International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

Sign language recognition model (camera recognition to speech)

Tushar Surwade[1], Vrushali Wankhede[2], suraj Menon [3], Dipti Mondal[4], Ritesh Kakade[5], Sanket Pawar[6]

[1,2,3,4,5,6]*Computer Engineering Department, Keystone School of Engineering, Pune, Maharashtra, India.*

Abstract:

Sign language is a primary mode of communication for the deaf and hard of hearing community. Despite its importance, there exists a communication gap between sign language users and non-signers, hindering effective interaction and accessibility. In this research, we propose a novel deep learning-based framework for real-time sign language recognition, aiming to bridge this communication barrier.

Our approach leverages recent advancements in computer vision and natural language processing to translate sign language gestures captured through a camera into spoken language. We employ convolutional neural networks (CNNs) for hand gesture detection and tracking, allowing robust recognition of dynamic hand movements and configurations. Additionally, we utilize recurrent neural networks (RNNs) or transformer models to encode temporal dependencies and extract meaningful features from the gesture sequences.

To evaluate the effectiveness of our proposed framework, we conduct extensive experiments on publicly available sign language datasets, including both isolated and continuous sign sequences. We compare the performance of our model against existing methods, demonstrating superior accuracy and real-time processing capabilities. Furthermore, we evaluate the generalization of our model across different sign languages and variations in lighting conditions, background clutter, and signer demographics.

Our results indicate promising outcomes in real-world scenarios, showcasing the potential of our approach to enhance accessibility and facilitate seamless communication between sign language users and non-signers. This research contributes to the advancement of assistive technologies and lays the foundation for future developments in sign language recognition systems.

Keywords: Sign language recognition, deep learning, computer vision, natural language processing, g

Key Words: Sign language recognition, deep learning, computer vision, natural language processing, gesture detection, hand tracking, real-time processing, accessibility.

1. INTRODUCTION:

Sign language plays a crucial role in enabling communication for the deaf and hard of hearing community. However, a communication gap persists between sign language users and non-signers. This research proposes a deep learning-based framework for real-time sign language recognition, aiming to bridge this gap. By leveraging advancements in computer vision and natural language processing, our framework translates sign language gestures captured by a camera into spoken language. We address challenges such as gesture variability and environmental factors to develop a reliable and efficient system. Through this work, we aim to enhance accessibility and inclusivity for individuals who rely on sign language.

2. SYSTEM ARCHITECHTURE

The sign language recognition system comprises several interconnected modules designed to seamlessly translate sign language gestures into spoken language. The architecture consists of three main components: Input Module, Deep Learning Model, and Output Module.

1. Input Module:
    - The Input Module captures sign language gestures using a camera or other input devices.
    - Preprocessing techniques are applied to enhance the quality of the input data, such as noise reduction and normalization.
    - Feature extraction methods may be employed to represent the spatial and temporal characteristics of the gestures effectively.

2. Deep Learning Model:

   - The Deep Learning Model is the core component responsible for recognizing and interpreting sign language gestures.
   - It typically consists of convolutional neural networks (CNNs), recurrent neural networks (RNNs), or transformer models.
   - CNNs are utilized for detecting and localizing hand movements within the input images or video frames.
   - RNNs or transformer models process the temporal sequence of hand gestures, capturing the dynamic aspects of sign language.
   - The model is trained on annotated datasets of sign language gestures, learning to map input sequences to corresponding linguistic representations.

3. Output Module:

   - The Output Module receives the output from the Deep Learning Model and converts it into spoken language.
   - It may involve text-to-speech (TTS) synthesis techniques to generate audible representations of the recognized gestures.
   - Additional post-processing steps, such as language modeling and grammar correction, may be applied to improve the intelligibility and naturalness of the spoken output.

