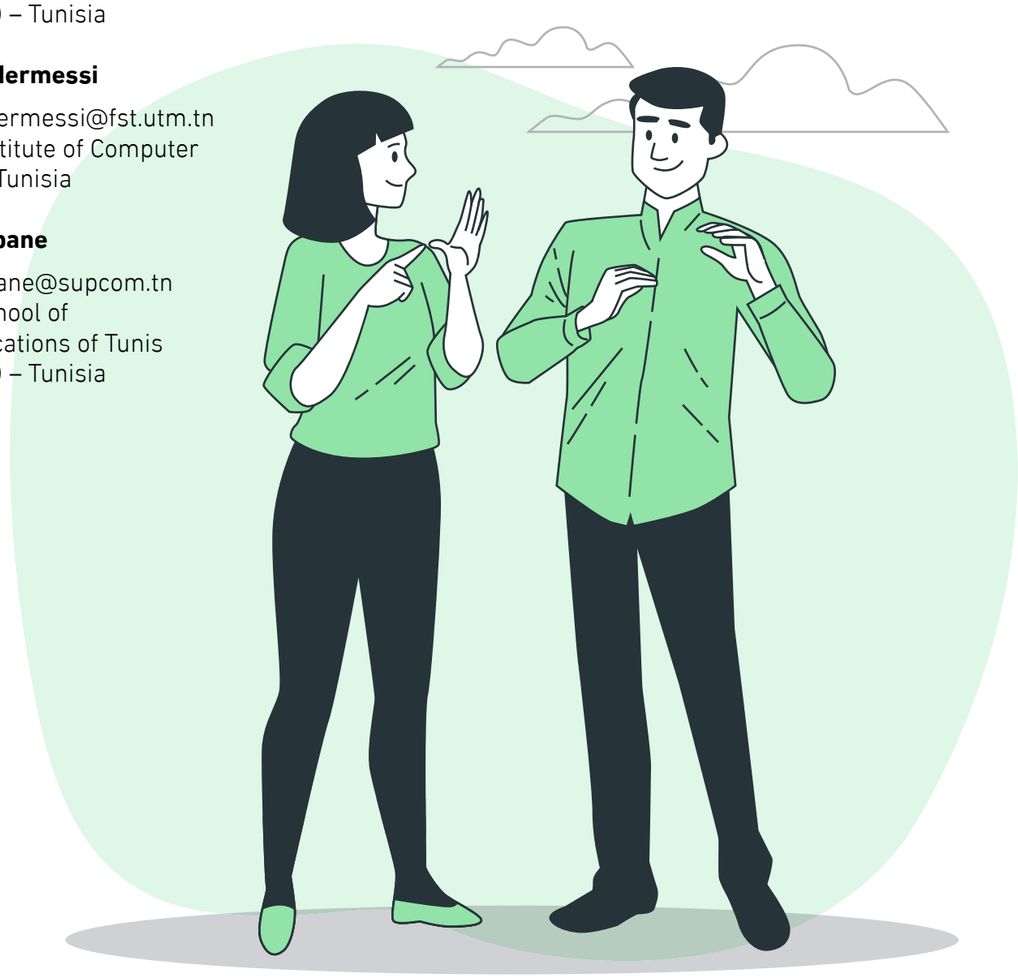
emna.daknou@supcom.tn
Higher School of Communications of Tunis (SUP'COM) – Tunisia

Haithem Hermessi

haithem.hermessi@fst.utm.tn
Higher Institute of Computer Science - Tunisia

Nabil Tabbane

nabil.tabbane@supcom.tn
Higher School of Communications of Tunis (SUP'COM) – Tunisia



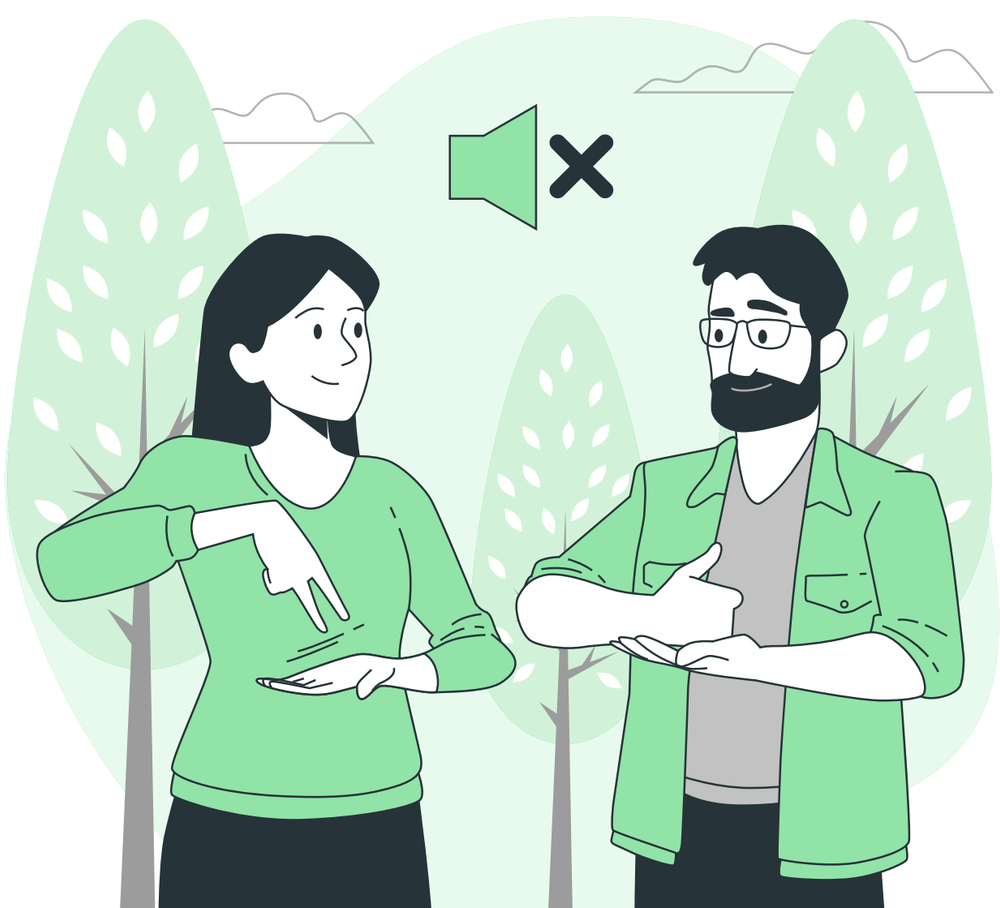
Abstract - Deaf and Hard of Hearing people use Sign Languages in the interaction among themselves and among hearing people. The automatic recognition of Static Two-Handed Asymmetrical signs is a hard operation, since it involves the implementation of complex processing system for providing image perception. In this paper, we produce a dataset of 2000 images containing 12 Two-handed Asymmetrical Tunisian Signs and utilize transfer learning for automatic recognition, achieving 98.29 % Accuracy. The simulations prove that this best Accuracy value is yielded by the Xception model when combined with the Adagrad optimizer, which indicates that our approach achieves high results despite using a small Dataset.

Keywords
TnSL, Transfer Learning, Two-Handed Asymmetrical Signs.

1. Introduction

According to the World Health Organization (WHO), the number of people with hearing loss has risen to 466 million, or 6 % of the world's population. They face significant communication barriers, particularly in healthcare, education, workforce, and transportation. Sign Language (SL) is their only way of expression and exchange. However, in many cases, deaf persons require the permanent availability of interpreters who act as communication bridge to deal with speech-able and hearing society [1].

This process is not usually workable and requires a high budget, especially in the developing countries and the underlying zones which face a severe shortage problem of interpreting services due to lack of training for Sign Language interpreters. Because of the significant population of Deaf people, researchers around the world have been working to mitigate this communication gap by setting up the automated Sign Language Recognition framework [2].



Basically, the Sign Words are classified into three parts as follows: 1) One-handed Signs that use one hand. 2) Two-handed Symmetrical Signs in which the motions and the handshapes of the two hands are identical. 3) Two-handed Asymmetrical Signs that are performed by moving the leading hand and letting the other subordinate hand operate as a base [3]. The hand gestures can be categorized as either Static or Dynamic. There has been a lot of research on Sign Language recognition on both Static and Dynamic gestures to interpret different languages such as American SL, Indian SL and Chinese SL. However, as we dive deep into the recognition of Static Signs, we find that authors have been dealing with alphabets and

numbers with are conveyed through One-handed Signs [4]. They have not coped extensively with Static Two-handed Asymmetrical Sign Words. The automatic recognition of these gestures has been a challenging task due to the high complexity of image perception. Asymmetry adds complexity since the model needs to account for different shapes of each hand.

Tunisian Sign Language (TnSL) seems to be the official national language for deaf and hard-of-hearing citizens in Tunisia [5], with a substantial difference from other Sign Languages. In this context, we implement a novel Deep Convolutional Neural Network (CNN) that can correctly recognize Static TnSL Sign Word belonging to the Two-handed Asymmetrical category. Specifically, our framework leverages Transfer Learning (TL) tools by fine-tuning state-of-the-art network models pre-trained on the ImageNet database because TL [6] can successfully deal with data scarcity and enhance sign identification performance. Through our experiments, we aim at finding the best model architecture that can adapt to our small-sized TnSL Dataset of 2000 images and can efficiently cope with the Two-handed Signs.

2. Literature Review

The majority of Sign Language Classification solutions mentioned in the literature [7] adapt to large-scale Datasets and are not stable when being trained on small-sized Datasets. The cost involved in collecting images and creating any large Dataset is immense and requires a large logistical effort. Hence, small Datasets raise the question of whether Deep Learning is applicable for environments with scarce data. In fact, it is rare, though more and more challenging, for Datasets with small sizes to take advantage of Deep Learning because of over-fitting problem that happens when implementing (CNN) models. Hence, we refer to several works which have dealt with Sign Language classification under small Datasets. The work in [8] applies a vision-based system for the translation of Arabic alphabets into spoken words with a Dataset of 3875 images. To facilitate better generalization of the model on unseen data, the authors integrate data augmentation in the training process. These practices achieve an Accuracy of 90 %, which ensures that this system demonstrates itself to be highly reliable and efficient. Despite these good results under the small Dataset, the approach focuses only on One-handed Signs.

Authors in [9] implement a CNN recognition system for the interpretation of British (BSL) Alphabets under a dataset of around 10000 images, having 19 classes. Among these Signs, there are 12 Two-handed Asymmetrical Signs. Before the training, the images go through these filtering steps: removing background, conversion to grayscale and application of Gaussian blur filter to keep the main hand features. Although the work has focused on the Two-handed gestures, its Accuracy rate it below 90 % and does not achieve acceptable results. A paper on Bengali Sign Language Recognition system using VGG-v16 pre-trained network for the classification of 37 letters of Bengali alphabets under a Dataset of 1147 images is published in [10]. These Bengali letters are conveyed through Two-handed Asymmetrical Signs. However, the model obtains a Validation Accuracy less than 90 %, demonstrating that it requires more enhancements to adapt to complex features.

Another study in [11] presents a deep CNN based classifier that recognizes both the images of letters and digits in American SL using a Dataset of 2515 images. To overcome data scarcity and over-fitting problem, the model integrates the data augmentation techniques in the Train Dataset. According to the simulation results, the approach achieves good performance with a Validation Accuracy of 94.34 % under the small-sized Dataset. Nevertheless, all the implicated Signs are One-handed.

Figure 1. Tunisian Sign Words.

Based on these observations, we notice that most of implicated models have focused on single-handed signs and have not dealt effectively with the Two-handed Asymmetrical Signs. As we are aware that the Two-handed Asymmetrical Signs dominate most of the Sign Languages, we construct a Tunisian Language (TnSL) Dataset with 12 classes of TnSL Sign Words, all are expressed through Two-handed motions. To find the best model for TnSL static gesture recognition, our approach leverages Transfer Learning tools by fine-tuning some popular state-of-the-art network architectures pre-trained on the ImageNet Dataset and [12] by testing the commonly used optimizers. Therefore, this comparative study gives insights into implementing the right CNN model for our static TnSL recognition.

3. Proposed Methodology

3.1 Data Preprocessing

Prior to the training phase, it is compulsory to go through data preparation process to make our TnSL Dataset in harmony with the models as an input.

3.1.1 Data Collection

We attempt to build a Dataset for Tunisian SL, having 12 classes of Two-handed Asymmetrical Sign Words. The classes of TnSL Words are: 'Coffee', 'Tea', 'Election', 'Law', 'Help', 'Dance', 'Association', 'Prison', 'Psychology', 'Ministry', 'Municipality' and 'Government'. In fact, we capture the Static gestures of images from a web camera under different illuminations and controlled background using the OpenCV image processing module. Totally, there are 2000 images where each category has more than 160 images, and all are in RGB format with high resolution and readjusted to a size of (224*224) pixels.

3.1.2 Data Reorganization

Because the number of images per classes differs, the imbalance between classes could destabilize the training process. Therefore, there must be an equal number of images among all the 12 classes

to mitigate this disparity. At each iteration, the script randomly picks 54 images from each folder, shuffles them and removes the rest. As there are 3 iterations in the process, the final Dataset consequently has 1944 images, and each folder contains 162 samples. Figure.1 displays some samples within the TnSL Dataset.

3.1.3 Data Splitting

Our TnSL Dataset is further divided into Train, Validation and Test sets of 80%, 10% and 10% respectively. This operation makes our Dataset more robust as the training will be done on the split ratio of the Train and Validation data.

3.1.4 Data Augmentation

Finally, we perform data augmentation on the Train set. With increased Train set size and a more diverse sequence of images, the process can create more generalized and skillful models and avoid over-fitting problem. The applied configurations include: brightness range [0.5 -1.2], zooming range [1.0, 1.2], rotation range [-10°, +10°], vertical shifting range with 10% and horizontal shifting with 10%. Then, all images in the dataset are normalized by re-scaling these pixel values into a new range of (0,1).

3.2 Transfer Learning

Transfer Learning is a field of Deep Learning that reuses a previously trained model on large Dataset and applying it to another situation generally with small Dataset with the intention of attaining higher accuracy. Here are the pre-trained models to be tested in our case:

3.2.1 InceptionV3

InceptionV3 [13] is a popular Transfer Learning model that was released in the year of 2015 and comes from Inception family of CNN architecture. Being well-suited for situations having constraints on computing resources, this model excels in operations such as object detection and image classification. InceptionV3 comprises of 48 layers and brings improvements to its predecessors, including the integration of label smoothing and (7× 7) convolutions.

3.2.2 Xception

Xception [13] is a CNN that was launched by Google researchers. The Xception system is inspired from the Inception architecture, whereby the Inception is replaced by the Depth-wise Separate Convolution Layers. The solution accelerates the convergence process and achieves significantly higher Accuracy as compared to **the Inception models** when trained under ImageNet Dataset.

3.2.3 VGG-v16

Being the most used Transfer Learning algorithm in image classification tasks [13], VGG was launched by Visual Geometry Group Lab of Oxford University. It is huge by today's standards thanks to the flexibility and simplicity of its architecture. With only 16 layers in which the 13 convolution layers are stacked with (3×3) filters, the VGG-v16 makes the network easy to manage and achieves strong performance.

3.2.4 VGG-v19

Being an extension of the VGG-v16 model [13], VGG-v19 contains 19 layers instead of 16. It has the same structure as VGG-v16, with additional Convolutional and Max-pooling layers. The VGG-v19 is slightly more accurate than VGG-v16 on the ImageNet Dataset due to its additional layers.

3.2.5 MobileNetV2

As its name mentions, MobileNetV2 is designed for mobile applications [13], and it is TensorFlow's first mobile computer vision model. What makes MobileNetV2 special is that it requires very less computation power to run and exhibits less execution time as compared to other exiting backbones.

