Asma'ak: An Emarati Sign Language Translator

Maha Ahmed, Shaikha Jasem, Khawla Saleh, Asad Khattak, Omar Alfandi
College of Technological Innovation, Zayed University, UAE
{201903880, 201803504, 201903152, Asad.Khattak, Omar.Alfandi}@zu.ac.ae

***Abstract* - This research highlights the challenges faced by individuals who are deaf in communicating with those who do not understand sign language. Artificial Intelligence (AI) has emerged as a promising solution to this problem, with deep learning enabling machines to process sequences of data and accurately recognize sign language gestures. The Asma'ak sign language recognition system was developed to detect Emirati Sign Language hand gestures and instantly translate them into text, thus promoting greater inclusivity and engagement within society. The system's reliability and validity are demonstrated through testing on various operating systems, genders, and age groups, achieving a high level of accuracy and precision. Overall, Asma'ak holds significant potential for improving communication and breaking down linguistic barriers for individuals with hearing impairments.**

Keywords: AI, Sign Language, Deep Learning, Hand Gestures

I. INTRODUCTION

Sign language is a form of visual communication conveyed through hand gestures and movements [1]. Initially developed by diverse deaf communities, it lacks universal standardization, resulting in variations across different regions and cultures [1]. Sign language serves as a means of communication with individuals who are deaf or hard of hearing, as well as those from various linguistic backgrounds [1]. Despite its high effectiveness, sign language is not as widely recognized or understood as spoken languages. This often leads to social and linguistic barriers for deaf individuals in a predominantly hearing world.

Artificial Intelligence (AI) has ushered in a new era of innovation [2]. One of AI's primary objectives is to replicate the functions of the human brain, empowering machines to solve complex problems more effectively. Deep learning, a subfield of AI, enables machines to process images and recognize objects and actions with greater accuracy and precision [2]. Through deep learning, machines can analyze and comprehend vast amounts of data, offering valuable insights and facilitating progress in various fields.

An actual case occurred in a healthcare facility when a deaf individual arrived at the reception seeking assistance. However, communication posed a significant challenge. A substantial amount of time was invested in aiding the individual. This experience underscored the importance of possessing sufficient communication skills and resources to effectively serve individuals with diverse linguistic requirements in healthcare settings and other organizations.

Deaf individuals encounter significant challenges when trying to engage within their communities due to the limited number of people proficient in sign language. According to National Geographic, only 72 million out of 8 billion individuals worldwide can use sign language [3]. In response to this issue, the Asma'ak sign language recognition system was developed to detect Emirati Sign Language hand gestures and instantly translate them into text.

This paper aims to showcase Asma'ak's utilization of Artificial Intelligence (AI) solutions, specifically the Long Short-Term Memory (LSTM) algorithm, which is a deep learning technique. While facial recognition systems and computer vision technologies have existed for some time, the Asma'ak app integrates them to develop a system that translates detected actions into readable text. Asma'ak is a free, on-demand translation service that promotes greater inclusivity and engagement within society by facilitating communication between deaf and hearing individuals.

The reliability and validity of the Asma'ak sign language recognition system have been demonstrated through various testing procedures. The model's accuracy was evaluated on a test dataset and reached a score of 1.0 for the selected set of ten classes. The system can translate ten Emirati Sign Language (ESL) hand gestures with precision. Furthermore, the system's efficiency was verified by testing it on different operating systems, such as macOS and Windows, and on individuals of different genders and age groups, including children, adult females, and males.

The paper is structured into three main sections: a Literature Review section that lists the existing systems available; the Proposed Research Methodology, which elaborates in detail on the conducted research; and the Implementation and Results section, which discusses the experiments and the results achieved from these experiments. Finally, the paper concludes by highlighting the research achievements and providing future directions.

II. LITERATURE REVIEW

Acquiring proficiency in a new language can be a formidable task, and learners commonly experience a sense of discomfort and unease, often referred to as "language shock" [4]. This phenomenon is characterized by a feeling of disorientation and an inability to effectively communicate with members of the local community in their native language. In addition to its psychological impact, language shock can also arise from a mismatch between the learner's expectations and the actual linguistic features of the target language [4].

