# R-SLR: Real-Time Sign Language Recognition System

Monalisa Ghosh
*Institute Of Engineering and Management*
*Department of Computer Science*
Kolkata, India
monalisa11@iitkgp.ac.in

Debjani De
*Institute Of Engineering and Management*
*Department of Computer Science*
Kolkata, India
debjanide2017@gmail.com

Lovely Anand
*Institute Of Engineering and Management*
*Department of Computer Science*
Kolkata, India
lovelyanand1201@gmail.com

Satyakam Baraha
*Sri Sivasubramaniya Nadar College of Engineering*
*Department of Electronics and Communication*
Chennai, India
satyakamb@ssn.edu.in

2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT) | 979-8-3503-7212-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIIoT58432.2024.10574553

***Abstract*—Sign language is the main medium of communication for people with hearing impairments. This language is mostly based on hand expressions complemented by non-manual gestures. It facilitates harmonious integration of the disabled into our society. In this paper, we present R-SLR, a sign language recognition system that can recognize the signs in real-time. R-SLR identifies the hand in a video stream and extracts the region of interest. We create our dataset by collecting the frames and pre-process them to enhance the frame quality and reduce noise. R-SLR detects sign language using Convolutional Neural Network. We extract the features from the pre-processed frames and classify the signs using the pre-trained DenseNet 201 model. The model performance is tested and it achieves 96.5% accuracy. The R-SLR then translates the identified sign language gesture into text or speech. Our designed R-SLR is an efficient and non-intrusive system.**

***Keywords*—Convolutional Neural Network, DenseNet 201, Hand tracking, Normalization, Sign Language.**

## I. Introduction

Sign language is a visual language used by people those who are deaf or hard of hearing in order to communicate with others. It is an important means of communication for millions of people worldwide [1]. However, for individuals who are unfamiliar with sign language, it can be challenging to understand and communicate effectively with them. A system that can recognize sign language gestures and convert them into speech or text can help bridge this communication gap and make better the life quality of individuals who are deaf or find difficulty in hearing. Current advancements in computer vision along with machine learning has made it possible to develop systems that can recognize sign language gestures and convert them into text or speech [2]. Image-based hand gesture recognition techniques using OpenCV [3] is one such approach that can be used to recognize sign language gestures. OpenCV (Open Source Computer Vision) is an open- source library which provides different tools and algorithms for computer vision and image processing.

Real- time sign language recognition is a promising research area that has gained a lot of attention in the recent years [4]. There are several reasons why this area of research is important and motivating. Firstly, sign language is a critical means of communication for deaf people with difficulties in hearing [5]. Secondly, traditional sign language recognition systems often require expensive and intrusive hardware, such as gloves or sensors, which can limit their accessibility and adoption. Real time image-based hand gesture recognition techniques using OpenCV can overcome these limitations as it is non-intrusive and does not require any additional hardware. Thirdly, computer vision and machine learning have advanced significantly in recent years, making it possible to develop accurate and efficient sign language recognition systems. These systems have the potent to be used in wide range of applications, like virtual assistants to gaming and robotics. Finally, the development of a real-time system for sign language identification using image-based hand gesture recognition techniques with OpenCV can contribute to the advancement of computer vision and machine learning research. It presents a challenging problem that requires the development of novel algorithms and techniques for hand gesture recognition and classification.

The goal of this paper is to develop R-SLR, a real time sign language recognition system using image-based hand gesture recognition techniques. Our R-SLR system uses a camera to capture video of the hand gestures and recognize them using image processing techniques. R-SLR then converts these recognized gestures into text or speech by using natural language processing (NLP) techniques. Our proposed system has the ability to enhance communication for individuals those who are deaf or have difficulty in hearing and facilitate their integration into society. This can also be used in several applications, including virtual assistants. R-SLR system can operate in real- time and is computationally efficient. Also, our proposed system is non- intrusive, means that it does not require any additional hardware or sensors to be worn by the users.

The main contributions of our work are as follows-

i) R-SLR identifies the hand in a video stream and extracts the Region Of Interest (ROI) containing the hand.
ii) R-SLR uses image processing techniques to extract inputs/ features from the ROI.
iii) R-SLR recognizes the sign language gestures employing DenseNet 201.