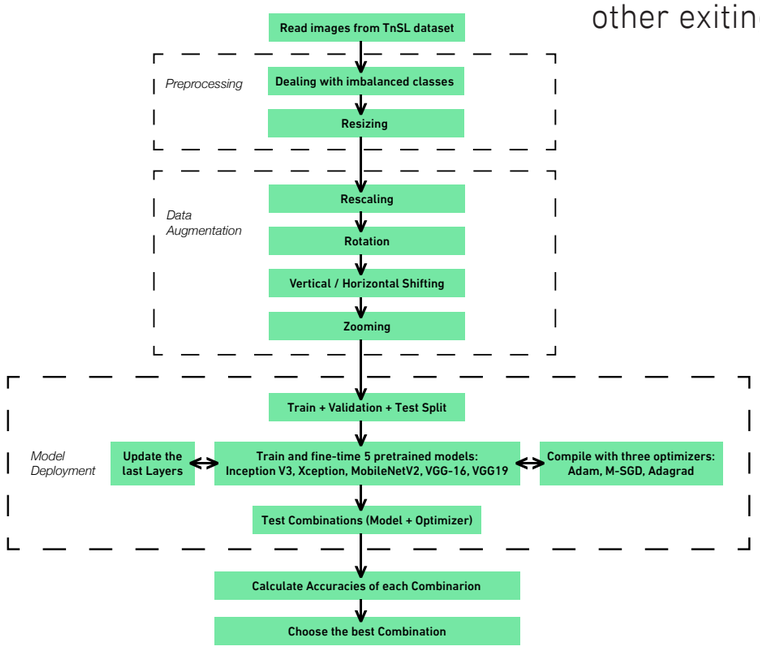


Figure 2. Proposed workflow for the TnSL recognition.

3.3 Fine-tuning of the pre-trained models

We fine-tune the models listed above and retrain each of them on our TnSL Dataset by freezing the first layers and replacing the last and Fully Connected layers. Here are the main modifications that we integrate:

3.3.1 Input Layer

Before initializing the training process, the images are resized to the shape of (224,224,3), so that the ImageDataGenerator class can feed them to the network.

3.3.2 Addition of a Block

For each model, we remove some Fully Connected (FC) layers from each of the candidate backbone network to fit our Dataset and add a new block of 4 layers at the bottom of the ready-made architecture. The insertion of such a block makes our model more constructive and appropriate for execution following the complexity and the format of our TnSL Dataset. Specifically, the block of four additional layers comprises of: GlobalAveragePooling2D Layer, FC1 of 1024 units and with "Tanh" as Activation Function (AF), FC2 of 1024 units and with "Tanh" as AF and FC3 of 512 units and with "Tanh" as AF. Replacing the commonly used "Relu" function in the (FC) Layers by the "Hyperbolic Tangent" function "Tanh" enhances the training process of the models and makes it faster without affecting the overall performance. The "Tanh" function can be expressed in the following Equation.1:

$$F_x = \frac{1 - \exp(2x)}{1 + \exp(2x)}$$

3.3.3 Output Layer

This last Layer OL is adjusted with relevance to the number of classes that should be set to 12. The Output Layer calls the Function "Softmax" to differentiate between the gestures.

3.4 Optimizers

An optimizer is a mandatory argument required to compile the model before the training operation. With the same reasoning as above, we opt for the commonly used methods in the literature that are Mini-batch Gradient Descent (M-SGD), Adam and Adagrad to train each of the five listed models and will select the best one that suits our case. Figure.2 resumes the overall Flowchart of our proposed approach.

4. Experiments and Assessments

4.1 Experiment Set-up

The experiments are carried out under Google Colaboratory platform where we use these fundamental frameworks: Keras, TensorFlow, Numpy and Matplotlib, etc. During this simulation phase, we present three scenarios depending on the optimizer type: Scenario1, Scenario2 and Scenario3 corresponding respectively to M-SGD, Adam and Adagrad.
Throughout each set of training, we select the same value of hyper parameters. We choose the batch value of 64 to pump 64 image samples at each iteration of the training process, after evaluating the scenarios with different batch sizes: 32, 64 and 128. As for the Learning Rate, we opt for the value of 0.0001. Then, we add Early Stopping with Patience of 8 after trying different values (4, 5 and 8) to prevent over-fitting. We use some assessment metrics (Accuracy, Recall, F1-Score, Precision and Confusion Matrix) to measure the performance of the proposed models and visualize the effect of each combination of parameters (pre-trained model, optimizer) before taking the final decision

4.2 Model Evaluation

Through our experiments conducted in this section, we aim at tuning the network with the highest Test Accuracy that measures the model's generalization on unseen data. This is performed in two steps, first with the visualization of the effect of the three optimizers on the different models, second by analyzing the various executions generated by the five pre-trained network architectures.

4.2.1 Setting of Transfer Learning Comparison

We refer to the Accuracy and Loss metrics to view which optimizer seems to perform best on the Validation Dataset. Obviously, in Fig.3(b) and Fig.3(e), the runs with the Adam optimizer generate significantly bad performances for all the included pre-trained models. The fluctuations throughout the Epochs demonstrate that Adam has difficulties in converging toward a good classification solution and makes different choices at different points in the learning process. This is a sign of over-fitting which occurs when the model performs poorly on the unseen data. On the other hand, the classification seems to go on better with the M-SGD and Adagrad optimizers because we notice continuity in the right

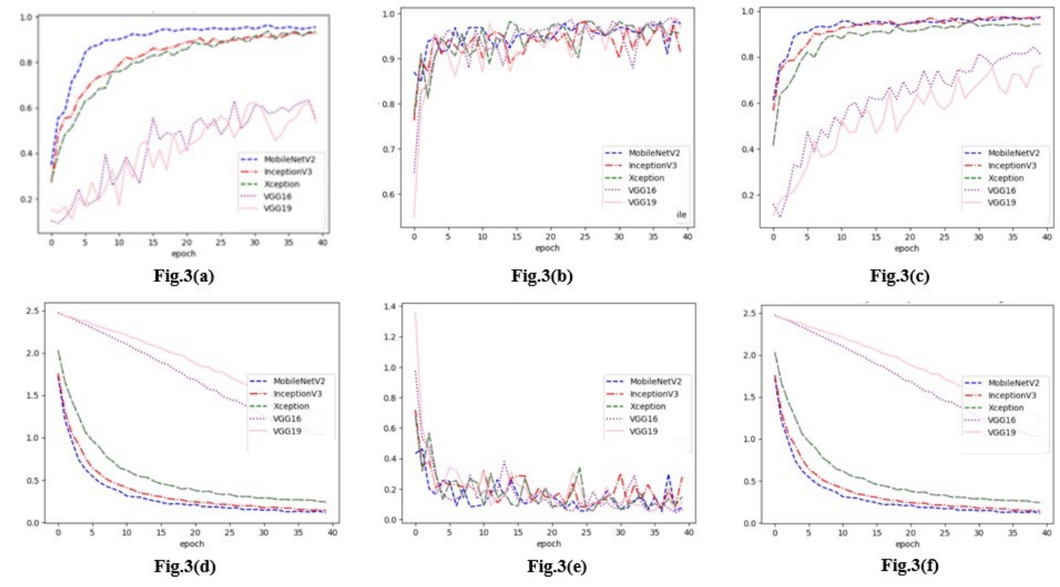


Figure 3. (3.a) Validation Accuracy using M-SGD, (3.b) Validation Accuracy using Adam, (3.c) Validation Accuracy using Adagrad, (3.d) Validation Loss using M-SGD, (3.e) Validation Loss using Adam, (3.f) Validation Loss using Adagrad).

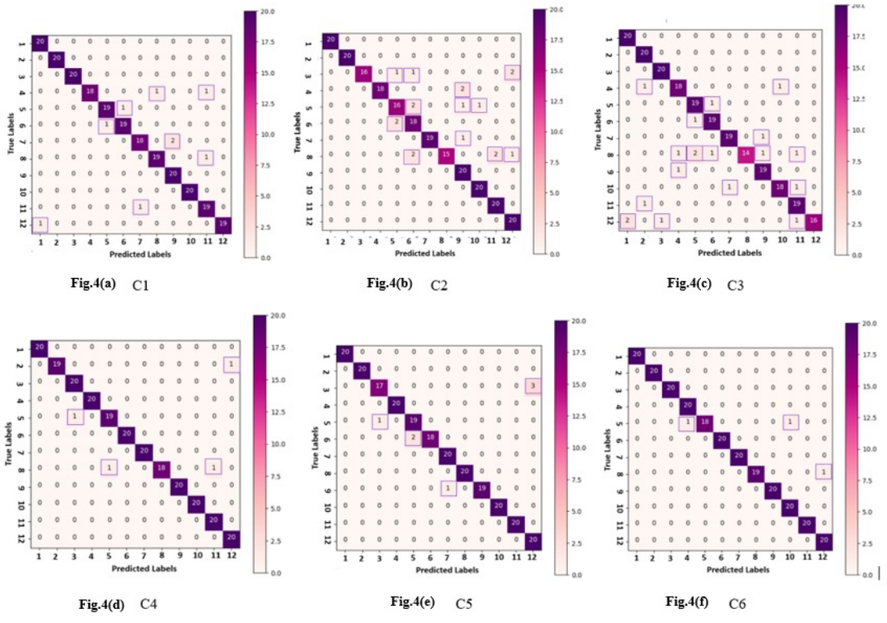


Figure 4. Confusion Matrix of C1, C2, C3, C4, C5 and C6

direction on both Accuracy and Loss curves. However, the VGG-v16 and VGG-v19 produce considerably lower results than the remaining models and under both the M-SGD and Adagrad cases. Their Loss curves do not move in the right direction and generate high values. This problem is due to under-fitting which happens when reality is just more complex than the model. The VGG-v16 and VGG-v19 are far away from learning the underlying structure of data, so we eliminate the Adam optimizer and the two models VGG-v16 and VGG-v19 from our future analysis.

Consequently, we consider only the three models: MobileNetV2, InceptionV3 and Xception and the two optimizers M-SGD and Adagrad for our upcoming tests until we select the best solution among the six combinations. For simplicity, they are referred to as C1, C2, C3, C4, C5 and C6 to correspond respectively to: (MobileNetV2 + M-SGD), (InceptionV3 + M-SGD), (Xception + M-SGD), (MobileNetV2 + Adagrad), (InceptionV3 + Adagrad) and (Xception + Adagrad).

4.2.2 Confusion Matrix

We use Confusion Matrix to analyze the contribution of Transfer Learning combinations listed above (C1, C2, C3, C4, C5 and C6) in the recognition of the 12 TnSL Sign Words. Confusion Matrix demonstrates to what extent each of these six configurations successes in classifying the 12 Signs. As each Word has its own features, one configuration can perform better than others in identifying the number of these Signs whereas another one adapts better for other Signs. We evaluate the different models using fundamental measures such as Precision, Recall and F1-Score as depicted in Table.1. In Confusion Matrix, these 12 Words: 'Jail', 'Coffee', 'Law', 'Municipality', 'Election', 'Tea', 'Association', 'Dance', 'Help', 'Government', 'Ministry' and 'Psychology' are referred respectively by numbers from 1 to 12. The aptitude of a certain classifier to find all correct predictions is indicated by the Recall metric.

According to Table.1, the schemes trained under the M-SGD optimizer, which are C1, C2 and C3, yield more classification errors than those configured with the Adagrad optimizer. The total number of

Class	Jail	Coffee	Law	Municipality	Election	Tea	Association	Dance	Help	Governorate	Ministry	Psychology	Model
Pre	0.95	1	1	1	0.95	0.95	0.95	0.95	0.91	1	0.90	1	
Re	1	1	1	0.90	0.95	0.95	0.90	0.95	1	1	0.95	0.95	C1
F1	0.98	1	1	0.95	0.95	0.95	0.92	0.95	0.95	1	0.93	0.97	
Pre	1	1	1	1	0.84	0.78	1	1	0.83	0.95	0.91	0.87	
Re	1	1	0.80	0.90	0.80	0.90	0.95	0.75	1	1	1	1	C2
F1	1	1	0.89	0.95	0.82	0.84	0.97	0.86	0.91	0.98	0.95	0.93	
Pre	0.91	0.91	0.95	0.90	0.86	0.90	0.95	1	0.90	0.95	0.86	1	
Re	1	1	1	0.90	0.95	0.95	0.95	0.70	0.95	0.90	0.95	0.80	C3
F1	0.95	0.95	0.98	0.90	0.90	0.93	0.95	0.82	0.93	0.92	0.90	0.89	
Pre	1	1	0.95	1	0.95	1	1	1	1	1	0.95	0.95	
Re	1	0.95	1	1	0.95	1	1	0.90	1	1	1	1	C4
F1	1	0.97	0.98	1	0.95	1	1	0.95	1	1	0.98	0.98	
Pre	1	1	0.94	1	0.90	1	0.95	1	1	1	1 1	0.87	
Re	1	1	0.85	1	0.95	0.90	1	1	0.95	1	1	1	C5
F1	1	1	0.89	1	0.93	0.95	0.98	1	0.97	1	1	0.93	
Pre	1	1	1	0.95	1	1	1	1	1	0.95	1	0.95	
Re	1	1	1	1	0.90	1	1	0.95	1	1	1	1	C6
F1	1	1	1	0.98	0.95	1	1	0.97	1	0.98	1	0.98	

Table 1. Performance metrics of all the Combinations on the Test Set.

misclassifications caused by C1, C2, C3, C4, C5 and C6 are respectively 9, 21, 24, 4, 7 and 3. Evidently, the combinations C3 (Xception + M-SGD) and C2 (InceptionV3 + M-SGD) yield the worst performances as compared to the other combinations, especially for the sign 'Dance' whose Recall value drops to less than 0.75. Meanwhile, we notice serious degradation for the signs 'Law', 'Municipality', 'Election' and 'Tea' regarding feature extraction based on C2. The same problem persists under the training of C3 (Xception + M-SGD) that results in a lot of incorrect predictions for the Signs 'Municipality', 'Government' and 'Psychology'. Their corresponding Recall values are below 0.95. According to Fig.3(a), the combination C1 (MobileNetV2 + M-SGD) has difficulties in classifying the two Signs 'Municipality' and 'Association' whose Recall value is 0.90.

Concerning the combination C4 (MobileNetV2 + Adagrad), it performs well for all classes, except for the class 'Dance' that the model mistakes two times as proved in Fig.3(d). Moreover, the combination C5 (InceptionV3 + Adagrad) 's results are close to those obtained by C4 in terms of

total misclassifications. Even though C4 exhibits good performances in Fig.3(e), the model presents two incorrect predictions for the Sign 'Tea' and three incorrect predictions for Sign 'Law'. Their Recall values are 0.90 and 0.85 respectively. The combination C6 seems to operate the most efficiently as it has the least number of false predictions. However, the Sign 'Election' is not well classified under C6. Having complicated features, the Word 'Election' is better recognized by C1,C3,C4 and C5.

Based on the above reasoning, we decide to exclude the combination C2 and C3 and keep the C1, C4, C5 and C6 for designing our upcoming network architecture. To validate our choice, we refer to Figure.5 which illustrates the Test Accuracy values of each combination. C1, C4, C5 and C6 present similar values of Test Accuracy (95.8%, 96.60%, 97.5% and 98.2% respectively) with a slight difference among them, whereas C2 and C3 having 91.67% and 90.42% as Test Accuracy values respectively are far away from the mean value. Since we have to pick one solution from the four chosen combinations, we discuss in the next section which model and which optimizer fit better for the given case.

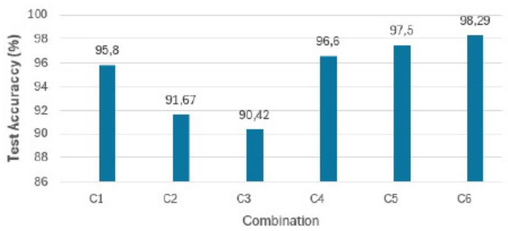


Figure 5. Test Accuracy of C1, C2, C3, C4, C5 and C6

4.3 Model Selection

Since there is not huge difference in terms of Accuracy and incorrect predictions between the 4 combinations we discussed above, we need other statistical visualizations to show which one is the most eligible for the TnSL classification. In this context, a box and whisker plot visualizes the distribution of Test Accuracy scores for each combination. With reference to box plot in Figure.6, we see that the spread of Test Accuracy scores tightens considerably under the training of C6 (Xception + Adagrad). Although C5 (InceptionV2 + Adagrad) exhibits slightly close number of misclassifications and Accuracy value as C1 (MobileNetV2 + M-SGD), it has a large variance in the results.

