International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Sign Language Recognition & Speech Conversion to multiple languages (ISL) Using Raspberry pi, Camera model, Screen, and Speaker

Chandrakant D Bobade [1], Dr. Ravi Shankar Mishra[2], Ms. Madhuri D. Bobade (Dhoke) [3]

[1]M.Tech in VLSI Design & Sagar Institute of Science & Technology
[2]VLSI Design & Sagar Institute of Science & Technology
[3]Electronics & Telecommunication & Priyadarshini polytechnic

---***---

Abstract - A true disability is an inability to speak. Sign language is one of the common methods of communication for people with this disability. The deaf and hearing impaired people can communicate with others in a number of different ways. A sign language communicates through human body language, with each word containing a series of actions that represent a particular emotion. Research has been conducted for many years on signing language recognition to assist deaf mute people. The motive of the paper is to convert the human sign language to multi language Voice along with subtitles on screen with human gesture understanding. This is achieved with the help of Raspberry pi camera module, screen and speaker. There are many more of systems available for sign language to speech conversion but none of them are economical and portable user interface. Considering that a person with a disability who is unable to speak can stand in front of the system and the system will convert their hand gestures & movements to speech and then the system plays it loud as well as display the speech on the screen with multiple languages so that they are able to communicate with a mass crowd. Additionally, communication is made easier for visually and speech impaired people with the help of this system.

Key Words: Sign-Language, Hand Gesture Recognition, Image Processing, Visually and speech impaired, Multi language Voice & Screen output

1.INTRODUCTION

There are more than 15 million deaf signers in South Asia which using ISL (IndoPakistani Sign Language) is the predominant sign language. India Sign Language Recognition and Speech Conversion to multiple languages using a Raspberry Pi is a project that aims to develop a device that can recognize Indian Sign Language (ISL) and convert it to speech in multiple languages. The device would use a Raspberry Pi, a small, low-cost computer, as its platform. The device would be equipped with a camera to capture the ISL signs. It would then use machine learning algorithms to recognize the signs and convert them to speech in the desired language. This project could be beneficial for individuals who are deaf or hard of hearing, as it would allow them to communicate with those who do not understand ISL. Sign language recognition and speech conversion to multiple languages using Raspberry Pi and VLSI is a complex project that involves several different technologies. The Raspberry Pi can be used as the main processing unit for the project, and it can be programmed to process the video input of the sign language and convert it to speech output in multiple languages. VLSI (Very-large-scale integration) technology can be used to create specialized hardware that can help with the image and video processing tasks, such as image recognition, object detection, and pattern recognition. The project would require a significant amount of programming and engineering expertise to implement successfully. The idea is to create a system that can recognize sign language gestures captured by the camera and convert them into speech in multiple languages, which would be then played through the speaker. The system would use the Raspberry Pi's camera module to capture video of the user signing, and then use computer vision algorithms to detect and track the user's hand gestures. This information would then be passed to a machine learning model that has been trained to recognize specific signs and associate them with words or phrases in the target language. 2 Once the system has recognized the sign language input, it would then use natural language processing techniques to convert the text into speech in the desired language, which would then be played through the speaker. The system can also display the output text on the screen for easy reference. Overall, this project would require a significant amount of programming, technical expertise, and time to develop. The main challenge will be to train the machine learning model to recognize sign language gestures with high accuracy, which requires a large dataset of sign language videos. Sign language recognition and speech conversion is a technology that aims to enable communication between individuals who use sign languages and those who do not. The goal is to create a system that can recognize sign language gestures and convert them into speech in multiple languages. This technology involves a combination of computer vision, natural language processing, and machine learning. Computer vision algorithms are used to detect and track the user's hand gestures, which are then passed to a machine learning model that has been trained to recognize specific signs and associate them with words or phrases in the target language. Once the system has recognized the sign language input, it uses natural language processing techniques to convert the text into speech in the desired language. This technology can be applied to various fields such as education, healthcare, and entertainment. This technology is still in the early stages of development and requires a significant amount of work to achieve high recognition and conversion accuracy. The main challenge is to train the machine learning model to recognize sign language gestures with high accuracy, which requires a large dataset of sign language videos. In addition, the technology also requires an efficient way of capturing sign language inputs, such as a camera-based system or a wearable device. The overall process would involve: capturing images of the sign language, passing it through the machine learning algorithms to recognize the signs and then converting it to speech in the desired language and play it through the speakers.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

