Enhanced Sign Language Detection with Deep CNN: Achieving Accuracy in Hand Gesture Recognition

Abdul Rehman
Department of Computer Science
University of Lahore
Sargodha, Pakistan
m.abdulrehman.shoukat@gmail.com

Muhammad Zaman
Department of Computer Science
COMSATS University Islamabad
Islamabad, Pakistan
mzamancui@gmail.com

Tanzila Kehkashan
Faculty of Computing
Universiti Teknologi Malaysia
Johor Bahru 81310, Malaysia
tanzila.kehkashan@gmail.com

Faheem Akbar
Department of Computer Science
University of Lahore
Sargodha, Pakistan
faheemgujjar658@gmail.com

Muhammad Hamza
Department of Computer Science
University of Lahore
Sargodha, Pakistan
muhamadhamza0301@gmail.com

Raja Adil Riaz
Department of Computer Science
University of Lahore
Sargodha, Pakistan
rajaadilx1@gmail.com

2024 5th International Conference on Innovative Computing (ICIC) | 979-8-3315-1929-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICIC63915.2024.11116573

***Abstract*— The Sign Language Detection (SLD) technology plays a critical role in dealing with communication challenges faced by individuals with hearing disabilities. While much of the existing research has concentrated on gesture recognition using deep learning techniques, difficulties persist in achieving consistent accuracy across intricate, multi-class datasets. This study introduces an innovative deep convolutional neural network (DCNN v2), designed to accurately classify 26 distinct hand signs from a diverse dataset of sign language images. Through use of advanced data augmentation and preprocessing methods, our model enhances classification performance while minimizing the risk of overfitting. The DCNN v2 achieved an exceptional 100% accuracy, consistently delivering high precision, recall, and F1 scores across all sign categories. The perfect classification and absence of any misclassification instances underscore the robustness and effectiveness of the model. The results emphasize the impact of this model in advancing sign language detection, greatly improving accessibility for individuals with hearing impairments and offering more accurate and dependable tools for inclusive communication solutions.**

Keywords—Deep Convolutional Neural Network, Deep Learning, Gesture Recognition, Hand Sign Language Detection, Image Classification

I. Introduction

Sign language plays a crucial role as a mean of communication for millions of deaf and hard-of-hearing people [1]. It enables them to actively participate and contribute to society. The advancement of technology in recognizing and translating sign language into text or speech is crucial for reducing the communication gap between deaf and hearing individuals, promoting inclusivity in areas such as education, healthcare, and everyday life. The need for efficient sign language recognition (SLR) technology is especially recognizable in those areas where communication support services are limited. In many developing nations, the shortage of skilled transcribers and the lack of educational tools for the deaf population highlight the massive value of automated SLR systems [2]. Such systems have the potential to close communication gaps in areas where human assets are not enough. Instead of significant progress in the field of SLR, present systems continue to face challenges. Historically, many approaches have relied on static image processing techniques, which often struggle to adapt to real-world scenarios [3]. For example, traditional image classification methods, like Otsu's thresholding, often fail to accurately differentiate sign language gestures from complex or dynamic backgrounds [4].

Additionally, many research efforts have focused on isolated and less diverse datasets, limiting the generalizability of their models. One study launched a system for feature extraction and classification using ORB and CNN but did not fully address the challenges caused by diverse sign language gestures and environmental unpredictability [5]. While this approach has capability, improvements are needed to enhance the model's ability to handle real-world complexities. Similarly, another study used Mediapipe for hand pose estimation in American Sign Language recognition, achieving high accuracy within particular datasets [6]. Although this approach is economical and doesn't require specialized sensors, it struggles with the variability in hand gestures and background complexity, likely affecting performance in dynamic environments.

Many existing solutions also face difficulties in deploying SLR systems on mobile platforms. A project by developing SUGO solved this, a real-time sign language translation system using depth cameras. While SUGO delivered good results, its performance in different real-world conditions, such as varying lighting and user movements, still needs improvement [7]. Ongoing Sign Language Detection (SLD) systems often face difficulties in accurately classifying hand signs due to variability in gesture recognition across different environments and user conditions. This research aims to address these issues by introducing a novel DCNN v2 model that improves gesture recognition accuracy while reducing computational delays, ultimately improving the effectiveness of sign language translation technologies.

