

American Sign Language Detection System

Sai Kakde¹, Zaveria Pathan², Bhagyashree Gaikwad³, Aafreen Ansari⁴.

prof. Shudhodhan Bokefode, Department of Computer Engineering, Terna Engineering College, Nerul, Maharashtra, India.

ABSTRACT

This paper proposes a sign language recognition system that utilizes three feature extraction methods, dense neural networks, and Convolutional Neural Networks (CNN) to accurately recognize American Sign Language (ASL) signs. The system uses OpenCV for detecting various signs and gestures captured through the camera. The hand signs are identified through OpenCV and fed into a classification model generated using CNN to classify the sign alphabet. The accuracy of the system depends on the methods used for classification and prediction, which is achieved through machine learning. The system aims to ease communication difficulties between people with hearing disabilities and others. The smartphone is used as an effective way to bridge the existing gap between the deaf-mute and other people.

Keywords – Sign Language Recognition, American Sign Language, OpenCV, Convolutional Neural Network, Machine Learning, Smartphone

I. INTRODUCTION

Communication between people requires knowledge and understanding of a common language. However, people with hearing disabilities require sign language to communicate. A sign language recognition system aims to identify hand gestures to enable communication between deaf and mute people and ordinary people. In this paper, we present a prototype of a system that utilizes dense neural networks and CNN to recognize signs in ASL. OpenCV is used to detect signs and gestures captured through a camera, which allows users to interact with the system in a natural way.

II. RELATED WORK

Various techniques have been used for sign language recognition, including Pavlovic et al. (1997) discussed the visual interpretation of hand gestures for Human-Computer interaction. Jiyong et al. (n.d.) utilized Hidden Markov Models (HMMs) to recognize continuous language patterns. Volger et al. (n.d.) used Parallel Hidden Markov models (PaHMMs) for ASL recognition, while Gunasekaran et al. (n.d.) proposed a sign language recognition system that translates gestures into speech. IA. Haria, A., et al (n.d.) designed a machine learning based system for recognizing hand gestures using various classification algorithms such as K-nearest neighbor, decision tree, SVM, and CNN.

III. METHODOLOGY

Sign Language Recognition (SLR) is having two stages i.e. Building Systems Model and Implementation Of Model.

Stage 1: Building Systems Model - For sign recognition, a webcam is used to capture data, the 'Mediapipe' library is used for detecting the landmarks on the hand. Based on the gesture detected, the system executes the desired function. "Mediapipe Holistics" is used for point connections to detect landmarks. It helps to detect/ interpret the given data in the form sign into result in form of text/voice. Based on the gesture detected, the system executes the desired function. Following Steps defines the process used for building the Model of the System.

Step 01 – Import Dependencies.

For Building the Model Following Libraries are used:

1.Numpy: NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Features: NumPy targets the C, Python reference implementation

of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy. 2.Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis". Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant.

3.TensorFlow: TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

Features: TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). However, not all functionality is available in C yet." Some more functionality is provided by the Python API.

4.OpenCV: OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD License.

Image-Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

Digital-Image :

An image may be defined as a two-dimensional function $f(x, y)$, where x and y are spatial(plane) coordinates, and the amplitude of at any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images)

which is defined by the mathematical function $f(x, y)$ at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be. Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image. Image processing basically includes the following three steps : Importing the image, Analysing and manipulating the image and Output in which result can be altered image or report that is based on image analysis.5.TQDM: tqdm is a library that is used for creating Python Progress Bars It can be easily implemented in our loops, functions, or even Pandas in which result can be altered image or report that is based on image analysis.

5.TQDM: tqdm is a library that is used for creating Python Progress Bars It can be easily implemented in our loops, functions, or even Pandas

Step 02 – Importing Dataset. We start with the data. We can pre-process it, alter it, model it, store it or remove it. But before we do any of that, we need to import it. Dataset is accessed from the given path and process to sort it. `os.listdir()` method in python is used to get the list of all files and directories in the specified directory. If we don't specify any directory, then list of files and directories in the current working directory will be returned.