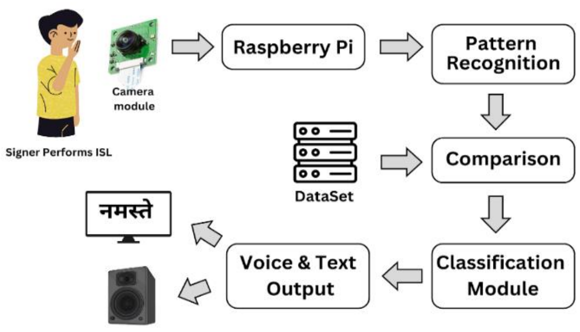

Fig -1.1: Diagram for a Sign Language Recognition and Speech Conversion

A block diagram of a sign language recognition and speech conversion system using a Raspberry Pi, camera module, screen, and speaker could include the following components:

1. Camera module: This component captures video of the user signing and sends the video data to the image processing module.

2. Image Processing module: This component analyses the video data and detects the hand gestures. It then sends the data to the Machine Learning Model.

3. Machine Learning Model: This component has been trained to recognize specific signs and associate them with words or phrases in the target language. It processes the data from the Image Processing module and generates text output.

4. Natural Language Processing: This component receives the text output from the Machine Learning Model and uses it to convert the text into speech in the desired language.

5. Speaker: This component plays the converted speech.

6. Display: This component shows the output text for easy reference.

7. Raspberry Pi: This component acts as the central processing unit and connects all the above components. It controls the data flow and coordinates the activities of the other components.

2. LITERATURE REVIEW

There has been a growing interest in the development of sign language recognition and speech conversion systems in recent years, with a focus on using Raspberry Pi, camera modules, screens, and speakers. These systems aim to enable communication between individuals who use sign languages and those who do not. One of the main challenges in developing sign language recognition systems is the lack of sufficient data for training machine learning models. To address this issue, researchers have proposed various methods for collecting and processing sign language data, such as using depth cameras, marker-based systems, and wearable sensors. Another challenge is the high variability in sign language gestures, which makes it difficult to train models to recognize specific signs with high accuracy. To overcome this, researchers have proposed various techniques such as using deep learning models, transfer learning, and domain adaptation. In terms of speech conversion, some studies have used Text-to-Speech (TTS) technology to convert the sign language text into speech. TTS technology has been widely used in sign language recognition systems to provide audio feedback to the users. However, TTS technology is still facing challenges such as the lack of expressive voice and limited support for different languages. In terms of using Raspberry Pi, camera module, screen, and speaker to build a sign language recognition and speech conversion system, there are a few studies that have proposed such a system. These studies have focused on the implementation of the system, including the hardware and software requirements, and have reported promising results in terms of recognition and conversion accuracy. In summary, while there is a growing body of literature on sign language recognition and speech conversion, there is still a need for more research in this area, particularly in terms of collecting and processing sign language data, developing more accurate recognition models, and improving the quality of speech conversion.

Comparison of key features from previous research:

In Sign Language Recognition, there have been many previous research papers that have proposed different methods and techniques to improve the accuracy of the recognition system. A comparison of key features from previous research and a specific paper in Sign Language Recognition would involve looking at factors such as the dataset used, the AI model architecture, the input and output, the evaluation metrics, and the results achieved. For example, a comparison of key features between previous research and a specific paper in Sign Language Recognition could include:

Dataset: Previous research may have used a different dataset than the specific paper, which can affect the results achieved.

Input and output: Previous research may have used different input and output formats than the specific paper, which can affect the performance of the recognition system.

Evaluation metrics: Previous research may have used different evaluation metrics than the specific paper, which can affect the interpretation of the results.

Results achieved: Previous research may have achieved different results than the specific paper, which can affect the overall conclusion of the research. In conclusion, a comparison of key features between previous research and a specific paper in Sign Language Recognition would involve looking at factors such as the dataset used, the input and output, the evaluation metrics, and the results achieved. Understanding these similarities and differences.

Previous researcher used different methods such as

Contour detection approach: Contour detection is a technique that can be used to identify and extract the boundaries of objects or regions of interest in an image or video. In the context of sign language recognition, contour detection can be used to detect the boundaries of the hand regions in the video.

