Enhancing User Experience through Intelligent Hand Gesture Identification Systems

[1]Jayashree R Pansare, [2]Sanyam Kudale, [3]Rohit Jadhav, [4]Swaraj Giri, [5]Ashish Kumar Pandey

Modern Education Society's Wadia College of Engineering, Pune

Abstract-- A This work aims to explain a new technique to hand gesture detection that offers a natural, creative, and contemporary way of nonverbal communication, with many applications in human-computer interaction as well as sign language. The user's gesture is captured by a single camera in this arrangement, and the hand image is used as input by the suggested algorithm. Tested on 780 photos, the constructed algorithm demonstrates its efficiency compared to previous methods with an approximate 97% recognition rate at an average elapsed time of 2.76 seconds.

Keywords-- Open-cv, Real-Time detection, Hand gesture identification, Deep learning, Gesture-based Interface

1. Introduction

Hand gesture identification is a fascinating field within computer vision and human-computer interaction (HCI) that aims to interpret human hand movements and gestures captured by cameras or sensors. It allows users to interact with devices or interfaces in a natural and intuitive way without the need for physical controllers or touch-based input. The prime objective of hand gesture recognition is for understanding and unravelling the meaning or intent behind various hand movements made by an individual. This can involve recognizing predefined gestures representing specific commands or actions, such as pointing, waving, thumbs-up, or complex sign language gestures.

Hand gesture recognition systems typically involve several stages, including hand detection, tracking, feature extraction and gesture categorisation. Various techniques and algorithms are used at each stage, extending from traditional computer vision approaches to deep learning methods. Utilizes Human-Machine Interface (HMI) to capture hand signs. Machine learning procedures are employed to recognize signs from images, with MATLAB's image analysis tools being commonly used for image analysis. The study employs ML algorithms to extract motion and signs, with color data also collected. ML algorithms undertake a pivotal role in gesture recognition technology [1]. Using a camera for input videos, RGB to HSV conversion, and centroid detection. This algorithm enables hand gesture recognition across various datasets, regardless of background restrictions. It facilitates easy hand identification and tracking under ordinary conditions [2].

Uses OpenCV and MATLAB, an Arduino stores data and controls servo motors based on gesture recognition commands. The system implements vision-based gesture recognition to interpret hand movements captured by a camera, enabling replication of gestures using a mechanical hand [3]. High resolution images in RGB are compressed to low resolution images in binary format by a preprocessing unit. Achieving an accuracy of approximately 99.84%, the framework consolidates a power-proficient acknowledgment center with two minimal classifiers for static motion acknowledgment and a larger part casting a ballot based grouping analyzer for dynamic sign acknowledgment[4].

A CNN classifier was employed for dynamic gesture recognition. They introduced a spatio-temporal data augmentation method and demonstrated its combination with spatial transformations. The classifier achieved a test set accuracy of 98.74%.The dataset used for training and for testing was 70% and 30% respectively[5]. The impact of body position on motion acknowledgment is investigated utilizing Ultra Wideband (UWB) innovation. The framework utilizes a receiver and transmitter working inside an expansive recurrence range with short heartbeat length, ideal for metropolitan circumstances. Results show better execution while sitting looked at than standing situations in arranging and perceiving hand signals and corresponded speeds[6].

The following sections provide comprehensive insights. Section 2 gives an overview of various hand gesture identification techniques which are already present. Section 3 dives into the methodology part in which workflow, system architecture and algorithm is discussed. 4th Section displays experimental data that shows the performance and efficiency of the system. The conclusion is given in Section 5 providing ultimate outcomes.

2. Various Hand Gesture Identification Techniques

Hand Gesture Identification for the Blind Using Machine Learning Algorithms maps various gestures to specific commands for controlling fans and lights. For instance, a single finger signifies turning on the fan, while two fingers indicate turning it off, and so forth. The system utilizes two hand gestures: open and closed hand, with the ability to accurately detect hand motions [7]. The methodology involves transforming the image to HSV color space, identifying skin pixels, applying background subtraction, enhancing edge detection, extracting hand contour, and so on. The system uses computer vision technology to control the mouse pointer using hand gestures and fingertip recognition [8]. The project aims to develop hand gesture identification system built on IOT. System recognizes signs and converts them into decimal numbers using conventional two-hand numbering symbols.

The authors developed a system to recognize signs representing numbers from 1 to 10 using a camera and a Raspberry Pi 3 [9]. They extricated objects features using the HSV color model, and afterward perceived the distinctions between various items through CNN. The program detects gestures effectively and converts gestures to text as well as speech and then opens website or audio file according to the gesture identified [10]. The proposed strategy perceives ceaseless hand signal utilizing three models: detector, pointing and non-pointing. The TD-Net model achieved accuracy of 84.98% and F1-score of 79.01%, surpassing other methods. The system worked better when the camera was not very clear and far from the hand. It also knew two-hand signs better than one-hand signs. It was quicker and more trustworthy than other systems that used colorful pictures [11].

The authors developed a system to know hand gestures using a camera and a computer. The system uses the YOLO model to find and name the hand signs. They used the YOLO-v5s model because it is faster and lighter than the YOLO-v4 model. They did not need any wearable device for the system. The system can help people who cannot hear or speak. The authors created a system to recognize hand signs and control a car using a camera and a computer [12]. The proposed method demonstrates utilization of computer vision and machine learning to identify gestures and apply them using Python with OpenCV and gives 99.04% and 99% accuracy for own and ASL dataset respectively [13]. Model prepared on 2250 hand datasets isolated into 3 classes for mouse development identification, the DNN offers a hand feature based on the normalized keypoint vector. The network design utilizes a multilayer structure with neuron sizes of (13, 12, 15, 14) and was trained over 500 epochs. For normalized features, it arrives at its best exactness of 98.5%. [14].