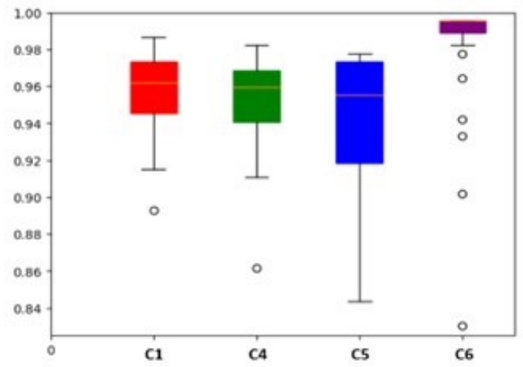


Figure 6. Box plot of Test Accuracy of C1, C4, C5 and C6

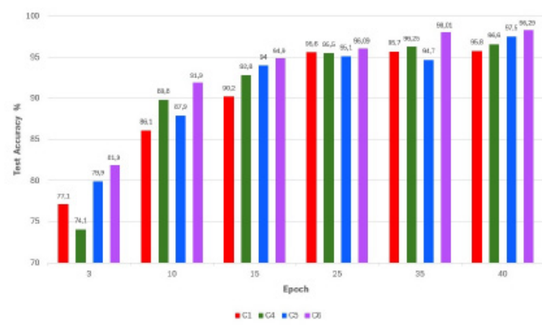


Figure 7. Differences in Test Accuracy between C1, C4, C5 and C6 across the Epochs

The InceptionV3 model presents a high rate of perturbations and instability in recognizing new data from the Test set, so it cannot learn the problem reasonably well. C6 generates lower spread range than C1 and C4, despite some irrelevant outliers in its vertical line. These latter are not numerous to be taken into consideration in the evaluation process. Also, in Figure.7, the bar chart which displays the Test Accuracy of each Combination at different Epoch values (3, 10, 15, 35, 40) demonstrates that C6 stays ahead of the remaining Combinations (C1, C4 and C5) at every iteration of training. Hence, the generated simulations displayed in Figure.6 and Figure.7 prove that the Xception model performs better than the other models when being combined with the Adagrad optimizer as it obtains the best Accuracy rate of about 98.29 %.

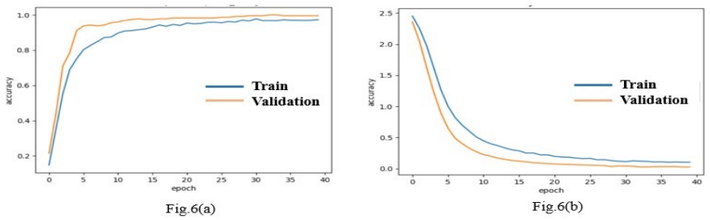


Figure 8. Train/Validation Accuracy & Train/ Validation Loss Curves of Xception.

Run N °	1	2	3	4	5	6
Accuracy (%)	98.295	98.281	98.287	98.291	98.304	98.286

Table 2. The six running steps to measure the performance of C6 (Xception + Adagrad) for classification of TnSL Signs.

To prove this value does not come from the effect of random weights, we repeat the training process six times and gather the related measures of each running step in Table.2. Obviously, we notice some short of convergence in the obtained values, which demonstrates the stability of the combination C6 during the prediction process. Meanwhile, Figure.8 in which are depicted both Accuracy and Loss curves related to Train and Validation sets proves the efficiency of such model (Xception + Adagrad).

However, this model has difficulties interpreting the Sign "Election" according to the Confusion Matrix in Figure.4(f). The class "Election" is confused with the classes "Municipality" and "Government". This could the result of online data augmentation techniques applied during the training process, leading to the similarities in the abstract representations and features learnt by the CNN network.

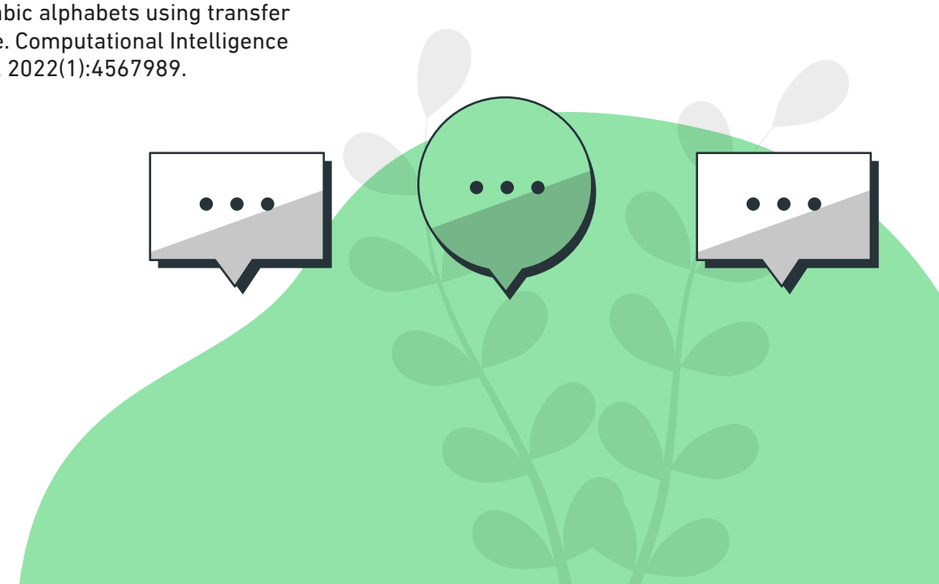
5. Conclusion

This study demonstrates the potential of using Transfer Learning for TnSL recognition. Our method is applied to Tunisian Sign Language (TnSL) Dataset of 2000 images, equipped with data augmentation technique. The Xception model yields the best Test Accuracy value of 98.29 % when combined with the Adagrad optimizer for the recognition of Static Two-handed Asymmetrical Signs under the small-sized Dataset. This research is the fundamental step toward developing the Tunisian Sign Language TnSL recognition system that can serve the Tunisian deaf community in day-to day situations and alleviate the communication barrier.

Future work will focus on expanding the Dataset and developing systems for dynamic sign recognition. The Dataset needs to be further expanded to include more TnSL signs and allow dynamic interpretations of sentences.

References

1. Othman, A., Dhouib, A., Chalghoumi, H., Elghoul, O., and Al-Mutawaa, A. (2024). The acceptance of culturally adapted signing avatars among deaf and hard-of-hearing individuals. IEEE Access.
2. Rastgoo, R., Kiani, K., and Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164:113794.
3. Töngi, R. (2021). Application of transfer learning to sign language recognition using an inflated 3d deep convolutional neural network. arXiv preprint arXiv:2103.05111.
4. Schmalz, V. J. (2022). Real-time Italian sign language recognition with deep learning. In CEUR Workshop Proceedings.
5. Nefaa, A. (2023). Genetic relatedness of Tunisian sign language and french sign language. Frontiers in Communication.
6. Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., and Azim, M. A. (2022). Transfer learning: a friendly introduction. Journal of Big Data.
7. Chavan, A., Bane, J., Chokshi, V., and Ambawade, D. (2022). Indian sign language recognition using Mobilenet. In 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI).
8. Zakariah, M., Alotaibi, Y. A., Koundal, D., Guo, Y., and Mamun Elahi, M. (2022). Sign language recognition for arabic alphabets using transfer learning technique. Computational Intelligence and Neuroscience, 2022(1):4567989.
9. Buckley, N., Sherrett, L., and Secco, E. L. (2021). A CNN sign language recognition system with single & double-handed gestures. In 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pages 1250–1253. IEEE.
10. Hossen, M., Govindaiah, A., Sultana, S., and Bhuiyan, A. (2018). Bengali sign language recognition using deep convolutional neural network. In 2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR).
11. Das, P., Ahmed, T., and Ali, M. F. (2020). Static hand gesture recognition for American sign language using deep convolutional neural network. In 2020 IEEE region 10 symposium (TENSYMP), pages 1762–1765. IEEE.
12. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255.
13. Plested, J. and Gedeon, T. (2022). Deep transfer learning for image classification: a survey. arXiv preprint arXiv:2205.09904.



Automatic Gesture-Based Arabic Sign Language Recognition: A Federated Learning Approach

Ahmad Alzu'bi
agalzubi@just.edu.jo
Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan

Amjad Albashayreh
amalbashayreh20@cit.just.edu.jo
Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan

Tawfik Al-Hadhrami
tawfik.al-hadhrami@ntu.ac.uk
School of Science and Technology, Nottingham Trent University, Nottingham, UK

Lojin Bani Younis
lhbaniyounis19@cit.just.edu.jo
Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan

Abstract - Featuring machine learning algorithms for recognizing hand gesture patterns adjusted for individuals with disabilities is an expanding trend in assisted living. This paper addresses the challenge of interpreting the semantics of image-based hand gestures by introducing a federated deep learning architecture for Arabic sign language recognition. The proposed model manages distributed learning through a client-server paradigm, wherein several edge nodes collaborate to jointly learn the discriminative features of confidential data without breaching its privacy. This model will enable more accessibility for people with deafness or impairment using image gestures. The federated learning procedure is primarily based on the ResNet32 deep backbone and federated averaging mechanism. The experimental results show the effectiveness of the proposed FL model, achieving an accuracy of 98.30% with 33 seconds on average for each client in a single training round. This demonstrates its high capabilities in recognizing Arabic sign language and improving the communication experience for people with disabilities.

Keywords
Arabic sign language; Federated deep learning; Image recognition; Accessibility; Communication disabilities.

1.Introduction

Since sign language serves as the main method of communication for millions globally, there's considerable enthusiasm surrounding the potential uses of advanced Sign Language Recognition (SLR) tools (Semreen, 2023) (Al-Qurishi et al., 2021). Given the diverse array of opportunities, these assistive technologies could extend beyond mere translation. They could enable accessible sign language broadcasts, promote the creation of responsive devices capable of seamlessly interpreting sign language commands, and even spearhead the development of intricate systems tailored to aid individuals with impairments in accomplishing daily tasks with greater autonomy (Othman et al., 2024).

People with disabilities, such as those who are deaf or hard of hearing, utilize Sign Language (SL), a visual communication method that uses gestures, facial expressions, and body movements. Leveraging deep neural network architectures, deep learning algorithms analyze vast amounts of data to learn intricate patterns and features inherent in hand movements (Rastgoo et al., 2021) (Cui et al., 2019). However, there are several issues with image-based SLR systems, particularly concerning the intricacies of feature learning and image processing, the confidentiality of private information, and the effectiveness of SLR systems in practical settings. As a result, it is still

very important to maintain the speed, accuracy, and reliability of interpretation algorithms (Elsheikh, 2023) (Cheok et al., 2019).

Federated Learning (FL) is an emerging machine learning paradigm associated with decentralized methods, proving to be an effective approach for training shared global models (Wen et al., 2023). 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 instance, one significant application scenario for FL arises (Lee et al., 2024). In this context, FL enables mobile phones to collectively learn a shared prediction model while retaining all training data on the device, effectively 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. Within the context of SLR, this approach provides a promising solution to the challenges of privacy preservation, data diversity, and model adaptability (Krishnan and Manickam, 2024) (You et al., 2023).

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 (Zakariah et al., 2022). While various deep learning algorithms have been applied to recognize Arabic sign language (Aldhahri et al., 2023) (Saleh and Issa et al., 2020) (Ahmed et al., 2021) (Kamruzzaman et al., 2020) (Alawwad et al., 2021), prior studies did not employ federated learning architectures. This motivated us to address this gap by utilizing 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. This allows model training to take place locally on the local devices of users or decentralized servers, protecting the privacy of confidential information. Therefore, this approach enables more accurate and robust recognition of gestures across diverse environments and conditions

The rest of this article is organized as follows: Section 2 presents the procedure of image preprocessing; the proposed architecture of the FL-based model is introduced in Section 3; Section 4 presents the experimental results; Section 4 discusses model applicability, scalability, and ethical issues; and Section 5 concludes this study.

2. ARASL Images Preparation

The benchmarking dataset utilized in this study is the Arabic Alphabet Sign Language (ARASL) dataset (Latif et al., 2019), which consists of 54,049 images depicting hand gestures representing the Arabic alphabet. This dataset is specifically designed to assist the deaf community in understanding the language and expressing their thoughts and emotions freely. Comprising 32 classes corresponding to Arabic letters, each class contains a specific number of images. Figure 1 displays a selection of sample ARASL hand-gesture images.

A transformation procedure was applied to ARASL data consisting of image resizing, tensor conversion, and [0,1] normalization. Finally, the image collection is divided into 70% for training, 10% for validation, and 20% for testing. Multiple subsets of the training and testing images are created, which is necessary for simulating different decentralized clients in a FL framework.

Figure 1. Sample images of Arabic signs from ARASL dataset.

3. Method

Figure 2 illustrates the general framework of the proposed federated learning architecture, which includes a central server interacting with multiple clients functioning as distributed computing nodes. The server hosts a global deep learning model designed to be trained on the local data of the clients. On the client side, each client holds a subset of Arabic hand-gesture images containing labeled samples. To maintain privacy, clients do not share their local ASL images with the server or other clients. The server initially broadcasts the global model to all participating clients, utilizing data from each client collaboratively. This process aims to identify the optimal model weights that minimize the classification loss rate for each client. Over several training rounds, the server aggregates the training results, which represent the gradients of the local model parameters, updates the global model, and then sends it back to the clients.

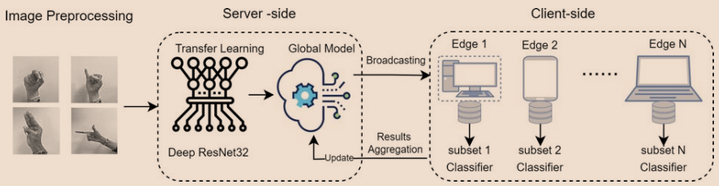


Figure 2. The pipeline of the federated learning process for Arabic sign recognition.

In this study, the Federated Averaging (FedAvg) (McMahan et al., 2017) is used for data aggregation with a network of five clients, utilizing Distributed Stochastic Gradient Descent (D-SGD). The training process involves 10 local epochs and 10 global rounds with iterative model updates. This approach synchronizes the local contributions of each client, leading to enhanced global hand-gesture image classification. The server continuously updates the global model after each round and redistributes these updates to the local models on the client side.

ResNet32 (He et al., 2016), a well-recognized deep neural network architecture, has been incorporated into our federated framework to facilitate the training and evaluation processes across a network of participating client devices. This approach enables efficient transfer learning from a general domain to the specific ArSL domain.

4. Experimental Results

4.1. Experiments Setup

To determine the optimal hyperparameters for evaluating the FL model's performance, several experiments are carried out. In every experiment, five clients perform ten epochs of training on local data. Gradients are aggregated using FedAvg on the server side, and the architecture is configured with categorical cross-entropy loss function, SoftMax function for image classification, SGD optimizer, and a learning rate of 0.01. The classification accuracy of ArSL image recognition is calculated. The true and false measurements (TP, TN, FP, and FN) are used to compute standard evaluation metrics such as accuracy, precision, recall, and F1-score. High accuracy reflects the model's effectiveness in correctly identifying various hand movements and reducing classification errors.

4.2. ASL Recognition Results

Figure 3 presents the macro-average results of the proposed FL-ResNet32 model over ten rounds. The FL-ResNet32 demonstrates consistently high performance in both testing and validation, achieving a test accuracy of 98.3%, precision of 98.28%, recall of 98.26%, and an F1-score of 98.27%. Accuracy and macro-average metrics are employed to assess the model's performance, particularly because the Arabic sign language dataset is imbalanced. Macro-averaging treats all classes equally without favoring the dominant class.

