JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Empowering Inclusive Communication: Sign Language Recognition through Computer Vision

Krupa Gawde, Smita Kadam, Vanita Varadkar, Ruchi Mhaddalkar, Swapnil Kadam

Student, Student, Student, Student, Assistant Professor
Computer Engineering Department,
SSPM College of Engineering, Kankavli, Sindhudurg, India

Abstract: Sign language serves as a communication bridge between normal and hard-of-hearing communities. People with difficulty communicating through speech rely on sign language to exchange language in society and with other people. Developing a sign language recognition system fills the communication gaps and enhances accessibility. Sign language recognition is a tool that can recognize and interpret hand gestures and convert them into text. Sign Language Recognition (SLR) systems take input from non-native English speakers and put it into a form for normal people. In this paper, we present an approach for real-time American Sign Language(ASL) recognition leveraging computer vision techniques and machine learning algorithms. Our system utilizes convolutional neural network (CNN) architecture for feature extraction and to achieve high accuracy. In addition, we created our dataset to train and evaluate our model, containing 2990 ASL characters in all. After training the model using TensorFlow, the CNN model outperformed the pre-trained model on the American Sign Language (ASL) dataset. The proposed system achieves effectiveness making it suitable for practical application in education and communication aids for the deaf and hard-of-hearing individuals.

I. INTRODUCTION

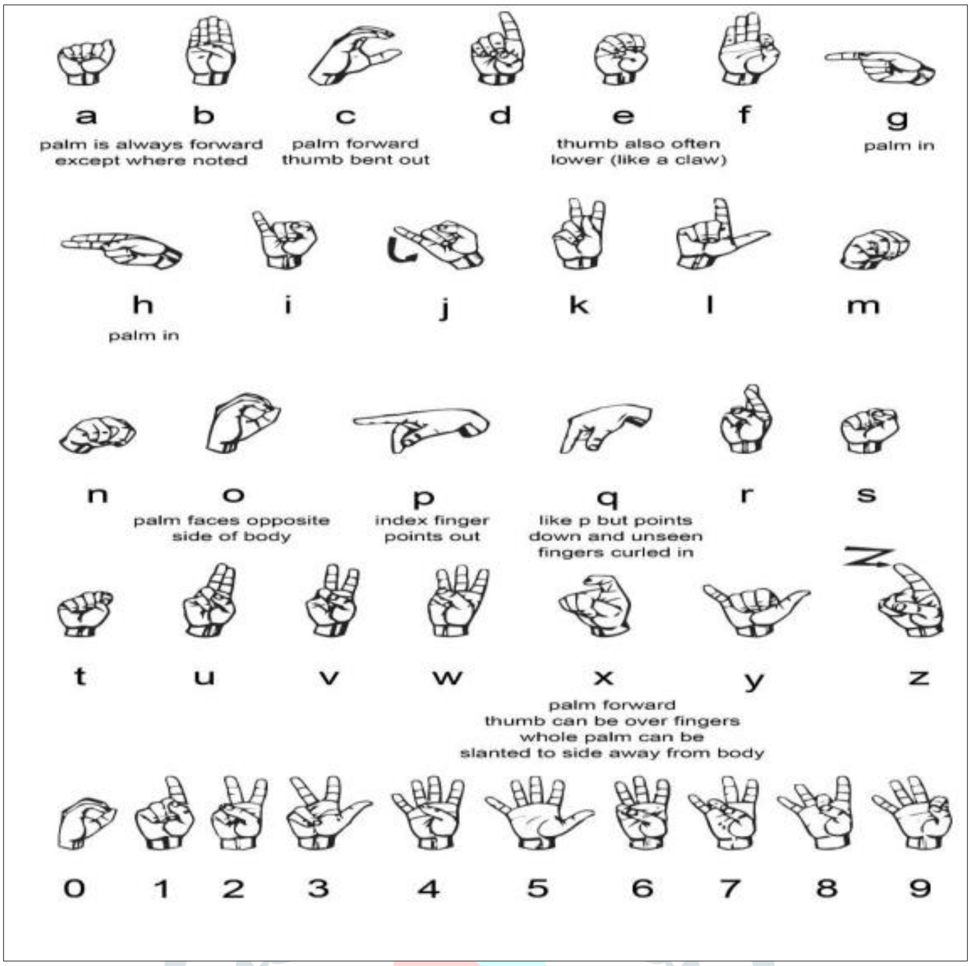
Language is a rich and expressive form of communication used by millions of people around the world. Sign language is a form of communication used by people with impaired hearing and speech. These differently-abled people use sign language as their primary means to communicate and express themselves to other people with hand gestures. But non-signers find it extremely difficult to understand, hence sign language interpreters are needed. Upon review, we found that there aren't many websites or apps that provide free information or education to deaf people. Our purpose is to foster the development that will make it easier for the deaf and hard of hearing to communicate with the hearing world.

