iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VII **Month of publication:** July 2024

DOI: https://doi.org/10.22214/ijraset.2024.62035

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

Bridging the Gap: Deep Learning Techniques for American Sign Language Recognition

Vanshika[1], Ankur Jain[2]
[1]*Student,* [2]*Assistant Professor, Dept. of Computer Science and Engineering, Bhagwan Mahaveer College of Engineering & Management, Sonepat, India*

Abstract: ***Communication stands upon the pillars of verbal and non-verbal conversations, and hence it holds the basis of human social relationships. Along with words, gestures are another component of nonverbal communication that achieves the purpose of conveying the intended sense and bridging the gab of the languages and cultures. People with speech or hearing problems usually read manual or verbal signs that often don't make sense to a hearing-challenged person. Gestures are the first sign of instruction that overcomes the speech gap. A kaleidoscopic patchwork of facial expressions comprising of facial movements and body language! Such variations occurring in linguistic areas overall are not surprising, as community cultures and tongues around the planet typically shape their language. In the United States and Canada, American Sign Language (ASL) is common and is an independent language from what is heard in the surrounding community but is also used as a way of communication between individuals and in groups that are deaf and hearing alike. There are some restrictions. Common language review and adequate practice are of crucial importance here, that is why it is so hard for deaf people to work outside. The accessibility of the translation tools decreases radically, which means it will be difficult to communicate, and it will further be hard to understand and to be understood. By integrating the AI technologies as neural networks and deep learning into the goals, the system will proceed to bridging various communication channels from manual writing to voice operation. The task goes with webcam installation together with gesture capture and then as an input it goes to the system. The proposed model will be divided into several stages namely, data acquisition, pre-training to the neural network, testing and the post-testing phases. This research project will do that through developing digital technology which in turn will enhance accessibility, encourage integration and let people who are film-blind or deaf to associate with the environment.***
Keywords: ASL, TensorFlow, Neural Networks, LSTM, MediaPipe

I. INTRODUCTION

The DGS (Deaf Gestural System) is the central mode of communication for deaf and mute individuals, making it a complicated aquain tance for those who are not versed in its unusual signs. The mental ineptness of the general public in understanding sign language serves as an obstacle that makes it difficult to receive the messages from the deaf or dumb people. Gesture recognition becomes an innovative tool that employs sensors to understand hand signs and turn them into text, therefore using signs to a much greater extent. As for gestures, there are various movements outlined which are necessary for the communication process and may include different and body gestures. Symbol language transmits the pictorial models by gestures, where each sign takes its place as specific communication mean. with this all becomes clear and understandable. On the parallel front, the Human-computer interaction (HCI) discipline has seen many new advances, but machine interactivity still mostly places reliance on fingers and eyes, thus notionally disregarding other human organs and tissues. The American Sign Language (ASL) system can be considered the pioneer of user-computer interaction since it avails users with options to interact with machines by making gestures that are precise as well as self-explanatory, thus leading to effective communication. ASL Recognition system is, in general, a tool that strives to respond as fast as possible to every possible hand movements pattern by applying algorithms and neural networks in order to get to know these patterns and how to this end the system can be used in many different industrial applications.

The designed ASL Recognition system rather relies on hardware equipment internally but intends for a straight forward interaction process where a working camera will be the only remote that is needed to determine ASLs accurately.

A. Problem Statement

In the present-day community effective communication is a necessity and it influences every sphere of life. However, people who have difficulties in speaking encounter problems in expressing themselves verbally making them to use other alternatives like sign language which in turn requires a lot of time and effort to be proficient due to lack of accessible learning aids.

Also, those who are good at it need interpreters for accuracy when relaying information through this means. To deal with these challenges, scientists intend coming up with new methods that employ technology so that seamless communication can be achieved among those with hearing as well as speaking impairments thus enabling them express their thoughts more effectively and take part in various activities within the society.