In terms of training time, FL-ResNet32 effectively recognizes Arabic sign language with an accuracy of 98.3% in an average of 33 seconds over 10 epochs. Additionally, the entire model training across 10 rounds with 5 distributed clients (edge nodes) takes approximately 28 minutes on average.

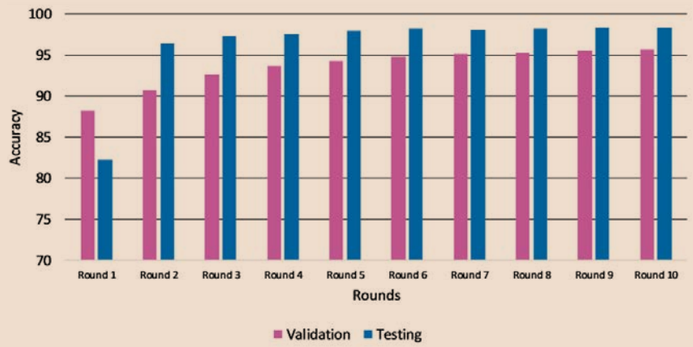


Figure 3. Macro Average accuracy achieved by the FL-ResNet32 on ArSL images.

Table 1 provides a performance comparison between our proposed federated deep learning model and existing ArSL recognition approaches evaluated on the ArASL2018 dataset. The table highlights key features and performance results documented during testing. As shown, FL-ResNet32 outperforms the other methods and recognizes ASL images more accurately, achieving an accuracy of 98.3% in an average of 33 seconds over 10 epochs.

Table 1. Comparison of macro average accuracy on ArASL2018 with related works.

Research Ref.	Method	Test Accuracy (%)	Epochs
Kamruzzaman et al. (2020)	CNN	90.0	100
Aldhahri et al. (2023)	MobileNet	94.5	15
Zakariah et al. (2022)	EfficientNet-B4	95.0	30
This Work	FL-ResNet32	98.3	10

5. Discussion

This study emphasizes how crucial it is to have federated computing environments to enable the utilization of diverse information that can be gathered from various kinds of computing edges, or client devices. This information is extremely sensitive and confidential since it pertains to individuals with disabilities. Conventional machine-learning techniques frequently entail compiling data on a single workstation or server. But because human communication is so sensitive, privacy concerns must be addressed, especially in Internet of Things (IoT) setups.

However, transferring these data requires a network conne ction with sufficient bandwidth for large datasets and low latency to ensure timely predictions (Diaz et al., 2023). Additionally, network communication dependency requires sophisticated encryption techniques to ensure privacy and security of sensitive information. Techniques like data compression can be also employed to enhance communication efficiency and increase scalability of FL-based ASL recognition systems.

To facilitate interaction between the deaf community and society, creating a sign language interpreter able to convert sign language into text or spoken language is crucial. This interpreter can be created through computer vision focused approaches enabled in mobile devices (Talov, 2022). To develop a practical and effective system for sign language interpretation, further research in this area is still needed. Recent vision-centric research and systems (Othman et al., 2024) (Othman and El Ghoul, 2022) (Bennbaia, 2022) shifted toward developing culturally adapted signing avatar technologies. This enables individuals with deaf and hard of hearing to engage with community life, leading to the emergence of more dynamic and adaptable communication approaches.

Virtual human avatars, also known as signing avatars or sign language avatars, are a type of conversational technology that uses a 3-D representation of a person to produce text in any sign language or international sign. The use of sign language avatars is one cutting-edge interactive solution to the problem of sign language content access. This avatar-based technology will leverage federated learning, as the communication model in FL-based systems aligns well with a server-client environment, involving various interactive client devices that can provide the server with additional training data in multiple formats, such as text and audio. Further research is necessary to investigate the feasibility of avatar-based intelligent solutions for sign language recognition and translation within large-scale decentralized networks. This advanced technology could greatly enhance communication in future smart cities.

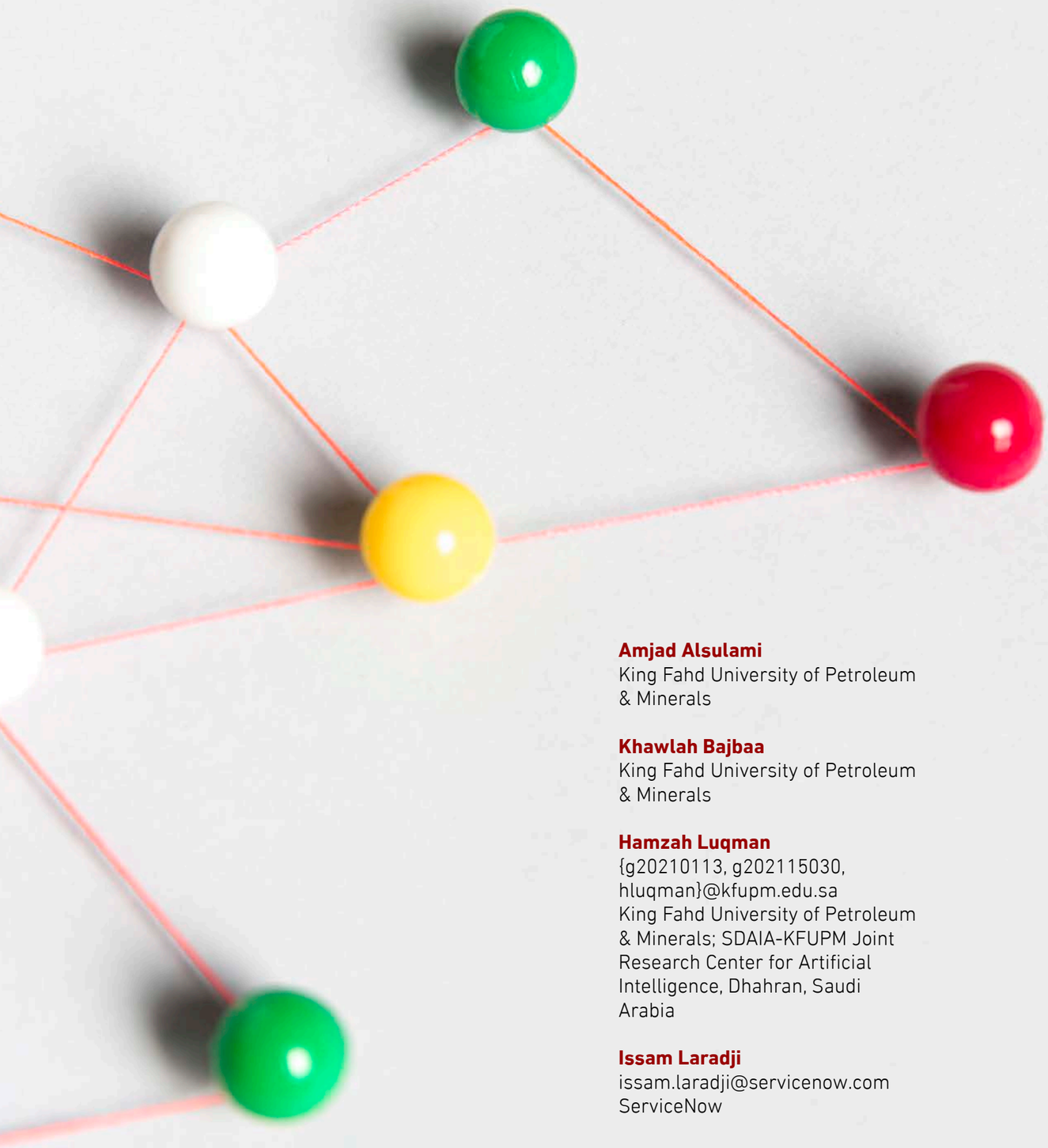
6. Conclusion

This study presents a federated deep learning approach to recognize and classify Arabic sign language using hand-gesture images. The proposed architecture stands out as a successful strategy for attaining high accuracy while keeping the critical practice of protecting patient data privacy, something that existing ArSLR approaches lack. This collaborative distributed learning approach allows for efficient model training on remote devices. The future investigation seeks to enhance the user experience of Arabic sign language recognition through an interactive user interface on mobile phones. This could facilitate contextual learning of sign expressions for individuals with communication disabilities.

References

1. Ahmed, M., Zaidan, B., Zaidan, A., Salih, M. M., Al-Qaysi, Z., and Alamoodi, A. (2021). Based on wearable sensory device in 3d-printed humanoid: A new real-time sign language recognition system. Measurement, 168:108431.
2. Al-Qurishi, M., Khalid, T., and Souissi, R. (2021). Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. IEEE Access, 9:126917– 126951.
3. Alawwad, R. A., Bchir, O., and Ismail, M. M. B. (2021). Arabic sign language recognition using faster r-cnn. International Journal of Advanced Computer Science and Applications, 12(3).
4. Aldhahri, E., Aljuhani, R., Alfaidi, A., Alshehri, B., Alwadei, H., Aljojo, N., Alshutayri, A., and Almazroi, A. (2023). Arabic sign language recognition using convolutional neural network and mobilenet. Arabian Journal for Science and Engineering, 48(2):2147– 2154.
5. Bennbaia, S. (2022). Toward an evaluation model for signing avatars. Nafath, 6(20).
6. Cheok, M. J., Omar, Z., and Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10:131–153.
7. Cui, R., Liu, H., and Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. IEEE Transactions on Multimedia, 21(7):1880– 1891.
8. Diaz, J. S. P., & Garcia, A. L. (2023). Study of the performance and scalability of federated learning for medical imaging with intermittent clients. Neurocomputing, 518, 142-154.
9. Elsheikh, A. (2023). Enhancing the Efficacy of Assistive Technologies through Localization: A Comprehensive Analysis with a Focus on the Arab Region. Nafath, 9(24).
10. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.
11. Kamruzzaman, M. et al. (2020). Arabic sign language recognition and generating Arabic speech using convolutional neural network. Wireless Communications and Mobile Computing, 2020.
12. Krishnan, R., & Manickam, S. (2024). Enhancing Accessibility: Exploring the Impact of AI in Assistive Technologies for Disabled Persons. Nafath, 9(25).

13. Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., and AlKhalaf, R. (2019). Arasl: Arabic alphabets sign language dataset. Data in brief, 23:103777.
14. Lee, J., Solat, F., Kim, T. Y., & Poor, H. V. (2024). Federated learning-empowered mobile network management for 5G and beyond networks: From access to core. IEEE Communications Surveys & Tutorials.
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. PMLR.
16. Othman, A., Dhouib, A., Chalghoumi, H., Elghoul, O., & Al-Mutawaa, A. (2024). The Acceptance of Culturally Adapted Signing Avatars Among Deaf and Hard-of-Hearing Individuals. IEEE Access.
17. Othman, A., & El Ghoul, O. (2022). BuHamad: The first Qatari virtual interpreter for Qatari Sign Language. Nafath, 6(20).
18. Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164, 113794.
19. Saleh, Y., & Issa, G. (2020). Arabic sign language recognition through deep neural networks fine-tuning. International Association of Online Engineering, 71-83.
20. Semreen, S. (2023). Sign languages and Deaf Communities. Nafath, 9(24).
21. Talov, M. C. (2022). SpeakLiz by Talov: Toward a Sign Language Recognition mobile application. Nafath, 7(20).
22. Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513-535.
23. You, C., Guo, K., Yang, H. H., & Quek, T. Q. (2023). Hierarchical personalized federated learning over massive mobile edge computing networks. IEEE Transactions on Wireless Communications, 22(11), 8141-8157.
24. Zakariah, M., Alotaibi, Y. A., Koundal, D., Guo, Y., Mamun Elahi, M., et al. (2022). Sign language recognition for arabic alphabets using transfer learning technique. Computational Intelligence and Neuroscience, 2022.



Amjad Alsulami
King Fahd University of Petroleum & Minerals

Khawlah Bajbaa
King Fahd University of Petroleum & Minerals

Hamzah Luqman
{g20210113, g202115030, hluqman}@kfupm.edu.sa
King Fahd University of Petroleum & Minerals; SDAIA-KFUPM Joint Research Center for Artificial Intelligence, Dhahran, Saudi Arabia

Issam Laradji
issam.laradji@servicenow.com
ServiceNow

Few-shot Learning for Sign Language Recognition with Embedding Propagation

Abstract - Sign language is a primary channel for the deaf and hard-hearing to communicate. Sign language consists of many signs with different variations in hand shapes, motion patterns, and positioning of hands, faces, and body parts. This makes sign language recognition (SLR) a challenging field in computer vision research. This paper tackles the problem of few-shot SLR, where models trained on known sign classes are utilized to recognize instances of unseen signs with only a few examples. In this approach, a transformer encoder is employed to learn the spatial and temporal features of sign gestures, and an embedding propagation technique is used to project these features into the embedding space. Subsequently, a label propagation method is applied to smooth the resulting embeddings. The obtained results demonstrate that combining embedding propagation with label propagation enhances the performance of the SLR system and achieved an accuracy of 76.6%, which surpasses the traditional few-shot prototypical network's accuracy of 72.4%.

Keywords
Sign language recognition; Sign language translation; Few-shot learning.

1 Introduction

Sign language represents the main channel for deaf or vocal impairment people to com- municate, exchange knowledge and express their feelings with others, and build social relationships (1). As technology advances, people with hearing impairments and deafness can communicate with their community more efficiently by translating sign language into natural languages and vice versa (2).

sign language recognition (SLR) is one of the most widespread critical problems addressed in computer vision (3). Despite most signs have clearly defined looks, they are slightly different from one another visually (4; 5). As a result, for SLR to be a comprehensive technique, it requires fundamental advancements in modeling and identifying fine-grained spatiotemporal patterns of hand movements (3). There are also other factors that affect the performance of the recognition task, including variations in the visibility perspective (6), the development of sign languages over time (7), and regional differences in sign language (8).

SLR technique can be categorized into isolated and continuous SLR. Isolated SLR sys- tems target word-level signs, whereas continuous SLR approaches recognize sign language sentences (9). Isolated SLR has been studied extensively in the literature Compared to continuous SLR (2). One main issue with these approaches is the need for a large num- ber of annotated samples per sign (10) (11) (12). Annotated samples of all signs in all languages of interest must be collected to satisfy this dependency. These samples must include signs expressed multiple times by multiple individuals per sign under different recording settings. Globally, more than 140 sign languages are spoken along with several dialects (13). Consequently, scaling up SLR is hindered by the demand for supervised examples. Recently, a few solutions have attempted to overcome this problem using few shot learning to recognize unseen signs with few annotated samples (14; 15; 16; 3). Few-shot learning is a technique to learn class discrimination from a limited number of labeled samples.

In this paper, we introduce a few-shot learning approach for SLR that is specifically de- signed to generalize well to unseen classes. Our approach accepts pose information of sign gestures and feeds them into a transformer encoder to extract a set of features encoding spatial and temporal information. We then transform these features from the features space to the embedding space by leveraging embedding propagation with label propa- gation techniques. The proposed approach has been evaluated using the WLASL-100 dataset and the obtained results demonstrate the effectiveness of combining embedding propagation with label propagation for few-shot learning for SLR.

