Grenze International Journal of Engineering and Technology, June Issue

Real-Time Sign Language Recognition System: Word Level

Malapriya S[1] and Manju N[2]
[1-2]Department of Information Science and Engineering, JSS Science and Technology University, Mysuru, India

***Abstract*— Communications are very important in the people community. This research paper aims to develop a real-time sign language detection system based on Recurrent Neural Networks (RNNs) that translates gestures into written words. To record the temporal dynamism involved in sign language, initially collecting and pre-processing a large dataset of different sign language motions, followed by designing and training an effective RNN model. A user-centric interface that would be accessible by individuals with various levels of technology expertise will be created to enhance usability and simplicity in order to integrate easily into everyday communication aids. The initiative focuses on testing with users and extensive testing and evaluation to ensure high accuracy and reliability, advancing deaf and hard-of-hearing inclusion and accessibility.**

***Keywords*— Sign Language Recognition, Recurrent Neural Networks, Real-Time Gesture Detection.**

I. INTRODUCTION

Communication is an inherent part of human interaction, and for deaf and hard-of-hearing individuals, sign language is an important medium. Sign language is a rich, complex way of communicating that employs hand movements, face expressions, and body postures to convey messages. Nonetheless, a considerable hurdle remains between individuals using sign language and those who do not, thereby affecting communication in social, educational, and working environments. Technological innovation such as sign language detection systems has bridged the gap, which is attributed to improvements in artificial intelligence and machine learning.

Recurrent Neural Networks (RNNs) are well adapted to sign language detection because they can handle sequential data and temporal dependencies. In contrast to regular neural networks, RNNs contain connections that create directed cycles, which allow them to keep a memory of previous inputs. This attribute makes RNNs perfect for applications involving sequences and time-series data, such as sign language, in which the context and order of signs are key to proper interpretation. With RNNs, one can design systems capable of learning and understanding the dynamic signs' patterns in sign language, enabling real-time translation and accessibility of communication.

Several key steps are involved in the deployment of an RNN-driven sign language recognition system. In the first place, a thorough dataset of sign language signs and expressions needs to be gathered and marked, detailing a wide variety of signs and expressions. The data is subsequently pre-processed to derive informative features, most commonly by utilizing Convolutional Neural Networks (CNNs) to transform video frames into feature vectors.

Grenze ID: 01.GIJET.11.2.771

These vectors are inputted into the RNN, which is trained to identify patterns and sequences for specific signs. As the model undergoes iterative training and fine-tuning, it learns to properly predict sign language gestures from new video inputs.

One of the major challenges to implementing such a system is guaranteeing real-time processing capability. This necessitates the optimization of the RNN architecture so that accuracy and speed are balanced, allowing the system to respond instantly. An easy-to-use interface is needed for practical applications, enabling people to use the system without interruptions. Through incorporating functionalities such as live video recording, instant translation, and accessible design attributes, the system can be rendered usable across a variety of settings and by users having different levels of technical expertise.

The contribution of a successful sign language detection system with RNNs is sizeable. It can improve communication and accessibility for the deaf and hard-of-hearing population, opening doors to a more inclusive society and fostering understanding in many areas of life. With further advancements in technology, the optimization and implementation of such systems promise to make the world a more inclusive and interconnected place where linguistic differences no longer stand in the way of interaction and cooperation.

II. Literature Survey

The author presents a sign recognition in real time architecture using gesture recognition using data gloves and accelerometers. The architecture operates in two tiers, segmentation and classification, to address the challenges of real-time processing. Despite efforts to handle issues like sensor noise and simplify training, segmentation problems persist, particularly when there are slight hand movements between signs or corrections during signing. To improve recognition rates, the authors suggest the creation of more complex classifiers. This research contributes valuable insights into the development of effective sign recognition systems for enhanced human-computer interaction [1]. Pala et al.'s research addresses the vital need for communication bridges between the deaf and voiceless communities and the wider population. By designing a system to interpret sign language, they aim to facilitate interaction between these groups.