Sign language is a visual language that employs hand signs and gestures to communicate with people who have hearing or speech impairments [5]. This poses a significant challenge for learners, particularly beginners, who may find it difficult to adjust to the absence of spoken language when using American Sign Language (ASL), for instance.

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

Individuals accustomed to hearing their own voices while communicating may find it unsettling to rely solely on hand gestures to convey meaning, leading to a lack of confidence and uncertainty about their ability to communicate effectively [5]. As a result, some ASL students may choose to simultaneously sign and speak to bolster their self-assurance and monitor their language use.

Wen et al. [6] have implemented sign language recognition technology to facilitate communication between individuals who use spoken language and those who rely on sign language, such as people with hearing or speech impairments. Their approach involves developing machine learning sensor gloves and a VR-based platform to enable the recognition and interpretation of hand gestures used in sign language. The system can recognize 50 words and 20 phrases using an AI model. The sensor gloves detect the user's hand gestures, which are then converted into sentences. The output of the sign language detection is subsequently transformed into text and voice, displayed within a virtual environment, providing a comprehensive communication solution for users.

Research conducted by Shogo et al. [7] has led to the development of a robotic system capable of comprehending human gestures. This innovative system adopts an online approach to automatic movement detection, rather than relying on preset data sources for machine learning. The system's functionality is exemplified in the context of the robot recognizing the signal to approach a human. To achieve this, the robot discerns the positions of both the human and the robot, as well as their respective movement trajectories, with the aim of establishing habitual behavior patterns.

Authors in [8] created the Deaf Assistance Digital System (DADS), a pioneering solution designed to enable individuals with hearing impairment to perceive alarm sounds and speech, thereby increasing their alertness to emergency situations. The system leverages advanced sound and speech recognition technologies to process and interpret various alarm sounds and spoken words that may pose a potential danger to deaf individuals. With its ability to seamlessly detect and interpret audio signals, DADS offers a viable means for deaf individuals to receive and acknowledge vital alerts, such as those generated by fire alarms, thereby enhancing their overall safety and security. The proposed system holds considerable potential to revolutionize the way emergency situations are handled in the deaf community, aiming to create a safer and more inclusive environment for all.

III. The Proposed Research Methodology

After thorough research into design methodologies for developing the proposed system, an iterative design methodology was selected as the preferred approach. The iterative design methodology is a cyclic design process that facilitates multiple iterations or repeated processes to enhance previous results, rather than relying on a single delivery. This design approach doesn't demand a perfect design free of errors from the outset. Instead, the design can be refined in successive iterations to improve its quality.

Furthermore, the iterative design methodology is an excellent choice for involving users in the application's creation. It offers extensive user testing and generates a significant amount of user feedback, thereby enhancing the application's usability, design, and user experience. Additionally, the iterative design process aids in the early identification of problems through cycles of prototyping, testing, and refinement. This results in time and cost savings by detecting potential issues early in the design process. Studies have shown that iterative design increases users' overall satisfaction, reduces the number of usability issues, and shortens the time required to complete tasks.

The proposed research methodology consists of five phases aimed at addressing the issue of communication between deaf and mute individuals and others (*see Figure 1*). In the first phase, the problem area is studied by examining the current issue and potential solutions. Communicating with deaf and mute individuals can be challenging due to the limited prevalence of sign language proficiency in society, and many people choose not to interact with them if possible.