Model uses Convolutional Neural Networks to track the complex actions and help in extracting major features. An average time of about 721 milliseconds was observed for gesture recognition [15]. Utilizing a convolutional neural network, the model proposed in this paper aims to achieve its objectives, a deep learning architecture widely used for image and video analysis. It can recognize among a few predominant and low-level highlights present in the input pictures. With a negligible model loss of 0.0504, the network achieves greater accuracy in classifying various hand gestures. [16]. The classification was done by CNN, achieving 98.3% accuracy, in hand gesture recognition, which utilizes machine learning algorithms. It involves two approaches: appearance-based and model-based[17].

Our proposed TSE-GRU achieves higher accuracy compared to other methods by using six channels for data acquisition, amplification, and filtering of raw sEMG signals, thus enabling full extraction of information from the signals[19]. The data unequivocally indicates that YOLOv5 combined with MediaPipe Hands outperforms the other three models across the board for each of the four measures. Using the Flask framework, the web application accesses the laptop's local camera to provide video input at 30 frames per second. After that, each video frame is sent as an image to our server, where our model categorizes the motions and highlights important areas on the input photos. For every model, average class precision and class recall, F1 score and mAP@0.5 are shown as metrics [20]. Three primary processes comprise the video-based gesture recognition process: hand detection and segmentation, hand tracking, and hand classification and identification. Using keypoints for detection and tracking the hand, we create belief maps with Convolutional Pose Machines (CPMs). Following AlexNet classification, these belief maps are employed, and the classification outcomes are shown on the belief maps[21].

Our prototype, utilizing a Convolutional Neural Networks based classifier guided via transfer learning on a pre-trained convolutional neural net, was put to the test across various backgrounds by seven untrained volunteers, each performing all hand gestures. Upon comparison with the AlexNet CNN architecture, our recorded accuracy stands at an impressive 93.09%, surpassing AlexNet's overall accuracy of 76.96%[22]. In the realm of computer applications, hand gestures present a captivating avenue for interaction. Dynamic hand gesture recognition demands greater computational resources when juxtaposed with static gestures. The framework proposed for hand gesture recognition entails harnessing data from the Kinect to retrieve depth information for every pixel within an image. This methodology streamlines the process of dynamic gesture identification[23]. This research makes use of the CamShift algorithm, which is well-known for its ability to track hand gestures even when the targets undergo deformations. This algorithm's simplicity allows for the real-time identification of hand gesture positions. Real-time acquisition of hand gesture areas is ensured by the system by utilizing the CamShift algorithm's tracking capacity based on movement and deformation aspects of hand gestures. The method is then finished by classifying the gesture areas using a CNN.[24].

3. Methodology

This algorithm plans the entire method of building machine learning based hand gesture identification system, covering data preparation, model training, evaluation, and real-time gesture recognition. It provides a comprehensive guide for implementing such a system, enabling users to understand and replicate the workflow effectively.

Below is workflow for machine learning based hand gesture identification system:

A. Initialization

Import necessary libraries and modules for data processing, model creation, and evaluation.

B. Data Preparation

Data preparation involves following steps: Define the actions (hand gestures) to be recognized and map them to numerical labels. Load hand gesture sequences from saved NumPy arrays, where each sequence contains hand gesture data captured over a period. Using the train_test_split function, divide the dataset in two sets (i.e. training and testing). Prepare target labels by encoding them into categorical format using one-hot encoding.

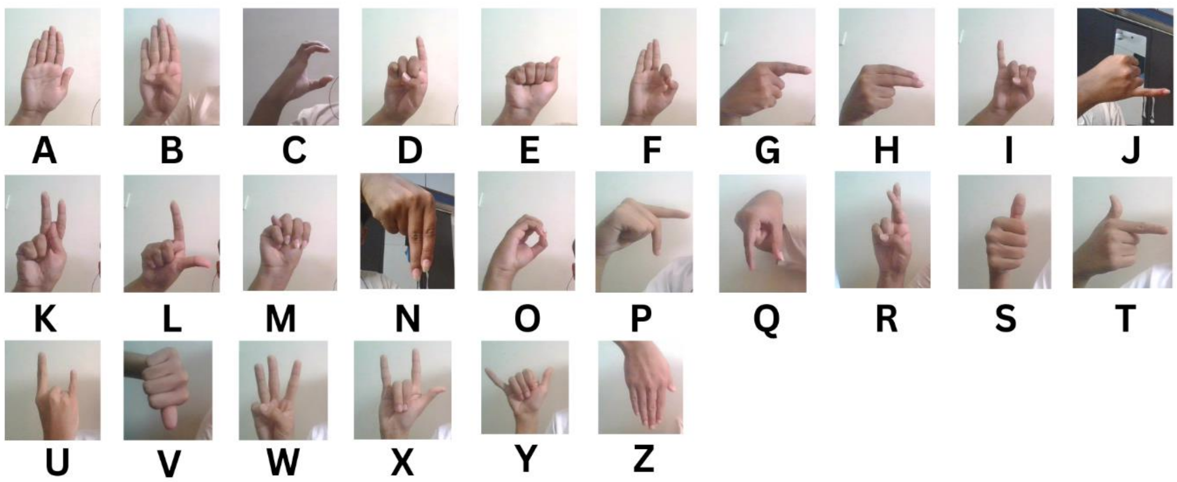

Fig.1. American Sign Language Symbols

C. Model Training

Create a sequential neural network model using Keras, comprising LSTM layers followed by densely connected layers. First, compile the model with the Adam optimizer and categorical cross-entropy loss function. Then, utilize the TensorBoard callback to log training metrics for visualization and analysis. Next, train the model on the training data (X_train and y_train) for a stated number of epochs. This sequence ensures a structured approach to model training and evaluation, enhancing clarity and efficiency.

