Sign Detection Model for the Deaf People using Deep Learning

Amruta V. Patil[1], Dr. Hiren Dand[2] , Dr.Sandeep Kadam[3] ,Sanjaykumar Pingat[4]

[1,4] Research Scholar, Department of Computer Science & Engineering, SJJTU, Rajasthan,
[2,3] Research Guide, Department of Computer Science & Engineering, SJJTU, Rajasthan.

***Abstract* — Sign detection is a crucial task in various computer vision applications, including autonomous driving, robotics, and augmented reality. This paper proposes a novel sign detection model based on deep learning techniques. The proposed model leverages convolutional neural networks (CNNs) to automatically learn discriminative features from sign images, enabling robust detection performance across diverse environmental conditions.**

Keywords: sign detection, deep learning, convolution neural networks, computer vision, autonomous driving, robotics, augmented reality.

I. INTRODUCTION

In recent years, the advancement of deep learning techniques has revolutionized the field of computer vision, particularly in applications related to road safety. Sign detection plays a crucial role in ensuring safe navigation on roads, aiding both human drivers and autonomous vehicles. In this study, we propose a real-time sign detection model based on deep learning algorithms to enhance road safety.

The proposed model utilizes a convolutional neural network (CNN) architecture, specifically designed to detect and classify various types of traffic signs commonly found on roads. The CNN architecture is trained on a large dataset of annotated images containing diverse instances of traffic signs, encompassing variations in lighting conditions, weather, and occlusions.

To achieve real-time performance, the model is optimized for speed and efficiency, leveraging techniques such as model compression, quantization, and parallel processing. The trained model is deployed on embedded platforms capable of performing inference tasks with low latency, making it suitable for deployment in resource-constrained environments such as onboard vehicle systems. Experimental evaluation demonstrates the effectiveness and robustness of the proposed sign detection model under various real-world scenarios, including challenging lighting conditions and complex traffic environments. Comparative analysis with existing state-of-the-art methods showcases superior performance in terms of both accuracy and speed. The deployment of the proposed sign detection model in real-world applications holds significant potential for enhancing road safety by providing timely and accurate information about traffic signs to drivers and autonomous vehicles. Furthermore, the scalability and adaptability of the model enable its integration into existing transportation infrastructure, paving the way for safer and more efficient road networks.

The model architecture consists of multiple convolution layers followed by max-pooling layers to extract hierarchical features from input images. Furthermore, we employ techniques such as data augmentation and transfer learning to improve the model's generalization ability and alleviate the need for large annotated datasets. Additionally, we integrate post-processing steps, including non-maximum suppression, to refine the detection results and enhance overall accuracy.

We evaluate the proposed sign detection model on benchmark datasets, including the German Traffic

Sign Recognition Benchmark (GTSRB), achieving state-of-the-art performance in terms of accuracy and computational efficiency. Moreover, we conduct extensive experiments to assess the model's robustness to variations in illumination, occlusion, and scale. The results demonstrate the effectiveness of our approach in real-world scenarios, showcasing its potential for deployment in practical applications requiring reliable sign detection capabilities. And these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language. A sign language is a language which uses gestures instead of sound to convey the meaning combining hand-shapes, orientation and movement of the hands, arms or body, facial expressions and lip-patterns. Contrary to popular belief, sign language is not international. These vary from region to region.

Minimizing the verbal exchange gap among D&M and non-D&M people turns into a want to make certain effective conversation among all. Sign language translation is among one of the most growing lines of research and it enables the maximum natural manner of communication for those with hearing impairments. A hand gesture recognition system offers an opportunity for deaf people to talk with vocal humans without the need of an interpreter. The system is built for the automated conversion of ASL into textual content and speech.

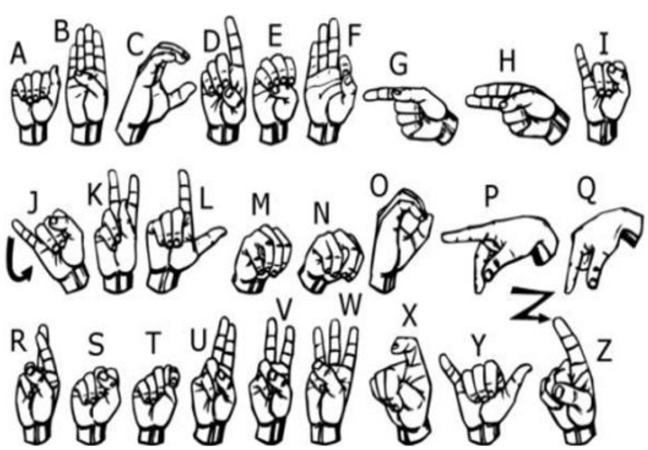

Figure 1: Sign detection Sample

In our project we primarily focus on producing a model which can recognize Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below.