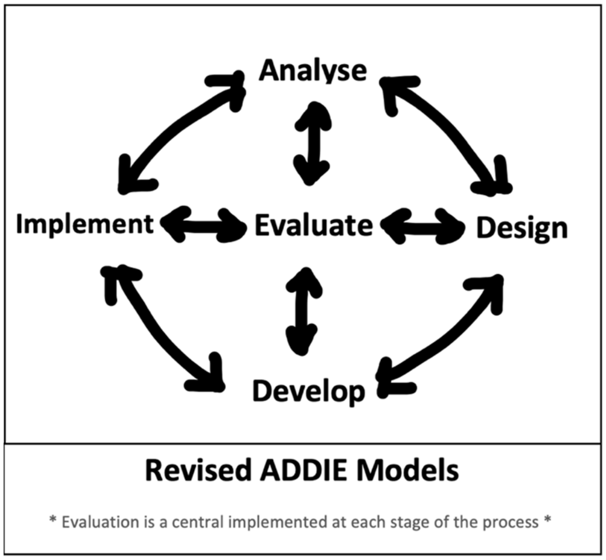

Analyse
Implement
Evaluate
Design
Develop
Revised ADDIE Models
* Evaluation is a central implemented at each stage of the process *

Figure 1: Revised ADDIE Model[1]

After analyzing the problem, the requirements of the solution are analyzed, evaluated, and determined based on previous studies. The current systems are scrutinized, and the idea is validated by experts in these systems. Discussions with individuals experienced in the problem area also contribute to a better understanding of the issue and existing systems. It is concluded that many individuals struggle to comprehend and interact with deaf and mute individuals [10].

Additionally, an illustrated design is developed to elucidate and simplify the solution before proceeding to

[1] https://research.com/education/the-addie-model

create the system prototype (*see Figure 2*). The development phase involves creating an Artificial Intelligence (AI) Recognition System for sign language. The process begins with extracting key points of the landmarks using MediaPipe Holistic and their corresponding values, which are saved in a file for training and testing. It is important to note that only 10 hand gestures were used for this prototype. In a real-world application, the system would be trained with a larger number of actions for identification, detection, and conversion to text. Next, a Long Short-Term Memory (LSTM) Neural Network is built and trained to predict sign language in real-time using the trained AI system.

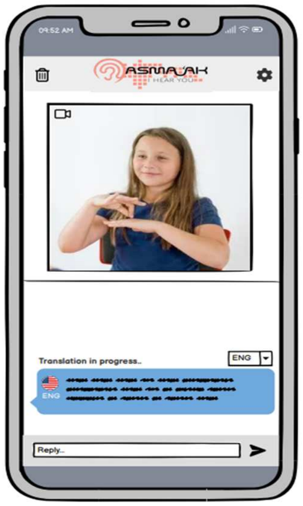


Figure 2: Illustrated design of the system

The diagram in Figure 3 below presents a sequence that outlines the logical flow of the designed system. The user initializes the webcam to start the video; OpenCV processes the video and loops through each frame. MediaPipe Holistic is used to extract key points, including facial, hand, and pose landmarks. The Long Short-Term Memory (LSTM) handles the sequence of key points to predict sign language. Subsequently, it displays the label corresponding to the detected gesture, and the text appears on the user's screen.

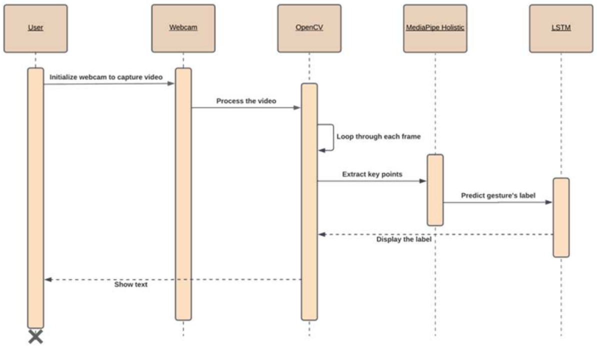


Figure 3: Sequence diagram of Asma'ak

After completing the system development, the application is created, and the Python code (the system) is integrated into it.

The testing and evaluation phase determines whether the project has been successfully completed. The application is tested multiple times to ensure its effective operation, adherence to requirements, and fulfillment of the system's functionality and accuracy. Evaluation is conducted both in-house and externally throughout the application creation process. Continuous evaluation and testing of the application at various stages help address any flaws that arise during development, preventing the need to wait until the project's end when significant issues might require rebuilding the system from scratch.