B. Objectives

People are given a speech handicap on the other hand, the process of communicating properly can lead to considerable difficulties, requiring the use of alternative means of expression like sign language. Despite its significance, well mastery and perception of sign language can be extremely time-consuming and complicated, thus affecting unobstructed communication between hearing people and those that are speech-impaired.

In this paper, the main objective is to consider the communication issues of the hearing disabled using the state of the art deep learning methods to develop a dependable American Sign Language (ASL) detection system. The use of advanced algorithms and technology is the main approach of the research, as it intends to develop a tool that is consistent and handheld, thus facilitating the conversion and transition of sign language into text and speech.

1) Bridging the Communication Gap: The study intends to fill the divorce between a deaf person who knows sign language and someone who uses spoken language as means of communication. Through signs language, the interpreter creates a text or speech, which works to facilitate communication between individuals with different communication abilities.

2) Enhancing Accessibility: The system, which has been developed, tries to facilitate access to people with hearing impairment by suggesting a way to communicate with others using gestures that can be innacurately interpreted and concluded by other members of the society. Furthermore, people with bleak eyesight who encounter the text that is translated into speech can also exploit machine learning in this case.

3) Training and Deployment of Deep Learning Models: The Study comprises training deep learning models using data gathered from ASL symbols captured by the camera. The models will be subjected to severe learning and testing regimes to make sure that the recognition of gestures is at the highest level of precision. In the future real-time gesture prediction will be enabled, so people will have no problems in communicating in a wide variety of situations.

4) Overcoming Previous Challenges: Through the novel ASL recognition system's development, the research trial is set to surmount the difficulties of learning and interpreting sign language. The system goes beyond basic interpretation in terms of real-time communication and translation, that help to overcome the barriers of accessibility and cultural differences.

II. METHODOLOGY

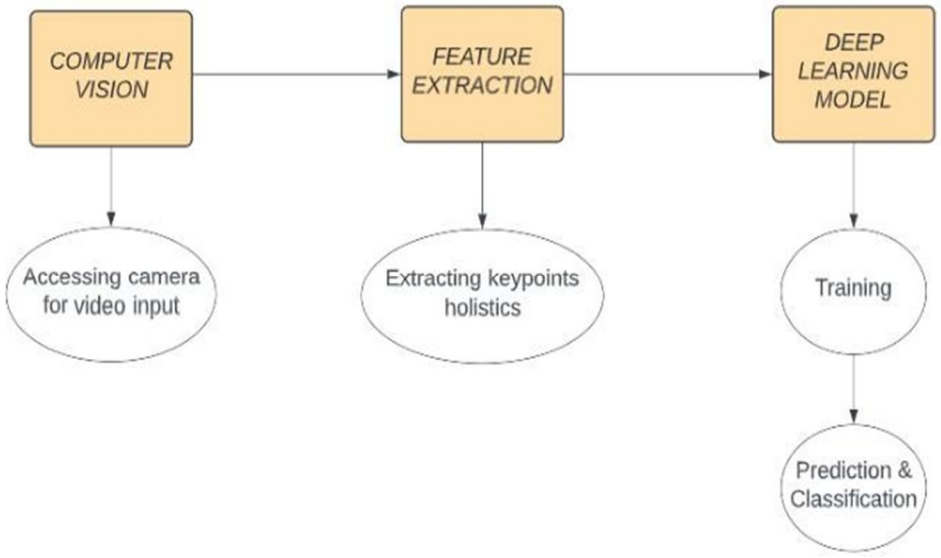


Fig 1: Methodology

This research project is applying a sequence of stages, each of which having its own objective and gradually leading to the development of our sign language recognition and translation system.

- *Stage 1: Data preprocessing and feature extraction*

The basic step involves data preprocessing and feature extraction with the help of MediaPipe tool. The mechanism of the pipeline applies data augmentation techniques internally to get facial, hand, and body feature marks and key points as the landmarks. Hence, MediaPipe Holistic works in the model-by-model fashion, as the specific models for hands, face, and pose are being processed and kept in the correct proportion for the image.