There are 138 to 275 different types of sign languages used globally such as British Sign Language (BSL), Chinese Sign Language (CSL), Arabic Sign Language (ArSL), and many more. In India, there are about 218 certified sign languages for a deaf population of around 7 million. Hence, American Sign Language(ASL) is the language chosen by all the deaf communities is a one-hand sign language. It can be expressed by the movements of the hands. ASL is the most commonly used language. In recent years, the intersection of computer vision (CV), and machine learning algorithms, especially convolutional neural networks (CNN), has provided a promising approach to increasing sign language knowledge.

As computer vision and machine learning continue to advance and are actively developing, computerized systems are capable of recognizing and converting sign language to text. The proposed method uses data containing multiple letters, allowing the model to learn and generalize gestures. TensorFlow is an open-source machine learning framework developed by Google and is a flexible and scalable platform for building, training, and deploying models, making it ideal for the complex tasks of language recognition. CNN architectures can be designed, optimized, and fine-tuned to achieve optimal performance.

In 2022, a system based on the Arabic Sign was used for handwritten documents. By the way, they used a lot of prioritization and training tools and their models reached 89 accuracy. ASL verification is also very helpful. The development of a camera lens begins with the collection of images to their distribution. Due to problems in obtaining the data set, personal data is requested to be created for the data collection strategy. Since CNN has better performance than other algorithms, a model using CNN to analyze ASL performance is planned for 2020. The proposed model uses computer vision, TensorFlow, and four layers to achieve accuracy. Our scheme works well with home data and we soon achieve an accuracy of 94.71. SLR poses a big difficulty in terms of computer vision because of a variety of factors, including:

I. Environmental disturbance (e.g., lighting sensitivity, background, and camera position)
II. Closure (e.g., some fingers or an entire hand can be out of the field of view)
III. Sign boundary detection (when a sign ends and the next begins)



In short, this project promises to use CV technology, TensorFlow, and CNN algorithms to support the development of language recognition. By delving into the mathematics of deep learning and exploring new applications, we want to contribute to ongoing efforts to create a more equal and equitable world for everyone, regardless of how they communicate. Our goal is to facilitate this development so that deaf and hard-of-hearing people can easily communicate with the hearing world.