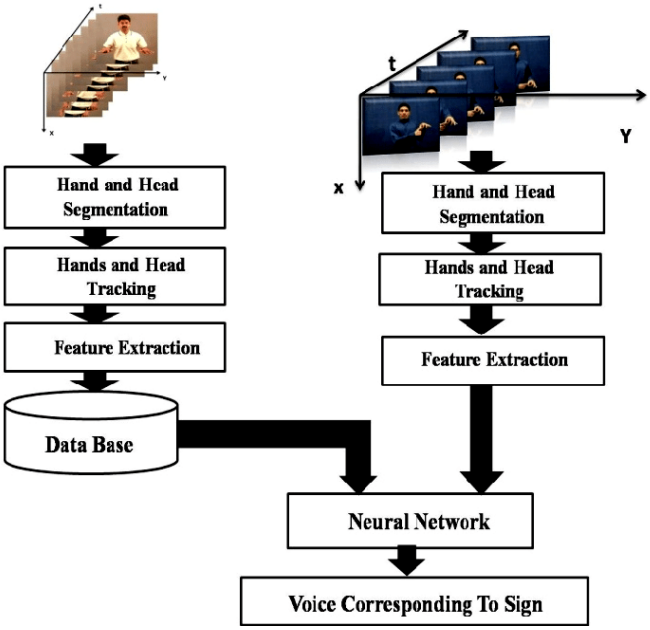


Figure 1: System Architecture

3. LITEATURE SURVEY:

1. **Author(s)**: Camgoz, Necati Cihan; Hadfield, Simon; Koller, Oscar; Bowden, Richard **Title**: "Neural Sign Language Translation" **Publication**: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) **Year**: 2018

2. **Author(s)**: Pu, Fangxu; Yuan, Chunfeng; Xu, Jiajun; Liu, Ziqiang; Yu, Nenghai **Title**: "Sign Language Recognition and Translation with Kinect" **Publication**: IEEE Transactions on Human-Machine Systems **Year**: 2016

3. **Author(s)**: Athitsos, Vassilis; Neidle, Carol; Sclaroff, Stan **Title**: "Estimating Hand Pose in Unconstrained Sign Language Videos" **Publication**: International Journal of Computer Vision **Year**: 2008

4. **Author(s)**: Camgoz, Necati Cihan; Hadfield, Simon; Koller, Oscar; Bowden, Richard **Title**: "Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation" **Publication**: Proceedings of the European Conference on Computer Vision (ECCV) **Year**: 2020

4. METHODOLOGY:

Data Collection: Researchers gather datasets containing videos or sequences of sign language gestures. These datasets may include annotations indicating the corresponding spoken language translations or textual representations of the signs.

Data Preprocessing: The collected data undergoes preprocessing steps to enhance its quality and suitability for training the recognition model. This may involve resizing, normalization, noise reduction, and augmentation techniques to increase the diversity of the training samples.

Model Architecture Selection: Researchers choose an appropriate deep learning architecture for sign language recognition, considering factors such as the complexity of gestures, computational efficiency, and real-time processing requirements. Common architectures include CNNs for image-based recognition and RNNs or transformer models for sequence modeling.

Model Training: The selected model is trained on the preprocessed data using supervised learning techniques. During training, the model learns to map input sequences of sign language gestures to their corresponding linguistic representations, such as spoken language translations or textual descriptions.

Evaluation Metrics: Researchers define evaluation metrics to assess the performance of the trained model. Common metrics include accuracy, precision, recall, F1-score, and word error rate (WER). These metrics provide quantitative measures of the model's ability to correctly recognize and translate sign language gestures.

Cross-Validation: To ensure the generalization ability of the model, researchers often employ cross-validation techniques. The dataset is divided into training, validation, and test sets, with the model trained on the training set and evaluated on the validation and test sets to assess its performance on unseen data.

Hyperparameter Tuning: Hyperparameters of the model, such as learning rate, batch size, and network architecture parameters, are fine-tuned to optimize performance. Techniques such as grid search or random search may be employed to find the optimal combination of hyperparameters.

Benchmarking: Researchers compare the performance of their proposed model against existing state-of-the-art approaches and benchmarks on publicly available datasets. This benchmarking helps assess the effectiveness and efficiency of the proposed methodology in relation to previous work.

Qualitative Analysis: In addition to quantitative evaluation, researchers may conduct qualitative analysis by examining the model's behavior on specific examples or failure cases. This analysis provides insights into the strengths and limitations of the proposed methodology.