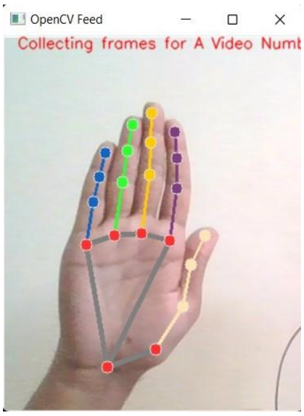

Fig.2

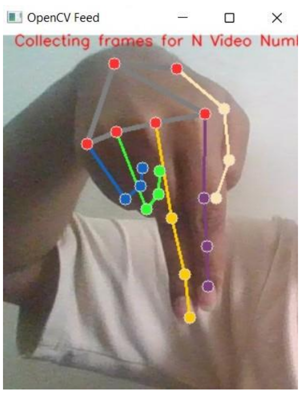

Fig.3

Fig.4

D. Model Evaluation and Saving

Print the summary of the trained model, displaying its architecture and parameter count.Save the trained model architecture as a JSON file (model.json) and its trained weights as a HDF5 file (model.h5).The saved model files can be loaded later to make predictions on new hand gesture data without retraining.

E. Real-time Gesture Recognition

Implement hand gesture identification model in real time by means of the trained model and live camera feed:

- Initialize the MediaPipe framework for hand landmark detection.Continuously capture video frames from the camera.
- Detect hand landmarks using MediaPipe and extract features.
- Feed the extracted features into the trained model for prediction.
- Display the recognized hand gestures and associated actions in real-time.

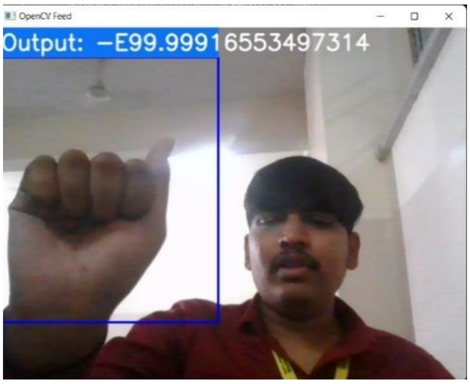

Fig.5 **Fig.6**

F. Termination

Close any open windows and release resources after the gesture recognition process is completed.

3.1 System Architecture

The Hand Gesture Recognition System architecture encompasses several interconnected stages, starting with Input Acquisition. Here, the system focuses on capturing the hand movements of the actor, who could be a user interacting with a device interface or a participant in a gesture-based study. Utilizing a specialized live camera, often equipped with infrared capabilities for depth sensing, ensures precise tracking of rapid and subtle hand movements due to its high frame rate and resolution.

Following Input Acquisition, the Pre-processing stage begins with Hand Detection. By employing skin detection algorithms and motion tracking techniques, the system distinguishes the hand from the background, often leveraging Convolutional Neural Networks (CNNs) for robust detection. Feature Segmentation further refines this process by isolating key hand features like the palm and fingers using contour detection or skeletal model ling techniques.

Subsequently, in the Processing phase, the system enhances the quality of captured images through various Image Processing techniques such as denoising and contrast enhancement. Feature Extraction becomes pivotal here, as critical gesture features like finger length, bending angles, and relative positions are extracted, along with dynamic features like speed and trajectory of movement.

Classification forms the next stage, where extracted features are passed to a classifier trained on a hand gestures dataset. Models in Deep learning are effected for their ability to handle the complexity of gesture data.

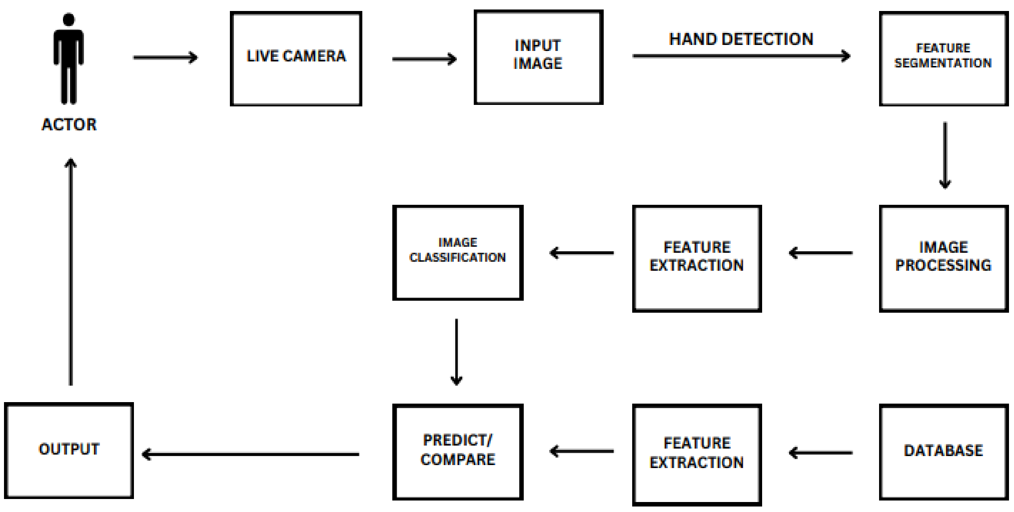

Fig.7. System Architecture of Real Time Hand Gesture Identification System

Further processing involves comparing classified gesture with a database of gesture patterns to determine the best match or predict the gesture's intent. This database contains a comprehensive collection of gesture patterns and their associated meanings or commands.