The study compares the effectiveness of KNN, SVM, and CNN algorithms for sign language recognition, considering the variability in hand gestures. Their findings indicate that CNN achieves the highest accuracy of 98.49% [2]. The authors describe segmentation scheme for hand sign recognition in attention images represents a significant advancement in the field, particularly in handling multiple deformable objects in complex backgrounds. The efficiency of the scheme is reflected in its capacity to draw on previous knowledge through a prediction-and-verification strategy. Through experiments on sequences of intensity images corresponding to hand signs, the researchers obtained remarkable results, with 95% accuracy [3]. The authors primarily concerned with work emphasizes the role of deep learning in computer vision, especially hand sign recognition. Their deployment of a CNN-based sign recognition system with high speed-ups on a TCPA (Tensor Processing Cluster Array) shows the potential of taking advantage of powerful accelerators such as GPUs and FPGAs in embedded computing.

By prototype implementation of TCPA as an overlay on a Xilinx Zynq System-on-a-Chip (SoC), researchers were able to achieve significant speed-ups over the conventional ARM Cortex-A9 processors. This work represents a substantial contribution to the embedded systems domain and provides room for further research into CNN acceleration methods for real-time applications [4]. The study primarily revolves around a new concept of real-time hand motion recognition through the fusion of Hidden Markov Models (HMMs) and Convolutional Neural Networks (CNNs). Through creating a 3-D projection of the angular correspondence between biological structures and representing it with HMMs, the authors constructed an accurate transition gesture model. Experimental outcomes exhibit the precision of the system in hand posture recognition with low inaccuracies, qualifying the system for real-time applications. Also, the fact that CNNs perform better than baseline models is a sign of how successful using deep learning approaches to solve this task has been [5].

III. Methodology

This work aims to create a precise and effective system for sign language gesture detection in real time using Recurrent Neural Networks (RNNs). The main objective is to interpret sign language into speech or written form, thus enhancing communication between sign language users and non-sign language users. With the use of cutting-edge deep learning techniques that are able to encode the sequential characteristics of gestures, the research aims to bridge the communication gap. The research encompasses building and preprocessing a large-scale dataset, creating a highly optimized RNN-based model, and developing an inclusive and user-friendly interface. Ultimately, strives to support greater inclusion and accessibility for individuals who are deaf or hard of hearing across various spheres of life, including education, work, and social interaction.

The process of creating a sign language detection system with Recurrent Neural Networks (RNNs) starts with the very important step of data gathering. A wide dataset of sign language videos is collected, including a variety of signs and expressions from various sign languages. The diversity is important so that the model can generalize well across different sign languages and differences in gesture performance. The videos thus captured go through intense preprocessing to normalize and annotate the data properly. Frames are captured from every video at regular intervals, transforming the raw video data into image frame sequences. This process ensures temporal coherence required for efficient sequence learning. Spatial features are then obtained from image frame sequences in the following stage of the system by utilizing Convolutional Neural Networks (CNNs). Owing to their effectiveness in extracting hierarchical representations from visual data, CNNs are particularly well-suited for identifying complex spatial patterns inherent in sign language gestures. Each input frame $It \in RH*W*C$, where t denotes the time step, is transformed into a corresponding feature vector $Xt\in Rd$. This transformation captures salient visual features, structuring the data for downstream temporal modelling.

These feature vectors that are extracted constitute a sequence dataset {x1, x2, x3,…. xT}, which is fed into a Recurrent Neural Network (RNN) structure. In order to properly model temporal relationships and context information over time steps, variations like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks are used.

$$ht=f(xt, ht-1;\theta)\ldots\ldots\ldots\ldots(1)$$

where ht represents the hidden state at time t, f is the recurrent function (LSTM or GRU), and θ denotes the model parameters. The recurrent layers record the temporal progression of gestures, allowing the system to process continuous motion as consistent sign sequences instead of disconnected frames. The last output layer projects the sequence of hidden state to a probability distribution over the set of sign language labels:

$$y\hat{}t=softmax\ (Wht+b)\ldots\ldots\ldots(2)$$

The model is trained in training mode by optimizing it using a labelled dataset divided into training, validation, and test sets. The objective function, generally categorical cross-entropy, is minimized to increase prediction accuracy. Optimization is performed using adaptive methods like Adam, and regularization techniques like dropout are used to avoid overfitting and improve generalization. To deploy in real-world environments, the system needs to be real-time capable. Live video input from a camera is processed frame-by-frame, where features are extracted by the CNN and on-the-fly predictions are made by the trained RNN. To enhance temporal stability and minimize prediction noise, temporal smoothing methods like a moving average or majority voting across a sliding windows are integrated. These enhancements guarantee strong and reliable interpretation, which is imperative for real-time applications in naturalistic environments.

