Development of a Pattern Recognition System for Interpreting Sign Language Using Convolutional Neural Network

H.O. Mahmud[1*], S.L. Ayinla[1], A.O. Jimoh-Mahmud[2], A.O. Ekogiawe[1]

[1]Department of Computer Engineering, University of Ilorin, Ilorin, Nigeria.
[2]Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria.

ABSTRACT: Sign language is essential for communication among the deaf, hearing-impaired, and the general public. However, many people with normal hearing struggle to understand sign language, resulting in ineffective interactions with the deaf community in their daily activities. Recent studies have employed machine learning algorithms and computer application software solely to convert sign languages into text, aiming to address these challenges and bridge the communication gap among the deaf, hearing-impaired, and the public. This study developed a device utilizing a convolutional neural network (CNN) and an embedded system to translate sign languages into text and audio, enabling daily interactions among the deaf, hearing-impaired, and others in both private and public settings. The system comprises hardware components, including an OV2640 camera sensor to capture images of the signers, an ESP32 microcontroller to process the captured images and convert them into text and speech after being trained with a CNN model, a display screen to read the interpreted images as text, and a speaker that outputs the interpreted images as speech. To validate the proposed framework, 34,628 images were collected from a publicly available American Sign Language dataset. In this dataset, 80% of the images were used as training data, and 20% were reserved for testing. The results show a training accuracy of 99.76% but a low validation accuracy. The system's testing accuracy could be reliably improved with good lighting, a plain background, a microcontroller that has increased memory size, and a higher resolution camera, considering the achieved training accuracy.

KEYWORDS: Pattern Recognition, Deep Learning, Convolutional Neural Network, Computer Vision, Sign Language.

[Received Feb. 4, 2025; Revised Feb. 22, 2025; Accepted March 10, 2025] Print ISSN: 0189-9546 | Online ISSN: 2437-2110

I. INTRODUCTION

According to a report from the World Health Organization (WHO), approximately 5% (about 430 million) of the global population experiences some level of severe or moderate hearing loss. Furthermore, it is projected that by 2050, over 2.5 billion people could suffer from hearing loss disabilities (World Health Organization, 2025). This portion of the total population can only communicate through their respective regional sign languages. These languages are not easily understood by the general hearing community, thus creating a communication gap between hearing and speech-hearing impaired communities. Sign language (SL) is a form of communication that conveys information through hand gestures, facial expressions, and body movements. It is commonly used, especially among individuals with hearing or speech impairments (Liu *et al*., 2023). Sign language recognition (SLR) is the use of technology such as machine learning, computer vision, and sensors to interpret gestures from sign languages into text, speech, or other forms of communication (Sharma & Singh, 2021). SLR bridges the communication gap between the deaf and hearing-impaired individuals and those who do not understand sign language, enabling real-time interactions. The common SLR methods include motion sensor-based techniques (which use a glove with sensors) and computer vision approaches that employ cameras (Halder & Tayade, 2021). The former approach produces good recognition results; nevertheless, it may be expensive, and users may experience inconveniences. The latter method is more user-friendly as the signals are acquired using cameras (Wangchuk *et al*., 2021).

Pattern recognition is a data analysis technique that employs machine learning algorithms to classify input data by recognizing patterns and characteristics. This technique automates the detection of data patterns in images, videos, text, and audio, and it is applicable in various fields, including astronomy, healthcare, robotics, and remote sensing (Amosa, bt Izhar, *et al*., 2023). The process consists of three steps: examining input data, identifying patterns, and matching them with existing data (Kanade, 2023). It encompasses two phases: the exploratory phase, where algorithms analyse data patterns, and the descriptive phase, where patterns are classified and linked to new data. Common methods in pattern recognition encompass statistical methods, syntactic approaches, neural networks, template matching, and fuzzy-based techniques. Computer Vision and Deep Learning are essential for pattern recognition in sign language interpretation (Kanade, 2023).

*Corresponding author: philemon@fuwukari.edu.ng
doi: http://dx.doi.org/10.4314/njtd.v22i1.3365

Computer vision is cantered on the automatic retrieval, analysis, and interpretation of important information from either a single image or a series of images (Amosa, Sebastian *et al*., 2023; Aromoye *et al*., 2025; Smits & Wevers, 2022). This involves processes like image preprocessing, feature detection, segmentation, 3D reconstruction, object recognition, and motion analysis. Deep Learning allows computational models with multiple processing layers to learn data representations at various abstraction levels (LeCun *et al*., 2015), enhancing fields like speech recognition and visual object detection. Communication emphasizes sharing and transmitting information for human interaction. It involves six key elements: sender (encoder), message, channel, receiver (decoder), noise, and feedback. Effective communication is essential in daily life; sign language serves primarily speech-impaired individuals (Oguntimilehin & Balogun, 2024). Its variations exist globally and are recognized as a natural language that aids interaction between deaf individuals and those who cannot speak (Halder & Tayade, 2021). Recent advancements in deep learning and computer vision have led to the development of automatic sign language recognition systems (Halder & Tayade, 2021), bridging this gap and promoting equality for hearing-impaired individuals. Interaction between individuals with speech impairments and others in public places has always been a recurring challenge; hence, there is a need to develop a pattern recognition system for effective communication.