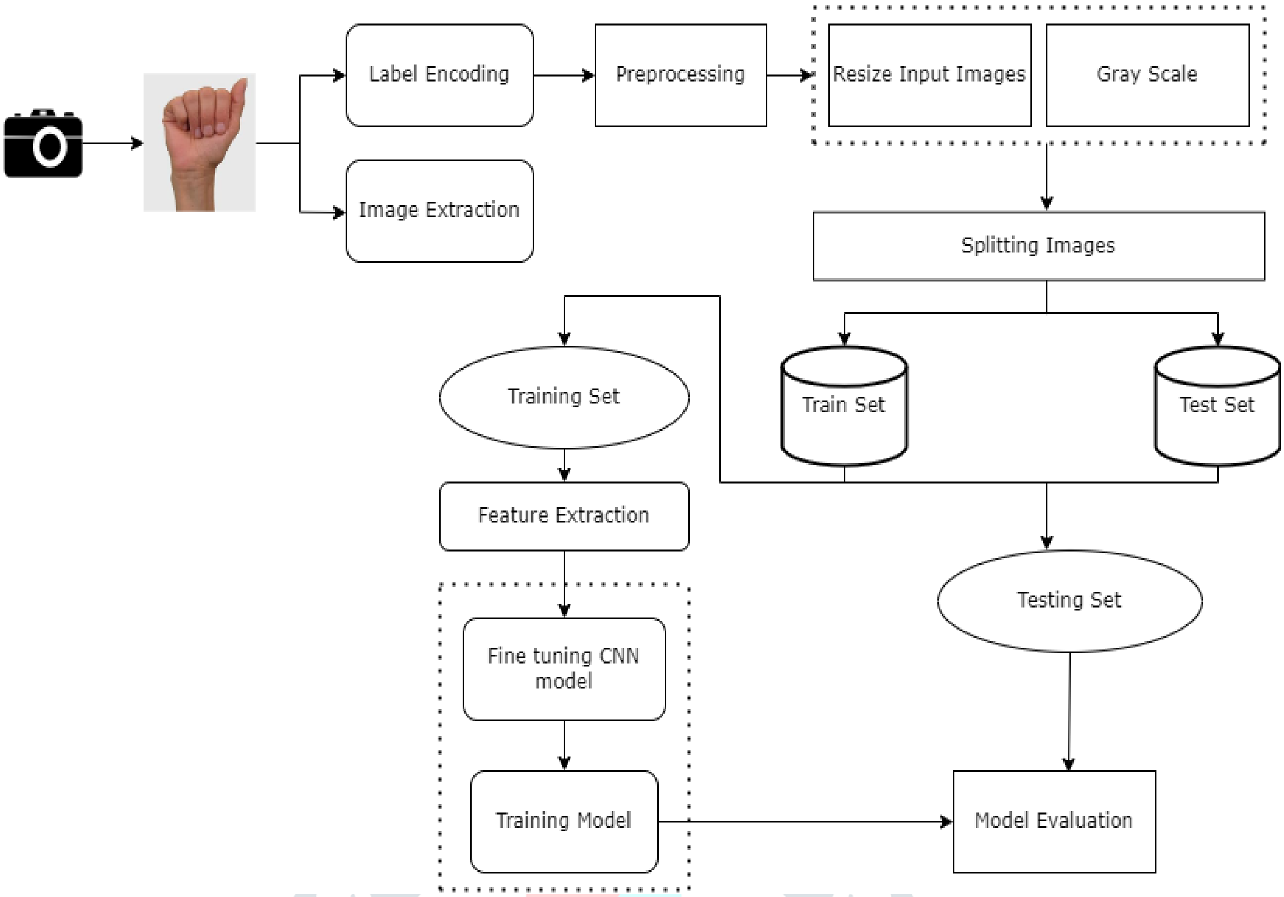
II. BIBLOMETRIC ANALSIS

Bibliometric analysis shows that there have been several approaches to address sign language recognition systems. Sign language uses lots of gestures so that it looks like movement language which uses fingers for its motions. In different countries, there are different sign languages and hand gestures. In [1] the author proposed a dynamic sign language system using machine learning methods and the OpenCV library to recognize characters. B.Pradyun Reddy [3] proposed a Hand Gesture Recognition System that uses CNN to recognize complex data and train the model. Safyzan Salim focused on translating hand gestures into text format achieving good accuracy. Sign Language Recognition (SLR) deals with recognizing hand-gesture acquisition and continues till text or speech is generated for corresponding hand gestures. Hence, recognizing hand gestures is relatively problematic. I.A. Adeyanjua [9] presented an action of retrieving an image from a camera for image acquisition. A similar kind of work was presented by [6] Aman Pathak and Radha Shirbhate defining how to implement and enhance the model to recognize expressions. There are specific gestures for each alphabet A to Z and for numbers between 0 to 9. Sign languages are divided into two, static gestures and dynamic gestures. Satwik Kondandaram [8] focused on using Computer Vision techniques to recognize static and dynamic hand gestures and translate them into text or speech. pon reviewing the literature and studying all the essential research papers and documentation thoroughly, we found that there are unexplored features that draw interest. From the research and literature, we found five significant gaps-

