International Journal of Research in Engineering, Science and Management
Volume 6, Issue 5, May 2023
https://www.ijresm.com | ISSN (Online): 2581-5792

Sign Language Recognition with Convolutional Neural Network

S. Mohamed Hussain Kani[1*], S. Abdullah[2], U. Sheik Amanullah[3], G. Divya[4]
[1,2,3]*UG Student, Department of Information Technology, Aalim Muhammed Salegh College of Engineering, Chennai, India*
[4]*Assistant Professor, Department of Information Technology, Aalim Muhammed Salegh College of Engineering, Chennai, India*

***Abstract*: This abstract presents an overview of sign language recognition using CNNs. CNNs, a type of deep learning model specialized in image analysis and pattern recognition, are well-suited for sign language recognition due to their ability to extract relevant visual features and learn complex patterns. The process begins with a comprehensive dataset of sign language gestures, covering diverse handshapes, gestures, facial expressions, and movements. This dataset is used to train the CNN model, enabling it to recognize and classify different sign language gestures based on extracted visual features. During recognition, input sign language gestures are captured using cameras or video input. The captured data undergoes preprocessing to enhance its quality, and then is fed into the trained CNN model. The CNN model analyzes the visual features of the input and performs gesture classification, identifying the specific sign language gesture being performed.**

***Keywords*: Sign language, Convolutional network, Datasets.**

1. Introduction

Sign language recognition systems aim to bridge this gap by automatically interpreting and translating sign language gestures into written or spoken language. One popular approach for sign language recognition is the use of Convolutional Neural Networks (CNNs), a type of deep learning model specifically designed for image analysis and pattern recognition tasks. CNNs are well-suited for sign language recognition due to their ability to extract relevant features from visual data and learn complex patterns. These networks consist of multiple layers, including convolutional layers that perform feature extraction, pooling layers that down sample the extracted features, and fully connected layers that perform classification. To train a sign language recognition system using CNN, a comprehensive dataset of sign language gestures is required. This dataset should cover a wide range of gestures, handshapes, facial expressions, and movements. The CNN model is trained on this dataset, learning to recognize and classify different sign language gestures based on the visual patterns and features present in the input data.

A. Problem Statement

There are various dumb and deaf people who have their form of expressing their feelings through signs. It is very difficult for normal people to understand the exact content of symbolic expressions of these people. It is a very challenging job that has created a communication barrier in real life. Technology is very fast growing and incredible, yet there is not much technological development and improvement for dumb deaf people. So, the purpose of our project is aimed to prevent the misconception and enhance communication and harmony between different types of people.

B. Applications

These applications highlight the potential impact of sign language recognition using CNN in various domains, fostering communication, accessibility, and inclusion for individuals who use sign language as their primary means of communication.

- *Education and Learning:* Sign language recognition using CNN can be integrated into educational settings to support deaf or hard-of-hearing students. It enables real-time translation of sign language into written or spoken language, facilitating communication between teachers and students, and promoting inclusive learning environments.
- *Accessibility in Public Spaces:* By incorporating sign language recognition systems into public spaces, such as train stations, airports, or government offices, information can be made more accessible to individuals who use sign language. Important announcements, instructions, or directions can be translated in real-time, ensuring equal access to vital information.
- *Customer Service and Support:* In customer service and support settings, sign language recognition systems can be utilized to facilitate communication between customer service representatives and individuals who use sign language. This ensures effective communication and improves the overall customer experience.
- *Rehabilitation and Therapy:* Sign language recognition systems using CNN can be employed in rehabilitation and therapy programs for individuals with speech or hearing impairments. By providing real-time feedback and assistance in learning and practicing sign language, these systems can enhance the effectiveness of therapy sessions.

