APPLICATION OF SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING

B PRADYUN REDDY, SAI TARUN REDDY, CH. CHAITANYA KRISHNA
MR. SHAIK MEERAVALI (Guide)
Sreenidhi Institute of Science and Technology
Hyderabad

ABSTRACT:

Many persons with illnesses including those that render them dumb, deaf, or blind are seen daily. Interacting with people is tough for them. Sensor-based approaches that have been previously developed all failed to provide the overall answer. The proposed project's main objective is to develop a system that employs machine learning to give voiceless communication through a system that is both affordable and effective. The recommended technique translates sign language into text by utilizing OpenCV and CNN. There will be easier communication between the two groups as a result.

INTRODUCTION

People who are dumb or deaf may find it challenging to communicate with them if they don't know sign language well. Social isolation and a challenge accessing information and resources are possible outcomes of this. People who are deaf or dumb may encounter difficulties in their educational endeavors because they may find it difficult to access spoken lectures or other audio-based materials. It's critical to acknowledge and solve the difficulties that deaf and dumb people confront, as well as to offer them assistance and resources so they can enjoy fulfilling lives, Hence this project's main goal is to develop a sign language recognizer that can be used to foresee the alphabet from hand gestures. This system aims to predict the alphabet, numbers, and a large array of additional hand gestures using machine learning techniques on recorded hand motions. To cut down on communication time, simple communication is necessary with the dumb and deaf.

CONVOLUTION NEURAL NETWORK (CNN)

"Convolutional Neural Network," or CNN , is a particular kind of artificial neural network created with the express purpose of processing data with a grid-like architecture, such as an image. For image categorization, object detection, and picture production tasks, CNNs are frequently used.

In a CNN, the input data is altered through a number of layers, each of which is made up of a collection of trainable filters that are applied to the input data using a method known as convolution. Each layer produces a collection of feature maps that show the existence of the patterns or characteristics that the filters are supposed to identify in the input data.

The ability of CNNs to learn hierarchical representations of the input data, which enables them to capture both low-level and high-level aspects of the image, makes them extremely good in image classification tasks. As a result, they can accurately recognise objects, patterns, and other elements in the image.

Convolutional Layer: This layer serves as CNN's framework and is in charge of carrying out convolutional operations. The element in this layer that does the convolution process is the Kernel/Filter. The kernel makes horizontal and vertical changes based on the stride rate until the entire image is scanned.While the kernel is smaller than a picture, it is deeper. The kernel height and width will be modest in size if the image has three channels, but the depth will cover all three.

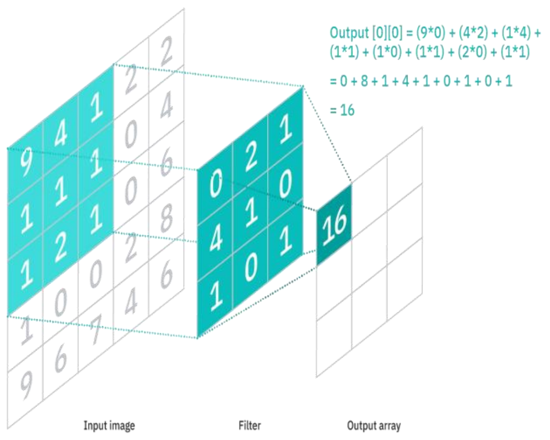
Output [0][0] = (9*0) + (4*2) + (1*4) + (1*1) + (1*0) + (1*1) + (2*0) + (1*1)
= 0 + 8 + 1 + 4 + 1 + 0 + 1 + 0 + 1
= 16
Input image
Filter
Output array

Fig2: Convolutional Layer

Pooling Layer: This layer, known as the pooling layer, is in charge of minimising dimensionality. It helps to lessen how much computing power is needed to process the data. Maximum and average pooling are the two categories into which pooling can be classified. Max pooling returns the highest value from the kernel-covered region of the image. Average pooling gives the average of all the values in the area of the picture covered by the kernel.