I. As per studies, first we found that the recognition model lacked the data, hence it is necessary to implement the model with the detailed dataset.
II. In ASL, finger spellings are used to represent English words, developing strong techniques and integrating them is an important gap.
III. In ASL, many times words or sentences involve continuous signing without pauses, thus recognizing and segmenting those signs is an important challenge.
IV. Then we found that gestures made by impaired people become difficult to understand for the hearing community, so recognizing the sign language and translating it into text or speech is important.
V. Finally, developing larger, more complex, and more effective datasets for training the model is an empirical gap.

III. PROPOSED ARCHITECTURE

ur proposed architecture aims to accurately interpret and classify ASL gestures in real-time. Our program focuses on hand gesture knowledge containing 26 English letters (A-Z). Our current goal is to improve static language recognition using computer vision and convolutional neural networks. Figure 1 illustrates the overall process of our SLR architecture. This diagram provides sequential steps involved in our system.



To achieve this goal, we use techniques such as image acquisition, hand tracking, subtraction, and classification. Since we use the ASL file at home, which contains images in pixel format, the CNN module is good for this data. Use CNN to detect the movement of the image. The diagram below shows a visual representation of the steps involved in our system. We present the model we developed using computer vision for image detection, TensorFlow for model training, and CNN for extraction and classification. Before separating the data into training and testing, some pre-processing is required. Finally, in this study, the use of TensorFlow training data is presented and we reach the most accurate version of the model.

IV. SYSTEM DESIGN

a. Design Details

The ASL sign language recognition design includes various components that work together to create a user-friendly experience. The first step of the system is to detect the real-time of the user's movement and capture the image

I. Camera Interface: The model connects to the webcam to capture images using OpenCV and stores them in Python's CV2 library.

II. Mapping: To identify hand movement you need to track the hand image accurately. Hand Tracker is the mapping library used to track hand movements and to identify key points on the hand.

III. UI elements: The user can perform different ASL gestures based on that, the system will generate text

b. Dataset

We created our dataset of still images called the "ASL dataset". We used the Open Computer Vision (OpenCV) library to produce our dataset. Our dataset comprises 26 ASL signs. Each class is represented by a substantial number of images, with 100 samples per class. All photos in our collection were taken in various environmental backgrounds (natural light, sunlight, artificial light). There are a total of 2990 alphabet (A to Z) pictures, each with different lighting. This large dataset size enables our proposed architecture to leverage a significant amount of data during the training process.

c. Algorithm

i. Convolutional Neural Network

Fine-tuning process using: CNN

First, the model demonstrates the algorithm for pre-processing the images and labels. The input photos are captured after normalization and resizing is done. We have fine-tuned the model using CNN so that we get improved accuracy and a more efficient model for sign recognition The Convolution Neural Network has two main phases namely feature extraction and classification. A series of convolution and pooling operations are performed to extract the features of the image. 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.

The main steps involved in convolution neural networks are:

I. Convolution Operation:
The convolution operation involves applying filter w to an input image x to produce a feature map. This can be expressed mathematically as:

$$zij = (x * w)ij = \sum m \sum n xi+m,j+n \cdot wm,n + b \qquad (1)$$

where x is the input image, w is the filter (kernel), z is the resulting feature map, b is the bias term, and i and j denote the spatial indices of the feature map

II. Activation Function:
We have used ReLU (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). ReLU calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time. It is defined as:

$$ReLU(x) = max(0, x) \qquad (2)$$

III. Pooling:
We apply Max pooling to the input image with a pool size of (2, 2) with the ReLU activation function. This reduces the number of parameters thus lessening the computation cost and reducing overfitting.