Mehedi & Arafat (2024) researched sign language recognition using hand gestures from images to enhance the reliability and effectiveness of a Hand Sign Detection System for real-world applications. They introduced a new dataset featuring hand signs from letters A to J, each with 100 images that varied in hand position, skin tone, and lighting conditions, compiled using the Media pipe framework. It was employed for data analysis, resulting in an overall accuracy of 87.3% in recognizing signs, though the system struggled with the sign 'I'. The study emphasized the challenges of using datasets typically gathered in controlled environments and the need for robust performance in diverse situations. Moreover, the system's accuracy was compared with other machine learning algorithms, demonstrating its effectiveness. The study by Lawal *et al*. (2024) on developing and evaluating an interactive mobile app for the English-Hausa sign language alphabet evaluated a mobile application designed to teach English to Hausa-speaking students with hearing impairments. The evaluation process included pre-test exams to assess baseline understanding, lecture sessions, and post-tests to measure performance improvement. Data analysis involved various statistical tests to determine the intervention's effectiveness. Student feedback revealed mixed perceptions, with some unfamiliarity with HSL (Hausa Sign Language) negatively impacting usability ratings. The study aimed to assess the mobile app's effectiveness in teaching English to the target population.

The work on hand gesture recognition for sign language transcription by Vazquez (2017) emphasizes the significance of hand gesture recognition in enabling communication for individuals with hearing impairments. It examines various sensors used for data collection, highlighting their strengths and weaknesses, and justifies the choice of a camera capturing both RGB and depth information for its advantages in gesture recognition tasks. The methodology involves developing a classification model using CNNs. The study aims to facilitate communication for the deaf and hearing-impaired persons by leveraging hand gesture recognition technology. The research on comparative analysis of sign language recognition systems by Swati *et al*. (2019**),** presents a comparative analysis of sign language recognition systems, which typically involve data gathering using RGB cameras, preprocessing to improve skin tone detection, feature extraction using methods like HOG, SIFT and PCA, and classification using machine learning techniques such as KNN, CNN or PNN to achieve real-time processing efficiency by employing algorithms that can efficiently process camera input and offer feedback while incorporating skin segmentation methods to enhance recognition precision. However, factors like differences in skin colour, backgrounds, and lighting can impact the system's accuracy, and the complexity of sign language, including facial expressions and body language, poses challenges for accurate recognition. The dependency on a well-structured dataset and the computational cost of deep learning models are also limitations that need consideration.

The work by Halder & Tayade (2021) was a real-time vernacular sign language recognition using Media Pipe and machine learning, highlighting Media Pipe's role in image preprocessing. Media Pipe is a versatile framework designed for developing cross-platform machine-learning pipelines that handle various data types, including video and audio. It includes models for human body detection and tracking, developed from extensive datasets at Google. The framework consists of a graph architecture defined in a pb.txt file, which connects C++ files that extend a base calculator class. The hand-tracking system relies on a two-part machine-learning pipeline comprising the Palm Detection Model and the Landmark Model. The Palm Detection Model produces a cropped image of the palm, which the Landmark Model uses for further processing. This approach simplifies detection, as it estimates bounding boxes around rigid objects like palms or fists instead of directly detecting hands, reducing the need for extensive data augmentation methods. Using an encoder-decoder architecture, the system extracts a broader scene context and employs the hand landmark model.

Unlike the previous studies, that utilized machine learning algorithms and computer application software only to interpret sign languages to readable texts that can only be accessed via the Internet using a mobile or window application, this work embedded the existing technology in a hardware device that can be deployed for use in private and/or public facilities in an offline mode, a critical advancement in facilitating communication between the deaf and hearing communities, thereby enabling them to voice their opinions and feelings seamlessly at an affordable rate. The system is developed using a hardware architecture, which comprises an IoT-based microcontroller unit, camera, TFT display, texts-to-speech module (DF mini-player), LM386, and power supply unit. However, the choice of the IoT-based microcontroller does not affect its offline performance; It is simply for cloud data storage for research purposes. The programs that enable the interactions between

the hardware components are developed using an Arduino IDE and Jupyter Notebook software architecture. The developed system is trained to interpret sign language on a CNN platform. The rest of the paper is organized as follows: Section II presents the methodology used to develop the sign language interpretation system. The results and discussion are presented in Section III, and finally, Section IV represents the conclusion.