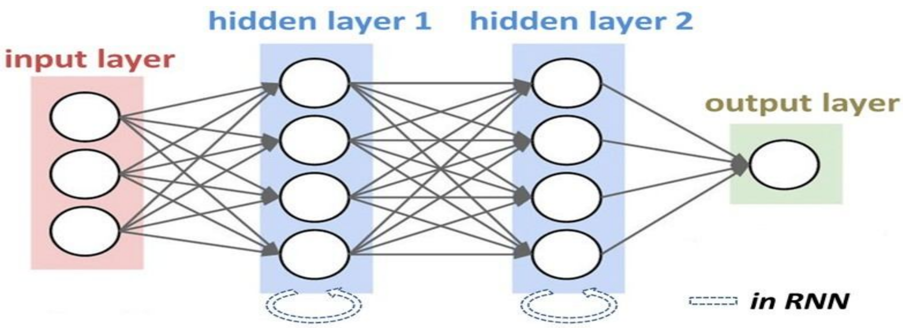

Fig 1: Architecture of RNN

The Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Networks (RNNs), is employed in this research to capture the temporal dependencies present in sign language sequences. LSTMs solve the vanishing gradient problem using gated memory cells, allowing them to keep context over long time steps. This feature is important for properly interpreting dynamic gestures that include hand movement, facial expressions, and body posture. Streams of image frames are provided as input to the LSTM model, and it learns discriminative temporal features through backpropagation through time (BPTT). The model updates its parameters iteratively by gradient-based optimization to reduce prediction error. Its memory-based structure enables it to distinguish between analogous gesture patterns through contextual clues in past frames. Once trained, the LSTM supports real-time sign language translation through class probabilities per time step as output. This real-time inference allows for smooth translation of visual-gestural input into written or spoken language, enhancing the deaf and hard-of-hearing populace's accessibility to communication.

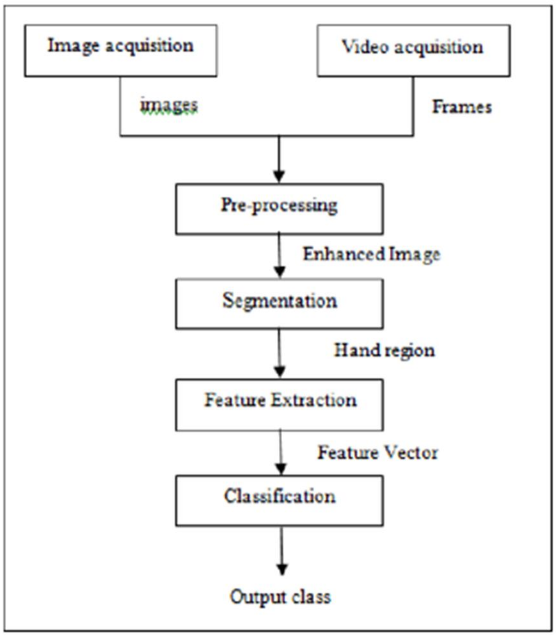

Fig 2: Proposed Model for Sign Detection System

The proposed model for Sign Detection System are:

A. Data Collection

The first step toward the deployment of the sign language detection system is the ordered acquisition of gesture data. Acquisition is made easy through the collection_data.py script, a building block meant to acquire images and video recordings of different sign language gestures, each appropriately labeled. Through its connection to a webcam or other types of video input devices, the script allows for live capture of user-executed gestures. Since gestures are recorded, the script breaks down the video stream into separate frames, where each frame represents a single instant in the sequence of gestures. These frames are subsequently categorized into directories, and each directory corresponds to a distinct sign language label. This labeled and organized dataset is the basis for further steps of pre-processing, model training, and evaluation to provide consistency and integrity in the learning process.