The workflow for this stage involves:

In order to apply this phase BlazePose's pose detector will be utilized to detect the position of the human and precise markings detection. For this experiment, we took 3 Region of Interest (ROI) crops from the face and hands and proceeded a customized cropping to augment ROI. Bringing out landmarks that are homologous and also match with the ROIs, thus making the tasks like hand and face modeling easier. Through an integration of all markers, one by one, to end up having all the landmarks, which collectively sum up to 540+ landmarks.

- *Stage 2: Data Cleaning and Tagging*

After the stages of data preprocessing and feature extraction, data cleaning and labeling are in order to preserve data integrity and ensure that it is fit for model training and testing.

The process includes:

Radius buffering and clustering (using a 500 m buffer radius) points, ending up with a dataset of 1662 points.

Finally, the data set will be saved to a file and any null entries will be removed to reduce the potential disruptions arising from failed feature detection in cases of fuzzy images.

Creation of labels for each class and saving relevant frame sequences with the purpose of preliminary training, testing, and validation.

- *Stage 3: Gesture Recognition and Speech Translation*

The later steps involve gesture recognition and speech translation which involves the use of a cleaned and labeled dataset used to train an LSTM model. Model is trained on recognizing sign language gestures in real time and deploying OpenCV module for implementing it. Besides, the identified text form sign language is converted into speech by using a Python text-to-speech module, the audio is played via the system's default audio device.

Indeed it is through these rigorous stages of this methodology that the correct recognition and translation of sign language gestures is ensured hence improving accessibility in communication for people with speech or hearing impairments.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 30, 64)            442112

 lstm_1 (LSTM)               (None, 30, 128)           98816

 lstm_2 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 3)                 99