The objectives of the proposed study are as follows:

i. To propose a novel architecture DCNN v2 especially developed for hand sign recognition, offering perfect accuracy over existing models.

ii. To implement techniques for preprocessing that enhance the model's robustness in handling real-world image variations.

iii. To perform a detailed analysis of a model's performance across different evaluation metrics.

This study looks to make a meaningful contribution to society by enhancing accessibility and comprehensive in communication for individuals with hearing and speech impairments. By developing the DCNN v2, the project aims to enable accurate and efficient hand sign classification in various environments, providing real-time communication for users in social, educational, and professional contexts. Optimizing image preprocessing techniques, such as maximizing brightness, contrast and others [8], will further enhance gesture recognition accuracy, ensuring reliability even in challenging conditions.

This research paper is organized into five key sections. The Introduction outlines the background and significance of SLD, as well as the rationale for using deep convolutional neural networks (DCNN). Literature Review examines previous work in the domain of gesture recognition and SLD detection methods, identifying gaps that this research addresses. The Methodology details the model's architecture, preprocessing techniques, and experimental setup. The Results and Discussion section analyzes the performance of proposed novel DCNN v2 model using various metrics. Finally, the Conclusion summarizes the findings, contributions, and potential avenues for future research.

II. Literature Review

Significant advancements have been made in SLD systems, several challenges persist. This section provides a critical analysis of key studies in this discipline, focusing on their procedures, participation, and limitations.

A. Early Approaches using Machine Learning Techniques

A system that uses video technology on mobile platforms to translate sign language into text, a study developed SUGO [9]. SUGO addresses privacy concerns associated with RGB cameras and attempts to achieve real-time translation by using depth cameras. System uses a 3D Convolutional Neural Network (3DCNN) for classifying video frame sequences and shows great accuracy of up to 91% [10], [11]. SUGO operates without the need for external sensors or cloud support, designed for mobile platforms. However, its dependency on depth cameras poses challenges, such as lower video resolution and increased sensitivity to user movements [12], [13]. Moreover, further validation is required to ensure the system's robustness under varying real-world conditions, such as different lighting and sign language variations [14].

B. Transition to Deep Learning Models

For sign language classification, a further study compared two models: one using ORB (Oriented FAST and Rotated BRIEF) [15], [16] for feature extraction and classification with SVM, and the other manipulating the CNN architecture [17], [18]. This study aimed to evaluate the effectiveness of both methods in SLR. While the ORB and SVM approach serves as a foundation for feature extraction, it struggles with variability in hand gestures and backgrounds. The CNN model shows improved performance but requires substantial computational power and fails to fully address the challenges posed by dynamic sign gestures and changing environmental conditions [19].

C. Advances in Computer Vision and Hybrid Models

Another study [20] introduced a bio-inspired model combining an attention-based Inception CNN with Bi-LSTM [21] for Arabic SLR. This approach integrates spatial feature extraction through the Inception CNN and temporal features using Bi-LSTM. The model demonstrated strong early detection performance, achieving results comparable to offline models. However, its focus on Arabic sign language limits the exploration of the model's adaptability to other languages. Additionally, the model's performance under real-world conditions—such as different lighting and background complexities—remains underexplored [22].

Another approach [23] applied the Mediapipe hands algorithm for hand pose estimation in SLD. This method derived features from hand joint estimations, using distances and angles to classify hand gestures via SVM and Gradient Boosting Machine (GBM). The method demonstrated high accuracy on datasets like the SLD Alphabet, Massey, and Finger Spelling. While cost-effective and computationally efficient, the approach's reliance on hand pose estimation presents challenges in managing variations in hand gestures and background clutter [24].