II. METHODOLOGY

This study contributes to the development of a sign language interpretation system to enhance communication between the deaf or speech-impaired individuals and others by gathering images for training (pre-test) and system evaluation (post-test), known as data collection. It preprocesses the images and extracts relevant features using Python libraries and software such as TensorFlow and Sklearn. The system employs CNN, a deep learning algorithm, to recognize patterns from the extracted features and train the system using the pre-test images, classifying these recognized patterns into corresponding meanings, displaying text on an LCD, and providing audio output via a speaker. The efficacy of the developed system is evaluated using a dataset reserved for the post-test and in real-life situations. The framework of this work is illustrated in Figure 1.

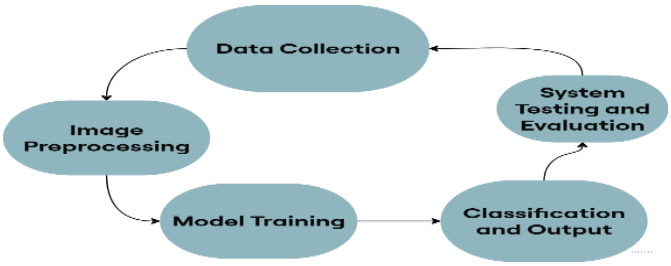

Figure 1: Framework of the sign language interpretation system

The hardware component of the system is developed using a microcontroller system which coordinates the interfacing of other peripherals and the interpretation of a signer's captures to audio and texts, as shown in the block diagram of Figure 2.

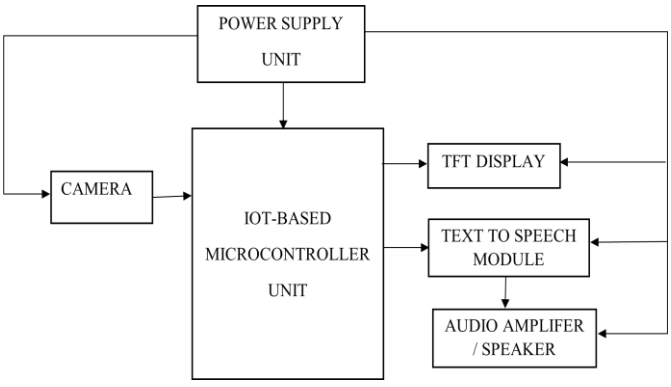

Figure 2: Block Diagram for the Sign Language Interpretation System

A. Hardware Component Description

Figure 2 shows the hardware architecture, which encompasses the study's physical components, including the power supply unit, microcontrollers, camera, display, and speaker, all packed in a plastic adaptable box. The microcontroller adopted is an ESP32 development board because of its Wi-Fi-enabled functionality. The Wi-Fi-enabled functionality is considered for cloud data storage in a Google Sheets platform for research purposes. The camera is an OV2640 of an ESP32-CAM, which is also relatively cost-effective when compared to other HD cameras. The Thin Film Transistor (TFT) display is a 16-bit RGB, 2.0-inch LCD that is selected to display the texts clearly due to its 65K colour depth. The Text-to-Speech module is a DF mini-MP3 player for microcontroller with UART functionality, which is relatively small, a low-cost MP3 module with a simplified output directly to the speaker. The audio amplifier is an LM386 IC, an IC with a low voltage, 200 gain audio power amplifier. Additionally, an LED light is added to increase the brightness of the images captured by the camera.

B. Data Collection

The American Sign Language dataset was easily acquired from the dataset files (Kagle Datasets Team, 2018). Thirty-four thousand six hundred and twenty-eight (34,628) images are from the American Sign Language dataset file, which are collected to train and test the system. These images include the alphabet in American Sign Language, as shown in Figure 3.

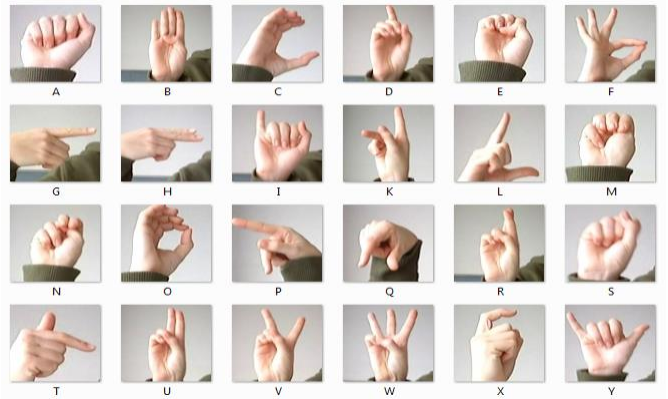

Figure 3: Images of alphabets in American Sign Language (Tecperson, 2018)

C. TensorFlow and CNN

TensorFlow is an open-source software library for machine learning and artificial intelligence (AI). TensorFlow is used in this study for image processing, as shown in Figure 4. It provides a flexible architecture for the CNN to train the pattern recognition system effectively. CNN is a deep-learning network architecture commonly used in computer vision. Computer vision is a field of AI that enables a computer to understand and interpret images or visual data. In this work, a Tensilica Xtensa LX7 dual-core microprocessor embedded in the ESP32 board is trained to interpret the sign language into readable texts and audio.