=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
```

Fig 2: Model Summary

III. LITERATURE REVIEW

Sign language recognition has emerged as an active domain for researchers in sensor and vision-based deep learning. Despite the remarkable advancement, several open issues persist and more challenges to face before making SLR systems more effective.

The exploration into gesture recognition for video streams has seen diverse methodologies. For instance, the study used Hidden Markov Models (HMM) and Gaussian Tree Augmented Naive Bayes Classifier (GNB) and Bayes Classifier (BC) to detect emotional expressions. Nevertheless, this algorithm has limitations on its use which are primarily associated with its inefficiency during the detection of moving gestures, as such, the performance of this method will be compromised. The corresponding works introduced by the authors – François et al. focused on Human Posture Recognition utilizing 2D and 3D appearance-based methods. PCA helped to obtain silhouettes and body posture per video prived a one 3D models through. However, this approach encounters issue of mediating gesture that can produce ambiguity for training decisions and then deteriorate the precision of decision.

In Human Posture Recognition, Francois et al. used both 2D and 3D appearance-based methods. They used a PCA to find silhouettes as well as 3D modeling techniques for posture recognition in static video scenes. Nevertheless, this tactic has a limitation on intermediary gestures which may lead to the creation of training errors, affecting the prediction accuracy.

Neural networks also have been utilized to process video clips by extracting visual information represented with feature vectors. Issues like hand tracking, segmentation, lighting variation, occlusion, and the human subject's body movement and position create these major barriers. Nandy and his team solved some of these problems by segmenting the datasets, extracting features, and classification based on Euclidean Distance and K-Nearest Neighbors, respectively. Precisely, the development of more advanced tracking systems is required.

The work of Kumud et al. has followed a step by step approach featured frame extraction, preprocessing, key frame extraction, feature extraction, recognition, and optimization. The preprocessing was done by replacing videos with RGB frames, segmentation of skin by using HSV model, binarisation, and gradient calculations for key frame extraction. Orientation histograms was used to quantify the features and different distance metrics were employed to classify the object.

The survey of the literature leads to the conclusion of Kshitij Bantupalli and Ying Xie that they should extract temporal and spatial information from video sequences. The model needed to face challenges when it was applied to the different skin tones and complex faces, and failed to be precise while working for a long time. Aju Dennisan proposed a design to detect ASL alphabets from RGB images, reaching a rate of 83 per cent. But there was a problem that those symbols were used only for writing alphabets.

K Amrutha and P Prabu developed a system for sign interpretation using a dataset and algorithm selection, but the model's accuracy was limited to 65% due to the small dataset size. A. Mittal et al. proposed a Modified LSTM Model for Continuous SLR using Leap Motion, which showed accuracies of 72.3% and 89.5%, constrained by the training dataset. M. Wurangian's work was confined to the recognition of the American sign language alphabet, while P. Likhar et al. developed a model specific to Indian Sign Language, not applicable to American Sign Language. Lastly, Shivashankara, Ss, and S. Srinath's model was restricted to alphabet recognition only.

The aggregated intelligence from these studies is a source that indicates the imperative value of AI in deepening the impacts of SLR. Consequently, they also emphasize the need for continuous research to overcome these limitations and to develop the cutting- edge technologies, culturally attuned and universally accessible them. The future of SLR proves to be a promising avenue for multi-disciplinary collaboration, user-oriented design and ethical AI approaches which could result in a convenient tool to be used for communication beyond barriers.

IV. PROPOSED SYSTEM

Our proposed method meets the gaps in the previous models and thus we ensure maximum precision and accuracy. Researchers constantly struggled with the error-prone background and the imperfect separation of finger movements from that noise. In order to overcome this obstacle, we constructed a novel dataset utilizing the Mediapipe technology from Google and OpenCV library to accurately detect hand landmarks. By using this method, it is possible to obtain sturdy markers in every type of environment; be it indoor, in a vehicle or outdoors.

Our proposed system comprises three interconnected modules: Dataset, Preprocessing and LSTM. The Dataset module allows for the extraction of landmarks and the making of a full dataset. Following on, Preprocessing module cleans up the data for LSTM module where the training for gesture detection takes place. When hyperparameter optimization methods are proven to be effective, the model will be ready for real-time deployment.