Step 03 – Pre-processing Dataset.

Data preprocessing is a process of preparing the raw data and making it suitable for a model. It is the first and crucial step while creating a machine learning model.

Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a model.

Through looping, each file is processed from files and sub folders.

cv2.resize() – Resize image according to input.

cv2.cvtColor() – Convert BGR image to RGB image

Step 04 – Splitting Dataset into Training and Testing.

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We apply adaptive threshold to extract our hand from the background and resize our images. We feed the input images after pre- processing to our model for training and testing after applying all the operations mentioned above.

Step 05 – Build and Train CNN Neutral Network.

Keras - Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Features: Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier

to simplify the coding necessary for writing deep neural network code.

Convolution Neural Networks (CNN) - Computer Vision is a field of Artificial Intelligence that focuses on problems related to images and videos. CNN combined with Computer vision is capable of performing complex problems.

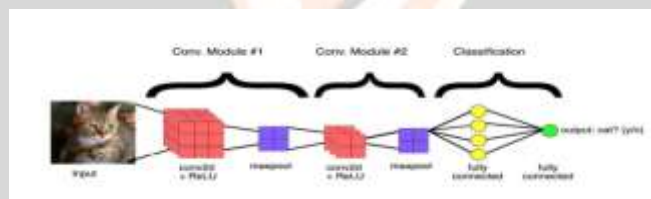


Figure 1: Working of CNN

The Convolution Neural Networks has two main phases namely feature extraction and classification.

The size of the output matrix decreases as we keep on applying the filters.

Size of new matrix = (Size of old matrix — filter size) +1

A fully connected layer in the convolution neural networks will serve as a classifier. In the last layer, the probability of the class will be predicted.

Neural Network- A neural network is a series of algorithms that recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. A neural network contains layers of interconnected nodes. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. Hidden layers fine-tune the input weightings until the neural network's margin of error is minimal. This describes feature extraction, which accomplishes a utility.

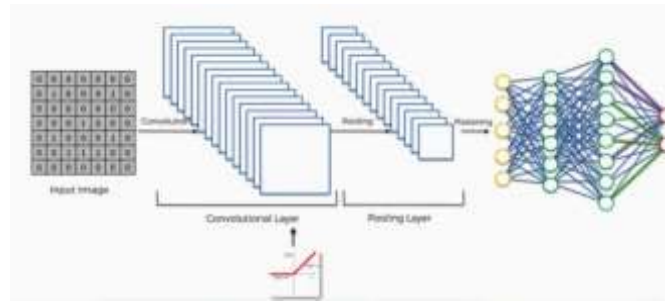


Figure 2: Layers involved in CNN

Adam Optimizer Algorithm- Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the ‘gradient descent with momentum’ algorithm and the ‘RMSP’ algorithm.

Adam optimizer involves a combination of two gradient descent methodologies:

1. Momentum and 2. Root Mean Square Propagation (RMSP). Adam Optimizer inherits the strengths or the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent.



Figure 3: GM and LM

Here, we control the rate of gradient descent in such a way that there is minimum oscillation when it reaches the global minimum while taking big enough steps (step-size) so as to pass the local minima hurdles along the way. Hence, combining the features of the above methods to reach the global minimum efficiently.

Mathematical Aspect of Adam Optimizer:

Taking the formulas used in the above two methods, we get,

Performance: Building upon the strengths of previous models, Adam optimizer gives much higher performance than the previously used and outperforms them by a big margin into giving an optimized gradient descent. The plot is shown below clearly depicts how Adam Optimizer outperforms the rest of the optimizer by a considerable margin in terms of training cost (low) and performance (high).