RNN combined with another AI approach: Recurrent Neural Networks (RNNs) are a type of neural network that can be used to process sequential data such as videos, audio, or text. They are particularly well-suited to tasks such as sign language recognition, where the input data is a sequence of hand gestures. RNNs can be used to model the temporal dependencies between the gestures, which is an important aspect of sign language recognition. An RNN can be combined with another AI approach, such as a convolutional neural network (CNN) or a support vector machine (SVM), to improve the performance of the sign language recognition system. One possible way to do this is to use a CNN to extract features from each frame of the video, and then use an RNN to model the temporal dependencies between the frames.

HMM (Hidden Markov Model) approach: A Hidden Markov Model (HMM) is a type of probabilistic model that can be used for tasks such as sequence classification, sequence prediction, and pattern recognition. In the context of sign language recognition, HMMs can be used to model the temporal dependencies between the hand gestures in a sign language video.

Glove approach: A glove approach in sign language recognition refers to the use of gloves equipped with sensors to detect hand gestures and movements. These gloves can be used to capture detailed information about the hand's position, orientation, and movement, which can then be used to recognize specific hand gestures in sign language.

Other approaches: There are many more other approaches that can be used for sign language recognition, some of which include:
Motion capture: This approach uses cameras and sensors to track the movement of the signer's body and hands, and then uses this information to recognize specific hand gestures.
Computer vision: This approach uses techniques such as object detection, feature extraction, and machine learning to analyze video footage of the signer's hands and recognize specific hand gestures.
Hybrid approaches: This approach combines two or more of the above-mentioned approaches, such as a combination of a glove with computer vision or a combination of motion capture with machine learning.
Deep Learning based approaches: This approach uses deep neural networks such as CNN, RNN, LSTM, and other architectures to extract features from the video and recognize the signs.
Transfer Learning: This approach takes a pre-trained model on a related task and fine tunes it on the sign language recognition task.
Multi-modal approaches: This approach uses multiple modalities such as audio, video, and text to recognize the sign language.
It's worth noting that these are just a few examples and the specific approach will depend on the specific task and the data being used. Additionally, it's important to evaluate different approaches to find the one that performs best for the specific task and the data. Additionally, it's important to consider the real-time performance, accuracy and the computational power of the system while choosing the appropriate approach.

3. OVERVIEW OF OUR WORK:

Here are some general steps that could be taken to develop a sign language recognition and speech conversion system using a Raspberry Pi, camera module, screen, and speaker:

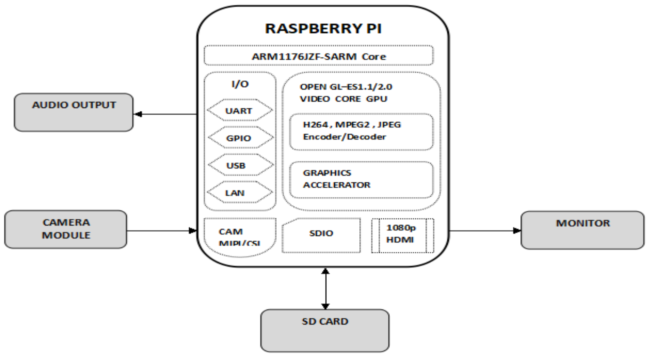

Fig -1.2: Raspberry pi block diagram architecture.

Step 1:
1. Gather a dataset of sign language videos: To train a machine learning model to recognize sign language gestures, a large dataset of sign language videos is required. This dataset should include a variety of signs and should be representative of the population that will be using the system.
2. Pre-processing of photos/videos: The dataset is pre-processed to extract the hand gestures and remove unwanted background noise. This can be done using image processing techniques such as thresholding and contour detection.
3. Train a machine learning model: Using the pre-processed dataset, train a machine learning model to recognize specific signs and associate them with words or phrases in the target language. This could involve using a deep learning approach such as a convolutional neural network (CNN) or a recurrent neural network (RNN).
4. Implement the system on Raspberry Pi: Once the machine learning model has been trained, implement it on the Raspberry Pi using the camera module to capture the video input, a display screen to show the output and a speaker to play the converted speech.