TABLE I
LIST OF EXISTING WORKS IN SIGN LANGUAGE RECOGNITION

| Serial No. | Paper | Author | Technique/ Network used | Dataset | Accuracy |
|---|---|---|---|---|---|
| 1 | VTN [6] | Wuyang Qin *et al.* | VTN (Video Transformer Network) | Chinese Sign Language-Bank & Station | 87.9% |
| 2 | Two-stream network [7] | Hamzah Lugman | Dynamic motion network + Accumulative Motion network + Sign recognition network | KArSL-190 and KArSL-502 Arabic sign language datasets | 15% increase over existing |
| 3 | Recognition of isolated gestures [9] | Mikhail G. Grif *et al.* | LSTM classifier | prepared by Novosibirsk State Technical University | 80-97 % |
| 4 | Static hand gesture [10] | Prangon Das *et al.* | CNN | American Sign Language (ASL) | 94.34 % |
| 5 | Privacy-Preserving BSLR [13] | Hameed *et al.* | VGG 16, VGG19, Inception V3 | British Sign Language (BSLR) | 93.33% |
| 6 | Dynamic Signs [14] | Arun Singh *et al.* | CNN | data collected using mobile devices | 70 % |
| 7 | Bangla Sign Language (BSL) [15] | Basnin *et al.* | CNN-LSTM model | Bangla Sign Language (BSL) | Train-90%, Test-88.5% |

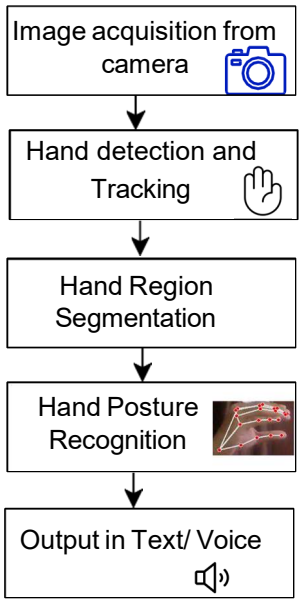



Fig. 1. Flow chart of R-SLR system

iv) R-SLR converts the recognized sign language gesture into text or speech using Natural Language Processing (NLP) techniques.

v) R-SLR is a real -time, efficient and non-intrusive system.

The rest of the paper is structured as follows. Section II reviews the related works. Section III discusses the proposed R-SLR system model. Section IV presents the R-SLR experimental results. Section V concludes the paper with future scope.

## II. RELATED WORKS

The authors in [6] constructed a sign language translation network using Video Transformer Net (VTN). In order to learn independent sign language, they set up a set up a two-way VTN and compared it with Inflated three Dimension (I3D). In [7], Luqman presented a deep learning network that can successfully record the spatio-temporal information with only a few frames of signs. First dynamic motion network (DMN) stream learns spatial-temporal information about signs via key postures. Then they combine crucial postures in both forward and reverse directions in order to create a cumulative video frame, which was given as input to accumulative motion network. These retrieved features were integrated with DMN features and for sign classification.

The work in [8] proposed an ASL recognition system for recognizing the hand gestures by employing the Deep Convolutional Neural Network (CNN). Their study was limited to experimental results only on the publicly available datasets. Grif *et al.* [9] used component based approach, which involves representing gesture in terms of the components, such as, palm configuration or position with respect to the body. Das *et al.* [10] studied the recognition of dynamic signs using CNN. They were able to attain accuracy of around 70%. Shi *et al.* [11] extracted the target and recognized gestures on the video frames using Jetson TX2 and yolov5 algorithm. Similar to [11], Sharma *et al.* used the yolov4 algorithm [12].

In [13], the authors developed a contact-less, but secured privacy based British SLR system with focus on six commonly used emotions like, state of confusion, happiness, loneliness, sadness etc. At first, the obtained data was displayed as spectograms. The spectograms were processed using VGG 16, VGG 19, and Inception V3 to to gather spatio-temporal information. The emotions were successfully determined by categorizing spectograms according to specific emotional index. This research in [14] proposes a dynamic SLR model using CNN. The suggested model has undergone training and testing using video clips of dynamic indicators. The works in [15] and [16] used a combined CNN-LSTM model for Bangla lexicals and Indian SLR, respectively. Koller *et al.* further extended their work by embedding combined CNN-LSTM model into each Hidden Markov Model (HMM) stream [17] using the hybrid methodology. This helps in the identification of features such that, when considered individually, do not possess enough discriminative strength.