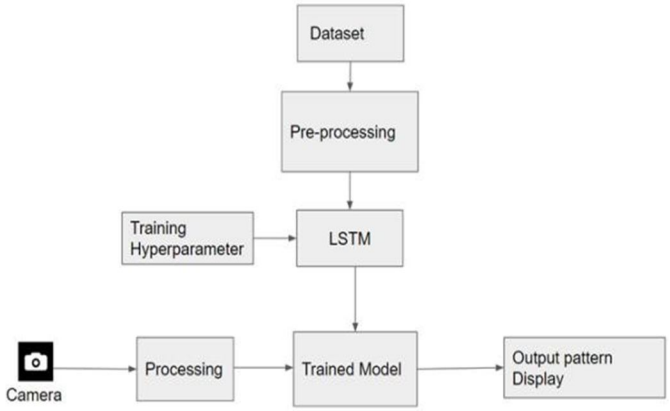


Fig 3: System Architecture

V. EXPERIMENTAL SETUP

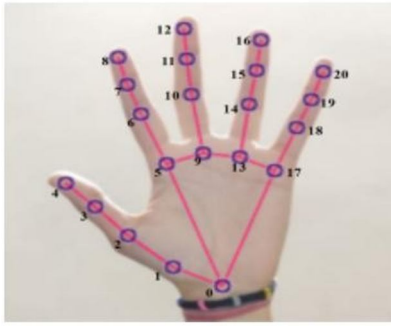
An experimental approach to this project will consist of the use of a computer system specified as Intel Core processor with 8 GB RAM and running the Windows 10 operating system. The development environment, initialization on Anaconda Navigator and would allow the setup of the appropriate environment for Jupyter Notebook, is created. The model is implemented using the Python programming language and libraries like OpenCV and MediaPipe with a deep learning model that is named LSTM(Long Short-Term Memory)are utilized. These types of software and hardware are selected with consideration of compatibility, efficiency, and fitness of the task at hand and, thus, which will enable a powerful model in recognizing the American Sign Language gestures. By being involved in the use of Anaconda Navigator, the control of the interdependence and environments is made easier thus reducing the chaos of development. Besides, Python programming language is a tool which gives you both freedom and implements it easily which is why it is a favorite one to create machine learning and deep learning models. The libraries like OpenCV and MediaPipe contribute to the integration process by providing indispensable functionalities that facilitate the processing of images and videos, while the LSTM network architecture helps the model to capture temporal dependencies of sequential data more efficiently due to the nature of its neural networks. In short, the experimental setup is constructed to allow for the implementation and evaluation of the ASL recognition algorithm to be carried out accurately and effectively with very little noise or errors occurring.

VI. MODULES AND IMPLEMENTATION

The system chooses the vision-based method recognize sign gestures taken from video frames. This sign recognition process encompasses three main phases: data acquisition, data preprocessing and feature extraction, as well as gesture recognition. Firstly, the accumulated data will be subjected to preprocessing and augmentation that will bring uniformity. Eventually, facial landmarks, hand landmarks, and body postures are extracted as key points that are generated from a sequence of input frames that are captured through a web camera. These features are further processed by the classifier that is trained to determine user gestures. Gesture recognition is being processed that is translated into words and displayed in the on-screen text.

A. MediaPipe Holistic Model

MediaPipe Holistic integrates three essential components for landmark extraction: posture, hand, and face. The 33 3D landmarks pose model carefully determines the true body pose with extreme accuracy, which is the core point of gesture recognition and motion analysis. Composed of 21 3D points, and it is a combination of both palm detection and keypoint localization which are used for real-time tracking of sign movements and gestures which are important for recognizing sign language and interactive interfaces.
Furthermore, the facial mesh model brings forward 468 3D points, through which we can track facial points and predict geometry, which is essential to tasks such as facial recognition, expression analysis as well as human-computer interaction experience design, making them more diverse and intuitive. The holistic landmark models that have been integrated by MediaPipe help the developers to be more potent. Researchers will further develop high-fidelity and multi-purpose perceptual systems to bring about progress in computer vision, and opening up new solutions for several domains, including assistive technologies and entertainment.
While upon holistic model utilized to gather major points including hand, face, and pose, we picked only hand landmarks that would be conveyed through the interface (see Figure 3) to ease the confusion of the overcrowded landmarks.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig 3: The hand landmarks model

B. Data Collection

The first step of data collection for the recognition framework is important as it is the initiation stage. The first step to achieving the vocabulary division of collected data is the application of refined data organization and data cleansing protocols which are tailored to guarantee data integrity and relevance. We created a set of 10 gestures that will be used to train our model, which will be used to recognize these gestures. As a result to the short size and irrelevant nature of the current datasets, which warrants a proactive approach, new data instances (in the form of video sequences) had to be created. This was the part where I picked my set of gestures associated with deaf community jargon and culture, as the "building blocks" of the following methodology.