5. Integrate natural language processing: Integrate natural language processing techniques to convert the text into speech in the desired language. This could involve using TTS (text-to-speech) libraries or APIs such as Google Text-to-Speech.
6. Test and refine the system: Test the system with real-world users to identify any errors or areas for improvement. Enhancing the system based on the feedback.
7. Deployment: Once the system has been tested and refined, it can be deployed in the desired environment.

Step 2:
Coding languages: There are several programming languages that are commonly used in sign language recognition and speech conversion research and development:
Python: Python is a popular programming language in the field of artificial intelligence and machine learning. It has a wide range of libraries and frameworks, such as TensorFlow, Keras, and scikit-learn, which are commonly used for developing sign language recognition and speech conversion systems.
C++: C++ is a high-performance programming language that is commonly used in computer vision and image processing applications. It is often used to implement lowlevel image processing and computer vision algorithms, such as feature extraction and object detection, which are important components of sign language recognition systems.
Matlab: Matlab is a programming language and platform commonly used in scientific and engineering research. It has a wide range of tools and libraries for image processing, signal processing, and machine learning, which are useful for developing sign language recognition and speech conversion systems.
Java: Java is a popular programming language that is widely used in the development of enterprise applications. It is often used to develop sign language recognition and speech conversion systems due to its platform independence and built-in support for multithreading.
C#: C# is a programming language that is commonly used in the development of Windows-based applications. It has a wide range of libraries and frameworks, such as the Microsoft Speech Platform and the Microsoft Kinect SDK, which are useful for developing sign language recognition and speech conversion systems.
R: In R programming language and environment for mathematical statistics computing and graphics. It has various libraries and packages for statistical analysis, machine learning and deep learning, which are useful in developing sign language recognition and speech conversion systems.
Lua: Lua is a lightweight programming language that is commonly used in video game development. Lua is also used in developing sign language recognition and speech conversion systems due to its fast execution and small memory footprint.
JavaScript: As a JavaScript is a popular programming language that is widely used for web development. It can be used to develop web-based sign language recognition and speech conversion systems using frameworks such as React and Angular.
Go: Go is a programming language that is designed for high performance and concurrency. It is used in developing sign language recognition and speech conversion systems due to its speed and efficiency.
Swift: Swift is a programming language that is commonly used for developing iOS and macOS applications. It is used to develop sign language recognition and speech conversion systems for mobile devices, such as smartphones and tablets.
Kotlin: Kotlin is a programming language that is commonly used for developing Android applications. It is used to develop sign language recognition and speech conversion systems for mobile devices, such as smartphones and tablets.
Rust: A Rust programming language used to design for concurrency, safety, and speed. It is suitable for low-level programming tasks and can be used in developing sign language recognition and speech conversion systems.
Scala: Scala is a programming language that is designed to be expressive and concise. It is used in developing sign language recognition and speech conversion systems due to its support for functional and object-oriented programming paradigms.
Shell Scripting: Shell Scripting languages such as Bash, are used to automate repetitive tasks, it can be useful in developing sign language recognition and speech conversion systems, to automate the process of data preparation and annotation, or to automate the deployment of the models.
Again, the choice of language depends on the specific task, the developer's familiarity with the language, the resources available and the system requirements. These are some common languages used in sign language recognition and speech conversion, however, the choice of language depends on the specific task, the developer's familiarity with the language, the resources available and the system requirements.

Step 3:
AutoKeras: AutoKeras is an open-source library that uses machine learning to automate the process of designing and training neural networks. It is built on top of the Keras library and can be used to train models for a wide range of tasks, including image classification, natural language processing, and time series prediction. AutoKeras has been used in several research studies on sign language recognition and speech conversion. For example, some studies have used AutoKeras to train models for recognizing American Sign Language (ASL) gestures from video data. The library's ability to automatically search for the best neural network architecture for a given task can make it a useful tool for developing sign language recognition systems, especially for researchers who may not have a lot of experience with neural networks.