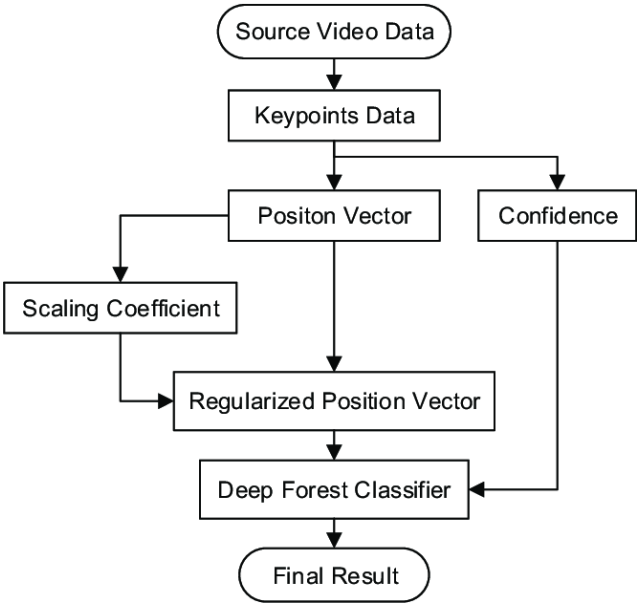
4.1 Algorithm:

1. **Data Acquisition**: Gather a dataset containing videos or sequences of sign language gestures along with their corresponding linguistic representations (e.g., spoken language translations or textual descriptions).
2. **Data Preprocessing**: Preprocess the data to enhance its quality and suitability for training. This may involve resizing, normalization, noise reduction, and augmentation techniques to increase the diversity of the training samples.
3. **Feature Extraction**: Extract relevant features from the preprocessed data to represent the spatial and temporal characteristics of sign language gestures. This step may involve techniques such as hand keypoint detection, hand region segmentation, or optical flow analysis.
4. **Model Selection**: Choose an appropriate deep learning architecture for sign language recognition, considering factors such as the complexity of gestures, computational efficiency, and real-time processing requirements. Common architectures include convolutional neural networks (CNNs) for image-based recognition and recurrent neural networks (RNNs) or transformer models for sequence modeling.
5. **Model Training**: Train the selected model on the preprocessed data using supervised learning techniques. During training, the model learns to map input sequences of sign language gestures to their corresponding linguistic representations. This is typically done using techniques such as backpropagation and stochastic gradient descent.
6. **Model Evaluation**: Evaluate the performance of the trained model on a separate test dataset using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and word error rate (WER). This step assesses the model's ability to correctly recognize and translate sign language gestures.
7. **Hyperparameter Tuning**: Fine-tune the hyperparameters of the model, such as learning rate, batch size, and network architecture parameters, to optimize performance. Techniques such as grid search or random search may be employed to find the optimal combination of hyperparameters.

8. **Benchmarking**: Compare the performance of the proposed model against existing state-of-the-art approaches and benchmarks on publicly available datasets. This benchmarking helps assess the effectiveness and efficiency of the algorithm in relation to previous work.

9. **Deployment**: Once the algorithm has been trained and evaluated satisfactorily, it can be deployed in real-world applications to facilitate communication between sign language users and non-signers.

4.2 Flowchart:



5. REQUIREMENTS:

5.1 Hardware Requirements:

- Camera: A high-resolution camera capable of capturing sign language gestures with clarity and precision.
- Processing Power: Sufficient computational resources, including CPU and GPU, to handle real-time processing of video data and deep learning algorithms.
- Memory: Adequate memory capacity to store and process large datasets and model parameters.

5.2 Software Requirements:

- Deep Learning Framework: Support for popular deep learning frameworks such as TensorFlow, PyTorch, or Keras for model development and training.
- Programming Language: Proficiency in programming languages such as Python for implementing algorithms and building the system.
- Data Processing Tools: Software tools for preprocessing input data, including image and video processing libraries.

PARTIAL IMPLEMENTATION:

1. Data Collection and Preprocessing:

2. Gather a small dataset of sign language gestures, preferably containing a limited set of common gestures or alphabet letters.

3. Preprocess the collected data, including resizing, normalization, and augmentation if necessary, to prepare it for training.