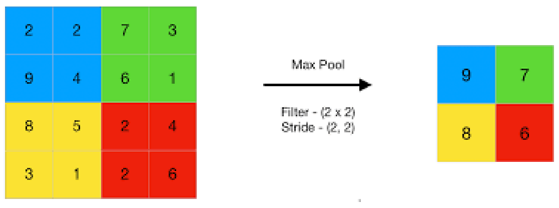

Fig 3: Pooling Layer

Flatten Layer: The convolutional and pooling layers' multidimensional output must be flattened into a one-dimensional vector so that it may be fed into the fully connected (FC) layer . Typically, the flatten layer comes before the fully connected layer and is put after the convolutional and pooling layers. The primary job of this layer is to transform the output of the convolutional and pooling layers into a one-dimensional vector so that it can be fed into the fully connected layer for additional processing. In a CNN, the flatten layer is essential because it enables the fully connected layer, which computes the final prediction based on the characteristics extracted by the convolutional and pooling layers, to handle the output of the convolutional and pooling layers.

Fully Connected Layer: This layer, known as the pooling layer, is in charge of minimising dimensionality. It helps to lessen how much computing power is needed to process the data. Maximum and average pooling are the two categories into which pooling can be classified. Max pooling returns the highest value from the kernel-covered region of the image. Average pooling gives the average of all the values in the area of the picture covered by the kernel.

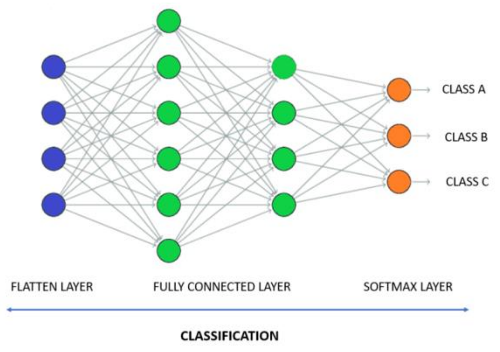

Fig 4: CNN Layers

OPERATION OF CNN

Multiple layers of a CNN are possible, and each layer trains the CNN to recognize the many aspects of an input picture. Each picture is given a filter or kernel to create an output that grows better and more detailed with each layer. The filters may begin as basic characteristics in the bottom levels.

To examine and identify characteristics that specifically reflect the input item, the complexity of the filters increases with each additional layer. As a result, after each layer, the output of each convolved picture becomes the input for the following layer. The CNN recognizes the picture or objects it represents in the final layer, which is an FC layer.

The input image is processed via a number of different filters during convolution. Each filter performs its function by turning on specific aspects of the image, after which it sends its output to the filter in the subsequent layer. The procedures are repeated for dozens, hundreds, or even thousands of layers as each layer learns to recognize various characteristics. Finally, the CNN is able to recognize the full object thanks to the picture data flowing via its numerous layers.

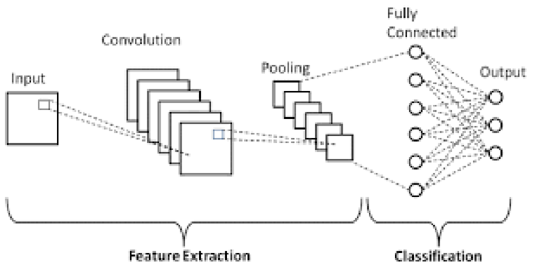

Fig 5: CNN Architecture

BENEFITS OF CNN

A particular class of neural network called a convolutional neural network (CNN) is made for processing input having a grid-like architecture, like an image. They have attained cutting-edge scores on several picture recognition benchmarks and are particularly well-suited for image classification tasks.