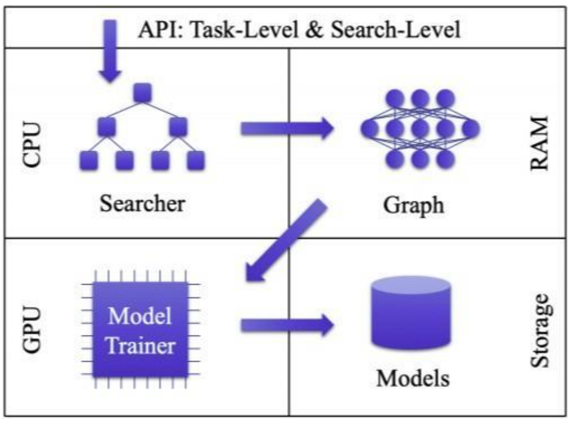

Fig -1.3: AutoKeras system architecture**.**

Additionally, AutoKeras can be used to train models for speech-to-text and text-tospeech conversion, which can be useful for people who are hard of hearing or who use sign

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

language as their primary means of communication. It is worth noting that, AutoKeras uses a search algorithm to find the best neural network architecture, which can be computationally expensive, and require a large amount of computational resources. Also, the performance of the model generated by AutoKeras is not always the best in comparison with a hand-designed architecture, especially in the case of sign language recognition, where the underlying data can be highly variable. In summary, AutoKeras is a useful tool for researchers and developers working on sign language recognition and speech conversion systems. Its ability to automatically search for the best neural network architecture for a given task can save time and resources, and make it easier for those with less experience in neural networks to develop sign language recognition systems. Additionally, the library can be used to train models for speech-to-text and text-to-speech conversion, which can be useful for people who are hard of hearing or who use sign language as their primary means of communication. However, it's important to note that the performance of the model generated by AutoKeras may not always be the best, especially in the case of sign language recognition where the underlying data can be highly variable and the best model architecture may require expert knowledge and human intuition to design.

Step 4:

MediaPipe: MediaPipe is an open-source platform for developing and deploying machine learning models for real-time, multimedia applications. It is developed by Google and has a wide range of pre-built components for tasks such as object detection, hand and facial landmark tracking, and gesture recognition. MediaPipe has been used in several research studies on sign language recognition and speech conversion. For example, some studies have used MediaPipe to track hand and facial landmarks in video data to recognize American Sign Language (ASL) gestures. MediaPipe's pre-built components make it easy to develop sign language recognition systems, as it provides a set of ready-to-use models that can be fine-tuned for specific tasks and data.

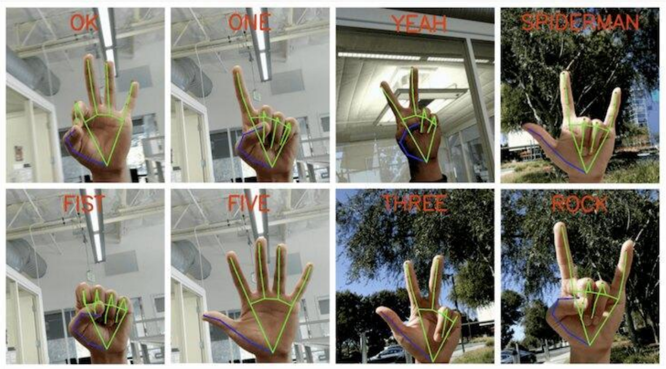

Fig -1.4: On top detection of real hands and below detection of hands.

Additionally, MediaPipe can be used to track facial expressions, which can be used for sign language recognition, as facial expressions are an important part of sign languages, and can convey additional meaning. It's worth noting that, MediaPipe is still a research project and is not as mature as other libraries, some features may not be available and it could be less stable than other libraries. Additionally, the accuracy of the models generated by MediaPipe may not be as good as models trained with other libraries, specifically in the case of sign language recognition, where the underlying data can be highly variable. In summary, MediaPipe is an open-source platform that can be used to develop real time, multimedia applications, including sign language recognition and speech conversion systems. It provides pre-built components for tasks such as object detection, hand and facial landmark tracking, and gesture recognition which can be used to track the hand and facial landmarks in video data to recognize American Sign Language (ASL) gestures. MediaPipe's pre-built components make it easy to develop sign language recognition systems, as it provides a set of ready-to-use models that can be fine-tuned for specific tasks and data. However, it's important to note that MediaPipe is still a research project, and some features may not be available, it could be less stable than other libraries and the accuracy of the models may not be as good as models trained with other libraries, specifically in the case of sign language recognition, where the underlying data can be highly variable.