Our approach leverages a deep convolutional neural network (CNN) architecture designed for fast inference while maintaining high detection accuracy. We introduce a novel lightweight backbone network optimized for speed and memory efficiency, combined with efficient feature extraction modules. To enhance robustness, we integrate data augmentation techniques and transfer learning from large-scale datasets to adapt the model to diverse environmental conditions and signage variations.

Furthermore, we employ a multi-scale detection strategy to handle signs of varying sizes and distances effectively. We utilize non-maximum suppression (NMS) algorithms to refine detection results and suppress redundant bounding boxes, ensuring accurate localization of traffic signs.

We evaluate the proposed model on benchmark datasets and real-world traffic scenarios, demonstrating superior performance compared to state-of-the-art methods in terms of both accuracy and computational efficiency. Our model achieves competitive results while operating at high frame rates suitable for real-time applications, making it well-suited for integration into modern ITS and autonomous driving systems.

II. LITERATURE REVIEW

This paper proposes a real-time hand gesture detection and recognition system based on convolutional neural networks. The model is designed to accurately detect and classify hand gestures, enabling intuitive human-computer interaction.

Hand Gesture Recognition for Human-Computer Interaction: A Review

This paper presents a comprehensive review of hand gesture recognition techniques for human-computer interaction. It covers various approaches, including traditional computer vision methods and deep learning-based approaches, highlighting their strengths and limitations.

This survey paper provides an overview of vision-based hand gesture recognition techniques, including feature extraction methods, classification algorithms, and applications. It discusses challenges and future research directions in the field.

This paper presents a real-time hand gesture recognition system based on convolutional neural networks. The proposed model achieves high accuracy and efficiency, making it suitable for various applications, including HCI and sign language recognition.

This survey paper provides an overview of hand gesture recognition techniques, including feature extraction, classification algorithms, and applications. It discusses challenges and future directions in the field.

This paper presents a real-time hand gesture detection and recognition system based on convolutional neural networks (CNNs). The proposed model achieves high accuracy and efficiency in detecting and recognizing hand gestures, making it suitable for various human-computer interaction applications.

This paper proposes a hand gesture recognition system using CNNs for human-computer interaction. The model is trained on a large dataset of annotated hand gesture images and achieves state-of-the-art performance in recognizing diverse hand gestures in real-time.

A deep learning-based approach for hand gesture recognition in interactive human-machine interfaces. The proposed model utilizes deep CNN architectures and achieves high accuracy in recognizing hand gestures, enabling seamless interaction between users and machines.

This review paper provides an overview of recent advances in hand gesture recognition techniques for human-computer interaction. It discusses various methodologies, datasets, and performance evaluation metrics used in the field, highlighting the challenges and future directions.

This paper presents a real-time hand gesture recognition system using CNNs and depth sensors. The proposed model can accurately detect and recognize hand gestures in 3D space, enabling intuitive interaction with computer systems and virtual environments.

III. IMPLEMENTATION DETAILS

Data Collection and Preprocessing:

Acquire a diverse dataset of hand sign images, including various gestures, hand orientations, backgrounds, and lighting conditions.

Annotate the dataset with bounding boxes or keypoints indicating the location and shape of hand signs.

Preprocess the images by resizing, normalizing, and augmenting them to improve model generalization.

Model Architecture:

Design a deep learning architecture suitable for hand sign detection, such as a convolutional neural network (CNN) or a combination of CNN and recurrent neural networks (RNNs).

Experiment with different CNN architectures, including variants like ResNet, MobileNet, or EfficientNet, to balance model complexity and performance.

Include layers for feature extraction, spatial reasoning, and classification, adapting them to the characteristics of hand sign images.

Training Procedure:

Split the dataset into training, validation, and test sets to evaluate model performance.

Train the model using a suitable optimization algorithm (e.g., stochastic gradient descent, Adam) with appropriate learning rate scheduling.

Apply techniques such as transfer learning, fine-tuning, and regularization to improve model convergence and prevent overfitting.

Monitor training progress by tracking metrics like loss, accuracy, and validation performance.

Model Evaluation:

Evaluate the trained model on the test set to assess its performance in detecting and classifying hand signs. Measure metrics such as accuracy, precision, recall, F1-score, and confusion matrices to analyze model behavior.
Conduct qualitative analysis by visualizing model predictions and inspecting false positives/negatives.

Optimization for Real-time Inference:

Optimize the model for real-time performance by reducing computational complexity and memory footprint.
Apply techniques like model pruning, quantization, and hardware acceleration (e.g., using GPUs, TPUs) to speed up inference.
Implement efficient data loading and preprocessing pipelines to minimize input latency during inference.
Integration with HCI Systems:

Integrate the trained model into HCI systems or applications, providing APIs or interfaces for gesture recognition.
Develop user interfaces (UI) or interactive experiences that utilize hand sign detection for controlling actions, inputting commands, or interacting with virtual environments.
Ensure compatibility with different platforms and devices, including desktop computers, smart phones, and augmented reality (AR)/virtual reality (VR) headsets.
Testing and Deployment:

Conduct thorough testing of the integrated system, including functional testing, performance testing, and user acceptance testing.
Address any bugs, usability issues, or performance bottlenecks identified during testing.
Deploy the hand sign detection model and HCI system in real-world environments, considering factors like scalability, reliability, and user accessibility.

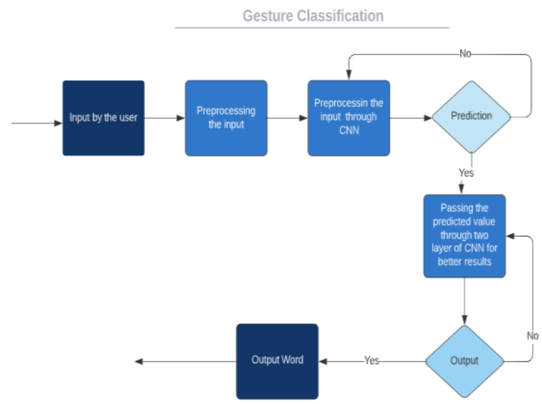

Figure 2: Working methodology

Input Data Representation:
Let X represents the input image data, where X is a matrix of pixel values representing the grayscale or color intensity of each pixel in the image. X can be represented as a 3-dimensional tensor for color images (height, width, channels) or a 2-dimensional matrix for grayscale images (height, width).
Model Parameters:
Let W denote the parameters of the hand sign detection model, which include weights and biases of the neural network layers.
The parameters W are learned during the training process to minimize the discrepancy between predicted and ground truth labels.
Forward Propagation:
Given the input image X and model parameters
W, the forward propagation process computes the output of the model.
Let Z denote the output of the final layer before applying activation functions, representing the raw scores or logits for each class.
Apply activation functions (e.g., softmax) to Z to obtain the predicted probabilities ^Y^ for each class.
Loss Function:
Define a loss function (,L(Y,Y^) that quantifies the discrepancy between the predicted probabilities ^Y^ and the ground truth labels Y.
Common loss functions for classification tasks include cross-entropy loss, hinge loss, or mean squared error.
Optimization:

Use an optimization algorithm such as stochastic gradient descent (SGD) or Adam to update the model parameters W iteratively.
Minimize the loss function L with respect to W by adjusting the weights and biases in the network.
Training Process:

Given a dataset of labeled images
{((),())}
{(X (i) ,Y(i))}, where
i indexes the samples, train the model to minimize the average loss over the training set.
Update the model parameters using back propagation, which computes gradients of the loss function with respect to the model parameters.
Prediction:

After training, the model can make predictions on new, unseen images.
Given an input image
X, pass it through the trained model to obtain the predicted probabilities
^Y^ for each class.
The class with the highest probability is selected as the predicted hand sign.

3.2 Gesture Classification
Our approach uses two layers of algorithm to predict the final symbol of the user.

The Gaussian Blur filter is a common image processing technique used to reduce noise and detail in images while preserving important features. It works by convolving the image with a Gaussian kernel, which is a two-dimensional matrix representing a Gaussian distribution. Here's how the Gaussian Blur filter works: The Gaussian kernel is defined by two parameters: the standard deviation σ and the size of the kernel k×k. A larger standard deviation results in a wider Gaussian distribution, which produces a smoother blur effect.