D. Development Cycle of Pattern Recognition

The pattern recognition system's developmental cycle, as shown in Figure 5, represents the pattern recognition training model. Figure 6 depicts an expanded flowchart of the step-by-step processes performed in the study. First, the images in the dataset were pre-processed after data collection; if the images were clear and sufficiently sized, then feature extraction and selection were done, and the images were used to train the model, subsequently stored in the database with their classified text. Figure 8 presents the flowchart for evaluating the system. When a signer performs any sign language, the camera captures the sign language image, preprocesses it, and checks

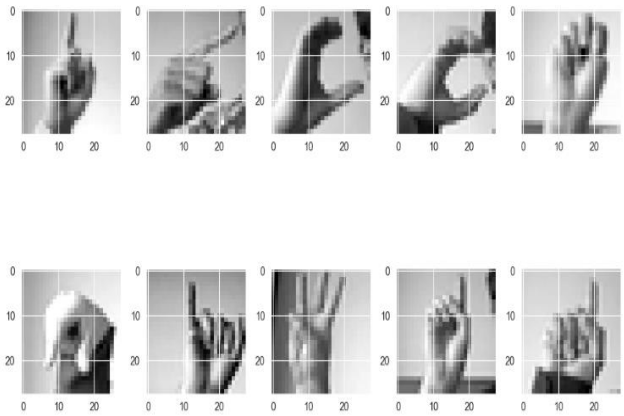

Figure 4: Image resizing

for clarity and proper resizing, and then feature extraction occurs, forming a pattern recognized by the trained model. The database is accessed to verify if the image pattern matches any in the database; if it does, the corresponding text is displayed and output through the speaker.

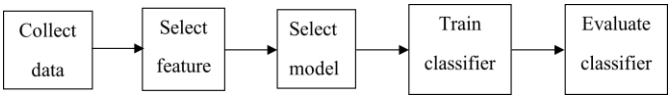

Figure 5: The development cycle of the pattern recognition framework

E. Data Processing and Feature Extraction

Data collected from the American sign language data sheet and the acquisition tool (OV2640 Sensor) were pre-processed. The selected features were extracted with TensorFlow, an open-source software library. Data preprocessing involved cleaning unwanted noise; the images are labelled, rescaled, and resized to 28 x 28 pixels, as shown in Figure 3. Once more, grayscale normalization is used to reduce the effect of variations in illumination and data augmentation to boost the volume of data. Finally, the pre-processed data is exported in a .csv file format. Figure 7 illustrates the graph of count versus label. The graph displays the frequency of each label in the training data. The letters are labelled from 0 to 24, excluding 9 and 25, corresponding to letters J and Z, respectively.