Step 5:

Recurrent Neural Networks: Recurrent Neural Networks (RNNs) are a type of neural network that are well-suited for tasks involving sequential data, such as time series data or natural language processing. In sign language recognition and speech conversion, RNNs can be used to process sequences of video frames or audio samples to recognize and translate sign language gestures or speech. RNNs have been used in several studies on sign language recognition, where they have been trained to recognize sign language gestures from video data. The network can process the video frames in a sequential manner and use the context of the past frames to improve the recognition of the current frame. Long Short-Term Memory (LSTM) networks, which is a type of RNN, are particularly well-suited for this task, as they are able to remember information over long periods of time, which is useful for capturing the temporal dynamics of sign language gestures.
Similarly, RNNs can be used in speech-to-text and text-to-speech conversion systems, where they process audio and text data in a sequential manner, to translate speech to text or text to speech. It's worth noting that RNNs are complex models and require a large amount of training data and computational resources to train effectively. Additionally, it's important to use the right type of RNN architecture that fits the task at hand, like LSTM or Gated Recurrent Unit (GRU) as they have different properties that can be better suited for a specific task.

In summary, Recurrent Neural Networks (RNNs) are a type of neural network that are well-suited for tasks involving sequential data, such as time series data or natural language processing. In sign language recognition, RNNs can be used to process sequences of video frames to recognize sign language gestures, by using the context of the past frames to improve the recognition of the current frame. Long Short-Term Memory (LSTM) networks, which is a type of RNN, are particularly well-suited for this task, as they are able to remember information over long periods of time, which is useful for capturing the temporal dynamics of sign language gestures. In speech-to-text and text-to-speech conversion systems, RNNs can be used to process audio and text data in a sequential manner, to translate speech to text or text to speech. However, it's important to use the right type of RNN architecture that fits the task at hand, like LSTM or Gated Recurrent Unit (GRU) as they have different properties that can be better suited for a specific task, and also RNNs are complex models and require a large amount of training data and computational resources to train effectively.

Step 6:

RNN with Python: Yes, it is possible to use Recurrent Neural Networks (RNNs) with Python for sign language recognition and speech conversion. Python is a popular programming language in the field of artificial intelligence and machine learning, and has several libraries and frameworks that can be used to implement RNNs

One popular library for implementing RNNs in Python is Keras, which is a high-level library that can be used to train and deploy neural networks. Keras is built on top of TensorFlow, and provides a simple, user-friendly API for defining and training neural networks. Another popular library for implementing RNNs in Python is Pytorch, which is an open-source deep learning framework that provides a dynamic computation graph, which allows for the modification of the graph on-the-fly and ease the usage of RNNs. Both of these libraries provide pre-built RNN layers that can be used to train models for sign language recognition and speech conversion. For example, in Keras, you can use the LSTM or GRU layers to create a RNN model, and in Pytorch you can use the nn.LSTM or nn.GRU module to create a RNN model.

It's worth noting that training a RNN model for sign language recognition or speech conversion can be a complex task and require a large amount of data and computational resources. Additionally, it's important to have a good understanding of the underlying data and the task at hand to properly design the architecture and set the hyperparameters of the model.

Step 7:

Approach: There are several approaches to sign language recognition and speech conversion. One approach is to use computer vision techniques to analyses video footage of sign language and convert it to speech or text. This can involve using machine learning algorithms to train a model to recognize and translate specific signs and gestures. Another approach is to use sensors, such as accelerometers or depth cameras, to track the movement of a person's hands and body and translate that movement into speech or text. A third approach is to use a combination of computer vision and sensor-based methods to improve the accuracy and robustness of sign language recognition and speech conversion.

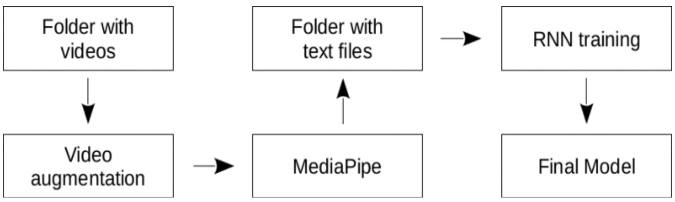

Fig -1.5: Steps followed in order to obtain a gesture recognition model.

