IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

ANALYSIS OF AUTOMATED SIGN LANGUAGE RECOGNITION USING DEEP LEARNING

Charitha Bhukaya[1], Dr Odugu Srinivasa Rao[2]
[1]M.Tech, CSE Department, UCEK, JNTU Kakinada, Andhra Pradesh, India
[2]Professor, CSE Department, UCEK, JNTU Kakinada, Andhra Pradesh, India

Abstract: Deafness and voice impairment have long been barriers to effective verbal communication, often resulting in the social exclusion of affected individuals from a predominantly speech-based society. While sign language offers a vital means of interaction, its limited understanding among the general population creates a significant communication gap. To address this issue, this paper presents a GUI-based real-time sign language recognition system utilizing a Region-based Convolutional Neural Network (RCNN). The system is designed to recognize hand gestures and convert them into textual outputs at the character, word, and sentence levels, along with real-time speech output for seamless communication. A dataset of 38,120 images representing American Sign Language (ASL) in that (A-Z) alphabets signs was collected, this dataset splits into two that 30,020 images allocated for training and 8,100 for testing. Hand gestures were detected using a webcam and processed through advanced image pre-processing techniques before being passed into the RCNN model. The system achieved a notable test accuracy of 96%, demonstrating its effectiveness. By integrating gesture recognition with an interactive GUI, this solution not only translates signs into readable and audible language but also provides a user-friendly guide for both the hearing and hearing-impaired communities, promoting inclusive communication.

Keywords: Deep Learning, RCNN, American Sign Language, Human-Computer Interaction GUI- based System, Speech Synthesis, Image Processing

I. INTRODUCTION

Communication is the heart of human interaction, enabling individuals to convey thoughts, emotions, and intentions. For people with hearing and speech impairments, sign language serves as a vital means of expression and understanding. However, in a world dominated by verbal communication, sign language users often face challenges in connecting with the broader community. Bridging this communication gap has become a key area of focus within the field of assistive technology.

Recent advancements in deep learning and computer vision have paved the way for innovative solutions that aim to make communication more inclusive. Specifically, vision-based sign language recognition (SLR) systems have emerged as a preferred approach due to their simplicity and ease of use. Unlike earlier sensor-dependent methods that required users to wear gloves or specialized equipment, contemporary systems leverage standard camera input to detect and interpret hand gestures.

The hand gestures used to represent the American Sign Language alphabet from A to Z. These gestures form the foundational dataset for sign language recognition, as shown in the figure1. In this project, we present a real-time sign language recognition system designed to translate American Sign Language (ASL) gestures into text and speech.
This dataset is crucial because it provides a standardized and diverse set of gestures, allowing models to learn consistent visual patterns across different signs. Additionally, ASL is one of the most widely used sign languages globally, making it an ideal choice for developing systems aimed at broad accessibility and real-world impact.

Fig1: American Sign Language (ASL) Alphabets

The system utilizes a Recurrent Convolutional Neural Network (RCNN), combining the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) with the temporal sequence modeling of Recurrent Neural Networks (RNNs). The proposed architecture is capable of recognizing dynamic and static gestures with high accuracy. To enhance usability, the system features a webcam-based graphical user interface (GUI) that captures hand gestures, predicts characters, forms sentences, and converts them into speech. This facilitates smoother communication between sign language users and non-signers. By integrating RCNN for real-time gesture recognition, this research offers a low-cost, efficient, and inclusive solution that advances accessibility in assistive communication technologies.

II. RELATED WORK

Several studies have explored sign language recognition using different machine learning and deep learning methods. Over the years, various machine learning and deep learning approaches have been proposed for sign language recognition, ranging from traditional classifiers to advanced neural network architectures. Durdi et al. [1] and Kulkarni et al. [2] explored CNN-based models incorporating stochastic gradient descent and Inception v3 with PCA to improve classification performance. Pigou et al. [3] implemented dual CNNs to capture both hand and upper body gestures, emphasizing automatic feature extraction. Kalangala and Prabu [4] enhanced recognition through grayscale conversion and brightness normalization. Adeyanju et al. [6] provided a broad analysis of machine learning methods for sign language recognition, highlighting challenges such as signer dependency and data scarcity. Bora et al. [8] utilized MediaPipe with deep learning for regional sign language (Assamese), achieving effective real-time recognition. Patil et al. [7] and Quinn et al. [16] employed CNN and SVM models for high-accuracy real-time gesture classification in varied environments.