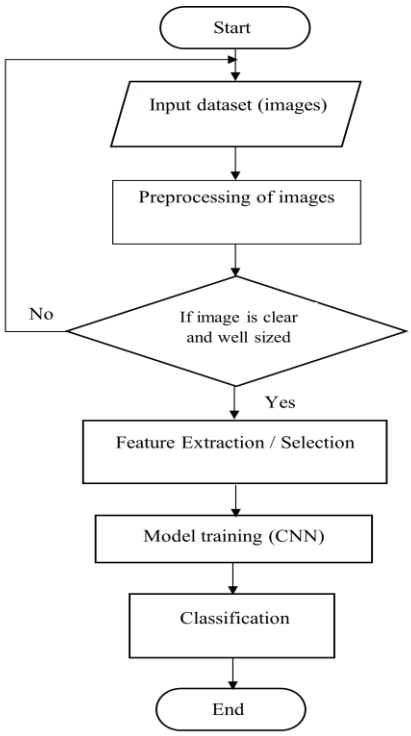

Figure 6: Training process of the pattern recognition system

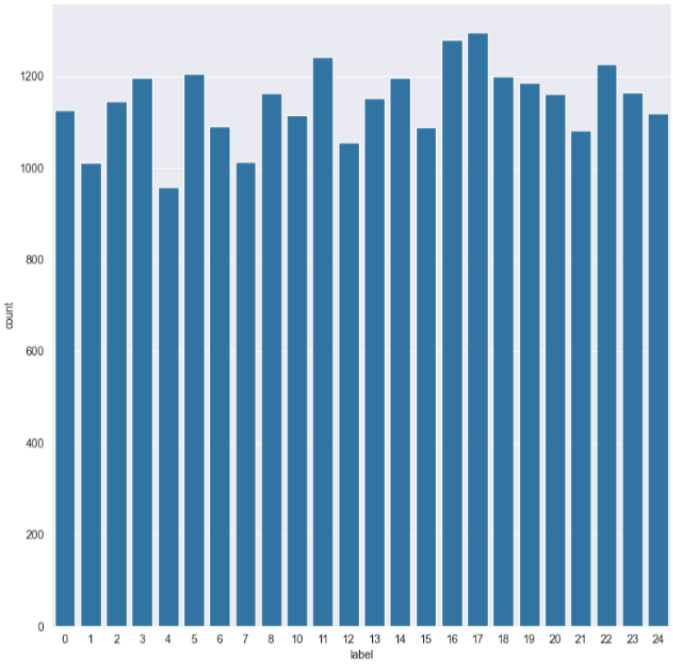

Figure 7: Labelling of alphabets excluding letters J and Z

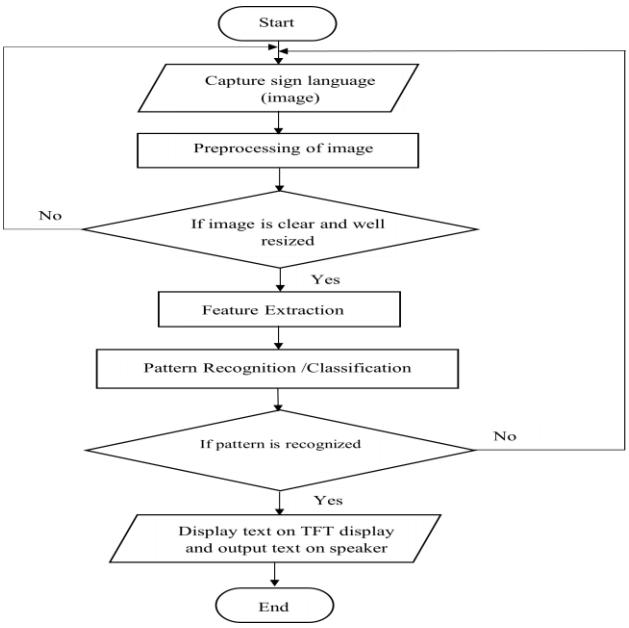

Figure 8: Implementation process of the pattern recognition system

F. *Training and Classification*

Figure 6 shows the training process of the sign language interpretation system. A CNN model is chosen to train the system. A deep learning algorithm, CNN is used primarily in image classification. This neural network algorithm trains the system with the selected features. After training the system, some dataset set aside for testing are used to analyse the system. Table 1 defines a CNN architecture using Keras, designed for image classification. It consists of three convolutional layers (with 65, 40, and 25 filters, respectively), each followed by batch normalization and max-pooling layers. Dropout layers (20% and 30%) are used to prevent overfitting. After flattening the output, a fully connected layer with 128 neurons and ReLU activation is added, followed by a softmax layer for classifying the 24 categories. The model is compiled with the Adam optimizer and categorical cross-entropy loss, and accuracy is tracked during training.

G. *Components Implementation*

The circuit diagram and wiring diagram depicting the connectivity between the various hardware components to enable the development of the pattern recognition system are illustrated in Figure 9 and Figure 10, respectively. The ESP32 CAM captures and preprocesses the image through the implementation process, as shown in Figure 8, subsequently sending the signal to the ESP32 C3 development board, which transmits the signal to the TFT display that shows the meaning of the sign and to the text-to-speech module that vocalizes the text through the speaker.

Table 1: CNN model training hyper-parameter setting

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 65)	650
batch_normalization (BatchNormalization)	(None, 28, 28, 65)	260
max_pooling2d (MaxPooling2D)	(None, 14, 14, 65)	0
Conv2d_1(Conv2D)	(None, 14, 14, 40)	23,440
Dropout (Dropout)	(None, 14, 14, 40)	0
batch_normalization_1 (BatchNormalization)	(None, 14, 14, 40)	160
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 40)	0
conv2d_2 (Conv2D)	(None, 7, 7, 25)	9,025
batch_normalization_2 (BatchNormalization)	(None, 7, 7, 25)	100
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 25)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 128)	51,328
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 24)	3,095

Total params: 88,059 (343.98KB)
Trainable params: 87,799 (342.96KB)
Non-trainable params: 260 (1.02KB)