A study [25] focusing on Indian Sign Language (ISL) proposed a hybrid approach that combined Features from Fast Accelerated Segment Test (FAST), Scale-Invariant Feature Transformation (SIFT), and CNN [16]. This method achieved high accuracy on datasets like ISL alphabets, MNIST, Jochen Trisech Dataset (JTD), and NUS-II dataset. Despite its effectiveness in feature extraction and recognition performance, the hybrid method faces challenges with computational complexity and the requirement for large-scale training data. [26].

III. Methodology

In this study, the objective is to establish a model for predicting Sign Languages using a dataset that includes different preprocessing techniques and a novel DCNN v2 for perfect accuracy. The methodology includes comparing a baseline CNN with the proposed novel DCNN v2 model, which incorporates specific improvements for better performance.

Baseline Method

The baseline paper [9] serves as a fundamental reference for our research. Its methodology consists of several critical steps aimed at enhancing the validity of sign word detection.

Initially, the dataset is meticulously prepared, dividing it into training and testing sets, with both sets featuring rotated, translated, and scaled (RTS) versions to improve model performance in varying conditions. A hybrid segmentation process is employed, incorporating Otsu Thresholding combined with YCbCr skin color segmentation to effectively separate the foreground from the background, followed by morphological analysis for enhancing the segmented images. The watershed algorithm is utilized to delineate gesture boundaries, optimizing the recognition of sign language gestures. For classification, a novel CNN architecture is applied, featuring 3 CNN layers with ReLU activation and max pooling layers, ultimately condensing feature vectors into a dense layer for classification across 20 sign classes. The model employs the Adam optimizer to enhance prediction accuracy. Our proposed methodology shares connections with this baseline method, particularly in the use of CNNs and advanced preprocessing techniques, providing a solid reference point for our research.

Proposed Model

The proposed model enhances the baseline by incorporating a deep CNN architecture, specifically designed to improve hand sign recognition. This model focuses on extracting key features from input images, using multiple convolutional layers to capture intricate gesture details, ensuring robust categorization across many conditions. Following are steps that are followed while development of our proposed model:

A. Data Collection

Dataset used in this research is the American Sign Language (ASL) dataset[1] which is taken from Kaggle. It includes a total of 8066 images, where 5646 are used for training and 1210 for validation purposes, covering 26 classes containing the ASL alphabet. Images are 200x200 pixels in resolution and in RGB format. It gives a different representation of hand signs, which is necessary for training the model to recognize the different hand signs effectively, thereby increasing model's ability to generalize in real-world scenarios. It includes images for all 26 letters of the ASL alphabet, with 300 images per letter. It provides a complete resource for building systems that can perfectly detect ASL hand signs.

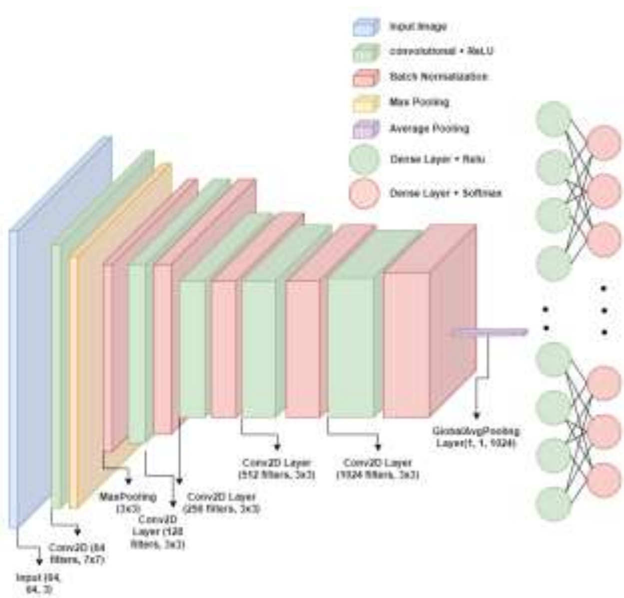

Fig. 1. Proposed model

B. Preprocessing

1) *Image Reshaping:* The input images are reshaped from 200x200 to 64x64 pixels to ensure balance across the dataset.

2) *Normalization:* Values of each Pixel are normalized between a range of 0 and 1 to standardize the input data.