In the Output stage, the recognized gesture is translated into actionable outputs, which could range from commands for devices to characters in sign language or control signals in user interfaces.

Application Integration follows, where the system interfaces with various applications to translate gestures into specific commands or actions, such as in gaming contexts.

Finally, a Feedback Loop is established to refine the gesture recognition algorithms based on user feedback. This iterative process is crucial for adapting to individual user preferences and improving the overall performance of the system. Overall, this architecture underscores the system's adaptability and user-centric approach, ensuring effective gesture recognition across diverse applications.

3.2 Algorithm

Utilizing deep learning methodologies, the algorithm harnesses recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) units, ideal for handling sequential data by capturing temporal dependencies effectively. Training the model involves employing standard techniques such as the Adam optimizer and categorical cross-entropy loss, which are well-established choices for classification endeavors. Furthermore, the implementation integrates TensorBoard for visualization and monitoring throughout the training process. This tool proves invaluable for tracking model performance and facilitating debugging tasks effectively.

The algorithm steps used for Train model are as follows:

1. Import necessary libraries and functions:

   - all functions from a custom module named function.
   - train_test_split from sklearn.model_selection.
   - to_categorical from keras.utils.
   - Sequential, LSTM, Dense from keras.models.
   - TensorBoard from keras.callbacks.

2. Define a dictionary label_map that maps labels to numerical values using a comprehension, assuming actions is a predefined list.
3. Initialize empty lists sequences and labels.
4. Iterate through each action in actions:
   - For each action, iterate through a range of no_sequences:
   - Create an empty list window.
   - Iterate through a range of sequence_length:
   - Load a NumPy array using np.load from a specific path based on the action, sequence number, and frame number.
   - Append the loaded array to the window.
   - Append the window to sequences.
   - Append the numerical label of the action to labels using label_map.

5. Convert sequences and labels to NumPy arrays X and y, respectively. Convert y to categorical using to_categorical.
6. Split the dataset into two sets (i.e. training and testing) using train_test_split.
7. Set up TensorBoard logging with a specified log directory.
8. Generate a successive model using Keras:

   - Add an LSTM layer with 64 units, returning sequences, ReLU activation, and input shape (30, 63).
   - Incorporate an extra LSTM layer with 128 units, configured to return sequences and activated by ReLU, into the existing architecture.
   - Add a third LSTM layer with 64 units, not returning sequences, and ReLU activation.
   - Include a Dense layer with 64 units and apply ReLU activation.
   - Afterwards, incorporate a final Dense layer with the number of action units and activate it with softmax.

9. Compile the model using categorical accuracy metric, categorical cross-entropy loss and Adam optimizer.
10. Fit the model to the training data for 200 epochs with the TensorBoard callback.
11. Print a brief of model architecture.
12. Convert model to JSON format and save it to a file named "model.json".
13. Save the entire model to an HDF5 file named "model.h5".

Overall, this algorithm is used for training a deep learning model, specifically a sequential model with LSTM layers, for a classification task.

4. Experimental Result

Performance evaluation of the hand gesture identification model is conducted by running input images in static, complex, and cluttered backgrounds. A total of 780 samples, representing 26 hand gestures, are stored in the database for this purpose.

4.1 Static Background

Alphabets	Accuracy in %	Time in sec
A	99	3
B	99.97	3
C	82	2.9
D	99.99	1.3
E	99.99	2.1
F	99.99	2.2
G	99.86	2.1
H	99.95	1.05
I	100	2.4
J	99.95	1.8
K	99.99	2.3
L	100	2.4
M	100	1.9
N	99.99	1.63
O	83.35	3.83
P	85	3.6
Q	99.99	2.4
R	99.42	3
S	100	2
T	100	2
U	95.52	2.7
V	100	2
W	99.99	3.2
X	80.56	2.9
Y	98.32	2.5
Z	100	1
Mean=	**97.03192308**	**2.354230769**