In addition, there are also some approaches that use deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to model the spatio-temporal dynamics of sign language. Lastly, it is important to note that there is not one standardized sign language, different countries and regions have different sign languages. Therefore, a sign language recognition and speech conversion system will usually be trained and tailored to a specific sign language.

Step 8:

Indian Sign language (ISL): Indian Sign Language (ISL) is a visual-spatial language used by the Deaf and hard-of-hearing community in the India. Recognizing and converting ISL to speech or text is a challenging task due to the complexity of the grammar and the large number of signs and variations. Indian Sign Language (ISL) is a sign language used by the deaf and hard-of-hearing community in India. It is an optical language that uses a combination of hand gestures, facial expressions, and body language to convey meaning. ISL is not a standardized language and can vary depending on the region and community. ISL is complex and nuanced, with a large number of signs and variations that can be used to convey different meanings. It has its own grammar, syntax and lexicon, which are different from spoken languages. ISL is not a direct representation of spoken languages and it is not possible to directly translate spoken languages into ISL Another approach is to use sensors, such as accelerometers or depth cameras, to track the movement of a person's hands and body and translate that movement into speech or text. This can be a more robust approach, as it is less affected by lighting and camera angles. A third approach is to use a combination of computer vision and sensor-based methods to improve the accuracy and robustness of ASL recognition and speech conversion.

In addition, there are also some approaches that use deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to model the spatio-temporal dynamics of ASL and improve the recognition performance.

Sign election: Sign selection refers to the process of choosing the appropriate sign or gesture to represent a specific word or concept in sign language recognition and speech conversion. One approach to sign selection is to use a predefined dictionary or lexicon of signs and their corresponding meanings. This can be a useful approach for simple systems that only need to recognize a limited set of signs or words. Another approach is to use machine learning algorithms to train a model to recognize and translate signs based on examples of sign language in a dataset. This approach allows the model to learn the relationship between signs and meanings and can handle a larger number of signs and variations. Additionally, some models use a combination of both: a predefined dictionary of signs as a starting point and then fine-tune it based on examples of sign language in a dataset. It's worth noting that sign selection is a complex task and still an active area of research. While some progress has been made in developing sign language recognition and speech conversion systems, they are still not as accurate or robust as human interpreters.

Step 9:

Database: A database is an important component in sign language recognition and speech conversion systems. It is used to store and organize the data used to train and evaluate the performance of the system. The database typically includes examples of sign language in the form of video or sensor data, along with the corresponding transcriptions or translations.

For sign language recognition, the database may include a large number of videos of people signing, along with annotations or labels indicating which signs or gestures are being used in each video. This data is then used to train machine learning models to recognize and classify different signs and gestures. For speech conversion, the database may include videos or sensor data of sign language, along with audio recordings of the corresponding spoken language. This data is used to train a model to translate signs or gestures into speech. In addition, some databases also include additional information such as facial expressions, head movements, and body language, which can be used to improve the recognition and conversion performance. It's worth noting that creating a good database for sign language recognition and speech conversion is a challenging task and requires a lot of data collection, annotation and preprocessing. Moreover, the database should include as much variation as possible to improve the robustness and generalization of the model.

Step 10:

A) Video recording: Video recording plays a crucial role in sign language recognition and speech conversion systems. It is used to collect data of people signing, which helps train and evaluate the system's performance. For sign language recognition, video recording captures signs and gestures in different settings and lighting conditions to train machine learning models. In speech conversion, video recording captures signs and gestures along with corresponding spoken language audio to translate them. High-quality video recording is important for accurate recognition and translation. It's essential to use good cameras and maintain consistent recording conditions. Collecting diverse data from various signers and situations ensures a representative sample for the system.

B) Video augmentation: Video augmentation is a technique used in sign language recognition and speech conversion systems to create more training data. It involves making changes to the videos, like rotating, scaling, and flipping, to generate diverse sign language examples. The aim is to enhance the system's adaptability by exposing it to different variations. This helps prevent the model from only performing well on specific data. Common video augmentation methods include random cropping, flipping, rotation, and adding various backgrounds. However, it's important to strike a balance between diversity and realism. Too many changes may create unrealistic examples, while too few may lead to overfitting.