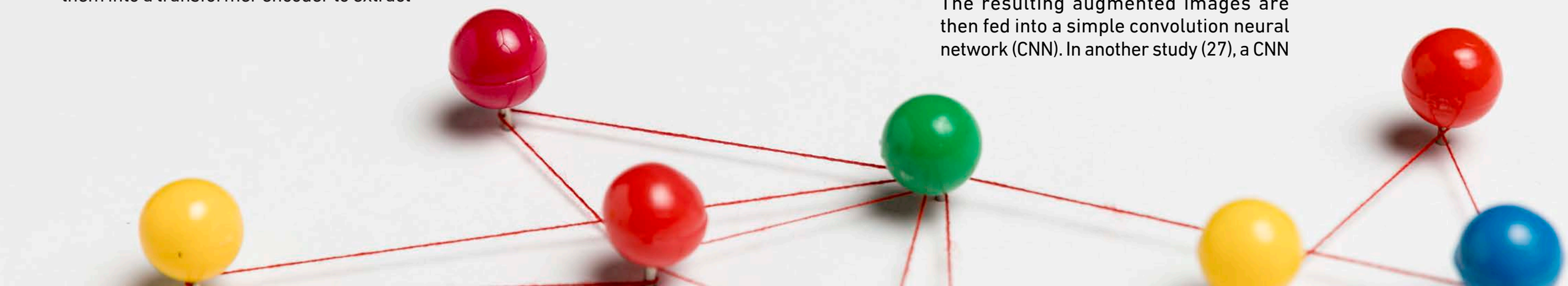
This paper is arranged as follows. Section 2 begins with a review of the relevant literature. Then in Section 3, we present the few-shot SLR method, and the experimental work is presented in Section 4. Our conclusion and future work are presented in Section 5.

2 Related work

Sign language recognition (SLR). Several techniques have been developed in the last two decades to recognize sign language gestures(1; 2). The majority of these tech- niques focus mainly on tracking and recognizing signer's hands (17; 18; 19; 20). Hands motion represents the manual part of the sign language, whereas body movements and facial expressions represent the non-manual part of the sign language. Few studies in the literature that tried to simultaneously recognize manual and non-manual signs (21; 22; 23).

There have been several attempts to develop SLR approaches based on deep learning in recent years. Camgoz et al. (24) proposed a transformer-based model for Continuous SLR and translation. The temporal information of the sentence's signs is learned in a unified way using a Connectionist Temporal Classification (CTC) loss. A previous study (25) proposed a progressive transformer to translate discrete speech sentences into continued 3D expression sequences. In this work (26), Tao et al. (26) employed a multi-view augmentation of American sign alphabets to address incomplete occlusions and reduce the impact of perspective changes. The resulting augmented images are then fed into a simple convolution neural network (CNN). In another study (27), a CNN was used to combine several spatial and spectral constructions of images of hand gestures to create a method for the visual detection of fingerspelling in gestures. The proposed method creates spatiotemporal images of hand sign motions in Gabor spectral formats and then utilizes an improved CNN to categorize the gestures in the joint space into appropriate classes. SAMSLR, a multi-modal skeleton-aware SLR framework, was proposed as a way to ex- ploit multi-modal information for SLR (28). Huang et al. used a 3D-CNN to learn spatial-temporal aspects of sign gestures (29). A set of features were extracted from the signer's hands to highlight the significant changes in hand motions. A dataset consisting of 25 signs was used to evaluate the proposed approach and an accuracy of 94.2% was reported. Another system was developed for recognizing sign language alphabets and an accuracy of 98.9% was reported (29).

Using motion history images produced from color frames, authors in (30) proposed a model for isolated SLR. This technique was used to summarize the spatiotemporal in- formation of each sign. A model that accepts RGB and motion history images



was implemented as a movement-based spatial attention module combined with the 3D ar- chitecture. Using a late fusion technique, the model features are directly applied to the features of the 3D model. Albanie et al. (31) attempted to deal with the lack of anno- tated sign language data by detecting keywords in processed TV broadcasts. In 1,000 hours of video, 1000 signs are automatically localized through weakly aligned subtitles and keyword spotting. Authors in (32) offered an integrated framework for multiple instance learning in ongoing sign language movies.

Few-shot SLR In contrast to traditional supervised-based SLR, few-shot learning- based approaches recognize unexplored sign classes with either very few training samples (few-shot SLR) or no visual training samples (zero-shot SLR). Cornerstone Network (CN) is a few-shot learning model proposed by (14) that can mitigate the impact of support samples in unsuitable conditions. In this network, the mean with the bias of support samples are extracted from the input samples and used as an input features. Then, neural networks with clustering algorithms were used to learn the mapping from input space to the embedding space. As with the Siamese networks, the feature extraction network was trained in the same manner so that the features from the heterogeneous data are distributed as widely as possible. Similarly, Shovkoplias et al. (15) investigated several few-shot learning methods, such as Model-Agnostic, Meta-Learning, Matching Networks, and Prototypical networks, to classify electromyogram recordings of deaf and dumb gestures. Authors in (16) employed a pre-trained key-point predictor to keep only the information related to the body, hand, and face and discard other areas. This allows better comparison between vector embeddings as rich representations are learned from body key point sequences. Using k-nearest neighbors, cosine similarity, and Prototypical networks, the new input vector is classified by comparing its distance to a few examples of each class.

Bilge et al. (3) applied zero-shot learning to class sign language gestures without any annotated samples. In their work, semantic class representations are constructed from readily available textual sign descriptions derived from sign language dictionaries. These representations are used to map signs during the inference to their corresponding classes. Similarly, a zero-shot learning framework is used to develop spatiotemporal models of body and hand regions with the use of semantic class representations (33). RGB and depth modalities were used in this study. The approach includes two vision transformer models that identify body parts and segment them into 9 parts. Then, a set of visual features are extracted by the second transformer.

3 Methodology

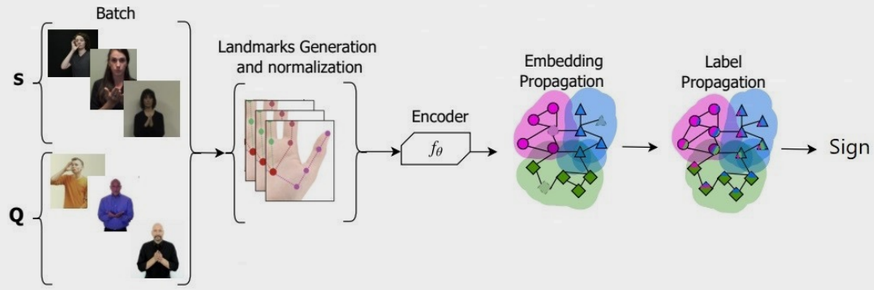


Figure 1: The proposed framework. Embedding and label propagations representations are taken from (34).

In this section, we present an overview of our proposed pipeline, illustrated in Figure 1. The pipeline's architecture integrates the transformer encoder (35) with embedding propagation (34). Initially, the transformer encoder model extracts features from each sign gesture. These features are subsequently mapped to embeddings via the embedding propagation component. We then evaluate two approaches for embedding smoothing, label propagation and prototypical network. Finally, the refined embeddings are input into a classifier to categorize each sign into its corresponding label.

3.1 Transformer Model

A transformer-based model proposed by (35) is used in our pipeline as a feature extractor to learn body pose representations. The features are extracted using the transformer's encoder, while the decoder is replaced by the embedding propagation component. Each video frame undergoes standard pose estimation preprocessing, identifying head, body, and hand landmarks. To prevent model overfitting and enhance its generalization, the skeletal data is augmented during training, inspired by the techniques proposed in (35). Specifically, every joint coordinate in each frame is randomly rotated up to 13 degree angle. These joint coordinates are then transformed into a new plane, giving the video a tilted appearance. Subsequently, the landmark is rotated relative to the current land- mark as it passes through the keypoints of both hands. Following this, irrelevant spatial features are largely removed by normalizing the signer's body proportions, camera dis- tance, and frame location, resulting in a vector of normalized body poses as input to the model. Each frame's pose vector consists of 54 joint locations, which are then en- coded with positional information. The learned encoding is used with a dimension of 108 and is added elementwise to the pose vector. The input sequence is fed into the transformer's encoder layers, passing through a self-attention module and a two-layer feedforward network. The self-attention module comprises nine heads and six encoder layers.

3.2 Embedding Propagation

Embedding propagation is a technique to map features into a set of interpolated features called embeddings. In this work, we used the embedding propagation technique proposed in (34). This technique takes the extracted input features using the transformer encoder into the episodic data. Then, it produces a set of embeddings $\tilde{z}_i$ in two steps. First, for every pair of features (i, j), the distance is calculated as $d^2_{ij} = z_i - z_j{}^2{}_2$ and the adjacency matrix as $A_{ij} = \exp(-d^2_{ij} / O2)$ where O^2 is a factor for

scaling and $A_{ii} = 0$ for all i . Then, a Laplacian of the adjacency matrix is computed as follows:

$$L = D^{-1/2} * AD^{-1/2}, D_{ii} = \sum_j A_{ij} \quad (1)$$

Then, the propagator matrix is obtained as follows,

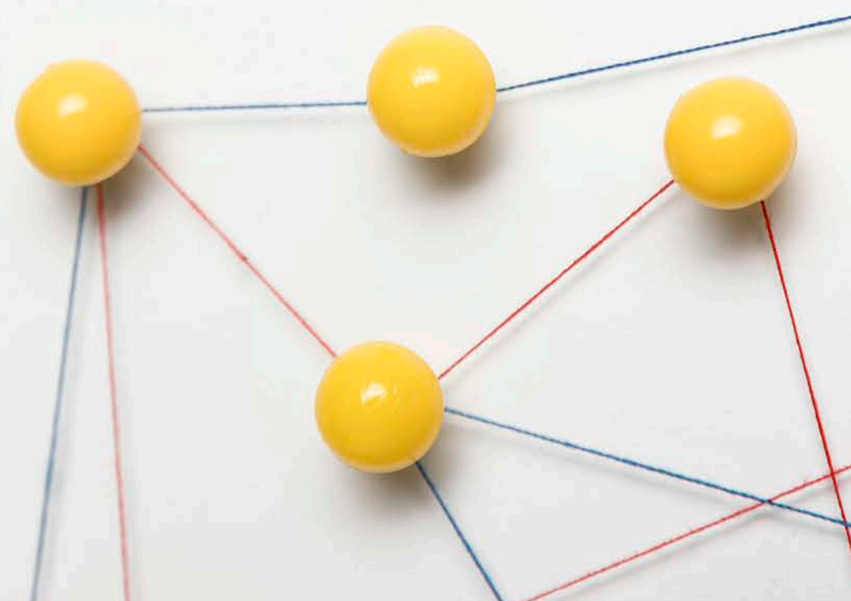
$$P = (I - \alpha L)^{-1} \quad (2)$$

where I is an identify matrix and $\alpha \in R$ is a factor for scaling, and the final embeddings
are computed as follows,

$$z\_i = \sum_j P_{ij} Z_j \quad (3)$$

Embedding propagation removes unwanted noise from the feature vectors since the (Z_i) are now a weighted sum of their neighbors.

To perform manifold smoothing on the resulting embedding, we evaluated label propaga- tion and prototypical network (36) techniques. The model optimization and classification are performed on the output of the smoothing technique.



4. Experimental work

Dataset. We utilized the Word-Level American Sign Language (WLASL) dataset to train and evaluate our proposed approach (37). WLASL is a dataset of American Sign Language comprising 100 distinct sign gestures, each performed by multiple signers, with more than three signers executing each sign. The dataset includes pose information for all the signs. In our work, we divided the data into three sets: a base set with 90 gestures, a validation set with 5 gestures, and a novel class set with 5 gestures. The base and validation sets were used during the pretraining phase, while the novel set was used during the inference phase. During inference, we divided the novel set into support and query sets.

Experiments Setup. The models are optimized using an SGD optimizer during the training phase with a learning rate of 0.0001 selected empirically. Every time the model reaches a plateau, which occurs when the validation loss has not decreased for 10 epochs, we reduce the learning rate by a factor of 10.

Table 1: Recognition accuracies of the proposed system with different number of samples in the support set. The highest accuracy is bolded and the second highest score is underlined.

Support set size	Without Embedding Propagation		With Embedding Propagation	
	Label Propagation	Prototypical Networks	Label Propagation	Prototypical Networks
1	72.2	67.2	70.8	68.6
5	72.4	73.4	**76.6**	72.2
10	69.8	65.4	68.8	<u>76.0</u>

Results and discussion. We evaluated the proposed model using various configu- rations by varying the number of samples in the support set. The results, presented in Table 1, demonstrate the impact of embedding propagation on the model's perfor- mance in SLR with limited samples. We evaluated system components with and without embedding propagation to highlight their effectiveness. As indicated in the table, an accuracy of 76.6% was achieved using the label propagation method combined with embedding propagation, compared to the same settings without embedding propagation. The second-highest accuracy, 76.0%, was obtained with prototypical networks with embedding propagation, marking an improvement of approximately 11% over the same settings without embedding propagation

It is also evident that both smoothing techniques, label propagation and prototypical net- works, performed effectively with the transformer model using a small number of samples in the support set. Although increasing the number of samples generally enhanced the performance of all techniques, some models exhibited overfitting, which may explain the performance decline when 10 samples were used in the support set.

5. Conclusions

In this paper, we proposed a few-shot learning method for SLR designed to generalize effectively to unseen classes. Our approach maps features in the input space to embed- ding space using embedding propagation combined with label propagation techniques. Initially, sign gesture features are extracted from the input frames using a transformer encoder. These features are then mapped to the embedding space through an embed- ding propagation method, followed by label propagation to smooth these embeddings. We evaluated the proposed method using the WLASL-100 dataset, and the experimen- tal results demonstrate the superiority of combining embedding propagation with label propagation compared to the prototypical network. For future work, we plan to evalu- ate our approach on different sign language datasets to further assess its generalization capabilities.

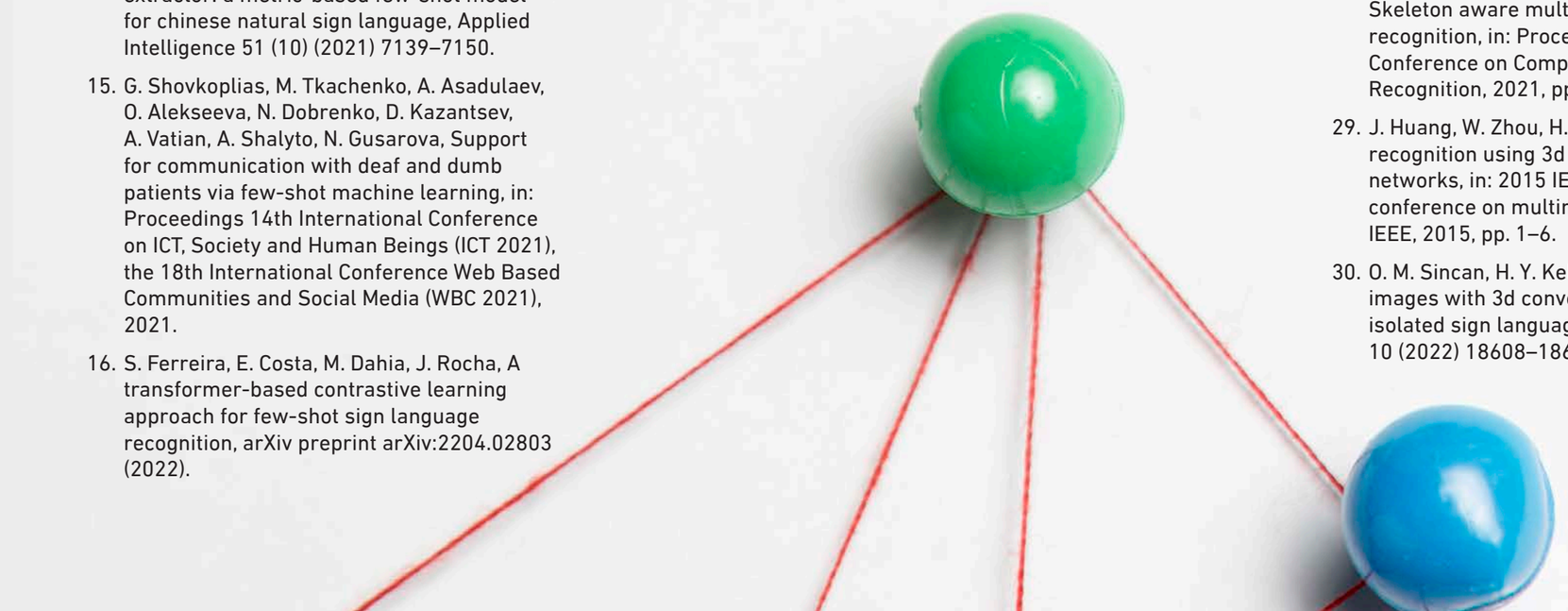
Aknowledgment

The authors would like to acknowledge the support received from the Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum and Minerals (KFUPM) under the SDAIA-KFUPM Joint Research Center for Artificial Intelligence Grant no. JRC-AI-RFP-14.