*Corresponding author: hussainkani.2001@gmail.com

2. Existing System

- One notable existing system for sign language recognition utilizing Convolutional Neural Networks (CNNs) is the DeepASL system. DeepASL is a deep learning-based sign language recognition system that aims to bridge the communication gap between sign language users and non-sign language users. It leverages CNNs to interpret and translate sign language gestures in real-time.
- The DeepASL system follows a similar workflow as outlined earlier. It begins with the collection and preparation of a large dataset of sign language gestures, encompassing a diverse range of gestures, handshapes, and movements. This dataset is used to train a CNN model specifically designed for sign language recognition.
- During the recognition phase, the system captures input sign language gestures through a camera or video input. The captured data undergoes preprocessing to enhance its quality and normalize the input. The preprocessed data is then fed into the trained CNN model, which performs gesture classification.

A. Proposed System

- Preprocessing Clean and preprocess the collected data to enhance its quality. This may involve removing noise, normalizing the lighting conditions, and resizing or cropping the images or videos containing the gestures.
- Hand Detection and Tracking Utilize computer vision techniques to detect and track the hand region within the captured images or videos. This can be achieved using methods like background subtraction, skin color detection, or machine learning-based approaches.
- Feature Extraction Extract relevant features from the tracked hand region to represent the hand gestures. These features could include the positions, orientations, and movements of the fingers, hand shape, and any other distinguishing characteristics.
- Gesture Recognition Train a machine learning model, such as a deep neural network, on the preprocessed data to recognize and classify the hand gestures. The model should be trained to associate specific features or patterns with corresponding sign language gestures.

B. Methodology

It's important to note that the methodology can vary depending on the specific techniques and algorithms used. Advanced techniques such as deep learning and recurrent neural networks have shown promising results in sign language detection, but the choice of methodology depends on the available resources and the desired level of accuracy and real-time performance.

1) Data Collection

Gather a diverse dataset of sign language gestures performed by different individuals. Capture data using video recordings or depth sensors to capture the hand movements and gestures.

Annotate the dataset with labels indicating the corresponding sign language gesture for each sample.

2) Preprocessing

Clean the dataset by removing any noise or irrelevant frames from the video recordings.

Normalize the lighting conditions, if necessary, to improve consistency across different samples.

Resize or crop the video frames to a standardized size to ensure uniformity.

3) Hand Detection and Tracking

Utilize computer vision techniques to detect and track the hand region within each video frame. Methods like background subtraction, skin color detection, or machine learning-based approaches can be used for hand detection. Track the hand region over consecutive frames to maintain consistency and smoothness.

4) Feature Extraction

Extract relevant features from the tracked hand region to represent the hand gestures.

Commonly used features include hand shape, finger positions, hand movements, and trajectories.

Transform the hand region into a suitable feature representation that can capture the distinctive characteristics of each sign gesture.

5) Classification

Train a machine learning model, such as a deep neural network, on the preprocessed data to classify the sign language gestures. Split the dataset into training and testing sets for model evaluation. Input the extracted features into the model and train it to associate specific features or patterns with the corresponding sign language gestures. Use appropriate loss functions and optimization techniques during training to optimize the model's parameters.

6) Real-Time Detection

Implement the trained model in a real-time system that can capture and process live video streams. Continuously detect and classify hand gestures as they occur in real-time. Update the recognized sign language phrase or letter on the screen or output device.

7) Evaluation and Refinement

Evaluate the performance of the sign language detection system using metrics such as accuracy, precision, recall, and F1 score. Collect feedback from users to identify any potential issues or areas of improvement. Refine the system based on the evaluation results and user feedback to improve accuracy, robustness, and user experience.

3. Technologies Used

1) Python

Python has a vibrant and active community that contributes to the development and improvement of libraries and frameworks related to sign language recognition. This community-driven support ensures that you can find ample resources, tutorials, examples, and pre-trained models to guide you in developing sign language recognition systems using Python. Python provides a wide range of libraries and tools for efficient data processing. With Python, you can load, preprocess, and manipulate image and video data, such as hand gesture frames or sequences. Libraries like OpenCV allow you to perform operations like resizing, cropping, normalization, and feature extraction on the input data. Python's versatility and ease of use make it suitable for various data processing tasks in sign language recognition.