3) *Data Augmentation:* Various techniques such as brightness, sharpness, zoom etc are used to increase diversity of dataset and improve the model's generalization.

C. Novel Deep CNN v2 Model Architecture

The proposed model developed a novel DCNN v2 architecture to increase the classification of language signs. It is developed to improve feature extraction and classification performance through special architectural choices and training methods. The steps that indicates the proposed methodology are as follows:

The model begins with some convolutional(Conv2D) layers with kernels (filters) of sizes 64, 128, 256, 512 and 1024. These layers capture essential spatial features from the input images such as given in eq. 1. Every convolutional layer is followed by batch normalization and ReLU activation to enhance learning stability and pattern recognition. After these layers there is Global Average Pooling (GAP) Layer which is further connected to dense layers with Relu and softmax activation respectively as shown in Figure 1. The novel DCNN v2 model is designed with multiple convolutional stages, followed by batch normalization and ReLU activation. This architecture is designed to handle complex feature extraction and improve the model's capacity to recognize different language signs. The convolution operation for specific location (i, j, k) in output feature map is given by the equation 1:

$$y_{i,j,k} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n} w_{m,n,k} + b_k \tag{1}$$

This equation represents sliding a convolutional filter over input image then performing element-wise multiplication and summation, finally we add a bias term to get output feature map. In deep learning models, activation functions are necessary for introducing non-linearity. Rectified Linear Unit (ReLU) is the most used activation functions. ReLU can be expressed mathematically in equation 2:

$$f(x) = \max(0, x) \tag{2}$$

When the input x is positive value, it is allowed to pass through unchanged due to the ReLU function, this retains information. When input x is less than or equal to zero, ReLU responds with zero and effectively deactivates the neuron. Batch normalization is a technique used to balance and accelerate deep neural networks' training by normalizing inputs of each layer. The batch normalization formula is defined as followed in equation 3:

[1]https://www.kaggle.com/datasets/konami9889/americanhand-sign-detection

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (3)$$

This normalization process helps in reducing internal covariate shift, leading to faster convergence during training. Max pooling is a down sampling technique that reduces dimensionality of feature maps while retaining the most salient information. The max pooling operation is defined in equation 4:

$$y_{i,j,k} = \max_{m,n}\left(x_{i+m,j+n,k}\right) \quad (4)$$

Max pooling helps in extracting dominant features and reduces the spatial dimensions, thus lowering computational complexity. In fully connected layer, each neuron is connected to every neuron in the previous layer. The output of the fully connected layer is computed as followed in equation 5:

$$z_i = \sum_j w_{ij} x_j + b_i \quad (5)$$

This layer is responsible for combining the learned features to produce the final output. The softmax function is used in the output layer of the neural network to convert the logits into probabilities. This function is defined in equation 6:

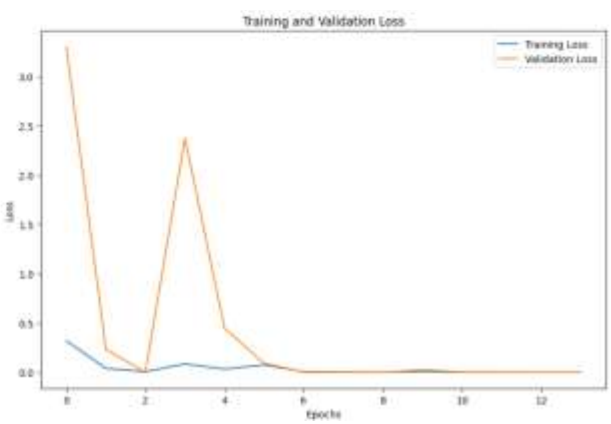

Fig. 2. Training and validation loss over epochs

Fig. 3. Training and validation accuracy over epochs

$$P(y = j \mid x) = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} \quad (6)$$

This function makes sure that the output probabilities sum up to one, allowing for probabilistic interpretation of the model's predictions. After applying the convolutional layers, feature maps are reduced by utilizing global average pooling. The dense layer includes dropout regularization to prevent overfitting.