The succeeding action is to design data portfolio frameworks that will help to collect gesture-specific data on an ongoing basis. Through the OpenCV library, video data are captured from the computer‘s built-in camera, by taking 30 video sequences for each gesture. Every one of the video sequences is disintegrated into 30 frames that allow the formation of an exquisite dataset subjected to further processing and analysis. Furthermore, the whole process of recognizing alphabets is given a framework which is accomplished by incorporating important libraries like TensorFlow, OpenCV, and Mediapipe. Such libraries give the user the ability to perform data manipulations, feature extraction and model training thus setting the pace for the rest of the recognition procedure.

Fig 4: Data Collection in folders

C. Gesture Recognition

It is important to note that a selective approach to the model training is required to replenish real-time gesture recognition. Feature extraction starts with the extraction of 543 properties per input word, among them 33 pose, 468 face, and 21 hand landmarks. Thus, we end up with the dataset of 9000 instances, including exactly 900 examples for each word in the input vocabulary. Each sample includes 30 videos, each shot consisting 30 frames. They form together into an npy file with class labels completed the package of a sample.

The LSTM network with a deep learning-based foundation is used as the model of choice due to its capability to effectively manage data time sequences. Crucial parameters such as batch size, number of epochs, and number of LSTM units are precisely adjusted to get the desired performance improvement. The batch size defines the amount of data processed per training step, where the experimenter should find a balance point between the computational cost and the model accuracy. With each epoch representing the full dataset passes and after weight modifications at the end each epoch to improve model accuracy while avoiding overfitting. The output space dimensionality and layer neuron count are determined by the LSTM units.

Upon parameter selection, we chose the categorical cross-entropy loss function, as it fully fits the model architecture. The Adam optimizer is chosen for its gradient-desendent optimization, which allows a steady learning rate, and also changes moment estimation according to the parameters utilized. This optimizer is highly beneficial thanks to its high performance, low memory utilization as well as its ability to accommodate large data set and dynamically moving targets.

The LSTM network is set out sequentially, and the first five layers involve the use of the sigmoid activation function. The network sums up by bringing the judgments of various groups to a single dense layer, and the softmax output function is responsible for the translation of these class probabilities. This setup allows the derivation of log odd ratios which are a way of quantifying the likeliness for each class. Model efficiency is frequently measured by the accuracy and loss metric recorded for both the training and validation sets following the end of each epoch.

D. LSTM Implementation

LSTM stands for Long Short-Term Memory which is a variant of RNNs (recurrent neural networks) that makes it possible for the machine to deal with sequential datasets or time-series. It contains three gates and a cell memorizer that is quite central for the handling of short-term as well as long-term dependencies in information. The construction of the LSTM single unit that is shown in Figure 3 includes the contain the forget gate, input gate, output gate, and memory cell.

LSTM will be applied after given data is arranged in the frames sequence and which frames contain the mark of any particular landmark. The LSTM movement from one time step to the other enables such sequences to be processed because they have their order. The LSTM network architecture comprises several layers with the first two layers activated using the "Sigmoid" activation function and the final layer for multiclass classification leveraging the "Softmax" approach.

While learning, the models were assessed by means of the accuracy and loss metrics. The metrics were recorded for both training and validation datasets using epoch iteration. Adam is among the adorable vectors which in fact are aimed at lifting the confidence of the doctor that getting drugs is possible. More importantly, batch size, number of epochs, and LSTM units that help to improve model are determined smartly.

The adoption of LSTM is indeed a very delicate process that takes into account multiple parameters choice and architectural features to guarantee performance and efficiency. The steps of model training need to be repeated until satisfying effects are obtained with the help of iterative adjustment of parameters and performance metric assessment.

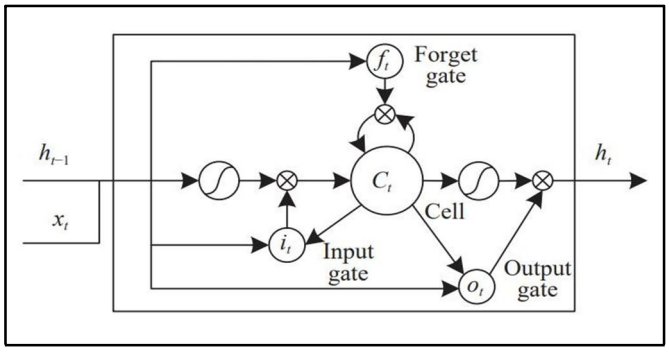


Fig 5: LSTM Architecture

VII. WORKFLOW

