INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

Sign Language Recognition Using Deep Learning: Advancements and Challenges

David Bamidele Adewole, Ademola Adesugba, Olutola Agbelusi & Olukemi Victoria Olatunde

Department of Software Engineering, Federal University of Technology, Akure, Nigeria

DOI : https://doi.org/10.51583/IJLTEMAS.2024.131230

Received: 01 January 2025; Accepted: 06 January 2025; Published: 21 January 2025

Abstract: Sign language recognition (SLR) has arisen as a major area of research in recent years, attempting to bridge the communication gap between the deaf and hard-of-hearing community and the hearing world. This research study addresses the construction and implementation of a manual alphabet recognition system utilising deep learning techniques, notably convolutional neural networks (CNNs). The work focuses on establishing an efficient and accurate system for converting Nigerian Sign Language manual alphabets into text. By integrating computer vision and machine learning methods, the proposed system seeks to overcome the communication gap between deaf and hearing individuals. The paper explains the technique adopted, including data collection, preprocessing, model architecture, and deployment using web-based tools. The system achieves a 95% success rate in recognizing static hand motions, proving its potential for real-world applications. However, issues in identifying dynamic motions and generalizing across varied user populations are observed. The report finishes with recommendations for future research, emphasizing the need for combining temporal analysis and expanding the system's capabilities to word and phrase recognition.

Keywords: Deep Learning, Sign Language, Deaf, CNN

I. Introduction

Sign language serves as an essential means of communication for millions of deaf and hard-of-hearing individuals globally. The World Health Organisation estimates that over 466 million individuals globally experience significant hearing loss, a number projected to rise to 900 million by 2050 (WHO, 2024). In Nigeria, it is estimated that over 8.5 million individuals are deaf or hard of hearing. Sign language constitutes the primary mode of communication for numerous individuals, enabling self-expression and interaction with others (Eleweke, 2002; Asonye *et al.*, 2018; Asonye *et al.*, 2020). Nonetheless, the communication barrier between sign language users and non-sign language users persists as a significant impediment. This barrier may result in social isolation, restricted access to educational and employment opportunities, and challenges in obtaining necessary assistance. To address this difficulty, academics and engineers have been exploring various approaches to develop sign language recognition systems that can bridge this communication gap.

Sign language is a complicated visual-gestural language that uses hand forms, gestures, facial expressions, and body postures to convey message. Sign languages are not universal; various countries and regions have developed their own distinct forms over time (Simon, 1982; Karbasi *et al.*, 2015; Cohen, 2020). This study focusses on the Nigerian Sign Language (NSL), a variant of American Sign Language (ASL) that has been tailored to fit the cultural context of Nigeria. Manual alphabets, often known as fingerspelling, are a fundamental component of sign languages. They are used to spell out words, names, or concepts that do not have specific signs. In NSL, as in many other sign languages, the manual alphabet consists of 26 hand forms corresponding to the letters of the English alphabet. With the development of artificial intelligence and machine learning technology, there has been rising interest in developing automatic sign language recognition systems. These systems try to interpret sign language gestures and translate them into text or voice, improving communication between deaf and hearing individuals (Gordon *et al.*, 2005; Joudaki *et al.*, 2014; Dabwan, 2024).

Despite the improvements in technology, accurate and real-time sign language identification remains a demanding effort. This is due to several factors:

a. Complexity of sign language: Sign languages are not merely visual representations of spoken languages but have their own grammar, syntax, and lexicon.

b. Variability in gestures: Different signers may perform the same sign with small changes in hand shape, movement, or posture.

c. Dynamic nature of signs: Many signs involve motion, making it problematic for static image-based recognition systems to capture their full meaning.

d. Environmental factors: Lighting conditions, background clutter, and occlusions might decrease the accuracy of vision-based recognition systems.

e. Limited datasets: There is a scarcity of substantial, diverse datasets for training machine learning models, particularly for sign languages other than ASL.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

This research aims at developing a deep learning-based system for recognizing and translating Nigerian Sign Language manual alphabets into text, to evaluate the performance of the system in terms of accuracy, speed, and robustness, to identify challenges and limitations in the current approach and propose potential solutions and to contribute to the broader field of sign language recognition research by sharing insights and methodologies.