Table 1. Static Background

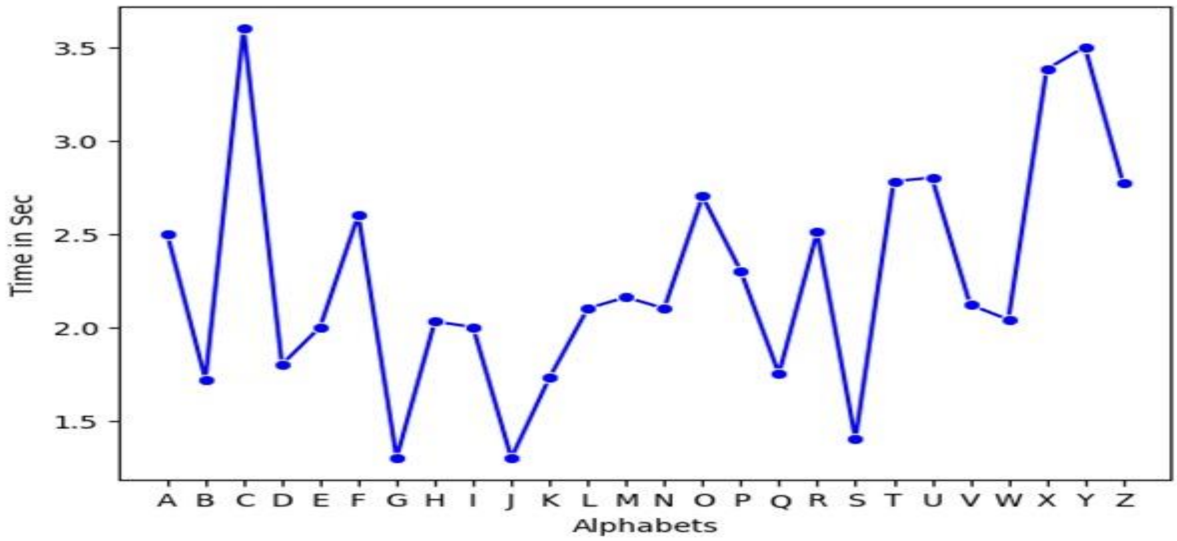

Fig.8. Alphabets v/s Time (Sec)

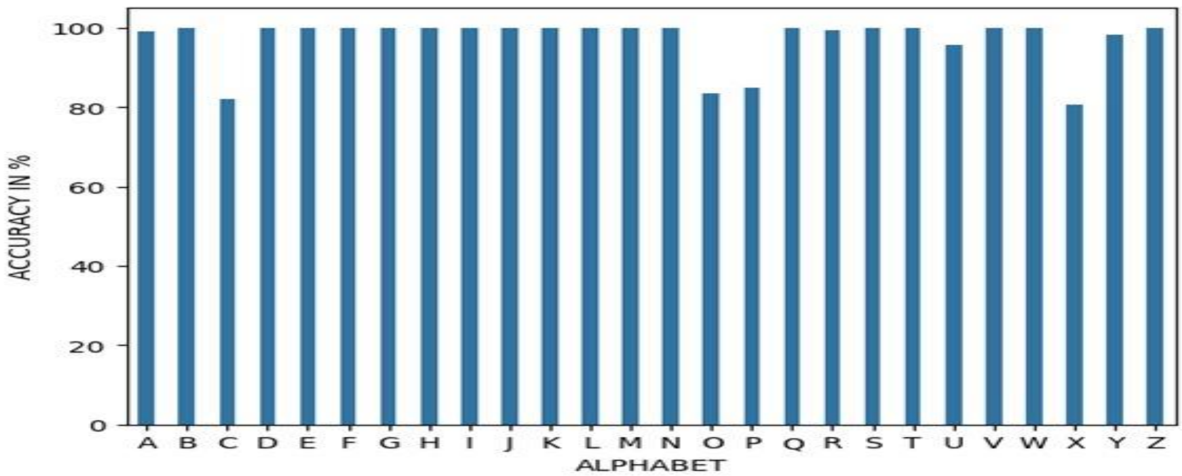

Fig.9. Bar Graph

In Table 1, recognition results based on the proposed algorithm are reported. The overall success rate, observed under natural light with a white background, is 97.03%. The recognition time, using a 720-pixel web camera at a distance of 45cm, averages about 2.35 seconds. Fig. 8 & Fig. 9 depict the graph of recognition results for 26 hand gestures.

4.2 Complex Background

Alphabets	Accuracy in %	Time in sec
A	96.65	2.5
B	99.95	1.72
C	89.65	3.6
D	98.87	1.8
E	99.99	2
F	99.97	2.6
G	98.76	1.3
H	90	2.03

I	99.99	2
J	99.97	1.3
K	99.27	1.73
L	100	2.1
M	98.65	2.16
N	100	2.1
O	94	2.7
P	99.99	2.3
Q	99.99	1.75
R	98.09	2.51
S	99.99	1.4
T	100	2.78
U	96.92	2.8
V	100	2.12
W	99.99	2.04
X	99.91	3.38
Y	99.94	3.5
Z	100	2.77
Mean=	98.48230769	2.268846154