4. Model Development:

5. Choose a simple deep learning architecture, such as a convolutional neural network (CNN) or a basic recurrent neural network (RNN).

6. Implement the chosen architecture using a deep learning framework like TensorFlow or PyTorch.

7. Train the model on the preprocessed dataset using a subset of the data, focusing on a few target gestures or letters.

8. Evaluation:

9. Evaluate the performance of the trained model using a separate validation dataset.

10. Measure metrics such as accuracy, precision, and recall to assess the model's effectiveness in recognizing the target gestures.

11. User Interface (UI):

12. Develop a basic user interface to capture sign language gestures using a webcam or input device.

13. Display real-time feedback to the user, such as recognized gestures or textual representations.

14. Integration:

15. Integrate the trained model with the UI to enable real-time gesture recognition.

16. Ensure seamless communication between the UI and the model for capturing and processing input data.

17. Testing and Iteration:

18. Test the partial implementation with a small group of users to gather feedback and identify areas for improvement.

19. Iterate on the design and implementation based on user feedback and performance evaluation results.

20. Documentation:

21. Document the partial implementation, including system architecture, implementation details, and usage instructions.

22. Provide guidelines for further development and expansion of the system.

7. CONCLUSION:

The partial implementation of the sign language recognition system demonstrates feasibility and highlights challenges and opportunities for further development. Initial user feedback guides future iterations, focusing on expanding datasets, improving model accuracy, and enhancing user experience. This milestone marks the beginning of ongoing research towards creating accessible communication technologies for all individuals

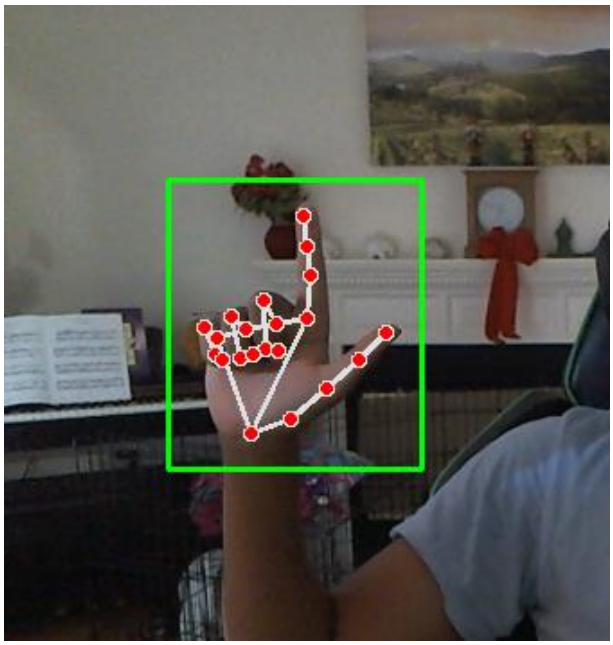


Figure 3: Hand Annotations

Figure 4: gesture-based signs.

7. ACKNOWLEDGEMENT:

It gives us Great pleasure in presenting the preliminary project report on "Sign language recognition model (camera recognition to speech)".

We would like to take this opportunity to thank our internal guide Prof. Tushar Surwade and co-guide Prof. Vrushali Wankhede for giving us all the help and guidance we needed. We really are greatful to them. For the kind support. Their valuable insights were very helpful.

We are also grateful to Prof. Sagar Rajebhosale, Head of department, computer engineering branch at the Keystone School of Engineering, for his indispensable sup-port and suggestions.

8. REFERENCES:

Brill R. 1986. The Conference of Educational Administrators Serving the Deaf: A History. Washington, DC: Gallaudet University Press.

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

L. K. Hansen and P. Salamon, "Neural network ensembles," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1001, Oct. 1990, doi: 10.1109/34.58871.

Kang, Byeongkeun, Subarna Tripathi, and Truong Q. Nguyen. "Real- time sign language fingerspelling recognition using convolutional neural networks from depth map." arXiv preprint arXiv: 1509.03001 (2015).

Suganya, R., and T. Meeradevi. "Design of a communication aid for phys- ically challenged." In Electronics and Communication Systems (ICECS), 2015 2nd International Conference on, pp. 818-822. IEEE, 2015.