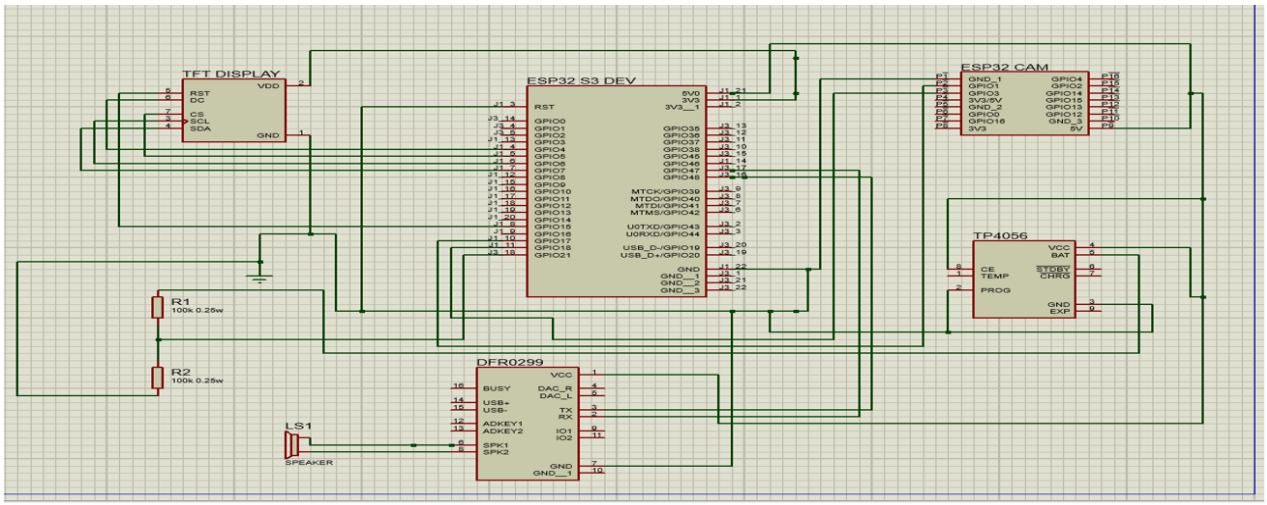

Figure 9: Circuit diagram implementation

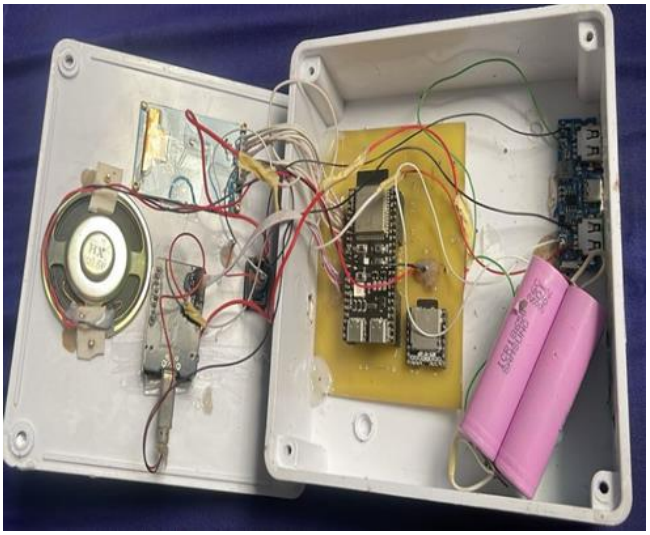

Figure 10: Wiring implementation of the hardware components.

III. RESULTS AND DISCUSSION

Appropriate figures are used to illustrate the findings of this study in this section. The developed prototype was tested with a fraction of the ASL dataset (Kaggle dataset Team, 2018) and validated with data generated from the implementation.

A. Developed Prototype

The developed prototype is shown in Figure 11. A total of 34,628 images from the American Sign Language dataset file were used in this study. Of these, 27,455 (79.29%) images were employed for training the system. In comparison, 7,172 (20.71%) images were used for the analysis (testing) of the trained model, with an accuracy of 99.76% achieved, as illustrated in Figure 12.

Figure 11: Prototype of the developed sign language interpretation system

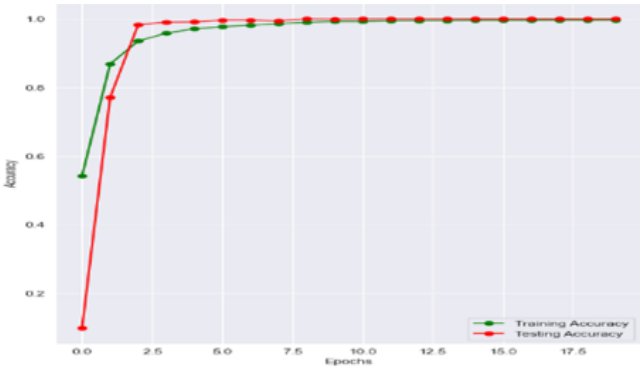

Figure 12: The training and testing accuracy

B. Validation of the Developed System

The system was further evaluated by testing the system with the data using real-time images captured using the camera, as shown in Figure 13. After converting the model into a C file, it was uploaded to the microcontroller using the Arduino IDE. During the validation of the developed system, several user responses were taken, and an accuracy of 50% was attained, as shown in Figure 14b. The low accuracy obtained is found to be a result of the processing power of the ESP32 microcontroller and the resolution of the OV2640 camera sensor.