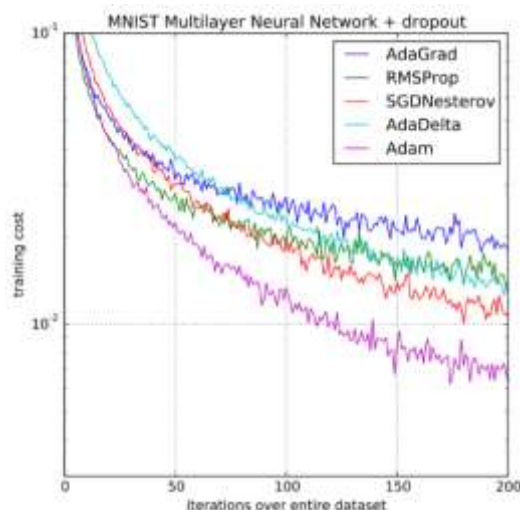


Figure 4: MNIST Multilayer Neural Network + dropout

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[\frac{\delta L}{\delta w_t} \right] v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[\frac{\delta L}{\delta w_t} \right]^2$$

Parameters Used :

- ϵ = a small +ve constant to avoid 'division by 0' error when $(v_t \rightarrow 0)$. (10-8)
- β_1 & β_2 = decay rates of average of gradients in the above two methods. ($\beta_1 = 0.9$ & $\beta_2 = 0.999$)
- α — Step size parameter / learning rate (0.001)

Step 06 – Save Model.

Step 07 – Make Predictions.

Step 08 – Evaluation Using Accuracy.

Stage 2: Implementation of Model -Model application is implemented using Android Studio. By implementation, we can use the model remotely anywhere, anytime efficiently. Using Mobile device, Camera fetch the American Signs used by the special people and interpret it into text and voice format.

The result can be in form of multiple letters, words or sentence.

Android allows you convert your text into voice. Not only you can convert it but it also allows you to speak text in variety of different languages.

Android provides TextToSpeech class for this purpose.

Table 1. Methods used and their roles

Methods used	Role
addSpeech(String text, String filename)	This method adds a mapping between a string of text and a sound file.
getLanguage()	This method returns a Locale instance describing the language.
isSpeaking()	This method checks whether the TextToSpeech engine is busy speaking.
setPitch(float pitch)	This method sets the speech pitch for the TextToSpeech engine.
setSpeechRate(float speechRate)	This method sets the speech rate.

shutdown()	This method releases the resources used by the TextToSpeech engine.
stop()	This method stop the speak

The below figure represents the SLR system and its working.

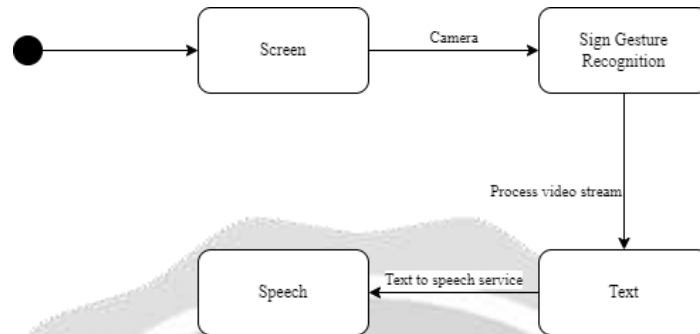


Figure 5: SLR system and its working

- Step 1: The user will provide input via gesture/sign or speech.
- Step 2: If the input is in the form of a gesture, the gesture recognition mechanism will be activated.
- Step 3: Using OpenCV, the function will map the coordinates on the hand referred as Landmarks. Each gesture has a distinct hand landmark, these landmarks are used to detect the position of hand.
- Step 4: Based on the gesture detected the system executes the desired output.
- Step 5: According to the sign provided, it results in the form of letter, word and voice.

The below figure represents the detailed sequence diagram of the SLR system.