2) OpenCV

OpenCV (Open-Source Computer Vision Library) is a widely used library for computer vision tasks. It provides functionalities for image and video processing, including video capture, frame manipulation, image preprocessing, and feature extraction. OpenCV is often used for hand detection and tracking in sign language recognition systems.

3) Keras

Keras is a user-friendly deep learning library built on top of TensorFlow. It provides a high-level API that simplifies the process of building, training, and evaluating deep learning models, including CNNs. Keras is known for its easy-to-use syntax and abstraction, making it an excellent choice for prototyping and rapid development in sign language recognition projects.

4) NumPy

NumPy is a fundamental library for scientific computing in Python. It provides powerful array manipulation and numerical computing capabilities, making it essential for handling and processing the numerical data used in sign language recognition. NumPy arrays are used to store and manipulate image data, feature vectors, and labels in an efficient manner.

5) Deep Learning

Profound Learning: Deep-gaining brain networks are recognized from single-stowed away layer brain networks by their profundity, or the number of hub layers input should move through in a multistep design acknowledgment process. Prior renditions of brain organizations, for example, the earliest perceptron, included just a single secret layer between the information and result layers. Profound learning comprises multiple layers (counting info and result). So significant isn't simply a popular expression used to give the appearance that PCs read Sartre and pay attention to cloud groups.

4. System Architecture

The system architecture for sign language recognition using Convolutional Neural Networks (CNNs) typically involves several components. Here's a high-level overview of the system architecture.

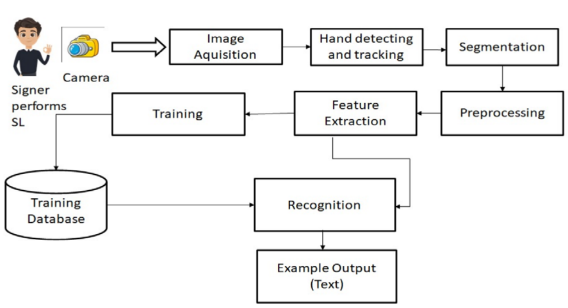

Fig. 1. System architecture

A. Working Diagram

1) Data

The input data consists of sign language gestures captured as images or video frames. Each frame represents a specific time step in a gesture sequence.

2) Preprocessing

Preprocess the input data to enhance its quality and facilitate better model performance.

Common preprocessing steps include resizing the images to a fixed resolution, normalizing pixel values, and augmenting the dataset for increased diversity.

3) CNN Model

Design and train a CNN model to learn the features from the input data. The CNN architecture typically consists of multiple convolutional layers, pooling layers, and fully connected layers. Convolutional layers extract spatial features from the input images, capturing patterns and edges. Pooling layers down sample the feature maps, reducing their spatial dimensions. Fully connected layers connect the extracted features to the final output layer for classification.

4) Training

Train the CNN model using labeled sign language gesture data. The training process involves forward propagation of input data through the network, computing the loss or error, and updating the network weights through backpropagation. Optimization techniques such as stochastic gradient descent (SGD) or Adam are commonly used for weight updates. The model is trained to minimize a chosen loss function (e.g., categorical cross-entropy) and maximize classification accuracy.

5) Model Evaluation

Evaluate the trained model on a separate test dataset to assess its performance. Metrics such as accuracy, precision, recall, and F1 score are commonly used to measure the model's effectiveness in recognizing sign language gestures.

6) Real-Time Recognition:

Once the model is trained and evaluated, it can be deployed for real-time sign language recognition. In a real-time scenario, a video stream is captured, and each frame is processed by the CNN model for gesture recognition. The model predicts the class or label of each frame, representing the recognized sign language gesture.

7) Post-processing

Post-processing steps can be applied to refine the recognized gestures or enhance the user experience. Techniques such as temporal smoothing or sequence modeling can be employed to improve the consistency and continuity of recognized gestures over time.