Table 2. Complex Background

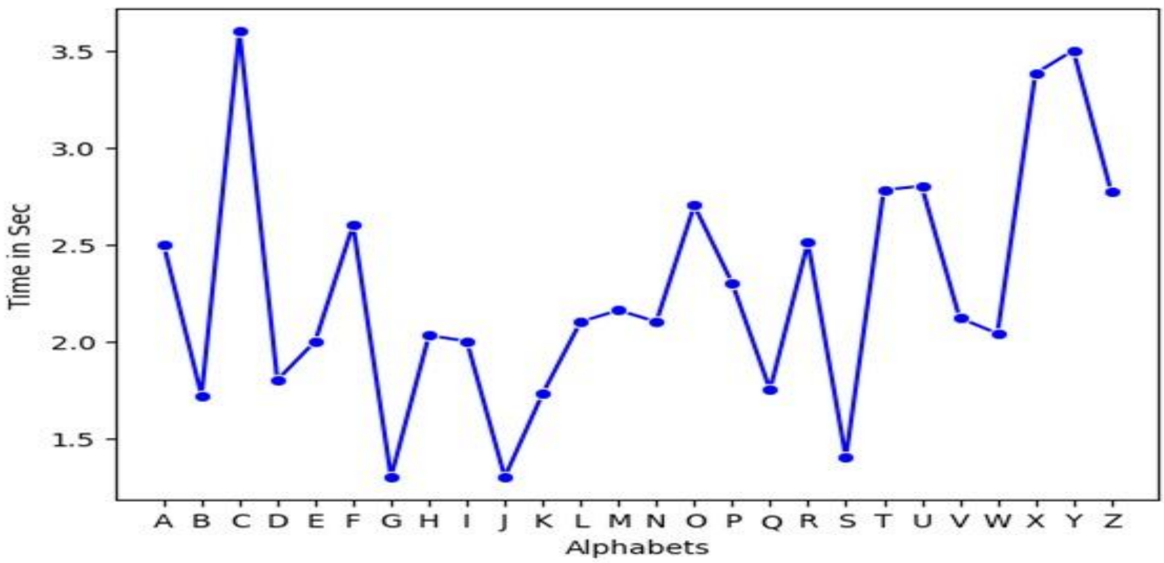

Fig.10. Alphabets v/s Time (Sec)

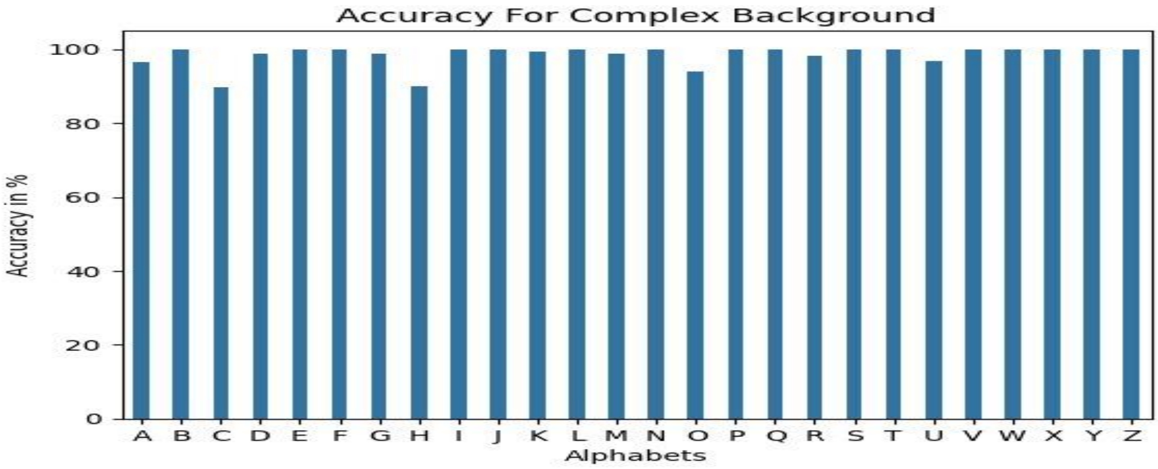

Fig.11. Bar Graph

The recognition results, are reported in Table 2. The success rate of 26 hand gestures in natural light with white tube light in a complex background is 98.48%. The recognition time is approximately 2.26 seconds at a distance of 30cm, using a 720-pixel web camera. Figures 10 and 11 show graphs of the recognition results.

4.3 Cluttered Background

Alphabets	Accuracy in %	Time in sec
A	95.32	3
B	98.96	2.25
C	86.12	4
D	95.34	2.4
E	98.12	3
F	95	3.1
G	97.78	2
H	90	2
I	98.32	1.8
J	99.01	1.9
K	98.97	1.6
L	99.98	1.9
M	97.21	3
N	100	2.5
O	92.23	3.6
P	97.86	3.9
Q	98.5	1.7
R	96.8	3
S	100	01:05

T	99.97	2.9
U	96	3.12
V	98.14	2.84
W	99..01	3.11
X	96.97	3.1
Y	97.5	2.8
Z	100	3.43
Mean=	96.964	2.61519765