The study in [18] presents a Chinese SLR system that utilizes the handy and cost-effective leap motion sensor and applies the k-NN algorithm, with a focus on feasibility. Reshna *et al.* presented a system [19] capable of accurately identifying sign language gestures in challenging contexts. They captured video recording of an individual, and a specific portion of the hand that was displaying the sign was isolated based on its skin color. Sign-specific characteristics were taken from the hand image and analyzed to identify the signs using Support Vector Machine. Table I lists the summary of works that has

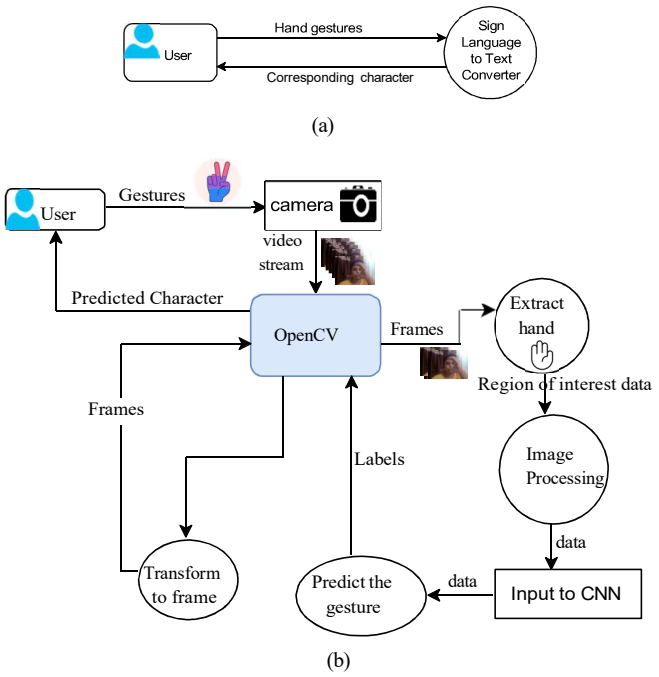



Fig. 2. (a)Data flow diagram 1 (b)Data flow diagram 2 of the proposed R-SLR system

been carried out in this sign language recognition system.

In summary, these studies underscore the potential of machine learning based hand gesture recognition techniques for sign language. While showing promise, opportunities for improvement remain in terms of accuracy, robustness, and real-time performance. The development lacks in terms of real-time implementation.

## III. Proposed R-SLR System Model

### A. R-SLR system model:

The image in Fig. 1 shows a flow chart that depicts the process of hand detection and tracking, hand region segmentation, hand posture recognition, and output in text or voice. The diagram shows the flow of information from image acquisition from a camera to the final output. The detailed data flow diagram in Fig. 2 showcases the process of capturing hand gestures through a camera, extracting the ROI data using OpenCV, processing the frames, and converting the gestures into corresponding characters using a CNN. It includes two diagrams labeled (a) and (b) for different levels of the data flow diagram.

Fig. 3 is a sequence diagram with the following components: User, Webcam, System, Model, video feeding, Image, Hand detector, Feature extraction, Feature matching, Matching result, and Result. The diagram illustrates a process involving video feed, image processing, hand detection, feature extraction and matching, and displaying the result.