The output dense layer utilizes a softmax activation function to classify images into one of the 26 hand language signs. GAP is a downsampling technique used in CNN to lessen the dimensions of feature maps. GAP operation computes average of each feature map and is defined by the following equation 7:

$$y_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c} \quad (7)$$

This pooling technique effectively captures the global context of the feature maps, making it particularly useful in classification tasks where spatial information is less relevant. By reducing the dimensions while retaining important information, GAP contributes to halt overfitting and improving the model's generalization capabilities.

D. *Experiments*

In this section, we outline experimental setup, covering training parameters, details of the dataset, and the evaluation methods used to measure performance of proposed model.

Computing Environment: All experiments were carried out on the Kaggle platform with a configuration of two NVIDIA Tesla T4 GPUs, using Kaggle's default standard RAM and disk space.

Frameworks: The research was carried out in Python, utilizing key libraries that are crucial for developing and evaluating models. TensorFlow, along with the Keras API, was employed to construct and train the deep learning model, providing ready-to-use functions for configuration, compilation, and optimization. For data manipulation and preprocessing, Pandas and NumPy were utilized, while scikit-learn assisted with model validation, metrics evaluation, and Stratified Kfold splitting. To visualize the metrics, Matplotlib and Seaborn were used, and Pathlib helped with file management. These libraries were chosen for their flexibility, strong community support, and compatibility with deep learning workflows. All code and scripts used for data preprocessing, model training, and evaluation are accessible at Github[2], enabling replication and adaptation of the proposed approach.

Training Parameters: This model is trained with various settings, which are fine-tuned through experimentation. The focus is on identifying the best-performing configuration of parameters for the dataset. (See Table I)

TABLE I HYPERPARAMETERS USED IN TRAINING WITH ASL DATASET

Hyperparameter	***Value***
Cross Validation	Stratified K-Fold
Epochs	15

[2] https://github.com/mani5100/Hand-Sign-Language-Detection

Hyperparameter	*Value*
Folds	8
Learning rate	0.001
Batch size	32
Optimizer	adam
Seed	42
Image size	200 x 200

IV. Results And Discussion

This section gives quantitative analysis and comparison of proposed novel DCNN v2 model with state-of-the-art methods, followed by in-depth discussion of model's performance through various metrics and visualizations. Figure 2 presents the validation and training loss curves. Initially, training loss decreases rapidly, stabilizing around epoch 5. The validation loss, however, shows fluctuations in the early epochs, peaking around epochs 2 and 4 before converging to near-zero values. This behavior suggests that the model experienced some difficulty generalizing during the initial training phase, but improved significantly after optimization. The convergence of both validation and training loss curves by epoch 7 indicates the model is not overfitting and has successfully minimized errors.

In figure 3, we observe a steady rise in training accuracy, reaching near-perfect levels (almost 1.0) by epoch 5. The validation accuracy, although initially fluctuating, stabilizes by epoch 7, closely matching the training accuracy. The consistent performance across both training and validation datasets highlights the model's generalization capability, reflecting its robustness in classifying unseen hand language signs.

Figures 2 and 3 likely represent the spikes that occur when training early models. High learning rate can at times produce fluctuations in loss and accuracy simply because of initial shuffling in weights in batches. There is usually instability at early epochs due to fluctuations of data patterns that might temporarily increase the validation loss and decrease the accuracy of the model. On top of that, overfitting on some patterns of data might cause such great variations at least before the regularization techniques start working. Such peaks tend to become stable as the model learns to converge, meaning that the initial changes are part of a learning process and not in any way reflect the performance of the whole model.

The precision and F1 scores metrics exhibit considerable fluctuations during the early epochs, but gradually stabilize. Precision oscillates more than the F1 score, indicating that the model faced difficulties in avoiding false positives in the initial training phases. By the final epochs, both precision and F1 score approach stable values, reflecting model's capability to maintain balance between recall and precision across all hand language sign classes.