Without explicit feature engineering, CNNs may automatically identify and extract key features from the incoming data . Even if pictures are translated or rotated during input, CNNs may still detect patterns in them. Convolutional layers and pooling layers, which apply filters and downsample the input data, respectively, are used to achieve this.The same set of weights is applied across all points in the input picture when using a CNN. As a result, the model has fewer parameters and is simpler to train. Even when pictures are translated during input, CNNs may still detect patterns in the data. Convolutional layers and pooling layers, which apply filters and downsample the input data, respectively, are used to achieve this. GPUs can effectively train CNNs, allowing for quicker training periods and the training of large and complicated models. CNNs are effective in a number of image identification tasks and can generalise well to fresh data.

OPEN SOURCE COMPUTER VISION

OpenCV, sometimes known as "Open Source Computer Vision," is a free and open-source collection of computer vision and machine learning algorithms created to assist developers in creating software that can process and comprehend visual data from the outside world.Intel originally created OpenCV, which is now maintained by a global development community.

A wide range of capabilities is included in the library, including image and video collecting and processing, fundamental and sophisticated image processing methods, object detection and recognition, 3D reconstruction, machine learning algorithms, and much more.Robotics, security systems, medical imaging, and many other industries can all benefit from using OpenCV.

The fact that OpenCV is free and backed by a sizable and engaged developer community is one of its main features.This implies that anyone who want to learn more about or utilise the library have access to a variety of information and resources.OpenCV is also cross-platform, which enables it to run on several operating systems including Windows, Linux, and MacOS.

PROCEDURE FOR IMPLEMENTATION

- Constructing Data Set
- Adding Libraries
- Data Preprocessing
- Feature Extraction
- Training a model
- Predict the Gesture

CONSTRUCTING DATASET

To start, a dataset of hand movements must be gathered. This may be accomplished by filming individuals performing hand gestures on camera and identifying each move. In order to have a representative dataset, it's critical to have a varied collection of individuals. In order to catch all the pertinent information, it's also crucial to record the motions in a well-lit setting and from several perspectives. Large amounts of data must be gathered in order to get reliable findings. It is appropriate to collect data from a wide variety of people since the person's age and colour might affect their ability to recognise the gesture.

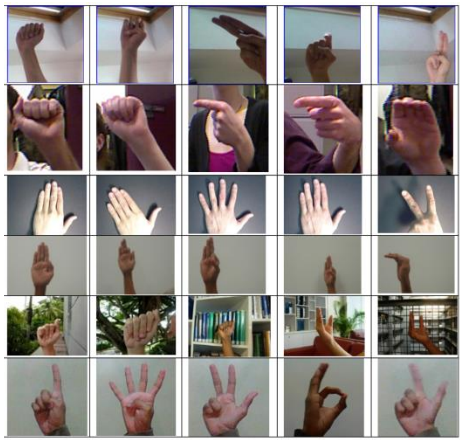

Fig5: Sample DataSet of this Project

ADDING LIBRARIES

In this project, we're utilizing a variety of libraries, including Tensorflow, Matplotlib, NumPy, openCV , sk-Learn is used to extract pictures and transform them into matrices and grayscale images.

Tensorflow: An open-source software library for artificial intelligence and machine learning is called TensorFlow. It was created by Google and is employed for many different machine learning tasks, such as image classification, natural language processing, and predictive modelling.

Matplotlib: A Python data visualisation package is called Matplotlib. In Python, it is used to provide interactive, animated, and static visualisations. You may design line plots, scatter plots, bar plots, error bars, histograms, bar charts, pie charts, box plots, and a variety of other visualisations with Matplotlib.

openCV: Applications for OpenCV include processing of images and videos, object identification, and augmented reality. It offers a variety of tools and capabilities for dealing with photos and videos, such as reading, writing, and displaying functions for images and videos, processing functions for images and videos, and object identification and tracking functions.

DATA PREPROCESSING

The area of the frame where we wish to detect the hand for gestures is called the ROI (Region of Interest) and This window is used to access the webcam's live stream., We compute the accumulated weighted average for the background in order to distinguish it from the foreground by subtracting it from the frames that have a distinguishable foreground object in front of the background.