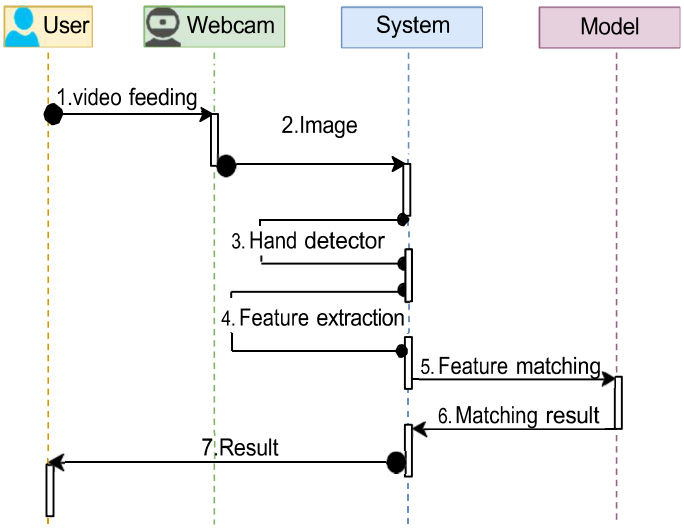
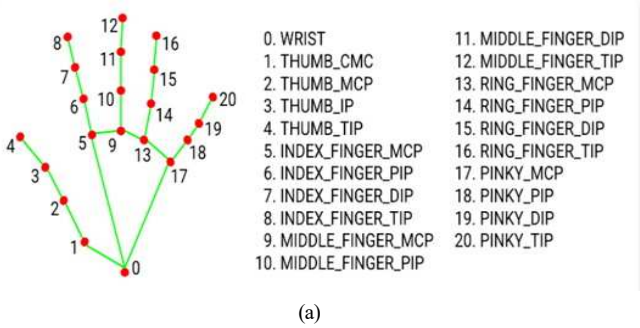



Fig. 3. R-SLR sequence diagram




(a)

Fig. 4. Media pipe Landmark System

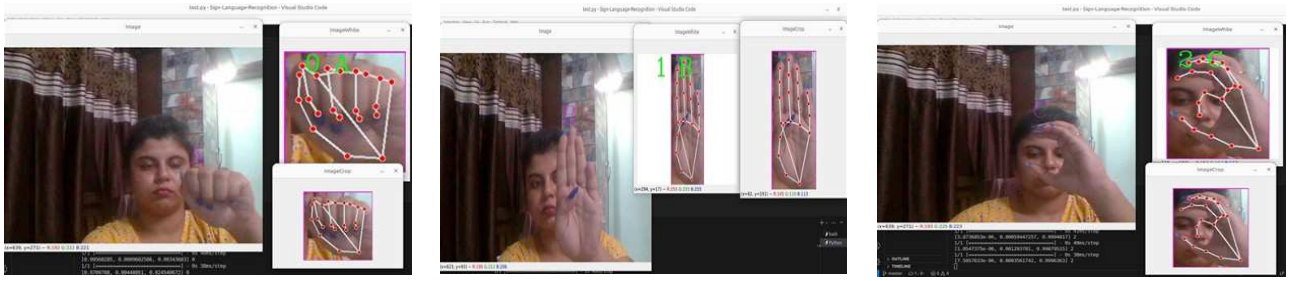
### B. R-SLR implementation details:

The implementation of R-SLR, a real-time sign language recognition system using image-based hand gesture recognition techniques can be divided into the following steps:

i) **Hand detection:** The first step is to detect the hand region in the image or video frame. This can be done using various techniques, such as skin color segmentation, contour analysis, and Haar cascades. OpenCV provides several functions and tools for hand detection, including cv2.CascadeClassifier and cv2.inRange. Fig. 4 shows the media pipe landmark system that detects the landmark of human hands.

We have used cvzone.HandTrackingModule to detect hand from the web stream. CvZone is a package that uses Mediapipe libraries and OpenCV at its core that can be used to detect face, pose, selfie, hands etc. Specifically the R-SLR hand tracking module works as follows:

a) Detects 21 key points/landmarks in our hand/palm and produces a graph out of it.
b) Counts the number of pendant graphs which represents the fingers.



(a) (b) (c)

Fig. 5. Pre-processing stages- Extracting the ROI and resizing the video frames one-by-one

Fig. 6. Real time data collection in R-SLR system

c) Calculate distance between specific landmarks on the first hand and draws it.
d) If this matches with the trained weights, it decides whether it is a hand or not.

ii) **Dataset Creation and Preprocessing:** Once the hand region is detected, preprocessing techniques can be applied to enhance the image quality and reduce noise. This can include operations such as smoothing, thresholding, and morphological operations. Fig. 5 shows the pre-processing steps of our proposed R-SLR system. For R-SLR setup, first we import necessary libraries (cv2 from OpenCV, HandDetector from cvzone.HandTrackingModule, numpy, os, traceback). Then we initialize a video capture object (capture) to access the default camera. We create two instances of HandDetector from cvzone.HandTrackingModule (hd and hd2), each configured to track a maximum of one hand.
For processing the video frames in loop, we perform the following steps:

a) We capture frames from the camera continuously using capture.read().
b) Flip the frame horizontally using cv2.flip.
c) Use hd.findHands to detect hands in the frame. If hands are detected, we proceed with further processing.
d) Extract the ROI around the detected hand from the frame.
e) Use hd2.findHands to detect hands within the ROI. If hands are detected, create a visual representation of the hand skeleton on a white canvas. The script uses a white canvas image (white) for drawing the hand skeleton, and this canvas is loaded from a saved image or frame before processing.
f) Draw lines connecting key landmarks of the hand to represent the hand skeleton on the white canvas.
g) Display the processed skeleton image using cv2.imshow.