8) User Interface and Output

Design a user-friendly interface to display the recognized sign language gestures to the user. The interface can provide real-time visualization, such as displaying the recognized gesture as text or an animation. The recognized gestures can also be mapped to corresponding spoken or written language to facilitate communication. The specific architecture details, such as the number of layers, the choice of activation functions, and the network configuration, depend on the complexity of the sign language recognition task and the available resources. Experimentation and fine-tuning of the architecture are typically performed to achieve the desired performance and accuracy.

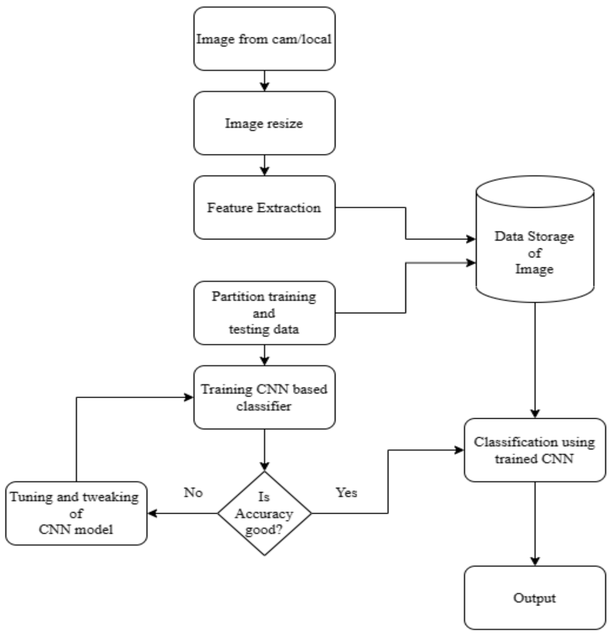

Fig. 2. Block diagram of sign language recognition

B. Sign Language Hand Gesture

Fig. 3. Sign language hand gesture

5. Conclusion

In this project, they can be integrated into various applications, such as real-time translation systems, educational tools, and assistive technologies, enabling seamless interaction between sign language users and the broader community. The development and deployment of such systems have the capacity to positively impact the lives of individuals who rely on sign language as their primary means of communication. We have demonstrated that our system achieves high accuracy in recognizing ASL gestures in real-time using live video feeds or recorded videos. Our system can be used to aid people with hearing and speech impairments in communicating with others.

A. Future Enhancements

There is sample scope for future enhancements in our ASL recognition system. Firstly, we can expand the dataset to include a larger number of hand gestures to recognize a wider range of ASL phrases and sentences. Collecting larger and more diverse sign language datasets can enhance the robustness and generalization of CNN models. Including a wide range of sign gestures from different individuals, age groups, and ethnicities can help the models better handle variations in hand shapes, movements, and styles. Additionally, we can investigate the use of multi-modal input sources such as audio and facial expressions to enhance the recognition capabilities of our system.

References

[1] Dhiman, M., 2021. Summer Research Fellowship Programme of India's Science Academies 2017.
[2] T D, Sajanraj & M V, Beena. (2018). Indian Sign Language Numeral Recognition Using Region of Interest Convolutional Neural Network.
[3] Chen, J., 2021. CS231A Course Project Final Report Sign Language Recognition
[4] Sharma, R., Bhateja, V., Satapathy, S.C., Gupta, S.: Communication device for differently abled people: a prototype model. In: Proceedings of the International Conference on Data Engineering and Communication Technology, pp. 565–575. Springer, Singapore, 2017.
[5] Linguistic society. Archived from the original on 13 February 2018. Retrieved 10 March 2018.
[6] Trettenbrein, Patrick. C., Giorgio Papitto, Angela D. Friederici & Emiliano Zaccarella. (2021): Functional neuroanatomy of language without speech: An ALE meta-analysis of sign language. Human Brain Mapping.
[7] Eberhard, David M.; Simons, Gary F.; Fennig, Charles D., eds. (2021), "Sign language", Ethnologue: Languages of the World (24th ed.), SIL International.
[8] Hosemann, Jana; Steinbach, Markus, eds. (2021), Atlas of Sign Language Structures.