To achieve this, we compute the accumulated weight over a certain number of frames, 'n', and then compute the accumulated avg for the background. To locate any object that covers the background, we deduct the background's accumulated average from each frame we read after the first 'n' frames. Now, we use CV2 to identify the contours for each frame and calculate the threshold value. Utilizing the function segment, find Contours returns the max contours, the object's outermost contours. We can tell if there is a hand in the ROI by looking at the contours to see if any foreground objects are being picked up there.We begin to save the picture of the ROI in the train and test sets, respectively, for the letter or number we are detecting for when contours are found or if the hand is present in the ROI.

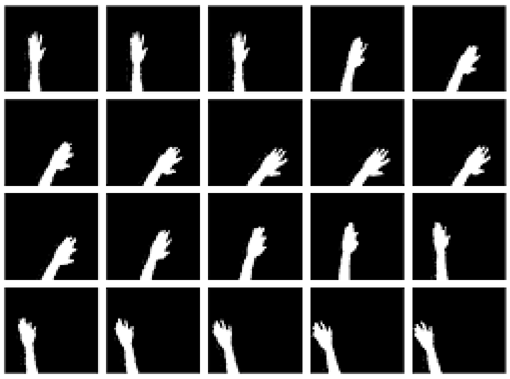
Fig 6: Sample of preprocessed data

FEATURE EXTRACTION

In order to create a model that can predict events with accuracy, it is crucial to extract the key characteristics from the processed pictures. We extract the hand's shapes, sizes, and finger positions from the preprocessed photos in order to gather more information that will help the model understand the gesture more precisely and reliably.

With the use of these characteristics, hand gestures are represented in a form that a machine learning model can comprehend. Convolution neural networks were utilised to automatically extract features from the provided data.

TRAINING THE MODEL

To create a model, we use the features that were extracted. To train the model, we use the whole dataset that was gathered as an input. In this project, the model has been trained with all 26 alphabets and 10 numbers. The alphabets and the numerals can be predicted by the model with accuracy.

WORKING OF APPLICATION:

Using the hand gesture data that has been gathered, we first construct a model.

The live feed is supplied to the processing code when a hand gesture is made in front of the camera by the user. The programme turns the video into a series of frames the moment it receives the stream. Each frame is then preprocessed. Each picture is first reduced in size to 50 × 50 pixels during the preprocessing stage, after which it is transformed into a matrix with 0s or 1s as its values. Based on the matrix value at that particular picture pixel, the frame is then changed to a grayscale version of itself. The purpose of grey scaling is to eliminate noise and outliers from the photos. Next, the model receives the greyscale picture. In order to anticipate the gesture from the training data, the model then makes use of neural networking.The predicted gesture is displayed to the end user in order to simplify the communication.

OUTPUT

We must first set up our hand histogram.

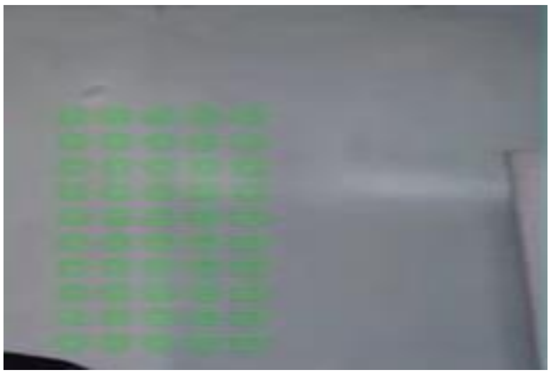

Fig 7: We should place our palm in the green box and ensure the palm is entirely covered in the green region