Table 3. Cluttered Background

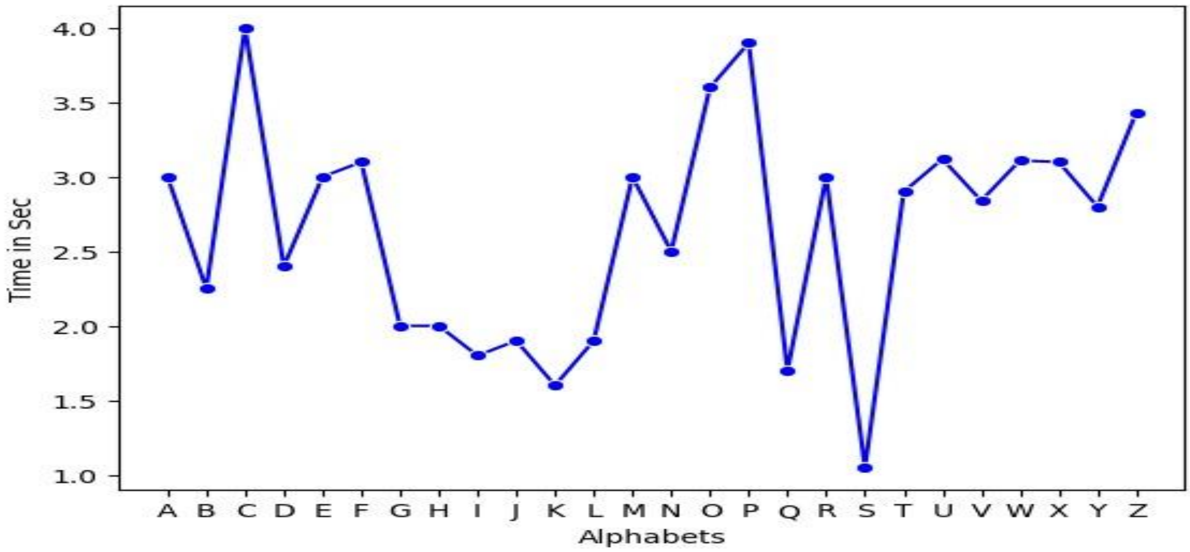

Fig.12. Alphabets v/s Time (Sec)

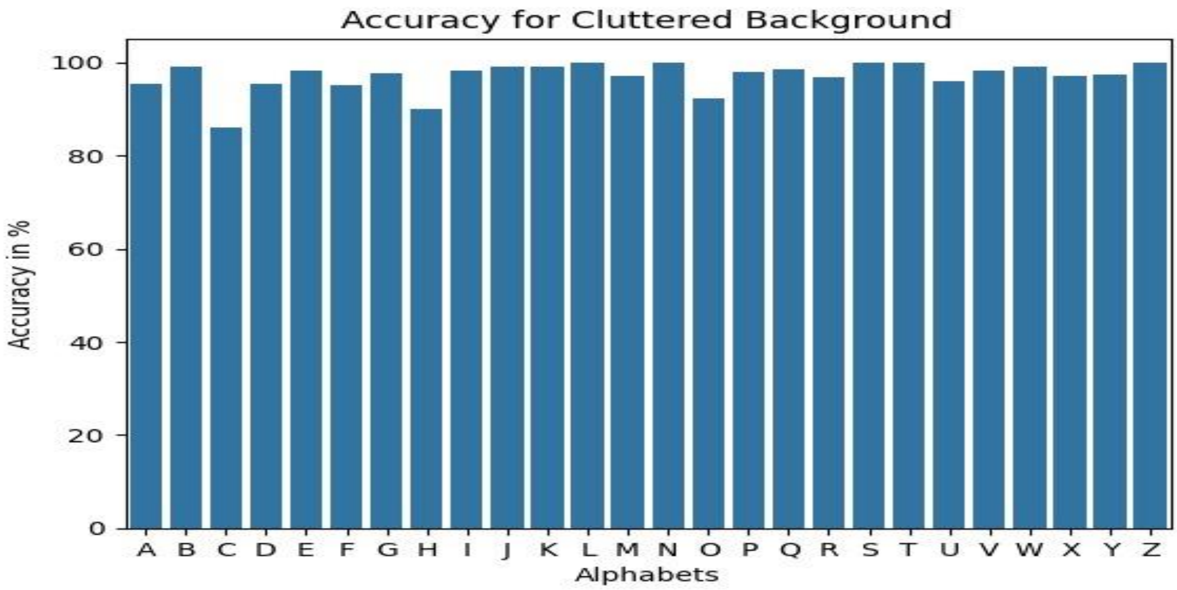

Fig.13. Bar Graph

The recognition results, are reported in Table 3. The success rate for 26 hand gestures in natural light with white tube light in a cluttered background is 96.96%. The recognition time is approximately 2.61 seconds at a distance of 30cm, using a 720-pixel web camera. Figures 12 and 13 show graphs of the recognition results.

5. Conclusion

The design and execution of a hand gesture identification model are described in this paper. The implemented system can achieve a high recognition rate of hand gestures, with accuracy reaching approximately 97%. The system works best from the distance of 30cm in natural light condition. Thus, the hand gesture identification model represents a significant advancement in human computer interaction (HCI) technology, offering promising opportunities for improving accessibility, efficiency, and user experience in numerous domains.

References

[1] K. K. S, A. Asokan, V. Velmurugan and B. M, "Hand Gesture Recognition for Blind Using Machine Learning Algorithms," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 765-768, doi: 10.1109/ICACCS57279.2023.10112770.

[2] S. Srivastava, R. Tiwari, A. Garg, Rishu and R. Tiwari, "Design and Development of Gesture Recognition Based Virtual Mouse System," *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, Ghaziabad, India, 2023, pp. 835-839, doi: 10.1109/CICTN57981.2023.10140390.

[3] P. Netinant, A. Mingkhwan and M. Rakhiran, "Two-Hand Gesture Recognition for User Information Interaction based on Internet of Educational Things," *2023 IEEE 6th Eurasian Conference on Educational Innovation (ECEI)*, Singapore, Singapore, 2023, pp. 321-324, doi: 10.1109/ECEI57668.2023.10105366.

[4] S. Nitnaware and A. Bagde, "Hand Gesture Recognition to Facilitate Tasks for the Disabled," *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, 2022, pp. 949-953, doi: 10.1109/ICAIS53314.2022.9743056.

[5] T. -T. Nguyen *et al.*, "A Continuous Real-time Hand Gesture Recognition Method based on Skeleton," *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, Hanoi, Vietnam, 2022, pp. 273-278, doi: 10.1109/ICCAIS56082.2022.9990122.