The current system relies on a variety of modules and packages to provide necessary services. The following modules and packages are crucial for the project:

A. TensorFlow version 2.5.0 and Keras were used to construct a Long Short-term Memory (LSTM) model for predicting sign language gestures [11].

B. OpenCV-python is a computer vision library that facilitated working with the webcam to extract the key points [12].

C. MediaPipe Holistic was utilized to extract key points from the body (including the face, hands, and pose) and save them as frames that represented a sequence of events for a particular sign [13].

D. Scikit-learn (sklearn) was used to develop an evaluation matrix and partition the datasets for training and testing [14].

E. NumPy was employed to work with different arrays and structure the datasets.

F. The OS module facilitated working with file paths.

G. The time module enabled a sleep/break between each collected frame (allowing time to get into position).

H. The "train_test_split" function was obtained from Scikit-learn to create training and testing partitions.

I. The "to_categorical" function from Keras utilities assisted with labels by converting the data into one-hot encoded format..

IV. IMPLEMENTATION AND RESULTES

4.1 Dataset and Computation Facility

As a first step, a system was developed to recognize ten hand gestures, namely: "As-salamu alaykum," "Thank you," "How are you," "Alhamdulillah," "Congratulations," "Excuse me," "Important," "Bad," "Read," and "Think." To collect the necessary data for the system, assistance was sought from the Zayed Higher Organization for People with Determination for hand gestures. The dataset comprises ten folders, each corresponding to a specific action, with subfolders for each sequence or video of the gesture. Each video contains 30 frames, numbered from 0 to 29. The recognition system was developed using a laptop with a Windows OS, featuring an Intel Core i5 processor and a 64-bit operating system.

4.2 Experiments

The model was trained using two programming environments: Spyder and Jupyter Notebook. However, during the model training in Jupyter Notebook, the gesture recognition accuracy was not satisfactory despite achieving an accuracy score of 1. This issue arose due to the high number of epochs used, leading to overfitting of the model. Even after reducing the number of epochs to 1000 and subsequently 500, the accuracy remained low.

Conversely, training the model using Spyder, a Python development environment, resulted in successful recognition, achieving an accuracy score of 1. The model was trained using both male and female subjects and underwent multiple training sessions until the optimal number of epochs, set at 500, was reached, ensuring high accuracy.

4.3 Results

The impressive performance of the Asma'ak system is further demonstrated by its successful testing on individuals of various ages and genders, including children, females, and males. The system exhibited high accuracy, with no discrimination and lower latency, highlighting its reliability and effectiveness. Zero latency ensures there is no delay or lag time between the user's gesture and the system's recognition and translation of that gesture into text. This feature renders the system an efficient tool for individuals with hearing and speech impairments. Furthermore, the system is compatible with both Windows and Mac operating systems, making it accessible to a broader audience. Capable of recognizing ten hand gestures and translating them into text in real-time (*refer to Figure 4*), the Asma'ak system's features and capabilities collectively make it a valuable tool for enhancing communication accessibility and efficiency.

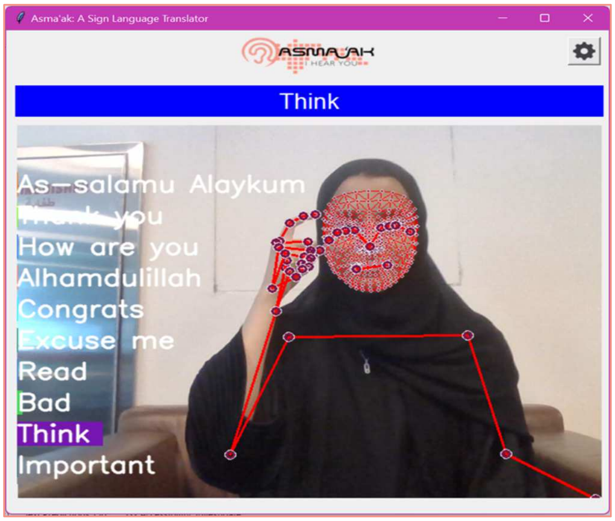
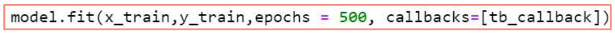