IV. Optimizer:
We have used the Adam optimizer for updating the model in response to the output of the loss function. Adam optimizer combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm

In addition, mathematical concepts such as unemployment and the regularization process also play an important role in improving the learning process and improving the ability of CNN. Convolutional layers are used to extract features from images. CNN architecture is applied to effectively process and analyze the features that are extracted from images. It learns from extracted data by analyzing the different patterns. We have used 4 layers for feature extraction, accuracy, and classification

d. Experimental Setup

i. VScode

Visual Studio Code (VSCode) is popular for its cross-platform support, multi-language support through extensions, user-friendly features such as syntax highlighting and Git support, integration and customizability. A large market for extensions and powerful IDE features such as debugging. and suggestions, community support, quality work, and the free and open nature of it.

ii. Python Language

Python is the best programming language because it has a lot of libraries and frameworks designed for machine learning and computing. Its simplicity and readability make it easy to quickly learn and try different algorithms. Community-wide support ensures adequate resources and support throughout the development process. Python's integration is seamless and compatible with hardware and other technologies.

iii. Python libraries

Python libraries provide tools and functions that simplify the process of building, training, and deploying machine, learning models. These are some libraries we used in our project:

I. Open CV:
OpenCV (Open-Source Computer Vision) is an open-source library of programming functions used for real-time computer vision. It is mainly used for image processing, video capture, and analysis for features like face and object recognition. It is written in C++ which is its primary interface, however, bindings are available for Python, Java, MATLAB/OCTAVE

II. TensorFlow:
TensorFlow is an end-to-end open-source platform for Machine Learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources. TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy

III. Keras:
Keras is a high-level neural network library written in Python that works as a wrapper for TensorFlow. It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objectives, activation functions, optimizers, and tools to make working with images and text data easier.

V. METHODOLOGY

a. Model Training

Machine learning has a revolutionary process compared to traditional machine learning techniques. It is designed to be used without deep knowledge of machine learning\cite{BR2}. Here's a simple summary of how training is typically done at Teachable Machine:

I. Data Collection:
Start by collecting information relevant to your goals. For example, if you are building an image classifier, you will need images related to the categories you want the model to recognize. Teachable Machine provides an interface for loading and tagging data. You can organize the items into groups or classes that you want the model to learn from.

II. Model Training:
Once your profile is uploaded and tagged, you can start training. Teachable Machine uses a predefined learning machine (usually a neural network) appropriate to the task you're working on (image classification, audio classification, etc.).The training process involves optimizing the model's parameters using the domain data you provide.

III. Lessons and Reporting:
Teachable Machine provides feedback on model performance during and after training. This includes accuracy, precision, recall, etc. It will include measurements such as. Based on this input, you can choose to adjust parameters such as training dataset size, number of training epochs, or sample size. - You can repeat the training process, make adjustments, and retrain the model until you are satisfied with its performance.

IV. Export your model
Once you're happy with the performance of your learning model, you can export it to a variety of formats, including TensorFlow Lite, TensorFlow.js, and Keras (can be saved as a .h5 file). This export model can be used in other applications or environments to make predictions on new data.

Overall Teachable The machine learning process is designed to be intuitive and user-friendly; It eliminates many of the complexities of the usual machine learning process while also allowing users to train models for a variety of tasks.