More recent advancements have focused on leveraging deep neural architectures like LSTM, ResNet, and 3D CNNs for complex gesture recognition. Montefalcon et al. [14] and Sharma et al. [15] combined ResNet and LSTM to improve temporal gesture modeling, while Sharma and Kumar [18] developed a 3D CNN-based framework for dynamic ASL gesture recognition. Al Moustafa et al. [17] conducted a systematic review of Arabic sign language systems, suggesting future directions in multimodal recognition. Additional innovations include DeepASLR by Kasapbaşi et al. [19], which uses CNNs for American Sign Language, and a gesture-to-voice system proposed by Mounika et al. [20]. Contributions from Luqman [23], Akshatha Rani & Manjanaik [24], and Petkar et al. [25] further reinforce the significance of integrating computer vision with assistive communication systems. These studies collectively underline the evolution of sign language technologies, guiding the design of inclusive, efficient, and user-friendly recognition systems.

III. METHODOLOGY

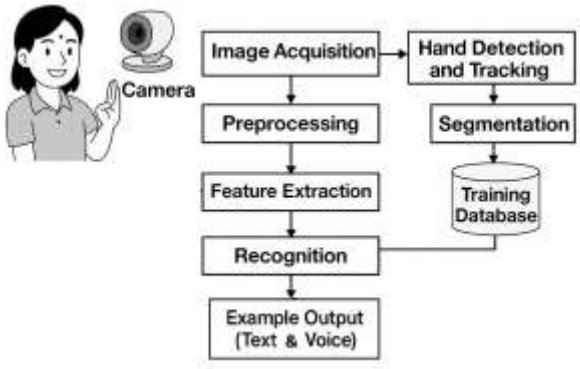

Fig2: Workflow of Sign Language Recognition System

The shown figure2 illustrates the complete methodology used in the sign language recognition system. It includes steps from image acquisition using a camera to hand detection, preprocessing, feature extraction, and recognition, followed by output in text and voice form. This pipeline enables real-time gesture recognition with high precision and fluency.

A. Image Acquisition

The image acquisition process begins with capturing hand gestures using a standard webcam integrated into the system, with OpenCV employed to access and manage the webcam feed. A defined Region of Interest (ROI) is displayed on the screen to guide the user in placing their hand correctly for consistent gesture capture. Continuous image frames are recorded, each labelled according to the displayed ASL gesture. A total of 38,120 images representing all 26 American Sign Language (ASL) alphabet signs (A–Z). The Hand Detector module from the *evzone* library was used to detect and crop the hand region, minimizing background interference. Each image was resized to 300 × 300 pixels using OpenCV and saved in JPG format. The dataset was divided into two main subsets: 80% (30,020 images) for training and 20% (8,100 images) for testing, with a portion of the training data further set aside for validation during model development to fine-tune and evaluate performance as shown in the figure3

Fig3: Bar graph illustrating the distribution of data for each class

B. Hand Detection and Tracking

The hand detection and tracking step ensures the system isolates the hand region from the background. This is done using color segmentation, contour detection, or region proposal methods. The system identifies and marks bounding boxes around the detected hand in each frame. The consistency in tracking ensures reliable gesture inputs for training.

C. Segmentation

Once the hand is detected, segmentation is applied to extract only the relevant hand region while eliminating the background. Techniques such as background subtraction and adaptive thresholding are used. The segmented hand is then resized and cleaned for further processing to enhance gesture clarity.