B. Data Pre-processing and Feature Extraction

After the data gathering stage, raw images are re-formatted to an optimized format for computational and model compatibility via the data.py script. This script re-converts image data into a NumPy array format (.npy files), which is one of the machine learning pipeline standards because it supports numerical computation with high efficiency. The pre-processing workflow entails loading the images from labelled folders, resizing them to a uniform size, and normalizing pixel values. The images are then converted into NumPy arrays, maintaining spatial characteristics while avoiding compactness and uniformity. These arrays are then organized into datasets, with each dataset paired with a matching gesture label. The dataset is split into training, validation, and test sets to facilitate model generalization and efficient evaluation. A Model Training and Evaluation integral part of the pre-processing script is the implementation of MediaPipe for hand landmark detection. The system uses OpenCV for video processing, NumPy for mathematical operations, and Media Pipe for obtaining (x, y, z) coordinates of hand landmarks. The detection pipeline contains methods for real-time prediction and display of landmarks, frame conversion from BGR to RGB, processing of frames, and reversing them for display. If landmarks are found, their coordinates are projected into a single feature vector; otherwise, an equal-length zero vector is returned. This formal method of feature extraction guarantees the consistency of input data during model training, translating complex gestures into useful, machine-readable representations.

C. Model Training and Evaluation

After the dataset is preprocessed, the subsequent key step is model training, which is undertaken by the train_model.py script. This script takes care of building, training, and testing a deep learning model, in this case, a Recurrent Neural Network (RNN) variant designed for sequence learning. Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) layers are normally used because they excel in modeling temporal dependencies found in gesture sequences. The model structure is established with input layers that handle sequences of feature vectors, hidden layers that extract temporal patterns, and output layers that classify gestures into pre-defined categories. Some key configurations are the application of softmax activation functions the Adam optimizer, and the categorical cross-entropy loss function, which is highly appropriate for multi-class classification. After training is complete, the model is tested using the test dataset.

IV. Results and Discussion

Sign language recognition system is to allow accurate and real-time identification of a pre-defined vocabulary of sign language gestures to make intuitive human-computer interaction through the interpretation of visual gestures as meaningful textual or spoken outputs. The system is designed to handle real-time video feed input from a camera, automatically identifying and labeling gestures like "Hello," "I Love You," "Yes," "Thank You," and "OK" in minimal delay while providing an integrated and intuitive user experience. Both high accuracy of recognition and high reliability are primary performance targets, with the system designed to have minimal false negatives and positives with various users and settings. For improving usability, the system has instant visual feedback through presenting the recognized gesture label on the display screen and can also superimpose hand landmarks or gesture boundaries on the video stream for improved transparency. In addition, recognized gestures are converted to textual and optionally verbal outputs through text-to-speech functionality, providing multimodal communication and accessibility. The system is also capable of recognizing a broad set of gestures, beyond a minimal subset, to provide greater linguistic coverage and flexibility for useful applications in a variety of communicative situations.