Step 11:
Using the model in Real Time: Using a sign language recognition and speech conversion model in real-time involves several considerations. One important consideration is the computational complexity of the model. Real-time processing requires the model to make predictions quickly and efficiently, which can be a challenge for complex models with many layers and neurons. To address this, it may be necessary to simplify the model architecture, use lightweight neural network architectures, or use specialized hardware such as Graphics Processing Units (GPUs) to accelerate the processing. Another consideration is the input data format. Real-time processing requires the model to process video data in real-time, which can be a challenge if the model is trained on still images. To address this, the model may need to be adapted to work with video data, or the video data may need to be pre-processed to extract still images that can be fed into the model. Additionally, it's important to consider the latency of the model. Real-time processing requires the model to make predictions with minimal delay, so the model should be designed to have a low latency. This can be achieved by using lightweight neural networks, or by using specialized hardware or software to accelerate the processing.

4. CONCLUSIONS

Sign language recognition and speech conversion are active areas of research in computer science and linguistics. Recent advancements in machine learning and computer vision have improved accuracy in sign language recognition, but challenges remain in real-time recognition and variability across sign languages. Speech conversion systems have made progress in converting spoken language into text or sign language, but factors like accents and noise impact their effectiveness. Ongoing research focuses on developing more accurate algorithms, user-friendly interfaces, and multimodal approaches that combine vision and language processing. With continued advancements, sign language recognition and speech conversion systems have the potential to enhance communication and inclusivity.

ACKNOWLEDGEMENT

The heading should be treated as a 3rd level heading and should not be assigned a number.

REFERENCES

1. T. Yang, Y. Xu, and "A., Hidden Markov Model for Gesture Recognition", CMURI-TR-94 10, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1994.
2. Pujan Ziaie, Thomas Muller, Mary Ellen Foster, and Alois Knoll "A Naive Bayes Munich, Department of Informatics VI, Robotics and Embedded Systems, Boltzmannstr. 3, DE-85748 Garching, Germany.
3. Mohammed Waleed Kalous, Machine recognition of Auslan signs using Power Gloves: Towards large-lexicon recognition of sign language.
4. Aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-ConvolutionalNeural Networks-Part-2/ [4] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham
5. Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32(4), 572–577 (2011).
6. N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nicograph International (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9
7. Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen" Real-time sign language fingerspelling recognition using convolutional

neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition

8. Kirsti Grobel and Marcell Assan. "Isolated sign language recognition using hidden Markov models". In: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. Vol. 1. IEEE. 1997, pp. 162–167.
9. Ruchi Manish Gurav and Premanand K. Kadbe. "Real time finger tracking and contour detection for gesture recognition using OpenCV". In: 2015 Inter- national Conference on Industrial Instrumentation and Control, ICIC 2015. Institute of Electrical and Electronics Engineers Inc., July 2015, pp. 974–977. ISBN: 9781479971657. DOI: 10.1109/IIC.2015.7150886.
10. Haifeng Jin, Qingquan Song, and Xia Hu. "Auto-keras: An efficient neural architecture search system". In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, pp. 1946– 1956.
11. Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: arXiv preprint arXiv:1412.6980 (2014).
12. Pradeep Kumar, Himaanshu Gauba, Partha Pratim Roy, and Debi Prosad Do- gra. "Coupled HMM-based multi-sensor data fusion for sign language recog- nition". In: Pattern Recognition Letters 86 (2017), pp. 1–8.
13. T Kuroda, Y Tabata, A Goto, H Ikuta, M Murakami, et al. "Consumer price dataglove for sign language recognition". In: Proc. of 5th Intl Conf. Disability, Virtual Reality Assoc. Tech., Oxford, UK. 2004, pp. 253–258.
14. T. Liu, W. Zhou, and H. Li. "Sign language recognition with long short- term memory". In: 2016 IEEE International Conference on Image Processing (ICIP). 2016, pp. 2871–2875.
15. S. A. Mehdi and Y. N. Khan. "Sign language recognition using sensor gloves". In: Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02. Vol. 5. 2002, 2204–2206 vol.5.
16. Arpit Mittal, Andrew Zisserman, and Philip HS Torr. "Hand detection using multiple proposals". In: BMVC. Vol. 40. Citeseer. 2011, pp. 75–1.
17. Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, and Valentino Zocca. Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd, 2019

BIOGRAPHIES

CHANDRAKANT D. BOBADE
Graduate Student Member, IEEE
M.Tech in VLSI Design
Sagar Institute of Science & Technology