D. Preprocessing

The segmented images undergo preprocessing operations to normalize the inputs. These include resizing to 224x224 pixels, converting to grayscale, applying histogram equalization, and normalizing pixel intensity values between 0 and 1. Normalization is calculated using the formula:

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$$

Where X is the original pixel value. This formula scales all pixel values to a common range, making the input data consistent and easier to process for the deep learning model.

E. Feature Extraction

The Region-based Convolutional Neural Network (RCNN) is used in this system to extract relevant spatial and temporal features from the segmented hand images. Unlike traditional CNNs, RCNN proposes candidate regions of interest (ROIs) likely to contain the hand gesture, extracts features from these ROIs, and classifies them. The RCNN architecture, as shown in Fig.4, combines convolutional layers for spatial feature extraction with an LSTM layer to capture temporal dependencies. This hybrid approach enables effective sequence modeling of image data for classification tasks.

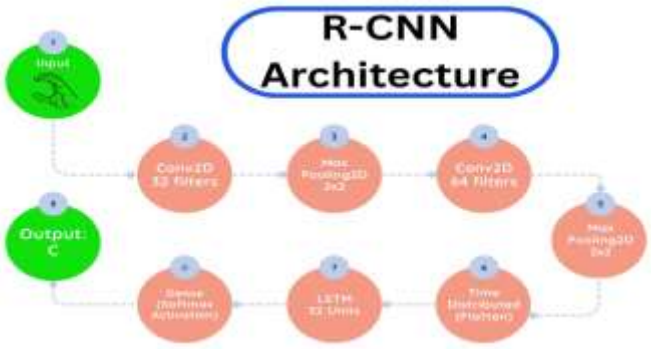

Fig4: R-CNN Architecture

The RCNN architecture includes the following main layers:

1. **Region Proposal Network (RPN)**: This network scans the image with a sliding window and suggests bounding boxes where objects (hand gestures) are likely to be present. It uses predefined anchor boxes and a small classifier to determine the objectness score for each region.

2. **Convolutional Layers**: These layers apply learnable filters to detect edges, patterns, and textures from the proposed regions. The size of the output feature map is determined using the following formula:

$$W_{out} = ((W_{in} - K + 2P) / S) + 1$$

$$H_{out} = ((H_{in} - K + 2P) / S) + 1$$

Where: W_{in} and H_{in} are the input width and height, K is the kernel size, P is the padding applied to maintain size, maintain size, S is the stride or step size

These layers use ReLU activation function defined as: $f(x) = \max(0, x)$

3. **ROI Pooling Layer**: This region of interest pooling layer ensures that all regions of different sizes are resized to a fixed size before being passed to the fully connected layers. It applies max pooling within each ROI, using the formula:

$$O_x = ((I_x - P) / S) + 1$$

Where: I_x is the input dimension (width or height), P is the pooling window size, S is the stride.
This ensures all features are of uniform size for classification.

4. **Fully Connected Layers**: These layers take the flattened ROI features and pass them through dense layers for classification. The output of a fully connected layer is calculated as:

$$\text{Output} = \text{Activation}(W \cdot X + b)$$

Where: W is the weight matrix, X is the input vector, b is the bias term.
This computes the weighted sum and applies an activation function like ReLU.

5. **Softmax Classification Layer**: The final dense layer outputs probabilities for each gesture class using the Softmax function:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

Where: Z_i is the raw score, n is the total number of classes, e is the base of the natural algorithm and the denominator is the sum of the exponentials of all logits. This step ensures that the output values are normalized and sum up to 1. This produces a probability distribution over all 26 classes (A–Z), and the class with the highest probability is chosen as the predicted output.

F. Recognition

The gesture corresponding to the highest softmax probability is selected and labeled accordingly. This approach ensures fast and accurate classification by leveraging the model's confidence scores. It enables seamless real-time prediction, essential for interactive applications.

G. Text and Voice Output

The recognized gesture is displayed on screen and converted to speech using pyttsx3, facilitating two-way communication for hearing-impaired users. The system dynamically forms words or sentences as gestures are recognized in sequence. This multimodal output greatly enhances accessibility and user engagement.