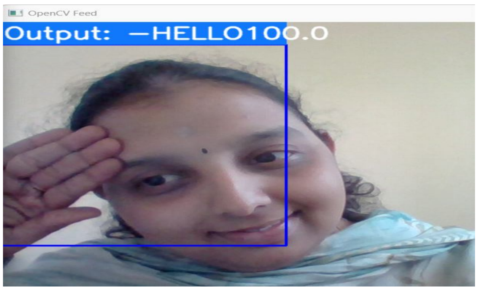

Fig 3: Hello sign detection

Fig 4: I LOVE YOU sign detection

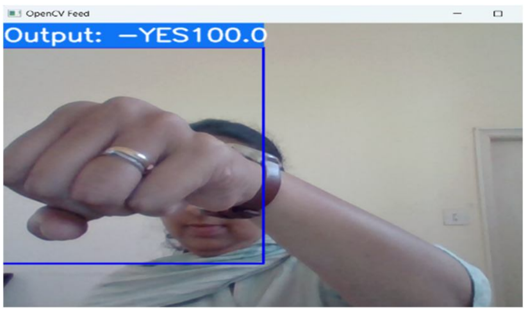

Fig 5: YES, sign detection

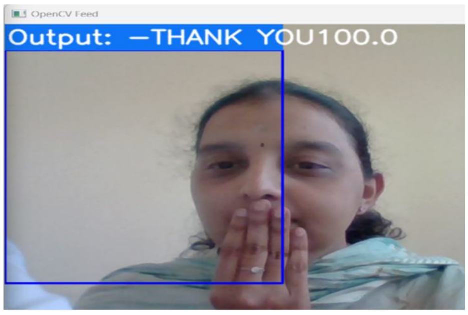

Fig 6: THANK YOU sign detection

V. Conclusion

Sign language recognition applications based on Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) models, constitutes a significant evolution in assistive technology for making communication more efficient for deaf and hard-of-hearing people. This initiative contributes to breaking the communication barrier in that it is able to do real-time translating of sign language into spoken words or written messages, thus being more inclusive and socially integrated. Its applications reach out to various fields such as healthcare, education, customer service, and work environment providing a scalable, cost-effective solution that can be adapted to sign languages.

REFERENCES

[1] Athavale, Y., Chavan, S., Gudadhe, S., Jain, A. (2021). Real-time Sign Language Detection Using Machine Learning. International Journal of Innovative Technology and Exploring Engineering, 10(5), 680-686.

[2] Ghaderzadeh, M., Czyz˙ewski, A., Palus, H. (2021). Hand Gesture Recognition Using Convolutional Neural Networks for Polish Sign Language. Sensors, 21(13), 4392.

[3] Majumder, A., Koushik, J. C. (2020). American Sign Language Detection Using Deep Learning Techniques. In Proceedings of the 2nd International Conference on Machine Learning and Soft Computing (pp. 167-175). Springer, Singapore.

[4] Mohan, S., Pal, A., Roy, S., Ramakrishnan, K. G. (2021). Sign Language Recog- nition System Using Deep Learning. In 2021 IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy (PESGRE) (pp. 1-6). IEEE.

[5] Perera, C. J., Jayasekara, P. P., Rajapaksha, H. N. (2021). Sign Language Recogni- tion using Convolutional Neural Network. In 2021 Moratuwa Engineering Research Conference (MERCon) (pp. 142-147). IEEE.

[6] Cai, Y., Guan, H., Lin, C., Zhao, X. (2021). Sign Language Recognition Using Deep Learning Techniques: A Review. IEEE Access, 9, 33545-33557.

[7] Huang, Y., Chen, Y., You, X. (2020). Sign Language Recognition Based on MultiModal Deep Learning. IEEE Transactions on Multimedia, 22(10), 2700-2712.

[8] Li, H., Huang, W., Zhang, X., Lao, S. (2020). Deep Learning for Sign Language Recognition and Translation: A Survey. IEEE Transactions on Human-Machine Systems, 50(4), 295-307.

[9] Liu, S., Ding, Y., Li, J., Tang, Y. (2021). A Novel End-to-End Sign Language Recognition System Based on Depth Image Sequences. IEEE Access, 9, 19235- 19244.

[10] Wang, S., Wei, Z., Wang, X., Gao, W. (2020). Sign Language Recognition via Attention-Based Temporal Convolutional Network. IEEE Transactions on Multi- media, 22(12), 3192-3203.

[11] Tan, X., Li, S., Wang, J. (2021). Sign Language Recognition Using Attention-Based Deep Neural Networks. IEEE Access, 9, 55518-55527.

[12] Feng, Z., Zhang, Z., Wu, Y. (2020). Sign Language Recognition Based on 3D Convolutional Neural Networks. IEEE Access, 8, 105759-105770.

[13] Deng, W., Wu, W., Hu, Z. (2021). Real-Time Sign Language Recognition Based on MobileNetV2 and ConvLSTM. IEEE Access, 9, 64790-64802.

[14] Liu, L., Zhang, Y., Zhang, S. (2020). Sign Language Recognition with Multi-View Graph Convolutional Networks. IEEE Transactions on Image Processing, 29, 6449- 6461.

[15] Kim, S., Oh, S., Kim, Y. (2021). Real-Time Korean Sign Language Recognition Using Deep Convolutional Neural Networks. Sensors, 21(7), 2552.