[6] A. Das, K. Maitra, S. Roy, B. Ganguly, M. Sengupta and S. Biswas, "Development of a Real Time Vision-Based Hand Gesture Recognition System for Human-Computer Interaction," *2023 IEEE 3rd Applied Signal Processing Conference (ASPCON)*, India, 2023, pp. 294-299, doi: 10.1109/ASPCON59071.2023.10396583.

[7] A. Kumar Sharma, A. S. Kumar, A. Nagpal, R. Singh, D. Reddy and A. Chadha, "Analysis of Machine Learning for Recognizing Hand Gestures," *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2023, pp. 1844-1847, doi: 10.1109/ICACITE57410.2023.10182676.

[8] M. Grami, "A New Algorithm for Hand Gesture Recognition in Color Videos for Operating System Commands," *2022 International Conference on Machine Vision and*

Image Processing (MVIP), Ahvaz, Iran, Islamic Republic of, 2022, pp. 1-6, doi: 10.1109/MVIP53647.2022.9738775.

[9] J. Hossain Gourob, S. Raxit and A. Hasan, "A Robotic Hand: Controlled With Vision Based Hand Gesture Recognition System," *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, Rajshahi, Bangladesh, 2021, pp. 1-4, doi: 10.1109/ACMI53878.2021.9528192.

[10] S. C. B. Jaganathan, K. R, T. P, K. Basak, S. Verma and A. Mital, "Dynamic Hand Gesture Recognition," *2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, Gunupur, Odisha, India, 2022, pp. 1-6, doi: 10.1109/iSSSC56467.2022.10051259.

[11] F. Zhan, "Hand Gesture Recognition with Convolution Neural Networks," *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, Los Angeles, CA, USA, 2019, pp. 295-298, doi: 10.1109/IRI.2019.00054.

[12] T. J. Daim and R. M. A. Lee, "The Effect of Body Position on IR-UWB Radar Sensor-based Hand Gesture Speed Recognition and Classification System," *2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Shah Alam, Malaysia, 2022, pp. 158-162, doi: 10.1109/I2CACIS54679.2022.9815477.

[13] M. Deepika, S. Choudhary, S. Kumar and K. Srinivas, "Machine Learning-Based Approach for Hand Gesture Recognition," *2023 International Conference on Disruptive Technologies (ICDT)*, Greater Noida, India, 2023, pp. 264-268, doi: 10.1109/ICDT57929.2023.10150843.

[14] H. Arwoko, E. M. Yuniarno and M. H. Purnomo, "Hand Gesture Recognition Based on Keypoint Vector," 2022 International Electronics Symposium (IES), Surabaya, Indonesia, 2022, pp. 530-533, doi: 10.1109/IES55876.2022.9888333.

[15] S. Shanmugam, L. S A, P. Dhanasekaran, P. Mahalakshmi and A. Sharmila, "Hand Gesture Recognition using Convolutional Neural Network," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-5, doi: 10.1109/i-PACT52855.2021.9696463.

[16] I. Dhall, S. Vashisth and G. Aggarwal, "Automated Hand Gesture Recognition using a Deep Convolutional Neural Network model," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 811-816, doi: 10.1109/Confluence47617.2020.9057853.

[17] A. Mohanarathinam, K. G. Dharani, R. Sangeetha, G. Aravindh and P. Sasikala, "Study on Hand Gesture Recoginition by using Machine Learning," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1498-1501, doi: 10.1109/ICECA49313.2020.9297513.

[18] W. B. Dou, W. H. Chin and N. Kubota, "Hand Gesture Communication using Deep Learning based on Relevance Theory," 2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS), Hachijo Island, Japan, 2020, pp. 1-5, doi: 10.1109/SCISISIS50064.2020.9322784.

[19] W. Ma, Q. Zeng, G. Song and M. Zou, "Design and implementation of traffic police hand gesture recognition system based on surface electromyographic signals," 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and

Automation Control Conference (IMCEC), Chongqing, China, 2022, pp. 1888-1894, doi: 10.1109/IMCEC55388.2022.10020092.

[20] Z. Wang, Z. Liu, Z. Li, Y. Jiang and F. Li, "American Sign Language Alphabet Recognition with YOLOv5 Enhanced by MediaPipe Hands," 2023 8th International Conference on Instrumentation, Control, and Automation (ICA), Jakarta, Indonesia, 2023, pp. 103-108, doi: 10.1109/ICA58538.2023.10273099.

[21] W. Wu, M. Shi, T. Wu, D. Zhao, S. Zhang and J. Li, "Real-time Hand Gesture Recognition Based on Deep Learning in Complex Environments," 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 5950-5955, doi: 10.1109/CCDC.2019.8833328.

[22] S. Hussain, R. Saxena, X. Han, J. A. Khan and H. Shin, "Hand gesture recognition using deep learning," 2017 International SoC Design Conference (ISOCC), Seoul, Korea (South), 2017, pp. 48-49, doi: 10.1109/ISOCC.2017.8368821.

[23] B. Nandwana, S. Tazi, S. Trivedi, D. Kumar and S. K. Vipparthi, "A survey paper on hand gesture recognition," 2020 7th International Conference on Communication Systems and Network Technologies (CSNT), Nagpur, India, 2017, pp. 147-152, doi: 10.1109/CSNT.2017.8418527.

[24] J. -H. Sun, T. -T. Ji, S. -B. Zhang, J. -K. Yang and G. -R. Ji, "Research on the Hand Gesture Recognition Based on Deep Learning," 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 2018, pp. 1-4, doi: 10.1109/ISAPE.2018.8634348.