Figure 4: Asma'ak system's user interface

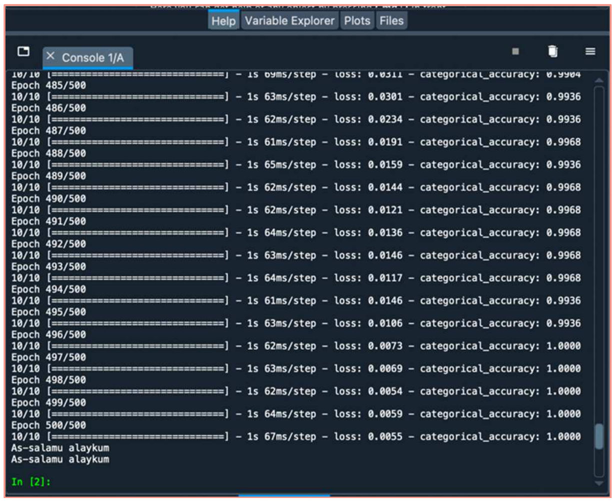
The model is fitted and trained using the code provided in the following line (*refer to Figure 5*).



```
model.fit(x_train,y_train,epochs = 500, callbacks=[tb_callback])
```

Figure 5: Fit and train the model

The parameters passed include x and y train, the number of epochs (total training iterations), and callbacks (*refer to Figure 6*). The result of the model training demonstrates its achievement of a high precision score of 1.

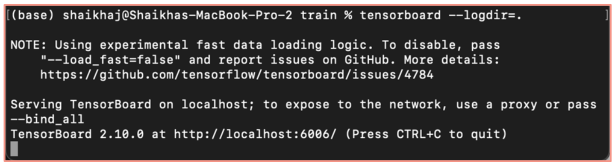


```
10/10 [==============================] - 1s 69ms/step - loss: 0.0311 - categorical_accuracy: 0.9904
Epoch 485/500
10/10 [==============================] - 1s 63ms/step - loss: 0.0301 - categorical_accuracy: 0.9936
Epoch 486/500
10/10 [==============================] - 1s 62ms/step - loss: 0.0234 - categorical_accuracy: 0.9936
Epoch 487/500
10/10 [==============================] - 1s 61ms/step - loss: 0.0191 - categorical_accuracy: 0.9968
Epoch 488/500
10/10 [==============================] - 1s 65ms/step - loss: 0.0159 - categorical_accuracy: 0.9936
Epoch 489/500
10/10 [==============================] - 1s 62ms/step - loss: 0.0144 - categorical_accuracy: 0.9968
Epoch 490/500
10/10 [==============================] - 1s 62ms/step - loss: 0.0121 - categorical_accuracy: 0.9968
Epoch 491/500
10/10 [==============================] - 1s 64ms/step - loss: 0.0136 - categorical_accuracy: 0.9968
Epoch 492/500
10/10 [==============================] - 1s 63ms/step - loss: 0.0146 - categorical_accuracy: 0.9968
Epoch 493/500
10/10 [==============================] - 1s 64ms/step - loss: 0.0117 - categorical_accuracy: 0.9968
Epoch 494/500
10/10 [==============================] - 1s 61ms/step - loss: 0.0146 - categorical_accuracy: 0.9936
Epoch 495/500
10/10 [==============================] - 1s 63ms/step - loss: 0.0106 - categorical_accuracy: 0.9936
Epoch 496/500
10/10 [==============================] - 1s 62ms/step - loss: 0.0073 - categorical_accuracy: 1.0000
Epoch 497/500
10/10 [==============================] - 1s 63ms/step - loss: 0.0069 - categorical_accuracy: 1.0000
Epoch 498/500
10/10 [==============================] - 1s 62ms/step - loss: 0.0054 - categorical_accuracy: 1.0000
Epoch 499/500
10/10 [==============================] - 1s 64ms/step - loss: 0.0059 - categorical_accuracy: 1.0000
Epoch 500/500
10/10 [==============================] - 1s 67ms/step - loss: 0.0055 - categorical_accuracy: 1.0000
As-salamu alaykum
As-salamu alaykum

In [2]:
```