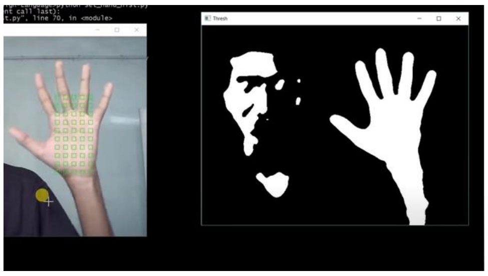

Fig 8: Greyscaling of the input

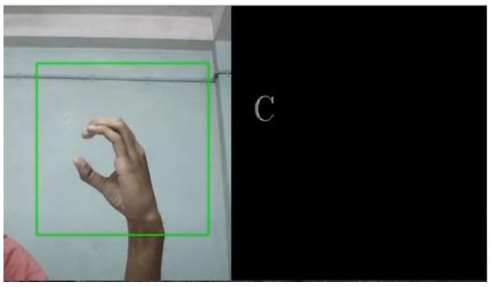

Fig 9: Final output

We've made progress in sign language recognition. This can be used to recognise letters, numbers, and certain hand gestures, and it can be expanded to recognise a sizable number of additional hand motions.

MERITS

When it comes to merits, the higher accuracy of our project makes it more reliable for decision-making. It makes the task a bit easier to communicate with deaf and dumb. As the system can recognize the gestures accurately we can use this in direct communication media, by this it can help the deaf and dumb to communicate effectively in a short time.

LIMITATIONS

The main limitation of the project is that it is often ambiguous , the same sign can be used to denote several different words or ideas in sign language, which makes it frequently unclear. Because of this, it could be challenging for sign language recognition software to understand what is being signed.

The other limitation is that the system cannot recognise other complex gestures as we have a very limited amount of data available to train the model.

SOFTWARE REQUIREMENTS

In this project we have used Python 3.7.4, Numpy, openCv, Keras, Tensorflow, h5py, pyttsx3.

HARDWARE REQUIREMENTS

In this project we have used 4 GB RAM , storage of 500 GB , 2 GHz Processor , 64 bit architecture , Nvidia graphics card

CONCLUSION

Hand gesture recognition can be helpful for deaf and dumb individuals as a means of communication and expression. Many deaf and dumb individuals use sign language as their primary mode of communication, and hand gesture recognition technology can facilitate this by allowing them to communicate with hearing individuals or with devices and systems that do not understand sign language.For example, hand gesture recognition technology can be used to translate sign language into written or spoken language, allowing deaf and dumb individuals to communicate with hearing individuals who do not know sign language. It can also be used to control devices or systems, such as smart home devices or assistive technology, using sign language gestures. Overall, hand gesture recognition technology has the potential to improve the accessibility and communication capabilities of deaf and dumb individuals, and can help to reduce barriers to communication and interaction with the broader world.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to our mentor Mr. Shaik Meeravali for providing us with the fantastic chance to work on this amazing project on this subject. This project also enabled us to do extensive research, which allowed us to learn a lot of brand-new information. They have our sincere appreciation.

As a follow-up, we also want to express our gratitude to my friends, who greatly aided us in completing this job in the allotted time.

REFERENCES

1. Ahmed W, Chanda K, Mitra S. Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language. In 2016 International Conference on Information Science (ICIS); 2016: IEEE. p. 120-125
2. Guo D, Zhou W, Wang M, Li H. Sign language recognition based on adaptive HMMS with data augmentation. Proceedings - International Conference on Image Processing, ICIP. 2016; 2016-August: p. 2876-2880
3. Jin CM, Omar Z, Jaward MH. A mobile application of American sign language translation via image processing algorithms. Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016. 2016;: p. 104-109.
4. Huang J, Zhou W, Zhang Q, Li H, Li W. Video-based sign language recognition without temporal segmentation. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018. 2018;: p. 2257-2264

AUTHORS

First Author - B. Pradyun Reddy ,
B-tech Final year , CSE
Second Author - Sai Tarun Reddy,
B-tech Final year , CSE
Third Author - CH. Chaitanya Krishna
B-tech Final year , CSE