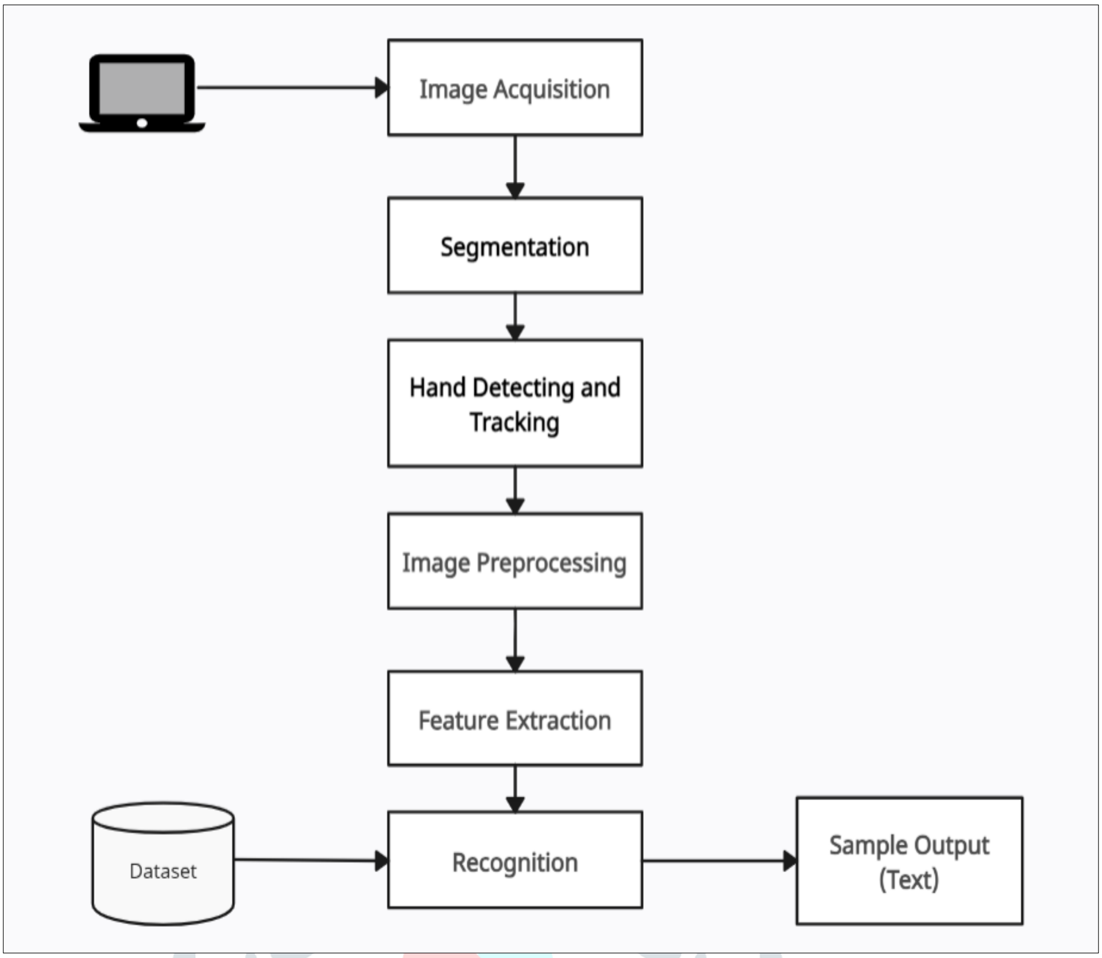
b. Steps

By following this methodology, leveraging CV techniques, TensorFlow, and CNNs, we aim to develop an effective and efficient sign language recognition system capable of accurately interpreting sign gestures in real-world scenarios.

I. Dataset Collection:
Here we will gather a dataset containing images of sign language gestures corresponding to different words, phrases, or letters. Ensure diversity and representative samples across sign languages.

II. Data Acquisition:
To develop an accurate Sign Language Recognition (SLR) model, it is crucial to acquire high-quality image frames that capture the gestures and movements of American Sign Language (ASL). The main equipment used in the monocular system is the camera. ASL information is in the form of hand movements that can be easily captured with a camera.

III. Image Tracking:
We are using the CV zone to perform accurate hand tracking. It tracks the movement of the hand in real time. From the hand tracking module, we extract landmarks for the hand, We ensure that the system can accurately recognize and classify ASL gestures.