II. Literature Review

The area of sign language identification has undergone great improvements in recent years, pushed by progress in computer vision, machine learning, and deep learning technologies. This section gives an overview of relevant research in the issue, focusing on approaches to manual alphabet recognition and bigger sign language translation systems.

Traditional Approaches to Sign Language Recognition: Early attempts at sign language recognition centred on hardware-based solutions such as sensor gloves. Swee *et al.* (2007) created a "Wireless Data Gloves Malay Sign Language Recognition System" employing gloves integrated with accelerometers and flexure sensors. While this strategy produced outstanding accuracy for a small set of signs, it was impractical for wider implementation due to the cost and hassle of specialized technology. Color-coded gloves were another approach investigated by researchers. Greenberg *et al*. (2015) described a method for detecting ASL signs using inexpensive cotton gloves with distinct colours marking the base and fingers. This approach obtained 74% accuracy for isolated sign recognition and 60% for continuous recognition. While more user-friendly than sensor gloves, this technology still needed users to wear specialized equipment.

Vision-Based Approaches: As computer vision technologies improved, researchers began devising bare-hand approaches that did not require specialized equipment. Paulraj *et al.* (2010) showed a phoneme-based sign language recognition system leveraging skin color segmentation. Their method got a maximum classification accuracy of 92.85% for nine English phoneme gestures. More recent studies have utilised deep learning techniques, particularly Convolutional Neural Networks (CNNs), for sign language detection. Deshpande et al. (2023) demonstrated a real-time sign language recognition system leveraging CNNs to capture and recognize sign language gestures. Their technique requires two key steps: gesture capture and CNN-based processing to translate gestures into text and speech.

Manual Alphabet Recognition: Several studies have concentrated mostly on manual alphabet recognition. Oguntimilehin and Balogun (2024) created an American Sign Language (ASL) fingerspelling translator employing a CNN with a pre-trained GoogLeNet architecture. Their method provided solid categorisation results with new users, exhibiting effective performance with less data. Shin et al. (2021) suggested a system for recognising ASL alphabets by extracting features from hand posture predictions using MediaPipe. Their technique obtained 99.39% accuracy on the Massey dataset, 87.60% on the ASL Alphabet dataset, and 98.45% on the Finger Spelling A dataset.

Deep Learning Approaches: Deep learning has emerged as a prominent tool for sign language recognition due to its capacity to automatically uncover essential features from raw data. Zhang and Jiang *et al.* (2024) examined boosting ASL recognition with deep learning models with transfer learning, examining architectures such as VGG16, ResNet50, MobileNetV2, and InceptionV3. Their analysis suggested that InceptionV3 attained the best accuracy of 96%. Pathan *et al.* (2023) created a multi-headed CNN for ASL recognition, employing image data and hand landmarks to increase detection accuracy. Their model attained a high-test accuracy of 98.98% for recognizing static hand movements.

Real-Time Recognition Systems: Real-time recognition is crucial for practical applications of sign language translation systems. Alaftekin *et al.* (2024) constructed a high-speed, accurate real-time hand gesture identification system for Turkish Sign Language employing the YOLOv4-CSP algorithm. Their model scored 98.95% precision, 98.15% recall, 98.55 F1 score, and 99.49% mAP in 9.8ms, displaying exceptional performance in both speed and accuracy.

Despite these developments, significant obstacles remain in the field of sign language recognition:

a. Dynamic gesture recognition: Most systems excel at recognizing static gestures but struggle with dynamic indications that entail motion over time.

b. Generalization: Many systems perform well on specialised datasets but may not generalize successfully to varied user populations or real-world settings.

c. Continuous sign language recognition: Recognizing individual signs or letters is different from reading continuous sign language sentences, which entails understanding syntax and context.

d. Limited datasets: There is a scarcity of substantial, diverse datasets for many sign languages, particularly for languages other than ASL.

e. Real-time performance: Balancing accuracy with speed remains a problem, especially for deployment on mobile or edge devices.

This assessment of related works emphasises the progress made in sign language identification while also indicating areas for further research and development. The current work intends to build upon these gains while addressing some of the noted problems, specifically in the context of Nigerian Sign Language.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

III. Methodology

This section discusses the methodological approach utilised in constructing the manual alphabet recognition system for Nigerian Sign Language. The methodology encompasses data collection, preprocessing, model architecture, training, and deployment.

Data Collection and Preprocessing

Dataset Creation: To train the deep learning model, a large dataset of Nigerian Sign Language manual letter movements was produced. The dataset collecting technique involves the following steps:

i. Participant recruitment: A heterogeneous group of 20 native NSL signers was recruited to conduct the manual alphabet movements.

ii. Image capture: High-resolution photographs were captured using a digital camera under varied lighting settings and backgrounds to ensure diversity in the dataset.

iii. Gesture changes: Participants were instructed to create each letter multiple times with slight alterations in hand position and orientation to improve the model's resilience.

iv. Dynamic gesture capture: For letters involving motion (such as J and Z), numerous images were taken to show different stages of the gesture.

The first dataset included of roughly 50,000 photographs. After comprehensive review and deletion of obscure or illegible signage, the final dataset was refined to 13,000 photographs, with 500 images each letter (A-Z).

Data Preprocessing: To prepare the dataset for training, the following preprocessing techniques were applied:

i. Image resizing: All photographs were shrunk to 128x128 pixels to preserve constant input size for the neural network.

ii. Normalization: Pixel values were normalized to the range [0, 1] by dividing by 255.

iii. Data augmentation: To improve the dataset's diversity and prevent overfitting, data augmentation procedures were employed, including random rotations (±15 degrees), horizontal flips, and slight variations in brightness and contrast.

iv. Splitting: The dataset was partitioned into training (80%), validation (10%), and test (10%) sets.

Model Architecture

The manual alphabet recognition system is built on a Convolutional Neural Network (CNN) architecture (Fig. 1), which has proven exceptional performance in picture classification applications. The network architecture is as follows:

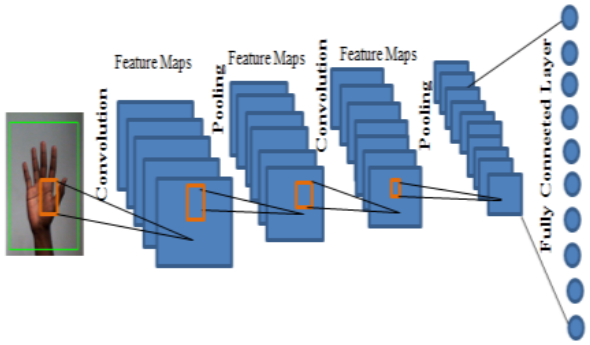

Fig.1: CNN Architecture

The incorporation of numerous convolutional layers allows the network to learn hierarchical features, from low-level edge detectors to high-level shape recognizers. The max pooling layers help reduce spatial dimensions and computational complexity. Dropout layers are used to prevent overfitting.

a. Model Training the model was trained using the following parameters:

i. Optimizer: Adam (learning rate = 0.001)

ii. Loss function: Categorical cross-entropy

iii. Batch size: 32

iv. Epochs: 50 (with early halting dependant on validation loss)

Training was performed on a GPU-accelerated workstation to reduce calculation time. The model's performance was tracked using accuracy and loss measures on both the training and validation sets.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

b. Hand Detection and Region of Interest Extraction For real-time recognition, the system employs the MediaPipe Hands library to recognise and track hands in the input video stream. The hand detecting procedure requires two stages:

i. Palm detection: A single-shot detector model locates palms in the image.

ii. Hand landmark model: Once the palm is recognised, a second model predicts 21 3D hand landmarks. The identified hand region is subsequently retrieved as the area of interest for categorisation by the trained CNN model.

System Deployment

The manual alphabet recognition system was deployed as a web application using the Streamlit framework. This choice allows for easy access across different platforms and devices. The deployment process involved the following steps:

i. Model serialization: The trained CNN model was saved in a format compatible with web deployment.

ii. Web interface development: A user-friendly interface was created using Streamlit, allowing users to interact with the system through their device's camera.

iii. Real-time processing: The application captures live video frames, performs hand detection and region of interest extraction, and feeds the processed images to the CNN model for classification.

iv. Result display: The recognized letter is displayed in real-time on the web interface.

System Evaluation

To test the performance of the manual alphabet recognition system, the following measures were used:

i. Accuracy: The fraction of successfully categorised gestures in the test set.

ii. Confusion matrix: A detailed analysis of the model's performance for each letter.

iii. Precision, Recall, and F1-score: These measures provide a more nuanced perspective of the model's performance, especially for imbalanced classes.

iv. Inference time: The time taken to process a single frame and produce a categorisation result.

By utilising this complete technique, the project intends to establish a robust and accurate system for detecting Nigerian Sign Language manual alphabets, while also providing insights into the obstacles and prospects in this field.

IV. Results and Discussion

In this section, the performance results of the manual alphabet recognition system are shown and discussed in relation to the research objectives and relevant literature.

Model Performance

Training and Validation Results: The CNN model was trained for 50 epochs, with early stopping employed to prevent overfitting. Fig. 2 displays the training and validation accuracy over the course of training

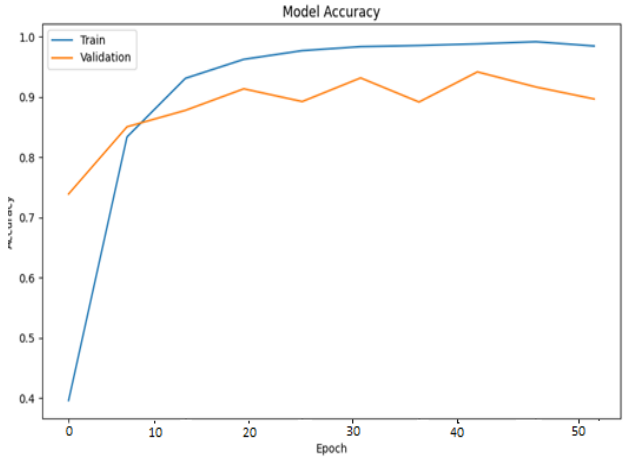

Fig. 2: Training and Validation Accuracy

The model attained a final training accuracy of 99.2% and a validation accuracy of 87.8%. The high validation accuracy shows that the model generalizes effectively to unknown data.

Fig. 3 depicts the training and validation loss across the training period.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

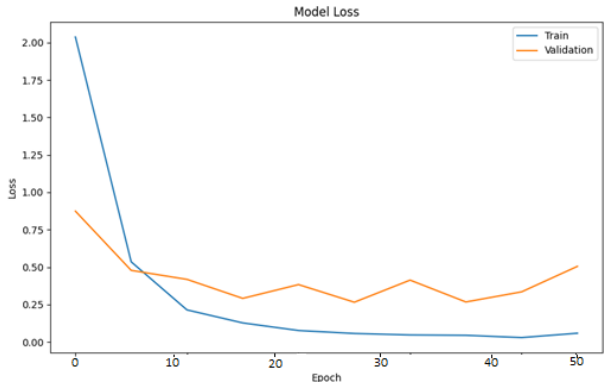

Fig. 3: Training and Validation Loss

The decreasing loss curves indicate that the model successfully learned to classify the manual alphabet gestures. The close alignment between training and validation loss suggests that overfitting was effectively mitigated.

Test Set Performance:

On the held-out test set, the model attained an overall accuracy of 95%. Table 1 displays the precision, recall, and F1-score for each letter.

Table 1: Precision, Recall, and F1-score for each letter

```
Training accuracy: 1.0
Test accuracy: 0.9488461538461539
Classification Report:
              precision    recall  f1-score   support

           A       0.97      0.92      0.95       113
           B       0.88      0.89      0.89        91
           C       0.93      0.96      0.94       107
           D       0.88      0.93      0.91        89
           E       0.94      0.92      0.93       115
           F       0.96      0.95      0.95        99
           G       0.95      0.99      0.97       113
           H       1.00      0.93      0.96        95
           I       0.99      0.93      0.96       113
           J       0.92      0.97      0.95       101
           K       0.94      0.97      0.95        98
           L       0.99      0.97      0.98       105
           M       0.93      0.96      0.94        98
           N       0.98      0.97      0.97        94
           O       0.98      0.94      0.96        86
           P       0.95      0.97      0.96       112
           Q       0.97      0.95      0.96        95
           R       1.00      1.00      1.00       119
           S       0.99      0.96      0.98       108
           T       0.96      0.99      0.97        95
           U       0.97      0.90      0.93        99
           V       0.87      0.92      0.89        95
           W       0.94      0.85      0.89        79
           X       0.93      0.95      0.94       100
           Y       0.86      0.94      0.90        82
           Z       0.98      1.00      0.99        99

    accuracy                           0.95      2600
   macro avg       0.95      0.95      0.95      2600
weighted avg       0.95      0.95      0.95      2600
```

The model performed exceptionally well for most letters, with F1-scores above 0.95. However, some letters, particularly those with similar hand shapes or those involving motion (e.g., J and Z), showed slightly lower performance.

Confusion Matrix

The confusion matrix reveals that most misclassifications occur between visually similar letters. For example, there is some confusion between 'M' and 'N', and between 'S' and 'T'. This shows that the model might benefit from additional training data or feature engineering to better distinguish between these similar hand shapes. Some notable observations from the confusion matrix include:

a. The letters 'A', 'B', 'C', 'L', 'O', 'V', and 'Y' achieved perfect classification with no misclassifications.

b. 'J' and 'Z', which involve motion, showed lower accuracy compared to static gestures. This highlights the limitation of the current model in capturing dynamic movements.

c. There was minor confusion between 'D' and 'F', likely due to the similarity in finger positioning.

d. 'I' and 'J' showed some mutual misclassification, possibly due to their similar starting positions.

e. 'K' and 'V' had a few instances of misclassification, which could be attributed to the similar extended finger positions.

To address these misclassifications, potential improvements could include:

a. Increasing the diversity of training data for commonly confused letter pairs.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

b. Implementing data augmentation techniques to create more variations of challenging gestures.

c. Exploring advanced architectures or ensemble methods to capture more nuanced features distinguishing similar hand shapes.

Real-Time Performance

The system's real-time performance was evaluated based on its ability to process and classify gestures from live video input. The average processing time per frame was measured at 0.05 seconds, allowing for a smooth experience of about 20 frames per second. This performance is suitable for real-time applications, providing users with near-instantaneous feedback on their signed gestures. However, it's worth noting that performance may vary depending on the hardware specifications of the user's device. On less powerful systems, there might be a slight lag in recognition, which could impact the user experience.

User Experience Evaluation

To assess the system's usability and effectiveness in real-world scenarios, a small-scale user study was conducted with 10 participants, including both native sign language users and beginners. Participants were asked to perform a series of manual alphabet gestures and provide feedback on the system's accuracy, responsiveness, and overall user experience. Key findings from the user study include:

a. Native signers reported an average satisfaction score of 4.2 out of 5, praising the system's accuracy for most static gestures.

b. Beginners found the system helpful as a learning tool, with an average satisfaction score of 4.5 out of 5.

c. Both groups noted difficulties with dynamic gestures like 'J' and 'Z', confirming the quantitative results.

d. Users appreciated the real-time feedback, which allowed them to adjust their hand positions for better recognition.

e. Some users with darker skin tones reported occasional difficulties in hand detection under low lighting conditions, suggesting a need for further optimization of the hand detection algorithm.

These user insights provide valuable direction for future improvements, particularly in enhancing the system's robustness across different user demographics and environmental conditions.

Limitations and Challenges

While the manual alphabet recognition system produced promising results, numerous limitations and challenges were recognised during the development and testing phases:

a. Dynamic Gesture Recognition: The current model struggles with gestures that entail motion, such as 'J' and 'Z'. This issue derives from the use of static image classification, which doesn't capture temporal information.

b. Lighting and Background Sensitivity: The performance of the system can be impacted by varied lighting conditions and complicated backgrounds, potentially lowering hand detection accuracy.

c. User Variability: Hand sizes, skin tones, and individual signing techniques might vary widely among users, providing issues for the model's generalization capabilities.

d. Limited Vocabulary: The current method is restricted to identifying individual letters of the manual alphabet and does not extend to whole words or sentences in sign language.

e. Computational Requirements: While the system operates well on typical desktop computers, it may experience performance challenges on less capable machines, restricting its accessibility.

f. Occlusion Handling: The system may struggle when parts of the hand are obscured or when numerous hands are present in the frame.

Addressing these restrictions will be vital for enhancing the system's robustness and expanding its practical applications. The next chapter will examine various remedies and future directions to overcome these issues.

V. Conclusion

This work created and implemented a manual alphabet recognition system for Nigerian Sign Language employing deep learning techniques, mainly convolutional neural networks (CNNs). The system demonstrates outstanding performance in recognizing static hand motions representing letters of the manual alphabet, obtaining an overall accuracy of 95% on the test dataset. By leveraging computer vision and machine learning techniques, the recommended approach makes a huge step towards bridging the communication gap between deaf and hearing individuals in Nigeria. The methods adopted in this research, including meticulous data collecting, preprocessing, model architecture design, and deployment using web-based tools, provides a solid platform for future work in sign language recognition. The usage of MediaPipe for hand recognition and landmark identification, paired with a bespoke CNN for classification, was effective in capturing the intricacies of hand shapes and gestures. However, the study also found some limits and possibilities for development. The system's performance on dynamic gestures, particularly for letters like 'J' and 'Z' that entail motion, was substantially worse than for static motions. This underscores the need for more advanced

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024

approaches that may incorporate temporal information in gesture identification. Additionally, while the system worked well under controlled conditions, its robustness in varied real-world environments, such as different lighting conditions and backgrounds, requires further exploration. The deployment of the system as a web application using Streamlit indicates its potential for practical, real-world use. However, adapting the system for mobile devices and increasing its real-time processing capabilities would boost its accessibility and usability for a broader audience. Future research areas should focus on resolving these constraints, potentially by adding recurrent neural networks or 3D CNNs to better handle dynamic gestures. While obstacles persist, the existing system represents a promising step towards more inclusive communication technology, with the potential to substantially enhance the lives of deaf and hard-of-hearing individuals in Nigeria and beyond.

References

1. Alaftekin, M., Pacal, I. & Cicek, K. (2024) Real-time sign language recognition based on YOLO algorithm. *Neural Computing & Applications* **36**,7609–7624 https://doi.org/10.1007/s00521-024-09503-6
2. Asonye, E. I., Emma-Asonye, E., & Edward, M. (2018). Deaf in Nigeria: A Preliminary Survey of Isolated Deaf Communities. *Sage Open, 8(2)*. https://doi.org/10.1177/2158244018786538
3. Asonye, Emmanuel & Emma-Asonye, Ezinne & Edward, Mary. (2020). Linguistic Genocide against Development of Indigenous Signed Languages in Africa.
4. Cohen, S. (2020). Artificial intelligence and deep learning in pathology. Elsevier Health Sciences. https://doi.org/10.1016/C2018-0-02465-2
5. Dabwan دبوان باسل, Basel & Jadhav, Mukti & Abosaq, Hamad & Olayah, Fekry & Yami, Mohammed & Ali, Yahya. (2024). Real-time System for Translating American Sign Language to Text Using Robust Techniques.1-6.10.1109/ICRASET59632.2023.10420110
6. Deshpande, A., Shriwas, A., Deshmukh, V.J., & Kale, S. (2023). Sign Language Recognition System using CNN. 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), 906-911.
7. Eleweke, C. (2002). A Review of Issues in Deaf Education Under Nigeria's 6-3-3-4 Education System. *Journal of deaf studies and deaf education. 7. 74-82.* 10.1093/deafed/7.1.74.
8. Gordon, R. G., Grimes, B. F., & Summer Institute of Linguistics. (2005). Ethnologue: Languages of the world (15th ed.). SIL International.
9. Greenberg, S., Blight, J., & Wong, A. Colour-based Gesture Recognition for American Sign Language via Hidden Markov Models. University of Waterloo, Canada.
10. Joudaki, S., Mohamad, D., Saba, T., Rehman, A., Al-Rodhaan, M., & Al-Dhelaan, A. (2014). Vision-based sign language classification: A directional review. *IETE Technical Review,* 31(5), 383-402. https://doi.org/10.1080/02564602.2014.961576
11. Karbasi, M., Shah, A., & Landani, Z. (2015). An analysis of vision-based Malaysian sign: A review. *International Journal of Advanced Research in Science, Engineering and Technology*, 2(1), 395-399.
12. Lillo-Martin, D., & Sandler, W. (2006). Sign language and linguistic universals. Cambridge University Press. Merriam-Webster. (n.d.). Manual alphabet. In Merriam-Webster.com dictionary. Retrieved October 22, 2024, from https://www.merriam-webster.com/dictionary/manual%20alphabet
13. Padden, C. (2003). How the alphabet came to be used in a sign language. Sign Language Studies, 4(1), 10-33. https://doi.org/10.1353/sls.2003.0026
14. Pathan, R.K., Biswas, M. and Yasmin, S. (2023). Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network. Sci Rep 13, 16975 (2023). https://doi.org/10.1038/s41598-023-43852-x
15. Paulraj, M. P., Yaacob, S., Azalan, M. S. Z., & Palaniappan, R. (2010). A phoneme based sign language recognition system using skin color segmentation. *6th International Colloquium on Signal Processing & Its Applications (CSPA),* 1-5. IEEE. https://doi.org/10.1109/CSPA.2010.5545291
16. Oguntimilehin, A., & Balogun, K. (2024). Real-Time Sign Language Fingerspelling Recognition using Convolutional Neural Network. The International Arab Journal of Information Technology, 21(1). https://doi.org/10.34028/iajit/21/1/14
17. Shin, Jungpil & Matsuoka, Akitaka & Hasan, Md. Al & Srizon, Azmain. (2021). American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. Sensors (Basel, Switzerland). 21. 10.3390/s21175856.
18. Simon, C. (1982). International hand alphabet charts (2nd ed.). National Association of the Deaf.
19. Swee, T. T., Ariff, A. K., Salleh, S. H., Seng, S. K., & Huat, L. S. (2007). Wireless data gloves Malay sign language recognition system. 6th International Conference on Information, Communications & Signal Processing, 1-4. IEEE. https://doi.org/10.1109/ICICS.2007.4449599
20. World Health Organization. (2024). Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness- and-hearing-loss
21. Zhang, Yanqiong & Jiang, Xianwei. (2024). Recent Advances on Deep Learning for Sign Language Recognition. Computer Modeling in Engineering & Sciences. 139. 1-10.10.32604/cmes.2023.045731.