Figure 2 and Figure 3 demonstrate that while the model initially faced overfitting, the applied data augmentation techniques and the architecture's deep layers improved its performance over time, converging both accuracy and loss to nearly perfect levels.

The model's overall performance of distinguishing classes (A-Z) is perfect. The final plot presents the ROC curve, which follows the top-left border of the plot, indicating that TPR of 1.0 at all FPR values. The AUC is a perfect 1.0, confirming that the model achieves perfect classification performance without any trade-offs between sensitivity and specificity. The ROC curve's perfect shape and AUC value of 1.0 shows that the model performs great in terms of accuracy, precision and also maintains perfect discrimination between the positive and negative instances in each classification task.

The cross-dataset validation experiment was conducted using an Indian Sign Language dataset, which is quite different from the original training data. Therefore, it will be able to test this model's ability to classify unseen hand signs correctly. This test verifies if the model's high accuracy is transferable to other datasets in the real world. Thus, its versatility in representing different sign languages can be distinguished. The model properly predicted labels for the Indian Sign Language dataset; all examples showed proper alignment between actual and predicted labels. The results show that the model performed very strongly and was fair while correctly classifying hand signs differently.

Table II Comparison with State-of-the-arts

Model Architecture	*Accuracy*	*Precision*	*Recall*	*F1-score*
DCNN v2	100	100	100	100
CNN + RTS (Baseline) [9]	98	96	97.1	96.4
CNN [27]	90	98	98	98
CNN [28]	96.3	96.8	97.3	96.2
CNN [10]	92.9	93.2	93.6	93.1

To evaluate the performance of the novel DCNN v2 on SLD, a comparative analysis is conducted against previous research models. The table II highlights the key evaluation metrics. As observed, our developed model outperforms baseline CNN and other models in Table II across all key metrics. The accuracy is perfect, with an F1 score and precision achieving 1.00, reflecting the model's robustness in identifying hand signs language.

V. Conclusion

Architecture DCNN v2 was highly accurate in classifying sign language gestures with high precision, recall, and F1 scores. The model did experience pretty major fluctuations at the start of training; however, the validation loss and accuracy were pretty good at generalization later on. Data augmentation greatly minimized overfitting and enhanced robustness supported strong performance in the confusion matrix, as well as the ROC curve. Future improvements may include advanced techniques for augmentation, transfer learning, adding extra diversity to the dataset, including diverse gestures or real-time data and adding attention mechanisms to deal with more complex gestures and extend the applicative range of the model.

References

[1] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep learning for sign language recognition: Current techniques, benchmarks, and open issues," IEEE Access, vol. 9, pp. 126 917–126 951, 2021.

[2] A. Deep, A. Litoriya, A. Ingole, V. Asare, S. M. Bhole, and S. Pathak, "Realtime sign language detection and recognition," in 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), 2022, pp. 1–4.

[3] P. R. Sanmitra, V. V. S. Sowmya, and K. Lalithanjana, "Machine learning based real time sign language detection," IJRESM, vol. 4, no. 6, pp. 137–141, 2021.

[4] S. Arooj, S. Altaf, S. Ahmad, H. Mahmoud, and A. S. N. Mohamed, "Enhancing sign language recognition using cnn and sift: A case study on pakistan sign language," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 2, p. 101934, 2024.

[5] P. Rakshit, S. Paul, and S. Dey, "Sign language detection using convolutional neural network," Journal of Ambient Intelligence and Humanized Computing, vol. 15, no. 4, pp. 2399–2424, 2024.

[6] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-Gonz´alez, and J. M. Corchado, "Deepsign: Sign language detection and recognition using deep learning," Electronics (Basel), vol. 11, no. 11, p. 1780, 2022.

[7] V. Munnaluri, V. Pandey, and P. Singh, "Machine learning based approach for indian sign language recognition," in 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022, pp. 1128–1132.

[8] A. Roy, P. Bhuyan, J. S. Balakrishnan, J. Mohan, V. T. Pradeep, A. R. Joshi, and P. K. Yadava, "Sign language recognition in augmented reality," J. Vis. Commun. Image Represent., vol. 76, p. 103002, 2021.