IV. Data pre-processing:
For more accurate results, we did some pre-processing on the images of our dataset. The pre-processing steps include:

a. Read images
b. Resize the captured image
c. Remove noise
d. Images are converted into pixel array

V. Data Training with Tensor Flow:
After creating a dataset, to train our data we used a Teachable Machine by Google Creative Lab. Teachable Machine is a web tool that makes it fast and easy to create machine-learning models. It recognizes images and then exports the model in the TensorFlow Keras format.

VI. Feature Extraction with CNN:
Utilize Convolutional Neural Networks (CNNs) to automatically learn discriminative features from sign language images.

$$Hi = ReLU(Wi * Hi-1 + bi) \quad (3)$$

These equations represent fundamental components of CNN-based sign language recognition models. CNN architecture is applied to model effective processes and analyze the features that are extracted from images. It learns from extracted data by analyzing the different patterns.

VII. Evaluation and Validation:
The dataset has been split into training, validation, and testing sets. Evaluated model performance on the validation set, monitoring metrics like accuracy, precision, recall, and F1-score.

VI. RESULTS AND DISCUSSION

The sign language recognition system uses deep learning techniques to interpret life lessons by focusing on American Sign Language (ASL). Using a Convolutional Neural Network (CNN) Extracts Spatial and Temporal Features Architectural model, by integrating two CNNs for spatial information, it enables accurate recognition of ASL characters for the hearing impaired.

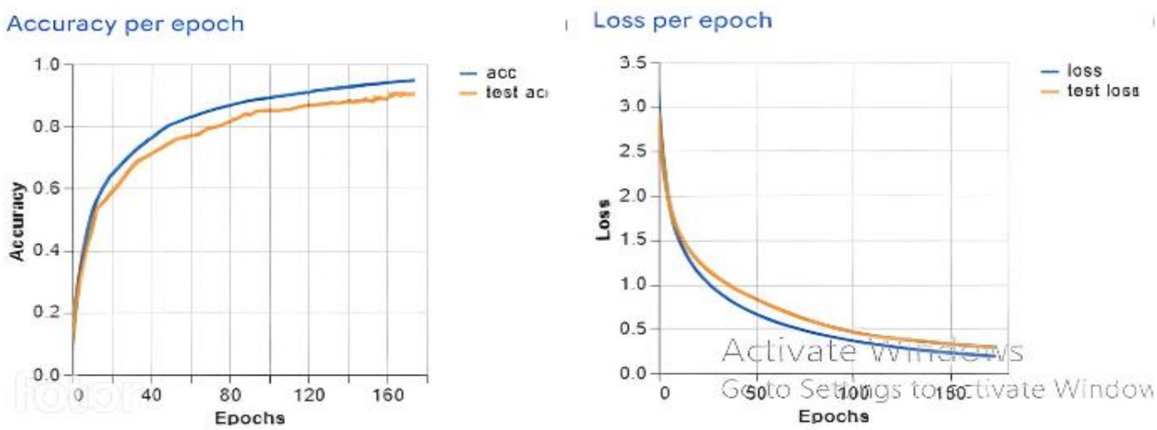
a. Results of Accuracy:

Table 6.1. accuracy of letters per class

Class	Accuracy	Samples
A	0.95	15
B	0.91	17
C	0.95	16
D	0.85	18
E	0.95	18
F	0.59	16
G	0.51	17
H	0.95	16
I	0.95	16
J	0.68	16
K	0.95	15
L	1.00	16
M	0.19	16
N	0.83	19
O	0.87	31
P	0.53	16
Q	0.75	17
R	0.85	16
S	0.59	15
T	0.69	15
U	0.94	19
V	0.98	15
W	0.98	22
X	0.99	17
Y	0.95	26
Z	0.68	25

Table 6.1 displays the accuracy of each letter we achieved as per class. There are a total of 26 letters with their respective accuracy.