Figure 13: Pattern recognition system for sign language interpretation in active mode

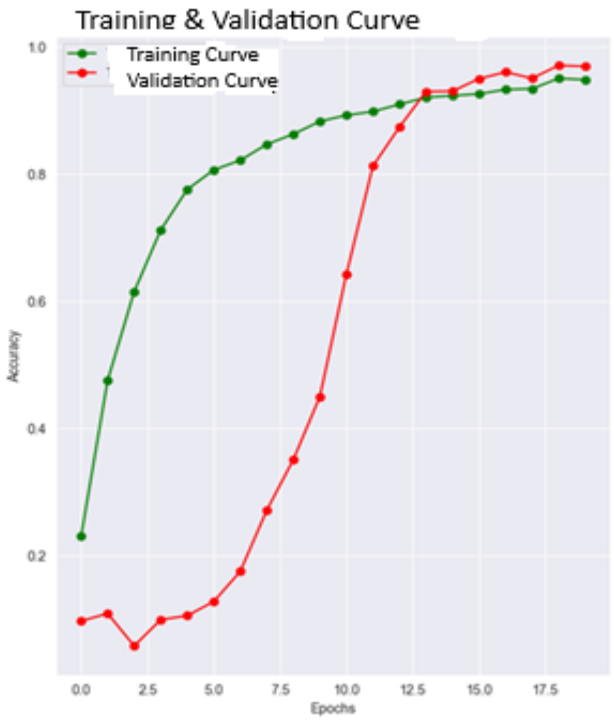

Figure 14a: Training and validation curve of the developed system

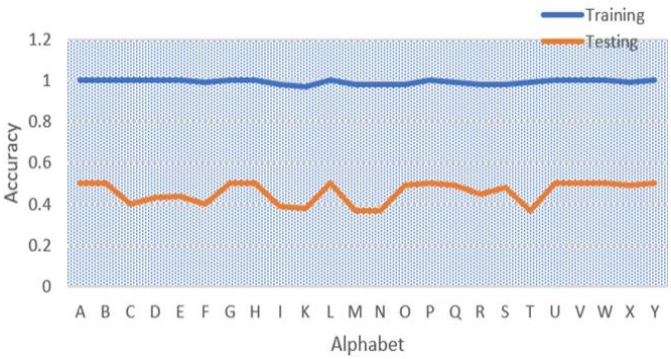

Figure 14b: Training and validation accuracy of the developed system

This project presents a significant improvement in developing a pattern recognition system for sign language interpretation**.** The successful accuracy of 100% achieved in the training and validation process is a pointer to the successful software implementation of the sign language pattern recognition system. However, the system implementation accuracy can be improved using an HD Webcam with high pixel sensor to enhance the quality of the image acquisition of the system. Furthermore, a larger dataset for model training could increase the accuracy of the pattern recognition system, and an IoT-based microcontroller with more memory space and high-speed processor such as the Raspberry Pi, good lighting conditions, and plain background could enhance the performance of the system.

C. System Visualization

The confusion matrix is a matrix that allows the performance of a classification model to be visualized. This makes it easy to see where the specific errors are coming from. The confusion matrix is a performance metrics tool used in machine learning for classification tasks. It helps to summarize the model performance by comparing actual classifications against predicted classifications as illustrated in Figure 15. Figures 15a and 15b are obtained from model testing and the system validation of the developed system, respectively.