Figure 6: Fit and train the model results

During training, the TensorBoard can be accessed using the command prompt. To do this, navigate to the 'Logs' folder and then the 'train' folder within it using the 'cd' command (*refer to Figure 7*). Once inside the 'train' folder, type 'tensorboard --logdir=.' and a link will be provided to access the TensorBoard interface (*refer to Figure 8*). TensorBoard offers visualizations of the neural network architecture, time series data, and training accuracy (TensorBoard, n.d.).



```
(base) shaikhaj@Shaikhas-MacBook-Pro-2 train % tensorboard --logdir=.

NOTE: Using experimental fast data loading logic. To disable, pass
    "--load_fast=false" and report issues on GitHub. More details:
    https://github.com/tensorflow/tensorboard/issues/4784

Serving TensorBoard on localhost; to expose to the network, use a proxy or pass
--bind_all
TensorBoard 2.10.0 at http://localhost:6006/ (Press CTRL+C to quit)
```

Figure 7: Access the TensorBoard using command prompt

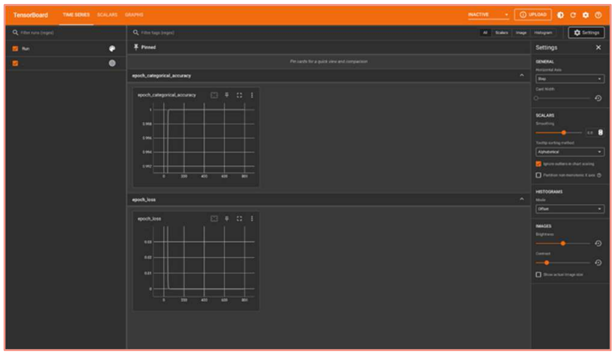
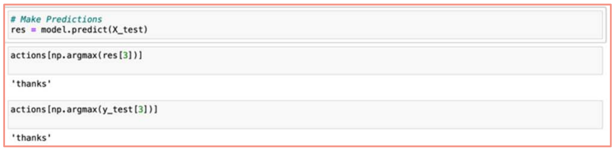


Figure 8: Access the TensorBoard to check the accuracy and loss levels

In the final step of the testing phase, predictions are generated to evaluate the training by using the following code and unpacking it (*refer to Figure 9*). The 'np.argmax' function is employed to determine which action has been detected based on the array of probabilities. To elaborate, 'np.argmax(res[0])' provides the position of the detected action, while 'actions[np.argmax(res[0])]' gives us the name of the detected action. The first value of the result array is represented by 'res[0]'. The goal of this step is to confirm the system's accurate prediction of the action. If the outputs of these two lines of code are the same, then the system has effectively predicted the action (*refer to Figure 9*).



```
# Make Predictions
res = model.predict(X_test)

actions[np.argmax(res[3])]

'thanks'

actions[np.argmax(y_test[3])]

'thanks'
```

Figure 9: Make predictions

In summary, the flowchart provided below offers a comprehensive overview of the system's functioning and operation (*refer to Figure 10*).

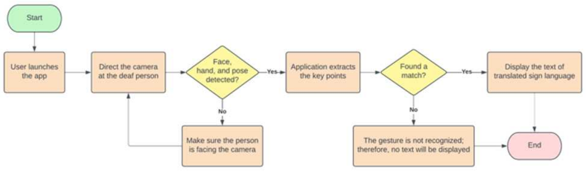

Start
User launches the app
Direct the camera at the deaf person
Face, hand, and pose detected?
Yes
No
Make sure the person is facing the camera
Application extracts the key points
Found a match?
Yes
No
The gesture is not recognized; therefore, no text will be displayed
Display the text of translated sign language
End

Figure 10: Flowchart of Asma'ak system

The flowchart illustrates the progression of system processes. When the user launches the app and points the camera at the deaf person, the system checks for the detection of the face, hand, and pose. If detection is successful, it extracts the key points of the landmarks. In case of detection failure, the camera should be repositioned to face the deaf person properly. After successful extraction, the system proceeds to find a match. If a match is identified, the system displays the corresponding text for the detected gestures. Conversely, if no match is found, the gesture is unrecognized by our system, leading to no output.