IV. RESULTS AND DISCUSSION

The proposed RCNN-based model was evaluated on a dataset consisting of American Sign Language (ASL) hand gestures representing 26 alphabets. The evaluation includes accuracy and loss trends, confusion matrix, and classification metrics. The following figures illustrate the model's performance along with detailed interpretation.

This paper Sign Language Recognition system, developed using Python and integrated with a user-friendly GUI built using the Tkinter library, demonstrates high performance and accessibility. By utilizing a significantly expanded dataset of 38120 labeled images covering ASL alphabets and digits, the system achieves improved accuracy and robustness. The RCNN-based model trained on this dataset attained a recognition accuracy of 96% The Python GUI enables real-time webcam-based hand gesture capture, instant character prediction, sentence generation, and speech synthesis, making the system practical and efficient for real-world usage, especially for the deaf and hard-of-hearing community.

The training and validation accuracy curves, as shown in Fig5, demonstrate that the model progressively learned the spatial and temporal features of the hand gestures. The training accuracy increased steadily to approximately 96.5%, while validation accuracy peaked around 95.3%, indicating strong generalization performance.
This consistent improvement highlights the model's ability to effectively capture and distinguish between subtle gesture variations. Moreover, the minimal gap between training and validation accuracy suggests limited overfitting.

The corresponding loss curves, illustrated in Fig6, show the training and validation loss steadily declining across epochs, indicating that the model's error rate is reducing over time. The validation loss remains low, confirming the model's generalization capability. Additionally, the smooth convergence of both curves reflects stable training dynamics and effective optimization throughout the learning process.

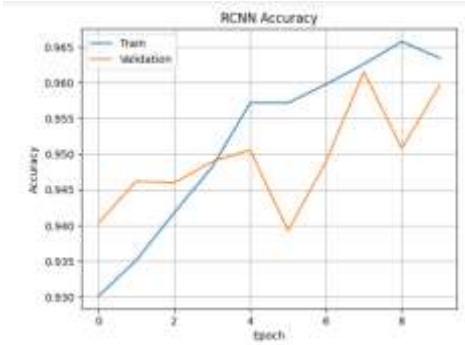

Fig5: Training and Validation Accuracy

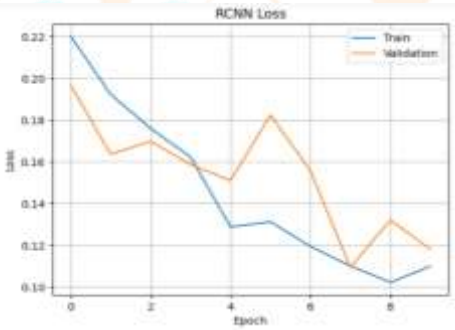

Fig6: Training and Validation Loss

A detailed evaluation of classification performance is provided by the confusion matrix shown in Fig7. The matrix indicates that the majority of classes were correctly predicted, as evidenced by the strong diagonal dominance. Only a few misclassifications occur, mostly in visually similar gestures. These patterns suggest that while the model is highly accurate, further enhancement in feature discrimination for similar gestures could improve precision.