The R-SLR user interaction steps are as follows:
1) Display information on the frame, including the current directory (c_dir) and the count of images in that directory.
2) Use keyboard inputs for user interaction:

a) Press ‘n’ to switch to the next directory (‘A’ to ‘Z’ looping back to ‘A’).
b) Press ‘a’ to start capturing hand poses.
c) Press ‘Esc’ key to exit the script.

3) The frame saving steps are as follows:

a) If the ‘a’ key is pressed and flag is set to True, save the current hand pose skeleton image to a directory based on the current letter (c_dir).
b) Images are saved in the format “<image_name>.jpg”.
c) The script tracks the count of saved images in each directory and updates the count accordingly.
d) The script stops capturing images when the ‘a’ key is pressed for 180 times or when the ‘Esc’ key is pressed.

We have collected images of different signs at different angles for sign letter A to Z, as shown in Fig. 6. There can be many loopholes in this collection process. Like, the hand might not be in front of a clean soft background and may lack proper lighting condition. In real world, we usually do not get good background everywhere and we may not get good lighting conditions either. So to overcome this situation in R-SLR, we tried different approaches. Then we reached one interesting solution in which first we detect the hand from the frame using a media pipe and obtain the hand landmarks present in that frame. Then we draw and connect those landmarks in a simple white image as described above.

iii) **Feature extraction and Gesture Recognition:** The next step is to extract features from the pre-processed image or frame. This can include features such as hand shape, orientation, and movement. In R-SLR, we extract

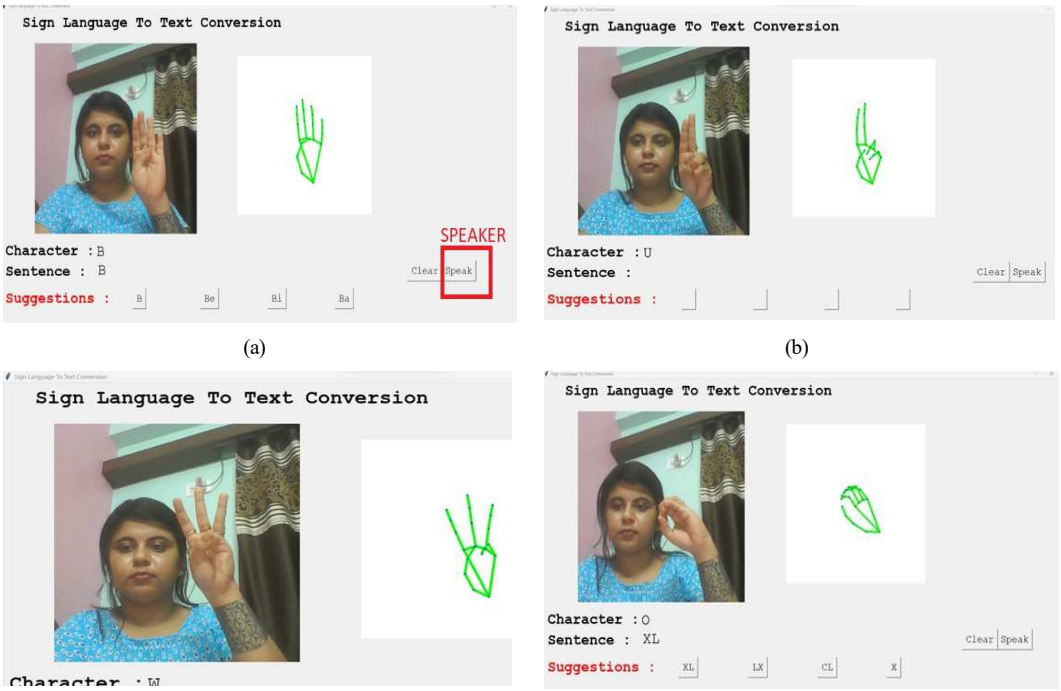



Fig. 7. Test results showing the real-time recognition of different signs

features that should represent an identifiable American sign language figure. The extracted features can be used to classify the hand gesture into a predefined set of gestures.