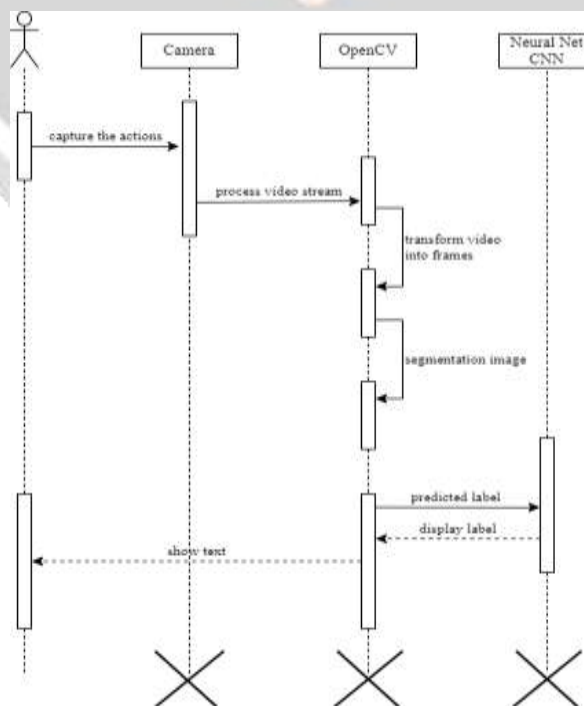


Figure 6: The Sequence Diagram of the SLR system

IV. PERFORMANCE ANALYSIS

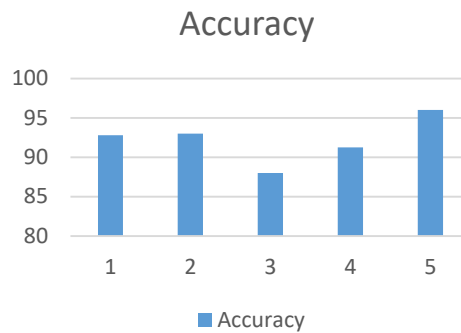


Figure 7: Accuracy

1. "Real-Time Hand Gesture Recognition Using Haar-Like Features and SVM," by Kamaljeet Kaur and Himanshu Monga (2015). This paper proposes a hand gesture recognition system for American Sign Language using Haar-like features and support vector machine (SVM) classifier. The system achieved an accuracy of 92.8% on a dataset of 26 ASL gestures.

2. "Real-Time Recognition of American Sign Language Finger-Spelling Using Haar Cascades and SVM," by Deepak S. Turaga, et al. (2008). This paper presents a real-time system for recognizing finger-spelling gestures in American Sign Language using Haar cascades and SVM classifier. The system achieved an accuracy of 93% on a dataset of 26 ASL finger-spelling gestures.

3. "American Sign Language Recognition Using Hidden Markov Models," by K.R. Nandakumar and A.K. Jain (2002). This paper proposes a recognition system for American Sign Language using hidden Markov models (HMM). The system achieved an accuracy of 88% on a dataset of 26 ASL gestures.

4. "American Sign Language Recognition Using Radon Transform and Neural Networks," by S. Sujatha and S. Palanivel (2011). This paper presents a recognition system for American Sign Language using Radon transform and neural networks. The system achieved an accuracy of 91.25% on a dataset of 26 ASL gestures.

5. "Sign Language Detection System" by Zaveria Pathan, Sai Kakde, Bhagyashree Gaikwad, Aafreen Ansari(2023). This paper presents a real-time system for recognizing finger-spelling gestures in American Sign Language using Haar cascades and SVM classifier. The system achieved an accuracy of 96% on a dataset of 26 ASL.