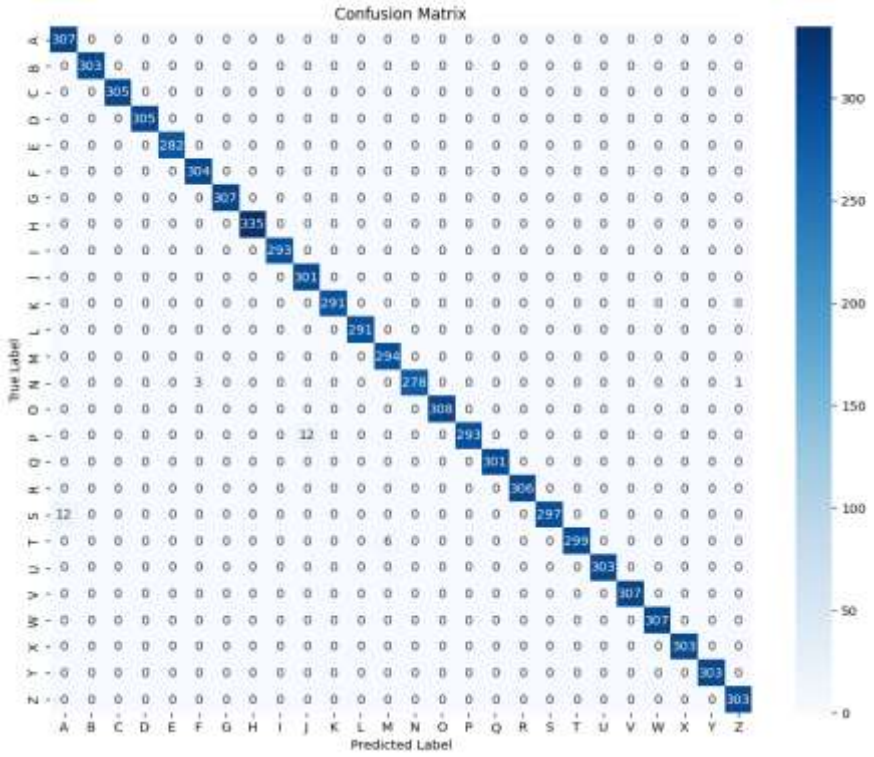

Fig 7: Confusion Matrix

The classification report presented in Table1 outlines the precision, recall, and F1-score for each class corresponding to the English alphabet (A–Z), commonly used in American Sign Language (ASL) recognition tasks. Each metric provides insight into the model's performance at a per-class level:

- **Precision** refers to the proportion of correct predictions for a given class out of all instances predicted as that class.
- **Recall** indicates the proportion of actual instances of a class that were correctly identified.
- **F1-Score** is the harmonic mean of precision and recall, offering a single measure that balances both metrics.

The evaluation metrics used to assess the model's performance include:

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

Where:
TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives
These metrics provide comprehensive insight into the model's classification performance.

Table1: Precision, Recall, and F1-Score

Class	Precision	Recall	F1-Score
A	0.96	1.00	0.98
B	1.00	1.00	1.00
C	1.00	1.00	1.00
D	1.00	1.00	1.00
E	1.00	1.00	1.00
F	0.99	1.00	1.00
G	1.00	1.00	1.00
H	1.00	1.00	1.00
I	1.00	1.00	1.00
J	0.96	1.00	0.98
K	1.00	1.00	1.00
L	1.00	1.00	1.00
M	0.98	1.00	0.99
N	1.00	0.99	0.99
O	1.00	1.00	1.00
P	1.00	0.96	0.98
Q	1.00	1.00	1.00
R	1.00	1.00	1.00
S	1.00	0.96	0.98
T	1.00	0.98	0.99
U	1.00	1.00	1.00
V	1.00	1.00	1.00
W	1.00	1.00	1.00
X	1.00	1.00	1.00
Y	1.00	1.00	1.00
Z	1.00	1.00	1.00

The model was trained for 10 epochs using cross-entropy loss and evaluated using accuracy, loss, a confusion matrix, and per-class precision, recall, and F1-score. This visualization reinforces the reliability of the model in maintaining high performance across the entire dataset.

As shown in Fig 8 that displays the precision, recall, and F1-score values for each class. The scores for most classes are close to 1.0, demonstrating the model's effectiveness in correctly identifying gestures with high reliability. Demonstrating reliable classification of hand gestures across 26 alphabets.