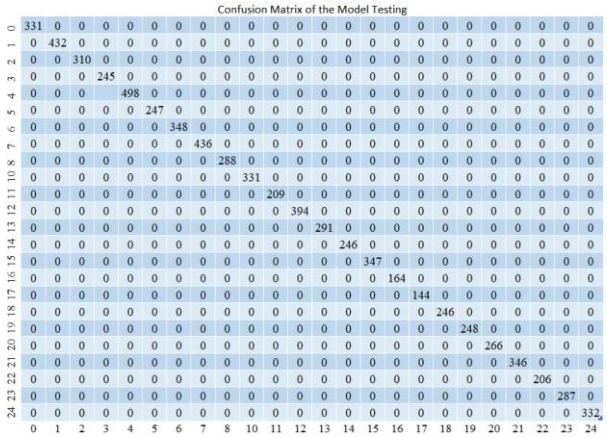

Figure 15a: Confusion matrix of the model testing

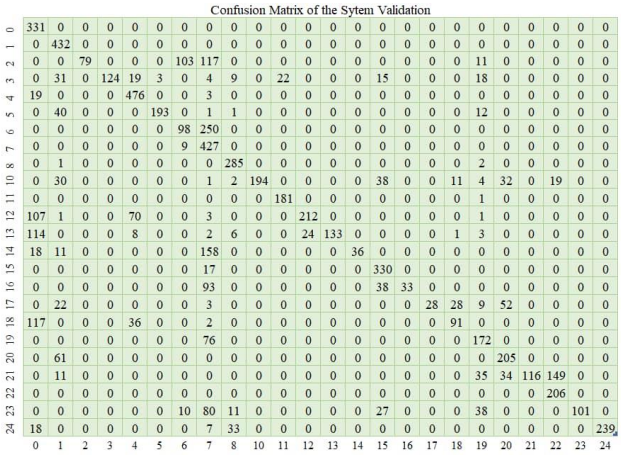

Figure 15b: Confusion matrix of the system validation

IV. CONCLUSION

This project represents a significant leap towards enhancing communication accessibility for the dumb community. By leveraging the CNN algorithm, this study has demonstrated the potential to create more accurate, real-time interpretation tools that aid communication between sign language users and others. The integration of these systems improves the quality of interaction and fosters an understanding of sign language for the hearing-impaired and the public. Moving forward, further research and refinement will be crucial in addressing current limitations and expanding the capabilities of this technology.

AUTHOR CONTRIBUTIONS

H.O. Mahmud and A.O. Ekogiawe: Conceptualization, Methodology, Software, Validation, Writing – original draft. **S.L. Ayinla and A.O. Jimoh-Mahmud:** Writing – review & editing.

REFERENCES

Amosa, T. I., bt Izhar, L. I., Sebastian, P., Ismail, I. B., Ibrahim, O., & Ayinla, S. L. (2023). Clinical Errors from Acronym Use in Electronic Health Record: A Review of NLP-based Disambiguation Techniques. *IEEE Access*, *11*, 59297-59316.

Amosa, T. I., Sebastian, P., Izhar, L. I., Ibrahim, O., Ayinla, L. S., Bahashwan, A. A., Bala, A., & Samaila, Y. A. (2023). Multi-camera multi-object tracking: a review of current trends and future advances. *Neurocomputing*, *552*, 126558.

A**romoye, I. A., Lo, H. H., Sebastian, P., Abro, G. E. M., & Ayinla, S. L. (2025)**. Significant Advancements in UAV Technology for Reliable Oil and Gas Pipeline Monitoring. *Computer Modeling in Engineering & Sciences (CMES)*, *142*(2), 1155-1197.

Halder, A., & Tayade, A. (2021). Real-time vernacular sign language recognition using mediapipe and machine learning. *International Journal of Research Publication and Reviews*, *2*(5), 9-17.

Tecperson(2018). Sign Language Modified National Institute of Standard and Technology (MNIST) [Data set]. retrieved from https://www.kaggle.com/datasets/datamunge/sign-language-mnist

Kanade, V. (2023). *What Is Pattern Recognition? Working, Types, and Applications*. Retrieved from https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-pattern-recognition/

Lawal, A., Cavus, N., Lawan, A. A., & Sani, I. (2024). Hausar kurma: Development and evaluation of interactive mobile app for the English-Hausa sign language alphabet. *IEEE Access*, *12*, 46012-46023.

L**eCun, Y., Bengio, Y., & Hinton, G. (2015).** Deep learning. *Nature*, *521*(7553), 436-444.

Liu, Y., Jiang, X., Yu, X., Ye, H., Ma, C., Wang, W., & Hu, Y. (2023). A wearable system for sign language recognition enabled by a convolutional neural network. *Nano Energy*, *116*, 108767.

Oguntimilehin, A., & Balogun, K. (2024). Real-time sign language fingerspelling recognition using convolutional neural network. *Int. Arab J. Inf. Technol.*, *21*(1), 158-165.

Sharma, S., & Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Systems with Applications*, *182*, 115657.

Smits, T., & Wevers, M. (2022). The agency of computer vision models as optical instruments. *Visual Communication*, *21*(2), 329-349.

Vazquez Lopez, I. (2017). *Hand Gesture Recognition for Sign Language Transcription* [Unpublished Master's thesis], Boise State University.

Wangchuk, K., Riyamongkol, P., & Waranusast, R. (2021). Real-time Bhutanese sign language digits recognition system using convolutional neural network. *ICT Express*, *7*(2), 215-220.

World Health Organization. (2025). *Deafness and hearing loss, Key facts*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss