Research Article GU J Sci33(x): x-x (2020) DOI:

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Sign Language Detection Using Computer Vision

Manak RAWAL, Kaustubh RANJAN[1,*]

[1]Students,IT-4th year, Bachelor of Technology, ABES Institute of Technology, Ghaziabad 201009, Uttar Pradesh, India

Highlights

- Sign Language and its importance among the society.
- Dataset and system for sign and gesture detection.
- Existing and proposed model for increased accuracy in sign language detection.

Article Info

Received: September 10, 2021
Accepted: March 2, 2022

Keywords

Computer Vision,
Machine Learning,
CNN, Sign Language,
Gesture Detection

Abstract

Sign Language plays a crucial role in assisting individuals with hearing impairments, enabling them to effectively communicate and interact with the hearing population. While it serves as a communication tool for both the hard-of-hearing and general communities, the complexity of sign language and gestures makes it challenging for non-signers to comprehend. Various tools have been developed to translate sign language into a format that can be understood by the general population. These tools involve the acquisition of images, pre-processing, hand gesture segmentation, feature extraction, and classification methods. In Sign Language Recognition (SLR), cameras serve as the primary component for translating sign language. Existing SLR systems predominantly employ high-quality cameras, but to increase accessibility, this paper suggests using commonplace cameras such as smartphones and webcams, which have improved over time and perform well in our proposed work due to the availability of more suitable datasets. This paper emphasizes the importance of integrating intelligent solutions into SLR systems and aims to fulfill the need for a universally accessible SLR web application. It is anticipated that this study will foster the development of intelligent-based SLR, contribute to the accumulation of knowledge, and provide researchers, practitioners, and readers with a roadmap for future advancements in this field.

1. INTRODUCTION

Sign language is a kind of communication that conveys meaning visually using facial expressions, hand gestures, and body language. For those who have trouble hearing or speaking, sign language is quite helpful. The gesture translation into the alphabets or words of formal spoken languages that already exist is called sign language recognition (SLR). Therefore, translating sign language into words using an algorithm or model can aid in closing the communication gap between those with hearing or speech disability and the rest of society. Computer vision and machine learning researchers are now conducting intensive research in the area of image-based hand gesture identification. Many researchers are working in this field since it is a natural method of human connection, having the aim of creating the interaction of human and computer, simple and natural by cutting off the need of additional gadgets. Therefore, the main objective of research in gesture recognition is to develop systems that are able to recognize and utilize particular human gestures. For example, to deliver information. For the same, interfaces that are vision based, need quick, extremely reliable hand detection as well as real-time gesture identification for this. Sign Language is a popular & effective way to facilitate the deaf and dumb community, but the primary limitation of lack of sign language interpreters results in non-availability and non-reachability of the interpreters. The expected achievement is to contribute towards research and development towards solutions for deaf and dumb community. The aim of this paper is to develop a Sign Language Recognition System accessible via a web portal, which takes video input and predicts hand actions into signs, achieving higher accuracy than existing solutions. Within this frame of reference, we have sign language recognition (SLR), the mode of communication used

*email: manak2019it008abesit@gmail.com, kaustubhranjan7@gmail.com

by the deaf and the silent. Hand gestures constitute a potent human communication method that has many possible uses.

.

1.1 Objective of Proposed Work

The main goal of proposed method is described as follows.

- To develop own benchmark Sign Language Dataset.
- To develop easy access application for Sign Language Detection.
- To achieve better accuracy at sign detection than existing solutions.

The rest of the paper from Section 2 to Section 7 is structured as: Pertinent works, tools and technique, proposed work, experimental results, advantage of proposed work, and conclusion respectively.

2. PERTINENT WORK

2.1 Image Acquisition

Cameras serve as essential tools for image acquisition in various domains, including photography, computer vision, and scientific research. On the other hand, webcams and smartphone cameras are designed for convenience, allowing users to capture images and videos quickly and easily. [1], [2]Camera/Webcam eliminates the requirement of equipping other devices and is easy to use, making it comparatively the best image acquisition method. A data glove is a cutting-edge and adaptable picture collection device that makes it possible to record hand and finger movements in three dimensions. It is a glove-like device with sensors and trackers that precisely tracks the wearer's hand and fingers' location and orientation.[3]Data glove is a feasible and easy feature extraction device as it prevents influence from outside environment. They are highly expensive with low convenience and fails to maintain originality of interaction. Kinetic is another image acquisition device which has high usage in various applications involving human computer interaction with limited range depth detection. By capturing a subject's motion and dynamics, kinetic imaging is a novel and cutting-edge method for acquiring images. Kinetic imaging, as opposed to conventional static photography, focuses on capturing the motion, gestures, and changes in an item or scene through time. The Leap Motion Controller was created specifically to capture and track hand and finger movements in three-dimensional space with extraordinary accuracy and speed, unlike conventional cameras or sensors. Leap motion controller has elevated speed processing with high accuracy in recognition, that facilitates in detecting hand and fingers.

2.1 Image Enhancement

[4]Using Histogram Equalization (HE) for images in grayscale gets best result and it has simple implementation. The downside is that image brightness gets modified due to which featured and noise become hard to distinguish. [5]Adaptive Histogram Equalization (AHE) has better performance than histogram equalization and is best suited to amplify edges and local contrast of image. By redistributing the pixel intensities, it is a potent image enhancement technique that seeks to boost the contrast and visibility of images. It is a potent approach of image enhancement that overcomes the drawbacks of conventional histogram equalization techniques. By shifting pixel intensities based on the local image content, CLAHE improves the contrast and details in an image.[6]Original brightness is retained in Contrast Limited Adaptive Histogram Equalization, and reduced noise can be noticed when comparing to HE and AHE. [7]Logarithmic Transformation is useful when high intensity pixel values are to be reduced into lower intensities pixel values. It is a key method used in image processing to increase an image's visual quality and dynamic range.

2.1 Image Filtering

[8]Mean filter implementation is easy, but shows significant impact in incorrect pixel value representation. In order to remove noise and smooth images, the mean filter is a popular and straightforward image filtering technique. It functions by substituting each pixel in an image with the average value of those closest to it. A prominent method in image processing for decreasing noise and maintaining significant image features is the median filter.[9]It counters problem of mean filter by retaining image sharpness and thin edges. The median filter, in contrast to linear filters like the mean filter, concentrates on the statistical order rather than the numerical average of the pixel values. [10]Gaussian type of noise is best removed using Gaussian filter. The Gaussian kernel, which is a matrix of values deriving from the Gaussian function, is convolved with an image in order for the Gaussian filter to function. This filter successfully blurs the image while keeping crucial structural information, making it mostly used to decrease noise and smooth images. Adaptive filters, as opposed to conventional filters with fixed coefficients, continuously adjust their coefficients or weights to reduce mistakes and enhance the desirable signal components. They are especially helpful in situations when the signal qualities change over time or when there is noise or interference because of their versatility.[11]Adaptive filter is better at preserving high frequency parts like edges than linear filter. [12]Wiener filter is a well-liked picture repair filter. Noise has no effect on it, so it is suitable for utilizing the image's statistical features. A signal or image's quality is improved by the Wiener filter by minimizing he mean square error between the original signal and the filtered version.

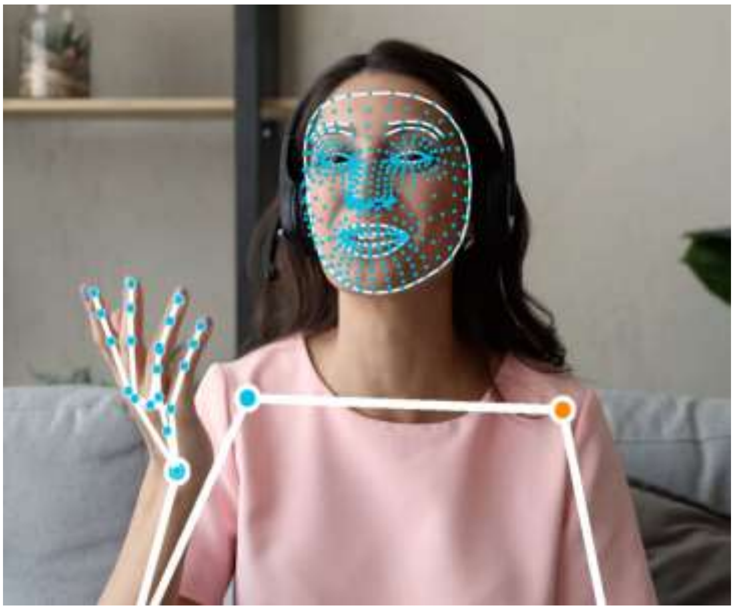
2.1 Image Segmentation

[13]Thresholding method as segmentation technique is an efficient and simple strategy. It can be used without any prior knowledge and requires less computation power. Thresholding is a widely used method for image segmentation, which seeks to distinguish objects or regions of interest from the background based on their pixel intensities. [14]Edge based method is suitable for pictures with better object contrast. The goal of edge-based methods for image segmentation is to recognize and define boundaries between various objects or regions within an image.[15]Region based method is more useful and less sensitive to noise when the similarity criteria are simple to define. This technique concentrates on segmenting an image into useful areas or parts based on how similar the pixels are within those areas. This technique divides the image into logical and visually meaningful portions by using clustering algorithms. Finding groups of pixels with similar properties, such as colour, texture, or intensity, is the fundamental idea.[16]Because of the usage of the fuzzy partial membership, Clustering method is more applicable to real-world problems.[17]Artificial Neural-Network based method can function without a complicated program, it is less noisy in nature and have become effective tools for picture segmentation problems.

3. TOOLS AND TECHNIQUES USED

3.1 Mediapipe Holistic

MediaPipe Holistic, developed by Google, is an advanced computer vision solution that enables real-time tracking and comprehension of human body movements and poses. By integrating multiple AI models and technologies, it offers a comprehensive approach that encompasses body tracking, hand tracking, and facial landmark detection capabilities.The foundation of MediaPipe Holistic is a deep learning-based model that can estimate the 3D pose of a person from a 2D video stream. This model combines convolutional neural networks (CNNs) and geometric reasoning techniques to accurately track the positions and orientations of various body parts, including the head, torso, arms, and legs. It demonstrates robust and reliable tracking performance, even in challenging scenarios involving occlusions and varying camera viewpoints.In addition to body tracking, MediaPipe Holistic includes a dedicated module for hand tracking. This module excels at detecting and tracking the 3D positions of the user's hands in real-time. This functionality is particularly useful for applications such as sign language interpretation, virtual reality experiences, and augmented reality interactions.

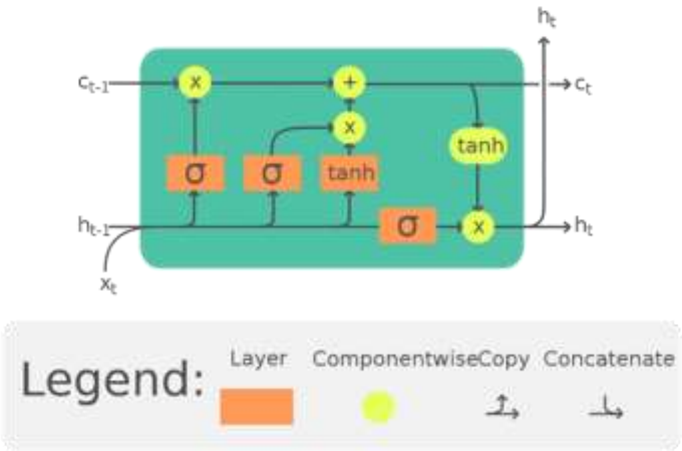


[18]**Figure 1.** Keypoints displayed using MediaPipe.

Moreover, MediaPipe Holistic incorporates a facial landmark detection module that estimates the 2D positions of crucial facial landmarks, including the eyes, nose, and mouth. This capability opens up possibilities for applications requiring facial analysis, emotion recognition, and augmented reality effects on the user's face. By seamlessly integrating these three modules, MediaPipe Holistic offers a holistic understanding of human movements and interactions. The solution is optimized for real-time performance, making it ideal for a wide range of applications, including fitness tracking, gesture-based interfaces, virtual try-on experiences, and immersive gaming. MediaPipe Holistic is built upon the MediaPipe framework, which provides a flexible and scalable pipeline for developing and deploying real-time computer vision applications. The framework offers a diverse array of pre-built components and processing modules, empowering developers to easily customize and extend the functionality of MediaPipe Holistic to suit their specific requirements.In summary, MediaPipe Holistic is an impressive computer vision solution that leverages body tracking, hand tracking, and facial landmark detection to achieve real-time understanding of human movements and poses. Its versatility, accuracy, and real-time performance make it a valuable tool in diverse fields, including entertainment, healthcare, and human-computer interaction.

3.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an architecture of recurrent neural networks (RNNs) specifically designed to overcome the limitations encountered by traditional RNNs in capturing long-term dependencies and preserving information across lengthy sequences. LSTM networks excel in processing and predicting sequential data, such as time series, speech, and natural language.The distinctive feature of LSTM lies in its ability to selectively retain or discard information from previous time steps using specialized memory cells and gates. These gates, namely the input gate, forget gate, and output gate, allow LSTM to regulate the flow of information within the network, effectively addressing challenges like the vanishing and exploding gradient issues often faced by traditional RNNs.The LSTM architecture comprises a series of memory cells responsible for storing and updating information over time. Each memory cell maintains a cell state, which ensures the preservation of information throughout the sequence. Additionally, three gates—input gate, forget gate, and output gate—govern the information flow. The input gate controls the amount of new information stored in the cell state, the forget gate determines the extent of discarding previous information, and the output gate governs the extraction of relevant information from the current cell state.



[19]**Figure 2.** LSTM Cell.

During training, the parameters of an LSTM network are learned through backpropagation through time, which propagates gradients from the output to the input across the entire sequence. This enables the network to adjust its internal state and update the memory cells, facilitating the effective capture of dependencies and patterns within the sequential data. LSTMs have demonstrated remarkable effectiveness in various tasks. In natural language processing, LSTM models have been successfully applied to language modeling,

3.3 Open CV

OpenCV, an open-source computer vision and machine learning software library, offers a comprehensive range of tools and algorithms for image and video processing, analysis, and comprehension. Developed in C++, it provides interfaces for various programming languages, including Python and Java.With its extensive collection of functions and modules, OpenCV facilitates numerous tasks such as capturing and preprocessing images and videos, detecting and extracting features, recognizing objects, calibrating cameras, and performing geometric transformations. It supports multiple image formats and delivers efficient algorithms for filtering, segmentation, and edge detection. Additionally, OpenCV includes methods for motion estimation, camera tracking, and optical flow analysis. OpenCV's greatest strengths lie in its versatility and adaptability. It finds applications in robotics, augmented reality, surveillance systems, facial recognition, and medical imaging. Its comprehensive documentation, vibrant community support, and cross-platform compatibility contribute to its popularity in computer vision projects.

[20]**Figure 3**. OpenCV logo.

Furthermore, OpenCV seamlessly integrates with renowned machine learning frameworks like TensorFlow and PyTorch, allowing users to combine the power of deep learning with computer vision algorithms. This integration enables tasks such as object detection, image classification, and semantic segmentation. Overall, OpenCV empowers developers and researchers to build advanced applications and systems in computer vision and machine learning domains. Its rich set of functions and algorithms make it a powerful tool for a wide range of computer vision tasks.

3.4 TensorFlow

TensorFlow, a machine learning framework developed by Google, is an open-source platform that offers a comprehensive suite of tools, libraries, and resources for the creation and deployment of machine learning models. It empowers developers to efficiently design, train, and deploy a diverse array of neural networks and machine learning algorithms. At its core, TensorFlow adopts a computational graph paradigm, where computations are represented as interconnected nodes that denote mathematical operations, and the edges symbolize the flow of data between these operations. This graph-based approach enables TensorFlow to leverage parallelism and effectively distribute computations across multiple devices, such as CPUs, GPUs, and TPUs, thereby accelerating the training and inference processes. It encompasses a wide spectrum of applications, including deep learning, reinforcement learning, natural language processing, and computer vision. Simplifying the development and training of neural networks, TensorFlow provides a high-level API called Keras.

[21]**Figure 4**. TensorFlow logo.

Another compelling aspect of TensorFlow is its extensive collection of pre-built models and tools. Within the TensorFlow Model Zoo, developers can access a repository of pre-trained models that cover various tasks, enabling efficient transfer learning and adaptation to specific applications. Moreover, TensorFlow includes TensorFlow Extended (TFX), a production-ready platform for the deployment and management of machine learning pipelines at scale. Furthermore, TensorFlow seamlessly integrates with popular libraries and frameworks, such as OpenCV for computer vision tasks and NumPy for numerical computations. It boasts compatibility with multiple programming languages, including Python, C++, and JavaScript, ensuring accessibility and adaptability across diverse environments. The TensorFlow community is highly active and fosters continual growth of the framework. Abundant resources such as documentation, tutorials, and online forums are readily available, promoting knowledge sharing and collaborative development. In summary, TensorFlow is an influential and flexible machine learning framework, effectively facilitating the construction and deployment of machine learning models. Its computational graph approach, rich selection of pre-built models, and compatibility with various languages and frameworks contribute to its popularity and make it a top choice for diverse machine learning tasks.

3.5 Numpy

NumPy, also known as Numerical Python, is a crucial open-source library designed for numerical computations in the Python programming language. It offers efficient data structures and functions that facilitate the manipulation of large, multi-dimensional arrays and matrices, along with a diverse range of mathematical operations for data analysis and manipulation. At its core, NumPy revolves around the ndarray (N-dimensional array) object, which enables optimized storage and manipulation of homogeneous data. The ndarray provides a robust interface for executing mathematical operations on entire arrays or individual elements. It supports various data types, exhibits efficient memory management, and boasts broadcasting capabilities that enable element-wise operations and mathematical computations across arrays with different shapes and sizes.NumPy encompasses an extensive collection of mathematical functions dedicated to array manipulation. These functions include arithmetic operations, linear algebra routines, Fourier transforms, random number generation, and statistical calculations. The performance of these functions has been optimized, making NumPy an indispensable tool for numerical computing tasks.

[22]**Figure 5**. NumPy logo.

A significant advantage of NumPy lies in its seamless integration with other scientific libraries within the Python ecosystem. It serves as the foundation for numerous high-level libraries such as Pandas, SciPy, and scikit-learn. These libraries build upon NumPy's functionality to provide advanced data manipulation, scientific computing, and machine learning capabilities. The simplicity and efficiency of NumPy make it an invaluable resource across a wide array of applications. It finds extensive use in scientific research, data analysis, machine learning, image processing, and signal processing. NumPy's intuitive syntax and comprehensive documentation cater to both novice and experienced programmers alike. The vibrant community surrounding NumPy ensures continuous development and support, with regular updates and contributions from a diverse group of users and developers. This active participation has solidified NumPy's popularity, stability, and high-performance, establishing it as a fundamental library within the Python ecosystem for numerical computations. In conclusion, NumPy is a potent library that provides efficient data structures and functions tailored for numerical computations in Python. Its ndarray object and extensive range of mathematical operations enable efficient manipulation of large arrays and matrices. NumPy's integration with other scientific libraries positions it as a cornerstone for various applications, making it an indispensable component of the scientific Python ecosystem.

4. PROPOSED SOLUTION

We propose the use of LSTM model for increased accuracy for sentence formation with sign language, with webcam as the primary source of image acquisition method, for both, testing and training of data.

4.1. Architecture

The architectural diagram in Figure 6 describes working infrastructure of the proposed work. It depicts a user logging in to the web portal provided for the user interaction and after accepting requests for required permissions, it briefly introduces to the functioning of camera as the primary means of interplay between detection system and user. Fleeting operations including pre-processing, key points extraction and database accessing takes place. Data from database is used for training and testing, where testing data is also the feeded input data that is converted from gestures to signs and finally the user experience is received as feedback.

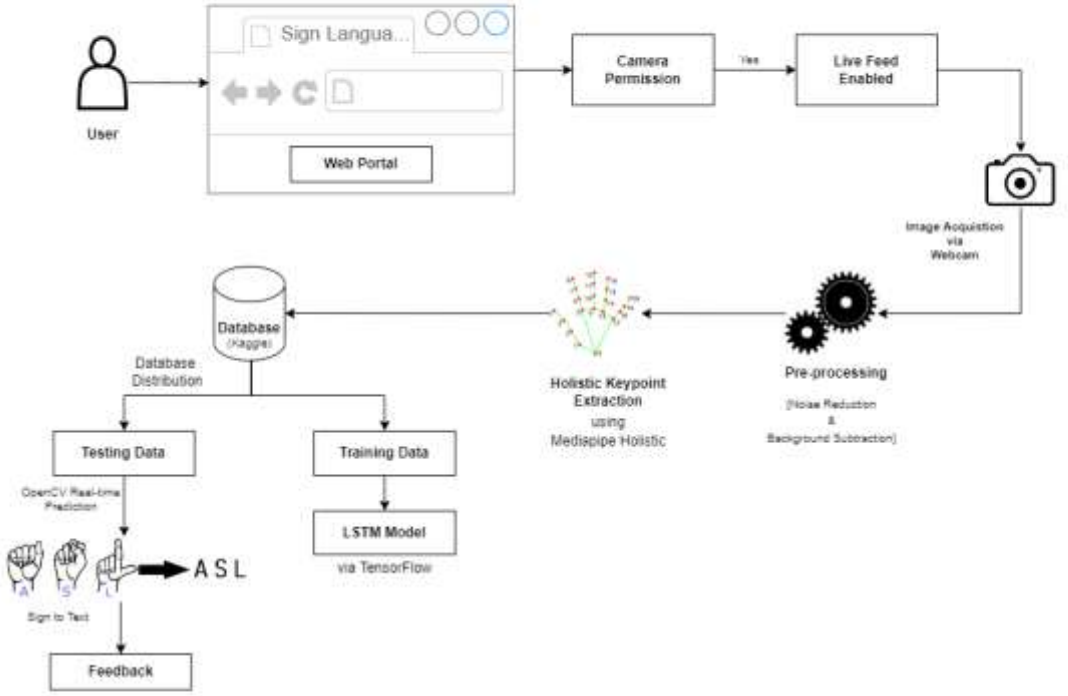


Figure 6. Architecture of proposed method.

4.2. Sequence Diagram

Shown above in Figure 7 is the sequence diagram which elaborates the working progression of the proposed work. Initially, the data as images is inputted by the user on the web portal. This data is stored in database and also used as training data. It is pre-processed and the model is trained. Trained model is saved and used on the same data to give output. The model is loaded and the inputted images after pre-processing is sent back to be classified. Finally, the output is received by the user in displayed text format.

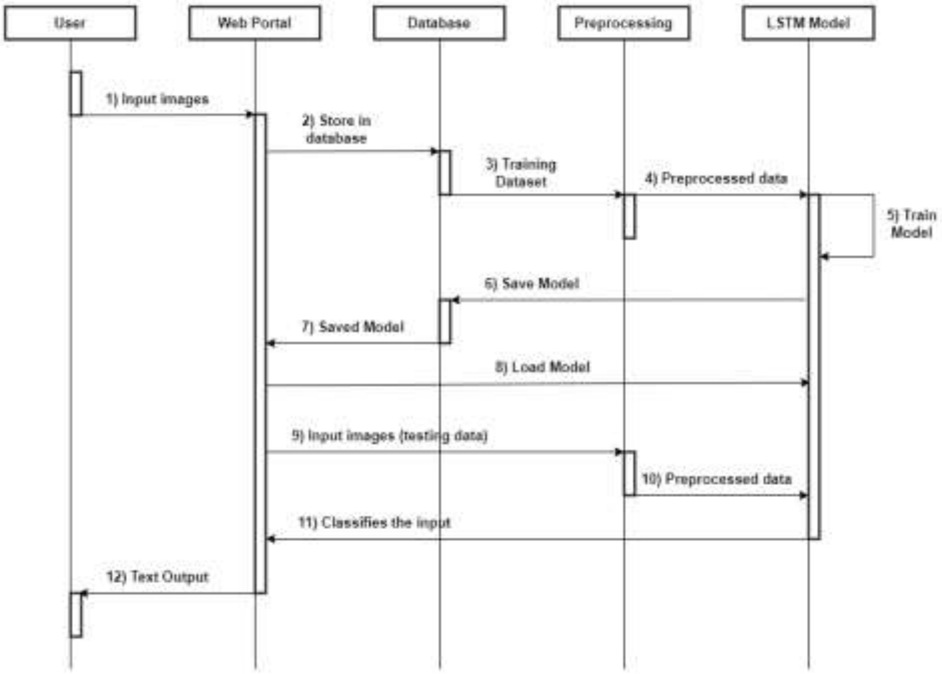


Figure 7. Sequence Diagram for Sign Language Detection.

4.3. Dataset Creation

The dataset used in this model was prepared exclusively by the use of webcams and smartphone cameras in order to achieve higher accuracy. The primary purpose of making own dataset was to mimic the wide variety of aspects that affect gesture recognition, such as noise level. The training was based on captured images by 2 people's hands, allowing a sense of variation for the generic purpose involving replication of actual webcam/smartphone camera like gesture.

A collection of digital photographs that has been assembled and organized for a particular use, such as testing and assessing computer vision or machine learning algorithms, is known as an image dataset. Image datasets are essential for creating and testing image-related applications because they offer a large variety of photos that represent various visual contexts and objects. One of our objectives was to create webcam and mobile camera generated datasets to achieve better results when predicting noise induced hand gestures. Following are instances of gestures we used for creating our dataset.

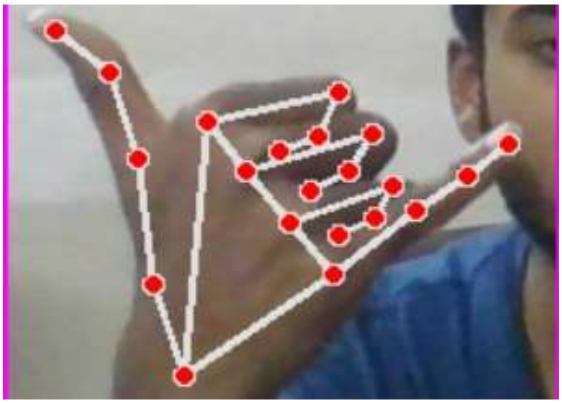


Figure 7. Hand gesture for the word 'Angry'.

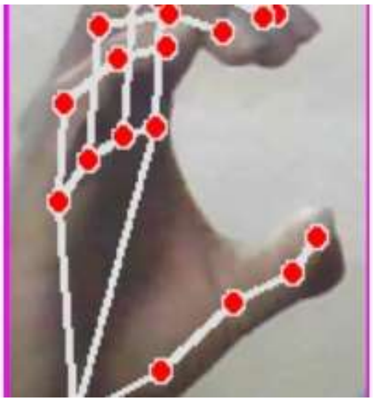


Figure 8. Hand gesture for the word ‘Christmas’.

Figure 9. Hand gesture for the word ‘Disciple’.

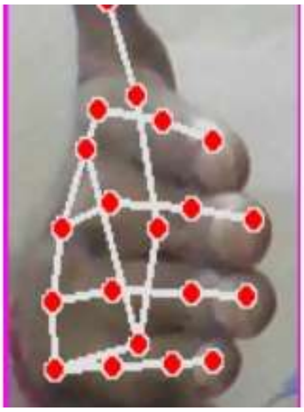


Figure 10. Hand gesture for the word ‘Major’.

Figure 11. Hand gesture for the word ‘Really’

Training of the model was done using these simple hand gestures, which sometimes included use of both hands also. For example, the hand gesture for the word ‘house’ required both of the hands to meet together at the apex of middle finger, which depicts the housetop.

4.4. Pseudocode

MP Holistic- Media Pipe Holistic combines distinct model of hands, poses and face.
Keypoints- Landmarks for object detection.
LSTM- Long Short-Term Memory Model processes sequence of data like video input.

1) import cv2, numpy as np, os, pyplot as plt, time, medipipe as mp
2) def mediapipe_detection, draw_landmarks, draw_styled_landmarks
3) capture video
    3.1) cap.read()
    3.2) draw_styled_landmarks()
    3.3) cv2.imshow()
    3.4) cap.release
4) extract_keypoints(results)
5) collect data as data_path = os.path.join()
6) save keypoint values as mp_holistic.Holistic()
    6.1) keypoints=extract_keypoints()
    6.2) npy_path=os.join()
    6.3) np.save(npy_path,keypoints)
7) import train_test_split, to_categorical for preprocess
8) import sequential ,LSTM,dense, tensorboard from tensorflow.keras for LSTM network
9) res=model.predict(X_test) for predictions
10) def model and model.save() saves weights
11) import accuracy_score , stats for accuracy statistics
12) make predictions
    12.1) keypoints=extract_keypoints()
    12.2) sequence.append(keypoints)
    12.3) sequenct=sequence[-30:]

5. EXPERIMENTAL RESULTS

In order to evaluate the accuracy of our trained gesture prediction model, we conducted a comprehensive test involving 10 participants. The objective was to compare the predicted gestures generated by our model with the actual gestures performed by the participants. By analyzing the results, we aimed to assess the effectiveness and reliability of our trained model. Each participant was assigned a unique identifier, ranging from 1 to 10. During the testing phase, we captured the gestures using advanced motion-tracking. This data was then fed into our trained model, which utilized machine learning algorithms to predict the gestures based on the input signals. Following are the test results obtained.

Figure 8. Number of times words predicted correctly

Word	Serial No. of Persons (Person abbreviated as “Per.”)									
	Per.1	Per.2	Per.3	Per.4	Per.5	Per.6	Per.7	Per.8	Per.9	Per.10
Sometimes	9	10	9	10	10	10	10	10	10	10
Angry	6	7	7	7	5	7	7	6	7	6
Christmas	9	9	10	9	9	10	10	10	10	10
Disciple	9	8	9	8	8	10	9	10	9	10
House	10	10	8	10	8	10	6	9	8	8
Small	8	8	8	7	6	7	7	8	8	7
Narrow	8	9	9	10	10	9	8	10	7	10
Really	9	7	7	8	10	8	7	8	8	8
I(me)	8	7	7	9	8	8	8	7	7	8
Major	9	9	10	9	9	9	10	10	8	9
Love	8	7	9	8	7	8	9	9	6	7

During our evaluation of the trained gesture prediction model, we examined the accuracy on a person-by-person basis for 11 different words. This analysis allowed us to gain insights into the model's performance and its ability to correctly predict specific gestures for each participant. For each participant, we recorded the accuracy percentage for each word they performed. The accuracy percentage represents the proportion of correct predictions made by the model for a particular word. Following mentioned are the results.

Person No.	Accuracy (in %age)
Person 1	83.63
Person 2	82.70
Person 3	84.54
Person 4	86.30
Person 5	81.80
Person 6	87.27
Person 7	82.72
Person 8	88.18
Person 9	80.00
Person 10	84.54

Figure 9. Person wise accuracy for 11 words.

During our evaluation of word prediction accuracy, we tested our trained model on a diverse group of 10 individuals. The objective was to assess how well the model performed in predicting specific words based on the participants' gestures. For each participant, we recorded the accuracy of word prediction by comparing the model's predicted word to the actual word performed by the participant. The accuracy was calculated as a percentage, representing the proportion of correct predictions made by the model. Results observed are as follows.

Word	Accuracy (in %age)
Sometimes	98
Angry	65
Christmas	96
Disciple	90
House	91
Small	74
Narrow	90
Really	80
I(me)	77
Major	92
Love	78

Figure 9. Accuracy of word prediction when tested on 10 persons.

6. ADVANTAGE OF PROPOSED MODEL

- It promotes more realistic datasets rather than real dataset.
- Sign Language System availability and accessibility.
- Empowers the deaf and dumb community even if the other person is not aware of sign language.

7. CONCLUSION

Our paper produces 84.63% accuracy when tested on 10 different people with multiple hand gestures. Sign Language Recognition is a subject of in-progress research which still has no extensive deployed system. This paper provides computer vision-based solution with better accuracy and accords to the associated knowledge domain. Literature Survey of this paper focuses on use of distinct methods for the solutions of gesture recognition and provides various perspectives on use of dissimilar methods and techniques. Future research will include relative analysis of this study with other existing and newly gleaned solutions incorporated with ever-evolving technologies.

CONFLICT OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] N. Verma, M. D.-J. of Engineering, T. (IRJET), and undefined 2017, "Contrast enhancement techniques: A brief and concise review," *academia.edu*, 2017, [Online]. Available: https://www.academia.edu/download/54173429/IRJET-V4I7375.pdf

[2] S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, "Technical Approaches to Chinese Sign Language Processing: A Review," *IEEE Access*, vol. 7, pp. 96926–96935, 2019, doi: 10.1109/ACCESS.2019.2929174.

[3] Suharjito, R. Anderson, F. Wiryana, M. C. Ariesta, and G. P. Kusuma, "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 441–448. doi: 10.1016/j.procs.2017.10.028.

[4] A. Chourasiya and N. Khare, "A Comprehensive Review Of Image Enhancement Techniques," *International Journal of Innovative Research and Growth*, vol. 8, no. 6, May 2019, doi: 10.26671/ijirg.2019.6.8.101.

[5] M. Dhanushree, R. Priyadharsini, and T. S. Sharmila, "Acoustic image denoising using various spatial filtering techniques," *International Journal of Information Technology 2019 11:4*, vol. 11, no. 4, pp. 659–665, Nov. 2019, doi: 10.1007/S41870-018-0272-3.

[6] F. Kasmin, "A Comparative Analysis of Filters towards Sign Language Recognition," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 4772–4782, Aug. 2020, doi: 10.30534/ijatcse/2020/84942020.

[7] M. Basu, "Gaussian-based edge-detection methods - A survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 32, no. 3, pp. 252–260, Nov. 2002, doi: 10.1109/TSMCC.2002.804448.

[8] R. Kaluri and C. Pradeep Reddy, "A framework for sign gesture recognition using improved genetic algorithm and adaptive filter," *Cogent Eng*, vol. 3, no. 1, p. 1251730, Oct. 2016, doi: 10.1080/23311916.2016.1251730.

[9] R. Kaluri and C. H. Pradeep Reddy, "Sign gesture recognition using modified region growing algorithm and Adaptive Genetic Fuzzy Classifier," *International Journal of Intelligent Engineering and Systems*, vol. 9, no. 4, pp. 225–233, 2016, doi: 10.22266/ijies2016.1231.24.

[10] S. Tania and R. Rowaida, "A Comparative Study of Various Image Filtering Techniques for Removing Various Noisy Pixels in Aerial Image," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 3, pp. 113–124, Mar. 2016, doi: 10.14257/ijsip.2016.9.3.10.

[11] M. Maru, P. G. Scholar, and M. C. Parikh, "Image Restoration Techniques: A Survey," 2017.

[12] S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Comput Vis Graph Image Process*, vol. 52, no. 2, pp. 171–190, Nov. 1990, doi: 10.1016/0734-189X(90)90053-X.

[13] H. D. Cheng, X. H. Jiang, and J. Wang, "Color image segmentation based on homogram thresholding and region merging," *Pattern Recognit*, vol. 35, no. 2, pp. 373–393, Nov. 2002, doi: 10.1016/S0031-3203(01)00054-1.

[14] W. Xu, Q. Li, H. J. Feng, Z. H. Xu, and Y. T. Chen, "A novel star image thresholding method for effective segmentation and centroid statistics," *Optik (Stuttg)*, vol. 124, no. 20, pp. 4673–4677, Nov. 2013, doi: 10.1016/J.IJLEO.2013.01.067.

[15] L. Dong, G. Yu, P. Ogunbona, and W. Li, "An efficient iterative algorithm for image thresholding," *Pattern Recognit Lett*, vol. 29, no. 9, pp. 1311–1316, Nov. 2008, doi: 10.1016/J.PATREC.2008.02.001.

[16] Rashmi, M. Kumar, and R. Saxena, "Algorithm and Technique on Various Edge Detection : A Survey," *Signal Image Process*, vol. 4, no. 3, pp. 65–75, Jun. 2013, doi: 10.5121/sipij.2013.4306.

[17] S. Bhardwaj and A. Mittal, "A Survey on Various Edge Detector Techniques," *Procedia Technology*, vol. 4, pp. 220–226, 2012, doi: 10.1016/j.protcy.2012.05.033.

[18] "GitHub - anujjain22/Holistic-detection-with-python: In this Project Holistic Detection, it detect Left and Right Hand, Face Mesh, and Pose Detection using Libraries Opencv and Mediapipe in Python." https://github.com/anujjain22/Holistic-detection-with-python (accessed Jun. 12, 2023).

[19] "File:LSTM Cell.svg - Wikimedia Commons." https://commons.wikimedia.org/wiki/File:LSTM_Cell.svg (accessed Jun. 12, 2023).

[20] "File:OpenCV Logo with text svg version.svg - Wikimedia Commons." https://commons.wikimedia.org/wiki/File:OpenCV_Logo_with_text_svg_version.svg (accessed Jun. 12, 2023).

[21] "Brief Introduction to Tensorflow for Deep Learning - Analytics Vidhya." https://www.analyticsvidhya.com/blog/2022/01/brief-introduction-to-tensorflow-for-deep-learning/ (accessed Jun. 12, 2023).

[22] "File:NumPy logo 2020.svg - Wikimedia Commons." https://commons.wikimedia.org/wiki/File:NumPy_logo_2020.svg (accessed Jun. 12, 2023).