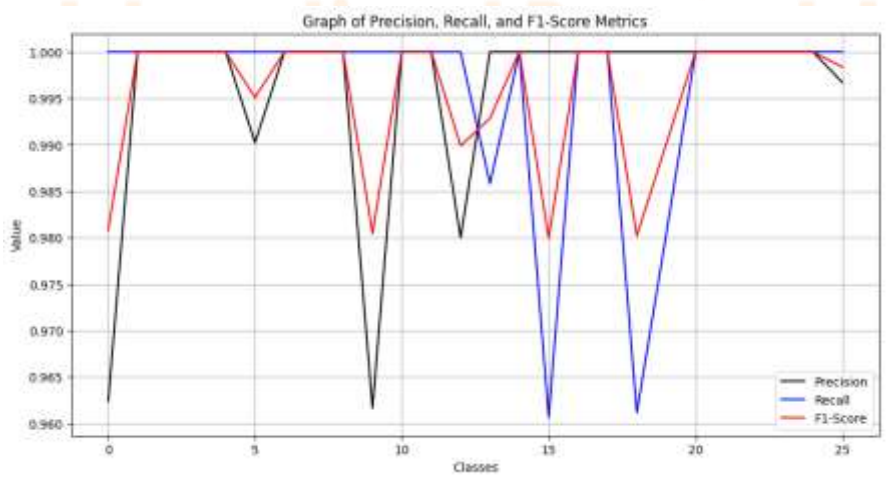

Figure 8: Line Graph of Precision, Recall, and F1-Score Metrics

Consistently high evaluation metrics and low validation loss highlight its robustness and make it well-suited for real-time, practical use in communication support systems.

V. CONCLUSION

This paper presents a real-time sign language recognition system utilizing a Region-based Convolutional Neural Network (RCNN) to identify ASL hand gestures for alphabets and digits. The system achieves high accuracy through a multi-stage pipeline that includes gesture detection, preprocessing, RCNN-based feature extraction, and output conversion to text and speech. Experimental results show a validation accuracy of approximately 96%, supported by strong precision, recall, and F1-scores across all classes. The proposed

solution serves as a practical and low-cost assistive tool for enhancing communication for individuals with hearing and speech impairments. By integrating deep learning with computer vision and speech synthesis, this system has the potential to foster more inclusive human-computer interaction. Future improvements may focus on recognizing dynamic gestures and supporting additional sign languages to broaden its real-world applicability.

Future improvements may focus on recognizing dynamic gestures, supporting additional sign languages, and integrating multilingual translation capabilities. By translating recognized signs into multiple spoken or written languages, the system can further enhance accessibility and bridge communication gaps across linguistic boundaries, making it even more effective in diverse, multilingual environments.

VI. REFERENCE

[1] Durdi, V. B., Kiran, A., Rao, A., Bhat, S. S., Santhosh, B., & Kumar, M.N. (2024). A novel method for recognizing hand gestured sign language using the stochastic gradient descent algorithm and convolutional neural network techniques. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 529–538.

[2] Kulkarni, A., Kulkarni, A., & Madhavan, P. (2024). Sign language prediction using deep learning. International Research Journal on Advanced Engineering and Management, 2(6), 1853-1859.

[3] Pigou, L., Dieleman, S., Kindermans, P., & Schrauwen, B. (2014). Sign Language Recognition using Convolutional Neural Networks. Ghent University, ELIS, Belgium.

[4] Kallingale, A., & Prabu, P. (2021). ML Based Sign Language Recognition System. In 2021 International Conference on Innovative Trends in Information Technology (ICITIIT) (pp. 1–6). IEEE.

[5] Matykiewicz, P., & Pestian, J. (2012). Effect of small sample size on text categorization with support vector machines. In Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012) (pp. 193–201). Montreal, Canada: Association for Computational Linguistics.

[6] Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications, 12, 200056.

[7] Athania, A., Gupta, K. S., Khan, K., & Patil, A. E. (2021). Recognition of Sign Language in Real Time. International Journal for Research in Engineering Application & Management (IJREAM), 7(1), 16-19

[8] Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time Assamese Sign Language recognition using MediaPipe and deep learning. Procedia Computer Science, 218, 1384–1393.

[9] Ansari, Z. A., & Harit, G. (2016). Nearest neighbour classification of Indian sign language gestures using Kinect camera. Sādhanā, 41(2), 161–182. Indian Academy of Sciences.