6.1.1 accuracy and loss



The model we created achieved an accuracy of 0.9460 and a loss of 0.5400.

b. Environmental Obstacles:

Monocular cameras create significant problems in computer vision due to a variety of factors, including:

1. Environmental interference (e.g. Light sensitivity, background, and camera position)
2. Occlusion (e.g. some fingers or all hands not visible)

VII. CONCLUSION

In this work, we have gone through an automatic sign language recognition system in real-time, using different tools, to recognize the sign language and convert it into the text. we were successfully able to develop a practical and meaningful system that can able to understand sign language and translate that to the corresponding text. This system can detect 0-9 digits and A English alphabet hand gestures but doesn't cover body gestures and other dynamic gestures.

VIII. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our supervisors, Prof. S.B. Kadam, and Dr. N. M. Shivsharan for their invaluable guidance, patience, and support throughout this research. His insightful feedback and constructive criticism have been instrumental in shaping this work. I would also like to thank the participants who generously gave their time and shared their experiences for this study. Finally, I would like to thank my family and friends for their unwavering support and encouragement.

REFERENCES

[1] Gesture recognition based on an improved local sparse representation classification algorithm, 2019.
[2] Geron Aurelien. Hands-on Machine Learning. O'Reilly Media, 2017. https://www.google.co.in/books/edition/Hands On MachineLearning with Scikit Le/bRpYDgAAQBAJ?hl=en&gbpv=0.
[3] CH. CHAITANYA KRISHNA B PRADYUN REDDY, SAITARUN REDDY. Application of sign language recognition using machine learning. International Journal of Scientific Research, 2022.
[4] Safyzan Salim, Muhammad Mahadi Abdul Jamil, Radzi Ambar, Rozaini Roslan, and Muhammad Ghazali Kamardan. Sign language digit detection with media pipe and machine learning algorithm. In 2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE), pages 180–184, Oct 2022.
[5] M.A. Adegboye I.A. Adeyanjua, O.O. Bello. Machine learning methods for sign language recognition. 2021.
[6] Aman Pathak, Avinash Kumar—priyam—priyanshu, Gupta—gunjan Chugh, and Editor Ijmtst. Real-time sign language detection. International Journal for Modern Trends in Science and Technology, 8:32–37,01 2022.
[7] Mr.Vedant Shinde and Prof.Radha Shirbhate. Sign language recognition using a machine learning algorithm. 07, March,2020.
[8]Satwik Ram Kodandaram, N. Kumar, and Sunil Gl. Sign language recognition, 07 2021.
[9] I.A. Adeyanju, O.O. Bello, and M.A. Adegboye. Machine learning methods for sign language recognition: A critical review and analysis.Intelligent Systems with Applications, 12, 2021.
[10] Akshit Tayade and Arpita Halder. Real-time sign language recognition using media pipe and machine learning. 05 2021.
[11] Mr.Satish Kumbhar Ms.Greeshma Pala, Ms.Jagruti Jethwani. Machine learning-based hand sign recognition. 2021.
[12] Andra Ardiansyah, Brandon Hitoyoshi, Mario Halim, Novita Hanafiah, and Aswin Wibisurya. Systematic literature review: (Iyangar, 2022)American sign language translator. Procedia Computer Science, 179:541–549, 2021. 5th International Conference on Computer Science and Computational Intelligence 2020.
[13] thirteen23.com. Asl dataset used for model training. Gesture Recognizer provides a quick and accessible set of models for categorizing hand gestures, identifying handedness, and recognizing characters.
[14] Hand landmarks, 2021.
[15] Prabu P Amrutha K. Ml-based sign language recognition system. 2021.
[16] Bhupendra Singh. Proposed CNN model to classify sign language for alphabet signs, 2021. Convolutional Neural Network in Deep Learning Techniques to improve the performance of convolutional models.
[17] N Rajasekhar, M Geetha Yadav, Charitha Vedantam, Karthik Pellakuru, and Chaitanya Navapete. Sign language recognition using a machine learning algorithm. In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), pages 303–306, 2023.
[18] Priyanka C Pankajakshan. Sign language recognition system. 2015.
[19] Prateek Kumar Ankita Wadhawan. Sign language recognition systems: A decade systematic literature. Archives of Computational Methods,2021.