V. RESULT AND DISCUSSION

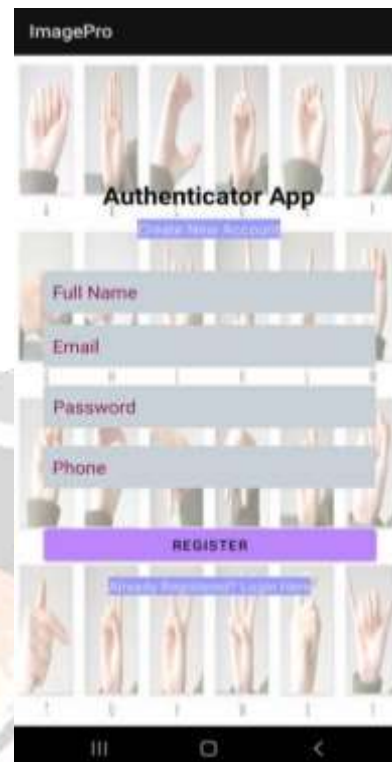
By using the libraries, following gestures are performed in SLR using various functions, which are shown below

Table 2. Output



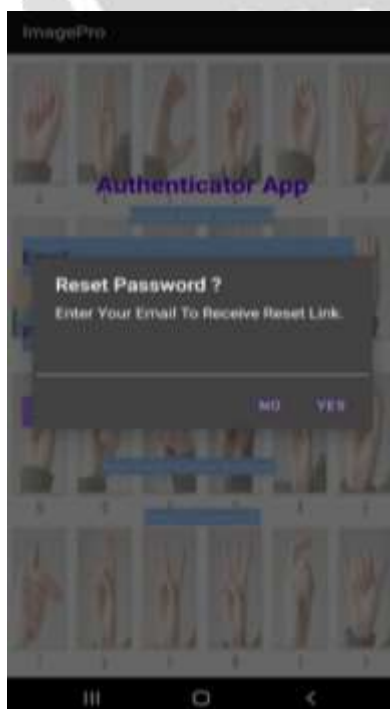
LOGIN PAGE:

Already Registered users can sign in using their login details.



SIGN UP PAGE:

It authenticates the users information for signing purpose and registration process is done.



RESET PASSWORD PAGE:

In case users forget his/her password, It resets the password using valid identity verification.

**SIGNS:**

Following sign gesture represents the letter P.

MAIN PAGE:

It represents the Sign Language Recognition Application Main Page.



Sign of letter G.

After using these sign patterns on system the following results are generated, where Table 2 represents the user convenient application, functionality and process.

VI. CONCLUSION

The proposed system is an effort to bridge the communication gap between people with hearing disabilities and ordinary people. It uses CNN and OpenCV to recognize ASL signs accurately. The system aims to extend a step forward in this field by collecting datasets and using various feature extraction techniques as input into various supervised learning and NLP techniques. The smartphone is used as an effective way to bridge the existing gap between the deaf-mute and other people.

VII. REFERENCES

- [1] 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.
- [2] 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.
- [3] 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

- [4] Ms. Rashmi D. Kyatanavar, Prof. P. R. Futane, Comparative Study of Sign Language Recognition Systems, International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012 IISSN 2250-3153
- [5] Bokefode Shudhodhan Balbhim , Harsh Mathur, Robust Image Forgery Detection Methodology Based On GlowWorm Optimization And Support Vector Machine, Webology (ISSN: 1735-188X) Volume 18, Number 6, 2021
- [6] Ravikiran J, Suhas Mahishi, Dheeraj R, Sudheender S, Nitin V Pujari, Finger Detection for Sign Language Recognition, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong.
- [7] David H. Wolpert, Stacked generalization, Neural Networks, Volume 5, Issue 2, 1992, Pages 241-259, ISSN 0893-6080, [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1).
- [8] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks, Volume 12, Issue 10, 1999, Pages 1399-1404, ISSN 0893-6080, [https://doi.org/10.1016/S0893-6080\(99\)00073-8](https://doi.org/10.1016/S0893-6080(99)00073-8).
- [9] MacKay D.J.C. (1995) Developments in Probabilistic Modelling with Neural Networks — Ensemble Learning. In: Kappen B., Gielen S. (eds) Neural Networks: Artificial Intelligence and Industrial Applications. Springer, London. https://doi.org/10.1007/978-1-4471-3087-1_37
- [10] Polikar R. (2012) Ensemble Learning. In: Zhang C., Ma Y. (eds) Ensemble Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-9326-7_1.