[10] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition (CVPR), 1, 886-893.

[11] Nagarajan, S., & Subashini, T. S. (2013). Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi-class SVM. International Journal of Computer Applications, 82(4), 28-34.

[12] Tolentino, L. K. S., Serfa Juan, R. O., Thio-ac, A. C., Pamahoy, M. A. B., Forteza, J. R. R., & Garcia, X. J. O. (2019). Static sign language recognition using deep learning. International Journal of Machine Learning and Computing, 9(6), 821–827.

[13] Chitra, S., Kokila, R., Suvithra, J., Krishnaveni, K., & Nandhini, A. (2022). Survey on static sign language recognition using deep learning. International Journal of Research Publication and Reviews, 3(5), 2339-2344.

[14] Montefalcon, M. D. L., Padilla, J. R., & Rodriguez, R. (2023). Filipino Sign Language recognition using long short-term memory and residual network architecture. In Proceedings of the 2023 Philippine National Computing Conference (pp. 45-55). National University, Manila, Philippines.

[15] Sharma, S., Kumar, K., & Singh, N. (2020). Deep Eigen Space based ASL recognition system. IETE Journal of Research.

[16] Quinn, M., & Olszewska, J. I. (2019). British Sign Language Recognition in the Wild Based on Multi-Class SVM. Proceedings of the Federated Conference on Computer Science and Information Systems, 18, 81-86.

[17] Al Moustafa, A. M. J., Rahim, M. S. M., Khattab, M. M., & Zeki, A. M. (2024). Arabic Sign Language Recognition Systems: A Systematic Review. Indian Journal of Computer Science and Engineering (IJCSE), 15(1), 1–16.

[18] Sharma, S., & Kumar, K. (2021). ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. Multimedia Tools and Applications, 80(21), 26319–26331.

[19] Kasapbaşi, A., Elbushra, A. E. A., Al-Hardanee, O., & Yilmaz, A. (2022). DeepASLR: A CNN-based human-computer interface for American Sign Language recognition for hearing-impaired individuals. Computer Methods and Programs in Biomedicine Update, 2, 100048.

[20] Mounika, K. K. S., Hemalatha, J. S., Sri, G. D., & Ram, M. K. (2023). Audio- To-Sign Conversion and Hand Gesture Recognition with An Air Board for Deaf and Dumb Using Deep Learning. International Journal of Research Publication and Reviews, 4(4), 1542-1550.

[21] Lee, L. (2022). The importance of learning Deaf culture through a BlackDeaf perspective in the field of communication sciences and disorders (Honors Thesis, Georgia Southern University). Digital Commons@Georgia Southern.

[22] Thakur, A., Budhathoki, P., Upreti, S., Shrestha, S., & Shakya, S. (2020). Real- time sign language recognition and speech generation. Journal of Innovative Image Processing (JIIP), 2(2), 65-76.

[23] Luqman, H. (2022). An efficient two-stream network for isolated sign language recognition using accumulative video motion. IEEE Access, 10, 93785-93798.

[24] Akshatha Rani K., & Manjanaik, N. (2021). Sign language to text-speech translator using machine learning. International Journal of Emerging Trends in Engineering Research, 9(7), 912–916.

[25] Petkar, T., Patil, T., Wadhankar, A., Chandore, V., Umate, V., & Hingnekar, D. (2022). Real-time sign language recognition system for hearing and speech impaired people. International Journal for Research in Applied Science & Engineering Technology, 10(IV), 2261-2267.

[26] Kuznetsova, A., Leal-Taixé, L., & Rosenhahn, B. (2013). Real-time sign language recognition using a consumer depth camera. 2013 IEEE International Conference on Computer Vision Workshops, 83-90. IEEE.

[27] Kakde, M. U., Nakrani, M. G., & Rawate, A. M. (2016). A review paper on sign language recognition system for deaf and dumb people using image processing. International Journal of Engineering Research & Technology, 5(3), 590-592.