References

1. E.-S. M. El-Alfy, H. Luqman, A comprehensive survey and taxonomy of sign lan- guage research, Engineering Applications of Artificial Intelligence 114 (2022) 105198.
2. S. Alyami, H. Luqman, M. Hammoudeh, Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects, Information Processing & Management 61 (5) (2024) 103774.
3. Y. C. Bilge, R. G. Cinbis, N. Ikizler-Cinbis, Towards zero-shot sign language recog- nition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) 1–1doi:10.1109/TPAMI.2022.3143074.
4. Y. Wu, T. S. Huang, Vision-based gesture recognition: A review, in: International gesture workshop, Springer, 1999, pp. 103–115.
5. A. a. I. Sidig, H. Luqman, S. A. Mahmoud, Arabic sign language recognition using optical flow-based features and hmm, in: Recent Trends in Information and Com- munication Technology: Proceedings of the 2nd International Conference of Reli- able Information and Communication Technology (IRICT 2017), Springer, 2018, pp. 297–305.
6. C. Neidle, A. Thangali, S. Sclaroff, Challenges in development of the american sign language lexicon video dataset (asllvd) corpus, in: 5th workshop on the represen- tation and processing of sign languages: interactions between corpus and Lexicon, LREC, Citeseer, 2012.
7. C. Lucas, R. Bayley, Variation in sign languages: Recent research on asl and beyond, Language and Linguistics Compass 5 (9) (2011) 677–690.

8. C. Valli, C. Lucas, Linguistics of American sign language: An introduction, Gal- laudet University Press, 2000.
9. [9] R. Rastgoo, K. Kiani, S. Escalera, Sign language recognition: A deep survey, Expert Systems with Applications 164 (2021) 113794.
10. N. Cihan Camgoz, S. Hadfield, O. Koller, R. Bowden, Subunets: End-to-end hand shape and continuous sign language recognition, in: Proceedings of the IEEE inter- national conference on computer vision, 2017, pp. 3056–3065.
11. N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, R. Bowden, Neural sign language translation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7784–7793.
12. S. Stoll, N. C. Camgöz, S. Hadfield, R. Bowden, Sign language production using neural machine translation and generative adversarial networks, in: Proceedings of the 29th British Machine Vision Conference (BMVC 2018), British Machine Vision Association, 2018.
13. M. P. Lewis, F. Gary, Simons, and charles d. fennig (eds.). 2013. ethnologue: Lan- guages of the world (2015).
14. F. Wang, C. Li, Z. Zeng, K. Xu, S. Cheng, Y. Liu, S. Sun, Cornerstone network with feature extractor: a metric-based few-shot model for chinese natural sign language, Applied Intelligence 51 (10) (2021) 7139–7150.
15. G. Shovkoplias, M. Tkachenko, A. Asadulaev, O. Alekseeva, N. Dobrenko, D. Kazantsev, A. Vatian, A. Shalyto, N. Gusarova, Support for communication with deaf and dumb patients via few-shot machine learning, in: Proceedings 14th International Conference on ICT, Society and Human Beings (ICT 2021), the 18th International Conference Web Based Communities and Social Media (WBC 2021), 2021.
16. S. Ferreira, E. Costa, M. Dahia, J. Rocha, A transformer-based contrastive learning approach for few-shot sign language recognition, arXiv preprint arXiv:2204.02803 (2022).
17. S. Ravi, M. Suman, P. Kishore, K. Kumar, A. Kumar, et al., Multi modal spatio temporal co-trained cnns with single modal testing on rgb–d based sign language gesture recognition, Journal of Computer Languages 52 (2019) 88–102.
18. K. M. Lim, A. W. C. Tan, C. P. Lee, S. C. Tan, Isolated sign language recogni- tion using convolutional neural network hand modelling and hand energy image, Multimedia Tools and Applications 78 (14) (2019) 19917–19944.
19. A. Wadhawan, P. Kumar, Sign language recognition systems: A decade systematic literature review, Archives of Computational Methods in Engineering 28 (3) (2021) 785–813.
20. S. Aly, W. Aly, Deeparslr: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition, IEEE Access 8 (2020) 83199–83212.
21. H. Luqman, E.-S. M. El-Alfy, Towards hybrid multimodal manual and non-manual arabic sign language recognition: Marsl database and pilot study, Electronics 10 (14) (2021) 1739.
22. P. Kumar, P. P. Roy, D. P. Dogra, Independent bayesian classifier combination based sign language recognition using facial expression, Information Sciences 428 (2018) 30–48.



23. A. Sabyrov, M. Mukushev, V. Kimmelman, Towards real-time sign language inter- preting robot: Evaluation of non-manual components on recognition accuracy., in: CVPR Workshops, 2019.
24. N. C. Camgoz, O. Koller, S. Hadfield, R. Bowden, Sign language transformers: Joint end-to-end sign language recognition and translation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10023– 10033.
25. [25] B. Saunders, N. C. Camgoz, R. Bowden, Progressive transformers for end-to-end sign language production, in: European Conference on Computer Vision, Springer, 2020, pp. 687–705.
26. W. Tao, M. C. Leu, Z. Yin, American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion, Engineering Applications of Artificial Intelligence 76 (2018) 202–213.
27. H. Luqman, E.-S. M. El-Alfy, G. M. BinMakhashen, Joint space representation and recognition of sign language fingerspelling using gabor filter and convolutional neural network, Multimedia Tools and Applications 80 (7) (2021) 10213–10234.
28. S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, Y. Fu, Skeleton aware multi-modal sign language recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3413–3423.
29. J. Huang, W. Zhou, H. Li, W. Li, Sign language recognition using 3d convolutional neural networks, in: 2015 IEEE international conference on multimedia and expo (ICME), IEEE, 2015, pp. 1–6.
30. O. M. Sincan, H. Y. Keles, Using motion history images with 3d convolutional networks in isolated sign language recognition, IEEE Access 10 (2022) 18608–18618.
31. S. Albanie, G. Varol, L. Momeni, T. Afouras, J. S. Chung, N. Fox, A. Zisserman, Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues, in: European conference on computer vision, Springer, 2020, pp. 35–53.
32. L. Momeni, G. Varol, S. Albanie, T. Afouras, A. Zisserman, Watch, read and lookup: learning to spot signs from multiple supervisors, in: Proceedings of the Asian Con- ference on Computer Vision, 2020.
33. R. Rastgoo, K. Kiani, S. Escalera, Zs-slr: Zero-shot sign language recognition from rgb-d videos (2021). doi:10.48550/ARXIV.2108.10059. URL https://arxiv.org/abs/2108.10059
34. P. Rodríguez, I. Laradji, A. Drouin, A. Lacoste, Embedding propagation: Smoother manifold for few-shot classification, in: European Conference on Computer Vision, Springer, 2020, pp. 121–138.
35. M. Bohácek, M. Hrúz, Sign pose-based transformer for word-level sign language recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 182–191.
36. J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Ad- vances in neural information processing systems 30 (2017).
37. D. Li, C. Rodriguez, X. Yu, H. Li, Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2020, pp. 1459– 1469.

Translate Arabic Text to Arabic Gloss for Sign Language

Doaa Alghamdi
doaaalghmdi@gmail.com
Department of Computer Engineering, College of Computer and Information Sciences, King Saud University

Mansour Alsulaiman
msuliman@ksu.edu.sa
Department of Computer Engineering, College of Computer and Information Sciences, King Saud University

Yousef Alohali
yousef@ksu.edu.sa
Department of Computer Science, College of Computer and Information Sciences, King Saud University

Mohamed A. Bencherif
mmekhtiche@ksu.edu.sa
Department of Computer Engineering, College of Computer and Information Sciences, King Saud University

Mohammed Algabri
mohmahgabri@gmail.com
Department of Computer Science and Information Systems, College of Applied Sciences, AlMaarefa University

Mohamed A. Mekhtiche
mabencherif@ksu.edu.sa
Department of Computer Engineering, College of Computer and Information Sciences, King Saud University

Abstract - Arabic Sign Language (ArSL) is a language used by the deaf community across Arab countries, but the lack of familiarity with ArSL among the hearing population often leads to social isolation for deaf individuals. The structural differences between ArSL and spoken Arabic pose significant challenges for machine translation. In this study, we enhance Arabic to ArSL gloss translation by employing data augmentation techniques, expanding the dataset from 600 to over 23,328 samples using sequence-to-sequence transformer models. Our approach achieved a substantial performance improvement, increasing the BLEU score from 11.1% in the baseline model to 52.72% on original test set. The best model achieved a BLEU score of 85.17% on augmented data test, underscoring the effectiveness of data augmentation in enhancing ArSL translation quality.

Keywords- Arabic Sign Language (ArSL); Gloss Text; Data Augmentation; Machine Translation; Sequence-to- Sequence Model; BLEU Score.

1. Introduction

The global deaf and hearing-impaired community, which constitutes over 5% of the world's population, relies heavily on sign languages for communication [1]. Sign languages are rich, visual-spatial languages that employ a combination of hand gestures, facial expressions, and body movements to convey meaning [2]. ArSL, in particular, serves as the primary mode of communication for the deaf community in Arab countries [3]. Despite its importance, ArSL remains largely unfamiliar to the hearing population, contributing to the social isolation of deaf individuals. Unlike spoken Arabic, ArSL has its own distinct syntax, grammar, and lexicon, making translation between these two languages a complex challenge.

The development of a semantic rule-based machine translation system for converting Arabic text to ArSL gloss, as demonstrated by [4], has laid an important foundation. However, these approaches have been constrained by the availability of training data and the inherent limitations of rule-based methodologies. The work was based on a relatively small parallel corpus of 600 Arabic sentences translated into ArSL gloss. While useful, this dataset is insufficient to capture the full variability of natural language. The rule-based system achieved BLEU score of 35%, highlighting the challenges in preserving the intended meaning and grammatical structure in translations. These limitations restrict the scalability and adaptability of the translation models, resulting in low accurate translations. The effectiveness of machine translation systems, particularly those designed for various language pairs such as Arabic language and ArSL, is heavily dependent on the availability of large, high-quality datasets. A more extensive dataset would allow for better training and generalization, leading to

more accurate translations [5]. Furthermore, advancements in Natural Language Processing (NLP) and machine learning techniques, such as sequence-to-sequence models, have demonstrated significant potential in improving translation accuracy by learning complex language patterns and relationships directly from data [6].

To address these limitations, our research aims to fill this gap by utilizing data augmentation techniques such as Blank Replacement, Synonym Replacement, and Sentence Paraphrasing to expand the original dataset from 600 to over 23,328 sentences. In addition, we evaluate the generated data by using the advanced Arabic sequence-to- sequence machine translation models and apply different data proportion techniques to examining the impact of dataset size on model performance. This approach makes the data more robust basis for training, capturing a wider range of linguistic diversity.

The contribution of this work is twofold: (1) We explore different data augmentation techniques to enhance the dataset size and quality of ArSL translation. (2) We investigate and compare different sequence-to-sequence machine translation models, by testing their performance on both the original test data and augmented test data.

2. Related Work

Translating Arabic text into ArSL is essential for integrating deaf individuals into their communities. However, developing effective translation systems faces challenges due to the scarcity of parallel corpora and incomplete documentation of ArSL's grammar and structure. ArSL translation research is still in its early stages compared to other sign languages [7] like American Sign Language (ASL) [8] and British Sign Language (BSL) [9]. Many existing systems rely on rule-based methods, requiring extensive linguistic knowledge to map spoken or written text to corresponding sign language expressions.

In the context of ArSL, several approaches have been explored. [10] focused on translating prayer-related Arabic sentences into ArSL using Sign Writing, limited by a small corpus and lack of coverage for various sentence structures. [11] used a chunk-based example-based machine translation (EBMT) approach, but their reliance on Google Tashkeel and example similarity led to high error rates. [12] developed a rule-based system that achieved high accuracy at the word level but did not adequately address sentence-level grammatical differences. [13] explored syntax transformations but were limited to specific grammatical structures.

Recent advancements have incorporated data-driven techniques, such as statistical ma- chine translation (SMT) and example-based machine translation (EBMT), which show promise but are limited by the availability of large, high-quality datasets. [4] developed a rule-based system for translating Arabic text into ArSL, utilizing a 600-sentence health domain corpus. While their system achieved over 80% accuracy, its limited dataset size restricted broader language applicability and generalization.

Additionally, efforts have been made to improve the availability and annotation of sign language data. The Jumla Sign Language Annotation Tool, described by [14], provides a web-based solution for annotating Qatari Sign Language (QSL) with written Arabic text, supporting the creation of annotated datasets such as the Jumla Qatari

Sign Language Corpus. [15] extended this work with the development of the JUMLA-QSL-22 corpus, containing 6,300 records annotated with glosses, translation, signer identity, and location. These tools and datasets are crucial for advancing sign language processing (SLP) and highlight the ongoing efforts to establish more comprehensive linguistic resources for Arabic-related sign languages.

Machine translation (MT) techniques have evolved significantly, with neural machine translation (NMT) models, such as sequence-to-sequence architectures, emerging as state- of-the-art methods. These models, including Recurrent Neural Networks (RNNs) and Transformer-based architectures, have demonstrated a strong ability to handle complex language pairs by learning from vast amounts of data to capture linguistic patterns and context [16]. For sign language translation, these models have been applied to other sign languages [17] and [18], achieving varying degrees of success. The challenge lies in effectively applying these models to ArSL, where data scarcity and linguistic differences pose significant obstacles. However, utilizing augmented datasets in conjunction with NMT models can enhance translation accuracy and help bridge the gap between spoken Arabic and ArSL.

In our previous work [19], we have shown the performance of the AraT5-V2 model for Arabic gloss machine translation which was evaluated using various data augmentation methods, including BR, SP, and SR. Experimental results shows that the BR method demonstrated superior performance, likely due to its larger dataset size of 22,404 samples. In contrast, the SP and SR methods, which utilized produced smaller datasets, exhibited higher validation losses and significantly lower BLEU scores. Further analysis was conducted by combining all three augmentation methods and compare with other models, including the original AraT5 Base and mT5 models, the AraT5 V2 model outperformed with test BLEU score of 90.93.

In this work, we extend the investigation by employing data augmentation techniques to expand the original dataset to over 23,000 samples and testing models on the original test set, while we tested the models by the augmented test set in our previous study. Moreover, in this study we apply data proportion techniques to study the impact of dataset size. In all experiment we compare the performance of AraT5 base, Arat5 v2 and mT5 models.

3. Methodology

In this section, we outline the methodology used to enhance the translation from Arabic text to gloss text. Our approach focuses on significantly expanding the dataset size through various data augmentation techniques and implementing advanced machine translation models. By enriching the dataset and leveraging sequence-to-sequence models, we aim to address the limitations of previous rule-based systems and improve translation accuracy and reliability.
Figure 1 shows an example of translation process from Arabic spoken language to sign language including an intermediate gloss representation, which serves as a crucial step in bridging the gap between the syntax of spoken Arabic and the grammar of ArSL. By translating spoken language into gloss text, the translation text to accurate sign language representations will be more easier for machine translation models.

اللغة المنطوقة Spoken Language	أعاني من التهاب في المعدة I suffer from inflammation in the stomach.			
النص الإشاري Gloss Text	المعدة Stomach	التهاب Inflammation	في IN	انا ME
[1]لغة الإشارة Sign Language				

Table 1: Illustration of the translation process from spoken Arabic to gloss text and then to sign language[1].