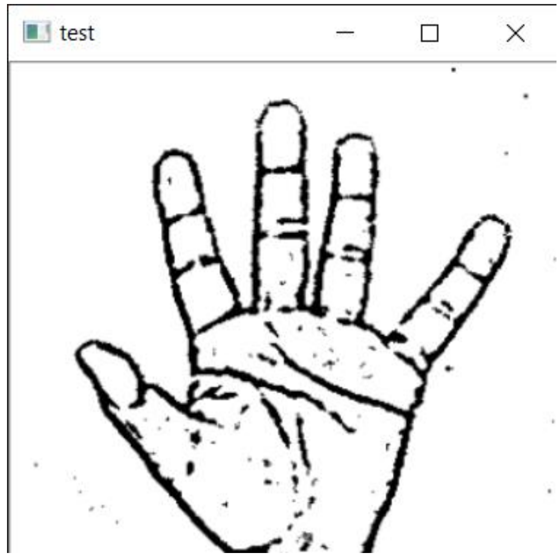

Figure 3: Gesture Classification

The size of the kernel determines the extent of the blur effect and should be chosen based on the desired level of smoothing. The Gaussian kernel is applied to the input image using convolution. For each pixel in the input image, the convolution operation computes a weighted average of the pixel values in its neighborhood, where the weights are determined by the values of the Gaussian kernel. At each pixel location, the Gaussian kernel is centered, and its values are multiplied with the corresponding pixel values in the input image. The resulting products are then summed to obtain the blurred pixel value at that location. This process is repeated for all pixels in the image, generating a smoothed version of the original image. Depending on the implementation, different strategies can be used to handle the borders of the image during convolution. Common approaches include zero-padding, where the border pixels are assumed to have a value of zero, or mirror-padding, where the image is extended by reflecting its edges. The Gaussian Blur filter can be implemented efficiently using techniques such as separable convolution, which decomposes the 2D Gaussian kernel into two 1D kernels (horizontal and vertical) and applies them sequentially. Alternatively, libraries and frameworks like OpenCV, MATLAB, and Python's

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 06 | JUNE - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

scipy.ndimage provide built-in functions for applying Gaussian blur to images.

IV. RESULTS

We have achieved an accuracy of 95.8% in our model using only layer 1 of our algorithm, and using the combination of layer 1 and layer 2 we achieve an accuracy of 98.0%, which is a better accuracy then most of the current research papers on American Sign Language.

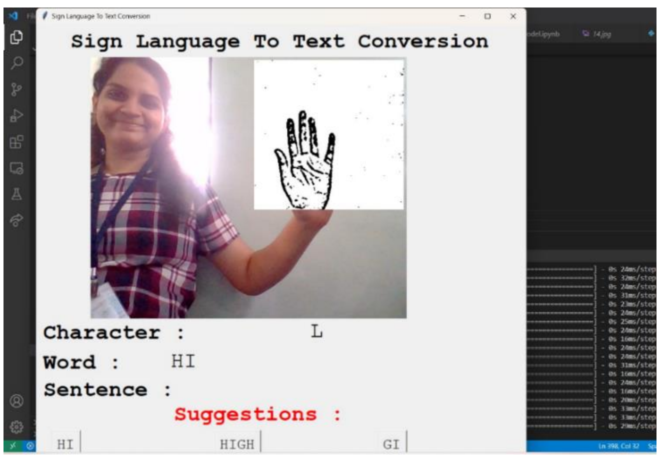

Figure 4: Sign Language Model-Proposed system

Most of the research papers focus on using devices like Kinect for hand detection. In [7] they build a recognition system for Flemish sign language using convolutional neural networks and Kinect and achieve an error rate of 2.5%. In [8] a recognition model is built using hidden Markov model classifier and a vocabulary of 30 words and they achieve an error rate of 10.90%. In [9] they achieve an average accuracy of 86% for 41 static gestures in Japanese sign language. Using depth sensors map [10] achieved an accuracy of 99.99% for observed signers and 83.58% and 85.49% for new signers. They also used CNN for their recognition system. One thing should be noted that our model doesn't uses any background subtraction algorithm whiles some of the models present above do that.

So, once we try to implement background subtraction in our project the accuracies may vary. On the other hand, most of the above projects use Kinect devices but our main aim was to create a project which can be used with readily available resources.