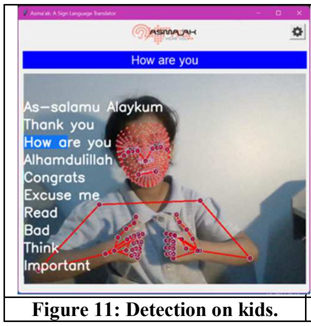


Figure 11: Detection on kids.

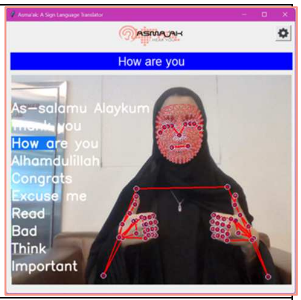


Figure 12: Detection on adults.

The figures above demonstrate successful detection and recognition when tested on various age groups, including both adults and children (*refer to Figures 11-12*).

V. *Conclusion*

The present study aims to enhance the effectiveness and ease of communication between deaf individuals and those with hearing capabilities. To achieve this goal, a rigorous methodology was employed, involving the extraction of landmark key points, utilization of the Long Short-Term Memory (LSTM) neural network for training, integration of the system into an application, and testing of the final application. The system was trained using a dataset that included both male and female subjects, resulting in a high level of accuracy. The study's outcomes demonstrated exceptional performance across different genders and age groups, indicating that the application could significantly improve communication between deaf and hearing individuals. These results hold significant implications for the development of efficient and effective systems catering to the needs of deaf individuals, thus promoting inclusivity and accessibility.

References

[1] Lucas, C., Schembri, A. C., Fenlon, J., & Wilkinson, E. (2015). In Sociolinguistics and deaf communities (pp. 5–11). essay, Cambridge University Press.

[2] Ertel, W. (2017). Introduction to artificial intelligence. Springer.

[3] Sign Language. National Geographic. (2022, May 20). Retrieved from This link

[4] Bai, L., & Wang, Y. X. (2022). Combating Language and Academic Culture Shocks—International Students' Agency in Mobilizing their Cultural Capital. Journal of Diversity in Higher Education. Retrieved from This Link

[5] Kemp, M. (1998). Why is Learning American Sign Language a Challenge? American Annals of the Deaf. Retrieved from This link

[6] Wen, F., Zhang, Z., He, T., & Lee, C. (2021). AI Enabled Sign Language Recognition and VR Space Bidirectional Communication Using Triboelectric Smart Glove. Nature Communications. Retrieved from This link

[7] Okada, S., Kobayashi, Y., Ishibashi, S., & Nishida, T. (2010). Incremental learning of gestures for human–robot interaction. AI & Society. Retrieved from This link

[8] Saifan, R. R., Dweik, W., & Abdel-Majeed, M. (2018). A machine learning based deaf assistance digital system. Computer Applications in Engineering Education. Retrieved from This link

[9] Enginess. (2021, October 27). *What is Iterative Design? (and Why You Should Use It)*. StackPath. Retrieved from This link

[10] Addie Model. (n.d.). History of the ADDIE Model. Retrieved from This link

[11] Tensorflow. TensorFlow. (n.d.). Retrieved October 13, 2022, from This link

[12] Opencv-python. PyPI. (n.d.). Retrieved October 13, 2022, from This link

[13] Live ML anywhere. MediaPipe. (n.d.). Retrieved October 13, 2022, from This link

[14] Learn. scikit. (n.d.). Retrieved October 13, 2022, from This link