3.1 Data Augmentation Techniques

In order to overcome the limitations of the original ArSL dataset, which consisted of only 600 sentences primarily from healthcare settings, we employed data augmentation methods to expand the dataset to 23,328 sentences. The primary augmentation techniques used include BR, SR, and SP. These techniques were selected based on their ability to enhance the variability and robustness of the training data, which is essential for capturing the complex linguistic features of Arabic and ArSL.

A crucial component of our methodology is the development and use of an indexing algorithm, which ensures systematic processing and accurate mapping between Arabic text and its corresponding ArSL gloss. The indexing algorithm assigns indices to each word in both the original Arabic sentences and their gloss translations, maintaining alignment and consistency throughout the data augmentation process. This alignment is essential for preserving the semantic meaning of sentences when applying augmentation techniques, as it ensures that any modifications made to the original text are accurately reflected in the gloss version. The use of the indexing algorithm facilitates seamless integration of new data samples, allowing for scalable and efficient data augmentation, and setting a foundation for creating a high-quality, diverse dataset.

[1]https://sshi.sa/

Blank Replacement (BR), is a data augmentation methodology used in NLP to simulate missing or unknown words and enhance the performance of machine learning models. This technique involves masking selected words in a sentence and predicting those words based on surrounding context using the fill mask tool[2], and the AraELECTRA model [20]. By creating masked versions of original sentences and generating new candidate words, BR enables enriching the dataset with up to 21,804 new samples.

Synonyms Replacement (SR), introduces lexical diversity by substituting words with their synonyms from a predefined dictionary, thus increasing vocabulary variety while maintaining the overall meaning of the sentences. This method leverages a custom-built dictionary based on the Saudi Sign Language dictionary and the ArSL dataset, which ensures that synonyms are contextually relevant to ArSL. By exposing the model to different lexical choices that convey similar meanings, SR enhances the model's ability to generalize across various linguistic expressions and improves its adaptability to different word usages, resulting in the generation of 684 new sentences.

Sentence Paraphrasing (SP), is used to provide the training data with some variation and help the model learn various ways to deliver the same information by creating paraphrased versions of sentences through back translation (i.e., translating sentences into English and then back into Arabic). This process generates alternative phrasings that preserve the original meaning but differ in structure. Such variability is crucial for handling the significant structural differences between Arabic and ArSL. By training on paraphrased data, the model becomes more flexible in recognizing and accurately translating a wide range of sentence structures, thereby improving its capacity to capture meaning and maintain grammatical consistency in translations. SP is resulting in ٨٤٠ new sentences.

[2]https://huggingface.co/tasks/fill-mask

Figure 2 Shows the examples of dataset with augmented data that span a wide range of vocabulary and sentence structures, providing the model with the necessary exposure to capture linguistic nuances.

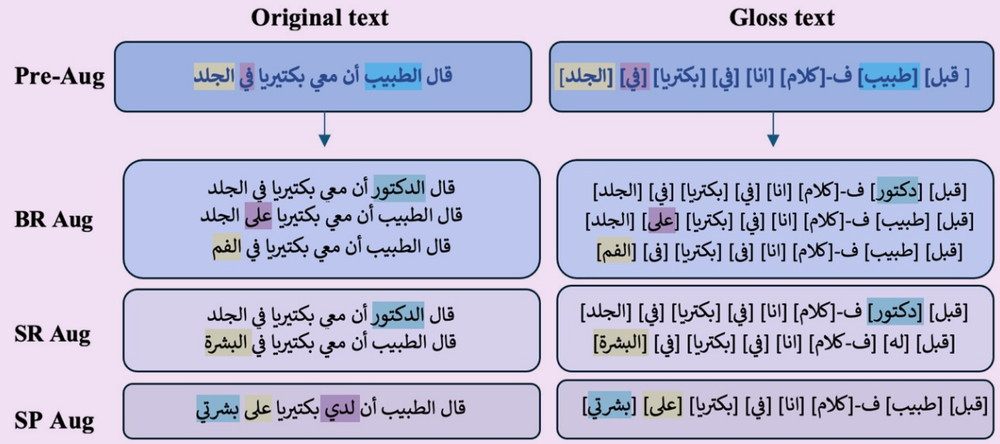


Figure 2. Arabic to Arabic gloss dataset samples.

Employing these data augmentation techniques, as detailed in our previous work [19], significantly enhances the dataset's size and diversity. This enriched dataset provides a solid foundation for training effective machine translation models, supporting the development of a robust system capable of accurately translating Arabic text into Arabic Sign Language gloss and improving accessibility and communication for the deaf community.

3.2 Sequence-to-Sequence Machine Translation Model

Sequence-to-sequence transformer models have become a cornerstone in natural language processing, particularly for tasks that involve transforming input sequences into output sequences, such as machine translation. These models are designed to handle input and output sequences of variable lengths, making them appropriate for translating text from one language to another while preserving the meaning of the samples. The T5 (Text-to- Text Transfer Transformer) model is a state-of-the-art Seq2Seq model that unifies various NLP tasks under a single framework by converting them into text-to-text tasks [6]. This architecture is particularly well-suited for translation tasks due to its ability to handle diverse linguistic patterns and context effectively.

In our study, we also utilize AraT5-V2, a variant of the T5 model training specifically for the Arabic language [21]. AraT5-V2 leverages the robust architecture of T5, optimized for handling the intricacies of Arabic syntax and semantics. The model consists of an encoder with multiple layers that each have a self-attention mechanism and a feed-forward network, followed by a decoder that also includes cross-attention to the encoder's output. This structure allows the model to generate translations that accurately reflect the source language's meaning and align with the target language's grammatical norms. Figure 3 shows the architecture of Arabic gloss machine translation using AraT5 Model.

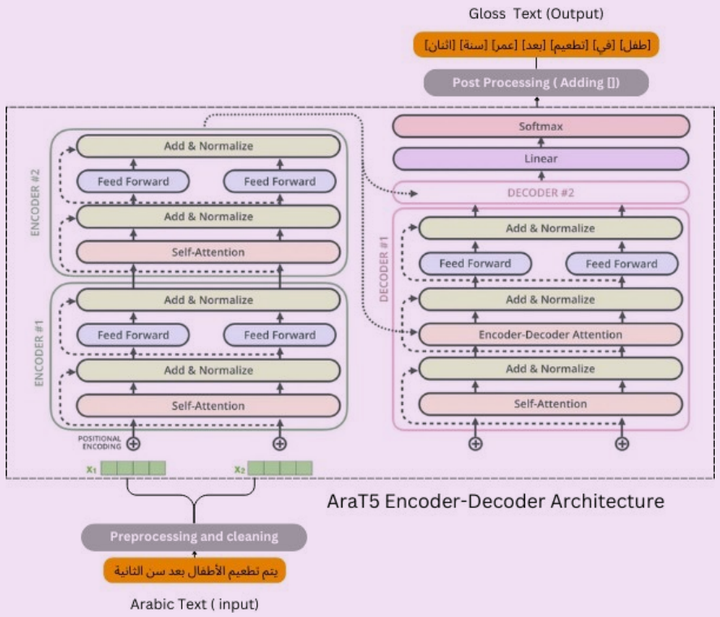


Figure 3. AraT5 Model architecture from gloss machine translation.

As shown in Figure 3, the process of training AraT5-V2 for translating Arabic text into Arabic Gloss Language involves a sequence of carefully structured steps to optimize the model's performance. Initially, the pre-trained AraT5-V2 model, which already understands general Arabic language structures, is further training on a parallel dataset of Arabic sentences and their corresponding Arabic gloss translations. During training, the encoder processes the input, converting it into a series of contextual representations that capture the sentence's meaning. These representations are then passed to the decoder, which generates the gloss output in an autoregressive manner—predicting one token at a time while using previously generated tokens to inform the prediction of the next. The model's parameters are adjusted using backpropagation, where the differences between the predicted gloss sentences and the actual gloss sentences are minimized using optimization algorithms like AdamW. Hyperparameters, including learning rate and batch size, are training to achieve optimal performance, and techniques such as early stopping are employed to prevent overfitting. Once the model demonstrates satisfactory performance on a validation set, it is considered ready for testing. Training AraT5-V2 model is capable of taking an Arabic sentence as input and generating its corresponding Arabic gloss translation. The encoder-decoder architecture ensures that the generated output maintains the semantic meaning and follows the syntactic rules of the gloss language, reflecting the knowledge gained during training. The performance of the model is evaluated using metrics such as BLEU scores, which measure the accuracy of the translated sentences against human-annotated references.

4. Experimental Results

We employed the AraT5 V2[2] model for our experiments, a state-of-the-art neural machine translation model tailored for Arabic text. Additionally, we evaluated two other models; mT5 [22], a multilingual transformer model capable of handling various languages, and AraT5 Base [21] a foundational version of the AraT5 tailored for Arabic but without the enhancements present in the V2 version. The primary evaluation metric used in these experiments is the BLEU score, which is commonly used in machine translation tasks to assess the quality of translations. In addition to BLEU scores, we also measure validation and test losses and compare model predictions against reference test sets. Training was conducted using an Adaptive Learning Rate with the AdamW optimizer, along with a dropout rate of 0.1 to

[3]https://huggingface.co/UBC-NLP/AraT5v2-base-1024

prevent overfitting. We employed a batch size ranging from 8 to 128, adjusted based on the dataset size, and a linear learning rate scheduler. The training ran for 22 epochs, with evaluation and model saving performed every 500 steps to monitor progress and prevent overfitting.

4.1 Performance of Data Augmentation techniques

To evaluate the performance of the trained AraT5-V2 model using different data augmentation methods, we calculated the BLEU scores for each method separately. Each data augmentation method was assessed using its own validation split during training, while the test BLEU scores were calculated using the original ArSL test set 90 samples to determine the overall impact on the translation task. The results of the AraT5-V2 model for each data augmentation method are presented in Table 1, alongside the results from the original dataset before augmentation.

Table 1. Performance Metrics of AraT5-V2 for Different Data Augmentation Methods.

Metric	BR	SP	SR	Original ArSL
Val. Loss	0.260	2.173	1.795	3.116
Val. BLEU	92.778	25.75	29.30	15.273
Test Loss	2.413	2.396	2.383	2.159
Test BLEU	52.71	13.33	12.90	11.069
Dataset Size	22404	1440	1284	600

As shown in Table 1, the BR method consistently demonstrated the best performance, with a validation loss of 0.260 and a validation BLEU score of 92.778. The test BLEU score for BR was 52.71, significantly higher than those of the other methods. The large dataset size of 22,404 samples for BR likely contributed to its superior performance, allowing the model to learn more robust translation patterns and effectively generalize to unseen data.

In contrast, SP and SR exhibited poorer performance, with validation losses of 2.173and 1.795, respectively, and evaluation BLEU scores of 25.75 for SP and 29.30for SR. The SR method, which used the smallest dataset size of 1,284 samples, had the lowest test BLEU score of 12.90. This indicates that limited data and vocabulary coverage significantly reduced its effectiveness. However, both SP and SR methods still performed better than the original unaugmented ArSL dataset, which had a test BLEU score of 11.069, demonstrating the value of data augmentation in enhancing translation quality.

Moreover, we combined all three data augmentation methods (BR, SP, and SR), resulting in a dataset of 23,328 samples which is used to train the AraT5 V2 model and compared against the AraT5 Base and mT5 models using the combined data augmented dataset. Note that we used the original test set to evaluate the models.

Table 2 demonstrates the results of the comparison of these models. As shown in Table 2, AraT5 V2 achieved the highest BLEU scores and the lowest validation and test losses, with a validation BLEU score of 86.16 and a test BLEU score of 69.41. The Base model also performed well in terms of validation BLEU score, reaching 35.190, but it had a significantly lower test BLEU score of 33.62. The mT5 model demonstrated moderate performance with a validation BLEU score of 72.380 and a much lower test BLEU score of 15.157.

Model	Val. Loss	Val. BLEU	Test Loss	Test BLEU
AraT5 V2	**0.492**	**86.16**	0.174	**69.41**
Base	1.610	35.190	0.979	33.62
mT5	0.586	72.380	**0.265**	15.157

Table 2. Comparison of Different Machine Translation Models.

4.2 Results of Data Augmentation Proportion

To further investigate the impact of data augmentation on the performance of our machine translation model, we conducted a data augmentation proportion experiment. This study examines how varying the proportions of augmented data—specifically 20%, 40%, 80%, and 100%—affects the AraT5 V2 model's translation accuracy. By systematically adjusting the amount of augmented data while maintaining consistent training setups, this experiment aims to identify the optimal balance between data variety and volume for enhancing model performance. The effectiveness of these proportions will be evaluated using BLEU scores, providing insights into how different levels of data augmentation contribute to the robustness and accuracy of the translation models. Table 3 shows the size of data proportion for different data augmentation methods.

Method	Split	20%	40%	80%	100%
BR	Train. Size	3,908	7,398	14,374	17,863
	Val. Size	526	962	1,834	2,270
	Test Size	526	962	1,834	2,270
	Total Size	4,360	8,722	17,443	21,804
SP	Train. Size	1,985	1,942	1,907	2,173
	Val. Size	18.45	22.45	23.55	25.75
	Test Size	106	123	157	174
	Total Size	168	336	672	840
SR	Train. Size	531	640	859	968
	Val. Size	103	117	144	158
	Test Size	103	117	144	158
	Total Size	137	274	547	684
All	Train. Size	4,995	8,728	16,191	19,923
	Val. Size	735	1,202	2,135	2,602
	Test Size	735	1,202	2,135	2,602
	Total Size	4,665	9,332	18,662	23,328

Table 3. Data Proportion and Dataset Sizes for Different Data Augmentation Methods.

[3]https://huggingface.co/UBC-NLP/AraT5v2-base-1024

Table 3 shows the dataset sizes for different data augmentation methods including BR, SP, SR, and the combination of them at proportions of 20%, 40%, 80%, and 100%. The Total Size represents the number of samples corresponding to each proportion from the complete dataset. Validation and test sizes were created by splitting the total dataset and including an additional 90 samples from the original ArSL dataset into the test set. This approach ensures that each augmentation method's impact can be evaluated under consistent conditions, providing insights into the effectiveness of varying data sizes and combinations on translation accuracy. Table 4 shows the results of test BLEU for each different data augmentation methods and proportions.

Table 4. Test BLEU Scores for Different Data Augmentation Methods and Proportions.

Method	20%	40%	80%	100%
BR	42.49	72.54	90.97	**91.57**
SP	12.63	16.52	20.27	21.77
SR	12.01	15.89	15.71	18.49
All	34.73	65.29	82.46	85.17

As shown in Table 4, the BR method outperformed other methods across all proportions, achieving the highest test BLEU score of 91.57 at 100% proportion. This indicates that the BR method provides the most robust training data for the model, likely due to its ability to capture diverse linguistic patterns effectively. Even at lower proportions, the BR method demonstrated substantial improvements, with a BLEU score of 42.49 at 20% augmentation, highlighting its strong impact even with less data.

In contrast, the SP and SR methods showed relatively lower test BLEU scores across all proportions, with the SP method reaching a maximum BLEU score of 21.77% at 100% augmentation and the SR method peaking at 18.4%. These results suggest that while SP and SR contribute to model performance, their impact is less pronounced compared to BR. The lower effectiveness of SP may be due to the inherent limitations of back- translation, which sometimes produces paraphrases that are too similar to the original or introduces noise that does not enhance the training process. For SR, the relatively poor performance could be attributed to the limited vocabulary coverage and context relevance of the synonym dictionary used, which might have resulted in substitutions that did not significantly vary the training data or, in some cases, distorted the sentence meaning.

The combined data augmentation methods (All) showed a balanced performance, achieving a test BLEU score of 85.17% at 100% augmentation, indicating that a mixture of augmentation techniques can yield high performance but may not surpass the effectiveness of BR alone. Overall, these findings emphasize the importance of selecting appropriate data augmentation methods and optimizing their implementation to enhance machine translation accuracy, as well as the need for more sophisticated approaches to improve SP and SR.