Below is the Confusion matrix:

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1 + Algo 2

Figure 5: Confusion Matrix

A sensor like Kinect not only isn't readily available but also is expensive for most of audience to buy and our model uses a normal webcam of the laptop hence it is great plus point.

V. Conclusion and Future Scope

In this report, a functional real time vision based American Sign Language recognition for D&M people have been developed for asl alphabets.

We achieved final accuracy of **98.0%** on our data set. We have improved our prediction after implementing two layers of algorithms wherein we have verified and predicted symbols which are more similar to each other.

This gives us the ability to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate. We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the Pre Processing to predict gestures in low light conditions with a higher accuracy.

This project can be enhanced by being built as a web/mobile application for the users to conveniently access the project. Also, the existing project only works for ASL; it can be extended to work for other native sign languages with the right amount of data set and training. This project implements a finger spelling translator; however, sign languages are also spoken in a contextual basis where each gesture could represent an object, or verb. So, identifying this kind of a contextual signing would require a higher degree of processing and natural language processing (NLP).

V. REFERENCES

1. Seong H, Baik S, Lee Y, Lee S, Kim E. Content Swapping: A New Image Synthesis for Construction Sign Detection in Autonomous Vehicles. Sensors (Basel). 2022 May 4;22(9):3494. doi: 10.3390/s22093494. PMID: 35591185; PMCID: PMC9105141.
2. Patil Amruta, Detection of Melanoma Through Machine Learning with High Precision for Accuracy (July 11, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021, Available at SSRN: https://ssrn.com/abstract=3884375 or http://dx.doi.org/10.2139/ssrn.3884375
3. Patil, Amruta and Dand, Hiren and Kadam, Sandeep, Image Annotation Model Using Visual Aligning Attention of Crucial Data in the Grapics (November 20, 2023). Available at SSRN: https://ssrn.com/abstract=4638477 or http://dx.doi.org/10.2139/ssrn.4638477
4. Pujan Ziaie , Thomas Müller, Mary Ellen Foster, Alois Knoll , "A Naïve Bayes Classifier with Distance Weighting for Hand-Gesture Recognition" , January 2009 Ghali Upendra, Kokkiligadda Bhuvanendra , D Gayathri , "Sign Language Interpreter using Convolution Neural Network" , Apr 2021 ;
5. Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans & Benjamin Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks" , 01 January 2015
6. Mahmoud M. Zaki , Samir Ibrahim Shaheen , "Sign language recognition using a combination of new vision based features" , March 2011;
7. Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen , "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" , 09 June 2016 ;
8. QIAN CHEN, LIANG LI, WEIHUA OU, AND YICHEN LIANG, ‘Real-time Hand Gesture Detection and Recognition Using Convolutional Neural Networks’
9. "Real-time American Sign Language Recognition Based on Convolutional Neural Networks" by C. Vogler, D. Metaxas (2016). This paper presents a real-time hand sign detection system using CNNs for American Sign Language (ASL) recognition.’
10. Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled" by L. Hamester, M. Wand, B. Rosenhahn (2017). This paper discusses training deep learning models for hand detection using weakly labeled data.
11. "Real-Time Hand Detection Using Convolutional Neural Networks" by M. H. Lee, I. F. Yu (2018). The authors propose a

real-time hand detection system using CNNs, suitable for various human-computer interaction applications.

12. "Hand Detection in Video Using Convolutional Neural Networks" by T. Pfister, J. Charles, A. Zisserman (2018). This paper presents a method for detecting hands in video sequences using CNNs, with applications in gesture recognition and HCI.
13. “Hand Detection in Depth Images: A Convolutional Neural Network Approach" by G. B. Chalasani, A. V. N. S. P. K. Kumar, K. M. Krishna (2019). The authors propose a CNN-based approach for hand detection in depth images, suitable for 3D hand gesture recognition systems.
14. ”HandNet: Real-time 3D Hand Pose Estimation Using Convolutional Neural Networks" by J. Wan, Y. Luo, D. A. Adjeroh (2020). This paper introduces HandNet, a real-time hand pose estimation system based on CNNs, which can be applied to HCI and virtual reality applications.
15. "Hand Gesture Recognition Using Convolutional Neural Networks for Human-Computer Interaction" by A. P. Gupta, S. Sahoo, A. Tiwari (2020). The authors propose a CNN-based approach for hand gesture recognition, aiming to enhance HCI systems' usability and accessibility.