In this work, we have used an advanced CNN model, DenseNet 201 [20] more specifically to recognize the Sign Language. DenseNet, short for Densely Connected Convolutional Networks, is a strong neural network architecture. DenseNet differs from conventional topologies in that each layer is directly linked to every subsequent layer, as opposed to layers merely linking to their immediate neighbours. This dense connection promotes feature reuse, allowing the model to pick up detailed information from the input images very well. The DenseNet 201 model was trained and tested on ASL dataset.

The ASL dataset is a very diverse dataset that covers a wide range of hand signs and gestures and was carefully designed to capture the subtle details of the language. Each image has undergone some preprocessing, ensuring it is suitable for the DenseNet 201 model. Dimensions of image used as dataset is 400 × 400 pixels as 8-bit unsigned integers, which ranges from 0 to 255. The dataset includes sticks figures with white background. The stick figures represents the key points of the palm such the knuckles, finger tips to represent gesture. The dataset was collected manually whereby snapshots of ASL gestures were taken, fine grained and imposed to stick figure. These stick figures were grouped by each gestures and the model was trained with the dataset along with the gesture they were representing. Then our collected images/pre-processed frames are given as input to this pre-trained DenseNet 201 model.

iv) **Post Processing:** Finally, the recognized gesture can be post processed to improve the accuracy and robustness of the system. This can include operations such as temporal smoothing, gesture verification, and error correction. As a part of prediction in post-processing, in R-SLR we use the same techniques that we used for pre-processing, where we draw a 21 key landmark graph and draw the skeleton in a white canvas. Then we determine which of the corresponding graph matches the existing datasets. Fianlly, we used NLP Technique to convert speech to text. For this, we have used the pyttsx3 library. It's a wrapper around several text-to-speech engines, including Microsoft's Text-to-Speech (TTS) engine.

## IV. R-SLR Experimental Results

TABLE II
List of Model Parameters

| #Parameters | Value |
|---|---|
| Optimizer | Adam |
| Dropout | 0.2 |
| Activation function | SoftMax |
| Pooling | Avg |

The DenseNet 201 is trained on the ASL dataset and tested on the same dataset with train:test ratio 80:20. We use the

TABLE III
MODEL PERFORMANCE

| Model | Test dataset | Accuracy |
|---|---|---|
| DenseNet 201 | ASL | 0.99 |
| | our collected | 0.965 |

evaluation metric i.e., accuracy to measure the performance of the model given as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where, $TP$, $TN$, $FP$ , and $FN$ represents True Positive, True Negative, False Positive and False Negative, respectively. The DenseNet 201 achieved 99% accuracy. Table II lists the parameters for DenseNet 201.

This pre-trained DenseNet 201 is then tested on our collected dataset. During testing, the pre-trained DenseNet 201 model achieved an impressive 96.5% accuracy when evaluated on our collected images/ preprocessed frames. Table III lists the performance of DenseNet 201 model. The model's impressive accuracy shows that it is good at recognizing and sorting ASL signs, making it useful for real-world tasks like sign language interpretation and communication. The close connections in the DenseNet 201 were crucial for capturing the subtle details in the dataset, which greatly contributed to its outstanding accuracy. Figure 7 shows the output results, wherein the signs are being identified and displayed in form of text or voice. The recognised signs are also audible from the speaker.

## V. CONCLUSION

We have successfully designed and developed the Real-Time Sign Language Recognition System (R-SLR). One key feature is that we are doing it in real time taking video as input. The system will be able to understand sign language and translate that to the corresponding text as well as voice. We have used ASL dataset and also created our own dataset. Extension of this work can include the inclusion of special characters and numbers. Preprocessing can also include better image enhancement and noise removal techniques. Working of the cvZone hand tracking module can be expected to be improved with better camera images.