5. Conclusion

In this study, we enhanced the translation of Arabic text into Arabic gloss text using advanced data augmentation techniques and the AraT5 V2 model. Our results showed that the Blank Replacement method provided the highest translation accuracy, while the combined augmentation method also improved performance but did not surpass BR. However, in this study we used small dataset which was developed in the heath field which does not cover a common of Arabic words. Future work will focus on complete the second phase to translate the gloss to text to Arabic sign language motions, which leads to develop more robust and accurate sign language translation systems to better serve the Arabic-speaking deaf community.

References

1. Kushalnagar, R. (2019). Deafness and hearing loss. Web accessibility: A foundation for research, pages 35–47.
2. Luqman, H., Mahmoud, S. A., et al. (2017). Transform-based arabic sign language recognition. Procedia Computer Science, 117:2–9.
3. Al-Fityani, K. and Padden, C. (2010). Sign language geography in the arab world. Sign languages: A Cambridge survey, 20.
4. Luqman, H. and Mahmoud, S. A. (2019). Automatic translation of arabic text-to-arabic sign language. Universal Access in the Information Society, 18(4):939–951.
5. Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872.
6. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67.
7. Sidig, A. a. I., Luqman, H., and Mahmoud, S. A. (2018). Arabic sign language recognition using optical flow-based features and hmm. In Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017), pages 297–305. Springer.
8. [8] Zhao, L., Kipper, K., Schuler, W., Vogler, C., Badler, N., and Palmer, M. (2000). A machine translation system from english to american sign language. In Envisioning Machine Translation in the Information Future: 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000 Cuernavaca, Mexico, October 10–14, 2000 Proceedings 4, pages 54–67. Springer.
9. Marshall, I. and Sáfár, É. (2003). A prototype text to british sign language (bsl) trans- lation system. In The companion volume to the proceedings of 41st annual meeting of the association for computational linguistics, pages 113–116.

10. Almasoud, A. M. and Al-Khalifa, H. S. (2012). Semsignwriting: A proposed semantic system for arabic text-to-signwriting translation.
11. Almohimeed, A., Wald, M., and Damper, R. I. (2011). Arabic text to arabic sign language translation system for the deaf and hearing-impaired community. In Proceedings of the second workshop on speech and language processing for assistive technologies, pages 101–109.
12. El, A., El, M., and El Atawy, S. (2014). Intelligent arabic text to arabic sign language translation for easy deaf communication. International Journal of Computer Applica- tions, 92(8).
13. Al-Rikabi, S. and Hafner, V. (2011). A humanoid robot as a translator from text to sign language. In 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2011), pages 375–379.
14. Othman, A., Dhouib, A., Chalghoumi, H., Elghoul, O., and Al-Mutawaa, A. (2024). The acceptance of culturally adapted signing avatars among deaf and hard-of-hearing individuals. IEEE Access.
15. Othman, A., El Ghoul, O., Aziz, M., Chemnad, K., Sedrati, S., and Dhouib, A. (2023). Jumla-qsl-22: Creation and annotation of a qatari sign language corpus for sign lan- guage processing. In Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments, pages 686–692.
16. Angelova, G., Avramidis, E., and Möller, S. (2022). Using neural machine translation methods for sign language translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 273–284.
17. Jang, J. Y., Park, H.-M., Shin, S., Shin, S., Yoon, B., and Gweon, G. (2022). Automatic gloss-level data augmentation for sign language translation. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 6808–6813.
18. Kayahan, D. and Güngör, T. (2019). A hybrid translation system from turkish spo- ken language to turkish sign language. In 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pages 1–6. IEEE.
19. Alghamdi, D., Alsulaiman, M., Alohali, Y., Bencherif, M. A., and Algabri, M. (2024). Arabic gloss machine translation through data augmentation. In Proceedings of the Third SmartTech Conference (Manuscript submitted for publication). King Saud Uni- versity.
20. Antoun, W., Baly, F., and Hajj, H. (2020). Araelectra: Pre-training text discriminators for arabic language understanding. arXiv preprint arXiv:2012.15516.
21. Nagoudi, E. M. B., Elmadany, A., and Abdul-Mageed, M. (2021). Arat5: Text-to-text transformers for arabic language generation. arXiv preprint arXiv:2109.12068.
22. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934.

The Development of AI-Powered Automatic Video Sign Language Translation Systems

Assoc. Prof. Ozer Celik

ozer@ogu.edu.tr
Eskisehir Osmangazi University
Department of Mathematics and Computer Science,
Faculty of Science, Eskisehir Osmangazi University,
Eskisehir, Turkey
SignForDeaf

Ahmet Avcioglu

ahmet.avcioglu@engelsizceviri.com
Eskisehir Osmangazi University ETGB Technopark Building No:44/106 Eskisehir/ Turkey

Abstract - People who are deaf or hard of hearing often face challenges in fully understanding written subtitles in videos due to the differences between sign language and spoken language, each having unique grammar and structures. While many social media platforms provide automatic subtitles, they are often insufficient for accessibility. Sign language interpretation should be included to make video content fully accessible. Traditionally, adding sign language to videos involves a time-consuming process of recording and embedding separate translation videos, which must be redone with any changes to the original video. The plugin simplifies this process by dynamically translating updates directly from subtitle files, significantly reducing the time, effort, and cost involved. It allows seamless sign language support for any video without modifying the original content. The plugin integrates with video players, providing a user-controlled, customizable sign language window that can be activated, moved, resized, or turned off as desired.

Keywords
Sign language translation; Artificial intelligence; NLP; Digital accessibility.

Introduction

The barriers faced by individuals who are deaf or hard of hearing often go beyond auditory challenges, excluding them from essential parts of daily life, especially when it comes to written language comprehension. Spoken and sign languages are inherently different, not only in terms of modality but also in their syntactic structures, grammars, and semantics. This fundamental divergence means that subtitles alone are insufficient in ensuring full accessibility for the deaf and hard-of-hearing communities. Even with the widespread adoption of automatic subtitles across social media platforms and streaming services, many individuals find it challenging to interpret these captions, as they do not fully reflect the grammar or nuances of sign language. Thus, digital accessibility efforts that rely solely on written subtitles overlook an essential aspect of communication— information conveyed through sign language.

Numerous studies have highlighted the importance of sign language in enhancing accessibility for deaf and hard-of-hearing audiences, emphasizing that subtitles, while helpful, are not a substitute for sign language interpretation [1]. For instance, a report by the World Federation of the Deaf highlights that over 70 million people around the world use sign language as their primary means of communication, underscoring the significance of providing accessible media that includes sign language translation [3]. As a result, it is critical to go beyond insufficient accessibility measures and incorporate sign language interpretations in video content to meet the needs of people who are deaf or hard of hearing.

This paper explores the traditional methods of integrating sign language into video content, the limitations of these methods, and how modern technology, particularly artificial intelligence (AI), can revolutionize this process. We propose a set of innovative web, video, and PDF plugins that allow for the real-time generation of sign language interpretations synchronized with subtitles. This system provides an affordable, sustainable solution that can adapt to video updates, ensuring that deaf and hard-of-hearing users have continuous access to fully accessible content.

Methods

The process of developing a fully accessible video content platform for the deaf and hard of hearing requires overcoming several challenges, both technical and practical. Traditionally, the inclusion of sign language interpretation in videos involves a multi-step process, which starts with the script being translated into sign language by a trained interpreter. This process necessitates a professional studio setup where the interpreter's video is recorded, edited, and then embedded into the original video content. Each time the video or subtitles are updated, the sign language video must also be redone, which is both time-consuming and expensive. This method limits the scalability and feasibility of adding sign language support to a wide range of video content.

This approach transforms this process by utilizing artificial intelligence (AI) and advanced subtitle synchronization techniques. The system we developed extracts information from the subtitle file of the video and uses it to generate real-time, frame-by-frame sign language interpretations. This eliminates the need for costly re-recordings or post-production editing whenever changes are made to the video content or subtitles (Figure 1).

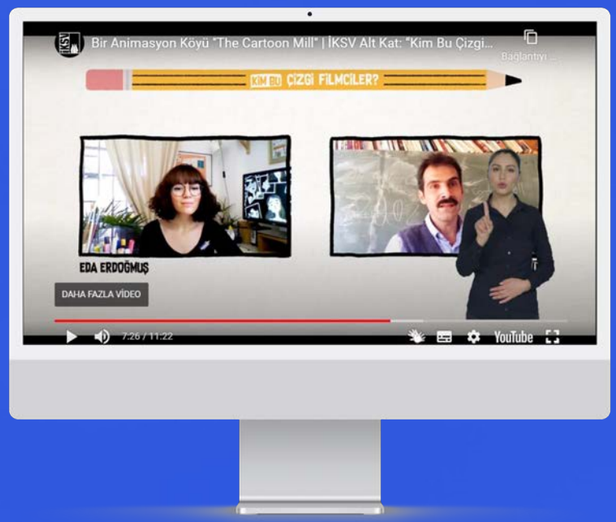


Figure 1. A screenshot from Istanbul Arts and Culture Foundation's video, using SignForDeaf's Video Sign Language Translation Plugin from their website (https://www.iksv.org/tr/haberler/iksv-alt-kat-yepyeni-bir-cevrimici-seriye-basliyor-kim-bu-cizgi-filmciler).

The core of this solution lies in its ability to sustainably translate updates from subtitles into sign language, providing a continuous and accurate interpretation that is synchronized with the video. The integration of this system with existing video platforms, such as YouTube, is simple and requires no modification to the original video file. Instead, the plugin acts as an overlay, allowing users to activate, move, resize, or disable the sign language window based on their preferences. This user-friendly feature ensures that the system can be tailored to the specific needs of each viewer, enhancing accessibility without compromising the visual integrity of the video.

Additionally, the plugin is designed with future adaptability in mind. As new sign languages are added, the system can easily be updated to accommodate different regions and languages, ensuring its applicability in diverse linguistic contexts. The modular design also allows for further AI advancements, such as improvements in real-time language recognition and varied interpretations of complex sentence structures, to be incorporated without requiring major changes to the existing framework.

Automatic Subtitle Synchronization

One of the most innovative aspects of our plugin is its ability to utilize artificial intelligence (AI) to synchronize subtitles with the appropriate sign language translation. In traditional systems, sign language videos must be carefully timed and manually synchronized with the content of the video, which is a tedious process, especially when dealing with videos that are frequently updated. Each time a new subtitle is added or modified, a complete re-recording of the sign language translation is required, followed by a re-integration of the video, which consumes both time and resources. SignForDeaf's AI-powered solution dynamically adjusts to any changes made in the subtitle files, automatically updating the corresponding sign language translation without the need for manual intervention. This feature allows for synchronization that responds to frequent updates, and the level of automation not only ensures accuracy but also provides the ability to handle large volumes of content, making it scalable for organizations that produce frequent and varied video outputs, a feature that previous solutions lack.

Customizable Sign Language Window

A key feature that sets our plugin apart is the fully customizable sign language window. Accessibility solutions are often criticized for being rigid, but this system prioritizes user experience in the design of the Video Sign Language Translation system. Viewers have complete control over the display of the sign language window, ensuring that it can be adapted to their individual preferences. This customization includes options to move, resize, or even disable the window as needed, giving users the flexibility to adjust the display based on their viewing environment and personal comfort. For example, a user watching a video on a small screen, such as a mobile phone, may prefer to reduce the size of the sign language window or move

it to a corner of the screen where it does not obscure important visual elements. Conversely, a user watching on a larger screen may choose to enlarge the window for greater visibility.

This user-centered approach ensures that the sign language interpretation does not interfere with the main video content, while still remaining easily accessible as needed. Additionally, users can enable or disable the sign language feature at any time, ensuring that those who choose to disable it can watch the video without distractions. This flexibility can significantly improve the accessibility and inclusiveness of video content, particularly for the deaf and hard of hearing communities.

Reduced Time and Cost

Traditional methods of adding sign language interpretation to videos are often prohibitively expensive and time-consuming. The process typically involves hiring a professional sign language interpreter, recording their translation in a studio, and then integrating the sign language video into the main content through post-production editing. This method requires significant human and financial resources, which can make it unfeasible for small content creators or organizations with limited budgets. Furthermore, any changes to the original video or subtitle text would require repeating this entire process, resulting in cost and time losses.

Our plugin eliminates these challenges by removing the need for constant reshooting and manual integration. Once the plugin is set up, the sign language interpretation is automatically generated based on the subtitles, meaning that any changes to the subtitles are immediately reflected in the sign language translation. This drastically reduces both time and cost losses required to maintain accessible video content, enabling more creators, educators, and organizations to provide sign language support without the financial burden. For instance, in the context of a large educational institution or media platform, where hundreds of videos may be produced each month, the time savings can be substantial, allowing resources to be allocated toward creating new content or improving other accessibility features instead.

Additionally, the reduced cost of implementation means that smaller organizations or independent creators, who previously might have been unable to afford sign language support, can now offer fully accessible videos. This broadens the reach of accessible content across different platforms and industries, from educational videos and online courses to entertainment and corporate training materials. Ultimately, this not only benefits the deaf and hard-of-hearing community by providing more inclusive content, but also encourages a more widespread adoption of accessibility practices across the media landscape.

Conclusion

The implementation of AI-powered sign language translation systems represents a significant advancement in making digital content more accessible to deaf and hard-of-hearing communities. By addressing the limitations of traditional subtitle-based accessibility, this solution offers a more inclusive, dynamic option to provide synchronized sign language interpretation. Not only does this approach alleviate the time and financial burdens typically associated with creating sign language videos, but it also ensures that accessibility features can be easily updated as content evolves. This is particularly important in educational, entertainment, and professional contexts, where content is frequently refreshed and updated.[2] Furthermore, this system opens new possibilities for the future of accessibility in media. As AI continues to advance in areas like NLP and gesture recognition, we can expect more sophisticated and contextually accurate translations, potentially bridging gaps between different sign languages and spoken languages worldwide.[5] This could be transformative not only for the deaf and hard-of-hearing community but for society as a whole, as it promotes inclusivity and breaks down communication barriers across linguistic and cultural lines.[4]

By integrating AI-powered sign language translation into mainstream video platforms, content creators and organizations have the opportunity to significantly improve the viewing experience for their audiences. The customizable, user-friendly nature of the Video Sign Language Plugin allows individuals to tailor their accessibility experience, making it a versatile tool that can accommodate a wide range of personal preferences and needs. Moreover, as more organizations adopt these technologies, we can anticipate a broader cultural shift toward the normalization of accessibility in digital content, benefiting not only the deaf and hard-of-hearing community but also society's collective understanding of inclusivity.[6]

In conclusion, the widespread adoption of AI-powered sign language translation tools has the potential to reshape the video landscape of the web, making digital content accessible to all. It is a solution that not only addresses current accessibility challenges but also paves the way for future advancements, ensuring that as technology progresses, no one is left behind.

Acknowledgments

We would like to thank our colleagues and research partners for their invaluable contributions to this project.

References

1. Wilson, M., & Moffat, P. (2018). The impact of subtitles and sign language on video accessibility. Journal of Deaf Studies and Deaf Education, 23(2), 204-215. doi:10.1093/deafed/eny012
2. Arfé, B., Rossi, C., & Sicoli, S. (2014). The role of sign language in reading comprehension for deaf individuals. Frontiers in Psychology, 5, 1174. doi:10.3389/fpsyg.2014.01174
3. World Federation of the Deaf (2021). Global accessibility report on sign language use in media. WFD Publications, pp. 12-36.
4. Liddell, S. K. (2003). Grammar, Gesture, and Meaning in American Sign Language. Cambridge University Press, pp. 45-89. ISBN: 9780521016505
5. Berke, J. (2020). AI and Accessibility: Bridging the communication gap for the deaf community. AI Journal of Linguistics, 15(3), 235-250. doi:10.1111/ail.153235
6. Napier, J., Leigh, G., & Goswell, D. (2016). Sign Language Interpreting: Theory and Practice in Australia and New Zealand. Federation Press, pp. 102-145. ISBN: 9781760021162