[9] A. S. M. Miah, J. Shin, M. A. M. Hasan, M. A. Rahim, and Y. Okuyama, "Rotation, translation and scale invariant sign word recognition using deep learning," Comput. Syst. Sci. Eng., vol. 44, no. 3, pp. 2521–2536, 2023.

[10] S. Sharma and K. Kumar, "Asl-3dcnn: American sign language recognition technique using 3-d convolutional neural networks," Multimed. Tools Appl., vol. 80, no. 17, pp. 26 319–26 331, 2021.

[11] K. P. Nimisha and A. Jacob, "A brief review of the recent trends in sign language recognition," in 2020 International Conference on Communication and Signal Processing (ICCSP), vol. 7, 2020, pp. 186–190.

[12] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), 2018, pp. 194–197.

[13] M. N. Saiful et al., "Real-time sign language detection using cnn," in 2022 International Conference on Data Analytics for Business and Industry (ICDABI), 2022, pp. 697–701.

[14] D. S. Bendarkar, P. A. Somase, P. K. Rebari, R. R. Paturkar, and A. M. Khan, "Web based recognition and translation of american sign language with cnn and rnn," Int. J. Onl. Eng., vol. 17, no. 01, pp. 34–50, 2021.

[15] B. D. Khuat, D. T. Phung, H. T. T. Pham, A. N. Bui, and S. T. Ngo, "Vietnamese sign language detection using mediapipe," in 2021 10th International Conference on Software and Computer Applications, vol. 2, 2021, pp. 162–165.

[16] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan, "Real-time sign language detection using human pose estimation," in Lecture Notes in Computer Science. Springer International Publishing, 2020, pp. 237–248.

[17] M. Muthuswamy, A. M. Ali, and A. Abdelhafeez, "Breaking the silence: Convolutional neural networks for sign language recognition in the deaf community," Sustain. Mach. Intell. J., vol. 1, pp. 1–9, 2022.

[18] M. M. Kamruzzaman, "Arabic sign language recognition and generating arabic speech using convolutional neural network," Wirel. Commun. Mob. Comput., vol. 2020, pp. 1–9, 2020.

[19] S. K. Paul, M. A. A. Walid, R. R. Paul, M. J. Uddin, M. S. Rana, M. K. Devnath, I. R. Dipu, and M. M. Haque, "An adam based cnn and lstm approach for sign language recognition in real time for deaf people," Bulletin of Electrical Engineering and Informatics, vol. 13, no. 1, pp. 499–509, 2024.

[20] A. S. M. Miah, M. A. M. Hasan, S. Nishimura, and J. Shin, "Sign language recognition using graph and general deep neural network based on large scale dataset," IEEE Access, 2024.

[21] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan, "Real-time sign language detection using human pose estimation," in Lecture Notes in Computer Science. Springer International Publishing, 2020, pp. 237–248.

[22] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," Neural Comput. Appl., vol. 33, no. 18, pp. 12 165–12 179, 2021.

[23] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, "Sign language identification and recognition: A comparative study," Open Comput. Sci., vol. 12, no. 1, pp. 191–210, 2022.

[24] N. Saquib and A. Rahman, "Application of machine learning techniques for real-time sign language detection using wearable sensors," in Proceedings of the 11th ACM Multimedia Systems Conference, 2020.

[25] K. Ayush and S. Mandal, "Identifying indian sign languages using decentralized deep learning," IJRESM, vol. 5, no. 11, pp. 241–246, 2022.

[26] S. Katoch, V. Singh, and U. S. Tiwary, "Indian sign language recognition system using surf with svm and cnn," in Array (N. Y.), vol. 14, 2022, p. 100141.

[27] R. S. L. Murali, L. Ramayya, and V. A. Santosh, "Sign language recognition system using convolutional neural network and computer vision," Int J EngInnov Technol, pp. 2582–1431, 2020.

[28] M. Hurroo , M. Elham, "Sign language recognition system using convolutional neural network and computer vision," Int J Eng Research & Technol, pp. 59–64, 2020.