## REFERENCES


[1] J.P. Sahoo, S.P. Sahoo, S. Ari and S.K. Patra, "Hand Gesture Recognition Using Densely Connected Deep Residual Network and Channel Attention Module for Mobile Robot Control," in *IEEE Transaction on Instrumentation and Measurement*, vol. 72, pp. 1-11, 2023.

[2] J.P. Sahoo, S.P. Sahoo, S. Ari and S.K. Patra, "DeReFNet: Dual-stream Dense Residual Fusion Network for static hand gesture recognition," in *Displays, Elsevier*, vol. 77, pp. 102388, 2023.

[3] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *IEEE proceedings of the 35th international convention MIPRO*, pp. 1725-1730, May, 2012.

[4] J.P. Sahoo, S.P. Sahoo, S. Ari and S.K. Patra, "RBI-2RCNN: Residual Block Intensity Feature using a Two-stage Residual Convolutional Neural Network for Static Hand Gesture Recognition," in *Signal, Image and Video Processing, Springer*, vol. 16, no. 8, pp. 2019-2027, 2022.

[5] K. Waldow, A. Fuhrmann and D. Roth, "Deep Neural Labeling: Hybrid Hand Pose Estimation Using Unlabeled Motion Capture Data With Color Gloves in Context of German Sign Language," in *IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, Los Angeles, CA, USA, pp. 1-10, 2024.

[6] W. Qin, X. Mei, Y. Chen, Q. Zhang, Y. Yao, Y. and S. Hu, "Sign language recognition and translation method based on VTN," in *IEEE International Conference on Digital Society and Intelligent Systems (DSInS)*, pp. 111-115, Dec. 2021.

[7] H. Luqman, "An Efficient Two-Stream Network for Isolated Sign Language Recognition Using Accumulative Video Motion," in *IEEE Access,* vol. 10, pp. 93785-93798, 2022.

[8] P. Das, T. Ahmed and M. F. Ali, "Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network," in *IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, pp. 1762-1765, 2020.

[9] M. G. Grif and Y. K. Kondratenko, "Recognition of Isolated Gestures of the Russian Sign Language Based on the Component Approach," *IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE)*, Novosibirsk, Russian Federation, pp. 1510-1513, 2023.

[10] P. Das, T. Ahmed and M. F. Ali, "Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network," *IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, pp. 1762-1765, 2020.

[11] D. Shi, L. Long, Y. Zhang, H. He and X. Liu, "Sign Language Recognition System based on Jetson TX2 and Yolov5," *4th International Symposium on Smart and Healthy Cities (ISHC)*, Shanghai, China, pp. 242-246, 2022.

[12] S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale, and S. Khurana. "Real-Time Word Level Sign Language Recognition Using YOLOv4," in *IEEE International Conference on Futuristic Technologies (INCOFT)*, pp. 1-7, 2022.

[13] H. Hameed *et al.*, "Privacy-Preserving British Sign Language Recognition Using Deep Learning," in *44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC),* Glasgow, Scotland, pp. 4316-4319, 2022.

[14] A. Singh, A. Wadhawan, M. Rakhra, U. Mittal, A. A. Ahdal and S. K. Jha, "Indian Sign Language Recognition System for Dynamic Signs," in *10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, pp. 1-6, 2022.

[15] N. Basnin, L. Nahar, and M. S. Hossain, "An integrated CNN-LSTM model for Bangla lexical sign language recognition," in *Proc. of International Conference on Trends in Computational and Cognitive Engineering*, Singapore: Springer Singapore, pp. 695-707, 2020.

[16] C. Aparna, and M. Geetha, "CNN and stacked LSTM model for Indian sign language recognition," in *Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA* , Trivandrum, India, Springer, pp. 126-134, 2020.

[17] O. Koller, N. C. Camgoz, H. Ney and R. Bowden, "Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306-2320, Sept. 2020.

[18] Y. Xue, S. Gao, H. Sun and W. Qin, "A Chinese Sign Language Recognition System Using Leap Motion," in *International Conference on Virtual Reality and Visualization (ICVRV)*, Zhengzhou, China, pp. 180-185, 2017.

[19] S. Reshna and M. Jayaraju, "Spotting and recognition of hand gesture for Indian sign language recognition system with skin segmentation and SVM," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, pp. 386-390, 2017.

[20] S.H. Wang and Y.D. Zhang, "DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification," in *ACM Trans. Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2s, pp.1-19, 2020.