1) Step1: The first thing to be done is install those necessary packages for the project to set up on. Specifically, six key dependencies are installed including TensorFlow, a machine learning library developed by Google, the driving force for deep learning and image classification computer vision tasks TensorFlow GPU, OpenCV-Python for webcam access, Mediapipe for landmark extraction, scikit-learn for evaluation metrics, and Matplotlib for image visualization. Installation is followed by library importation which include cv2, NumPy, os, pyplot, time and Mediapipe among others for more effectiveness.

2) Step2: Through the use of OpenCV, we made sure that for webcam access both video capture and frame from each frame are iterated every time. Then, the module Mediapipe Holistic comprising landmark detection is set up, where the webcam feed is converted from BGR to RGB format for understanding. These points of interest displayed on the image frame are further represented using different methods.
3) Step3: To extract the keypoint values for left hand, right hand, face and pose, NumPy arrays are concatenated. This step is the one which makes the algorithm capable of handling missing values and places the data in an organized way that is suitable for further processing. Each frame's landmark values shall be saved in arrays rendering further action detection far simpler.
4) Step4: Folders are set up that are used to put data into them which consists of 30 videos per action and each contains 30 frames of landmark values. Data flow diagram stands out, and an iteration is used to create folders by action and video sequences. The acquired frames are stored as NumPy arrays for later applications.
5) Step5: Every video sequence collection is separated by a pause during it to reposition themself as well as reset. The specified data path location is where frames get saved as NumPy arrays and are later used for training the model.
6) Step6: Lastly, preprocessing comprises of dividing the data into training and testing set via the noise function train_test_split. Labels are encoded using the categorical feature approach and the resultant dataset is arranged into two forms of data, i. e. features arrays (X) and label arrays (Y), for model training.
7) Step7: Through the sequential layers consisting of LSTM and dense nodes, this LSTM-based neural network is designed in Keras API of TensorFlow platform. The model consists of the optimizer settings and loss function parameters alongside previously prepared dataset as training data.
8) Step8: The trained model will be used to make predictions for both training and testing datasets. Evaluation metrics like accuracy are obtained using the scikit-learn function accuracy_score. This shows the model's performance.
9) Step9: The trained model weights are saved for future use, so that they can be used again and the results can be reproduced.
10) Step10: Loaded saved model is executed for real-time testing in which input video frames are processed to elicit sign language prediction. The conversion from text-to-speech is done with the pyttsx3 module to provide an audio output of the recognized sign language.

VIII. RESULTS

To test the effectiveness of our American Sign Language Recognition System, we carried out a thorough testing on a validation dataset which was obtained from the original dataset, this dataset was the one that was previously divided between training and testing data. The main evaluation criteria here is accuracy as we use this. The LSTM (Long Short-Term Memory) model, a variant of the recurrent neural network (RNN), turned out to be very effective, as the model demonstrated an astonishing accuracy to be around 85% on the test dataset.

Apart from offline evaluation, our system was also tested in real-time to assess its practical effectiveness in dynamic conditions. Surpassing the memorized stored gestures and taking advantage of the live video feed inputs, the recognition system displayed its real-time gesture recognition capability by generating instant text feedback conforming to the performed gesture movements. Real time testing involved processing real-time streamed data with subsequent recognition showing that the approach patterns in check have been detected. We have developed a system that is able to make predictions based on the 30 frames' worth of key points and then seamlessly integrate that with the real-time video feed processing pipeline.

The following paragraph outline the methodology employed for real-time gesture recognition:

A vacant array, sequence = [ ], is created to store the sequence of key points. Using OpenCV, the real-time video is captured, and the key points are extracted by the methods which were addressed previously in this research, i.e, mediapipe_detection and extract_keypoints. These points are added into the empty array to the array, the array is truncated to retain only the last 30 frames key points, to keep the continuity and relevance. After the sequence array has a length of 30 frames, the model is used to predict the gesture output based on the accumulated key point sequence.

The figure 6 indicates the results screens after an actual test, which shows that the system is able to precisely understand gestures under diverse, real-time scenarios.

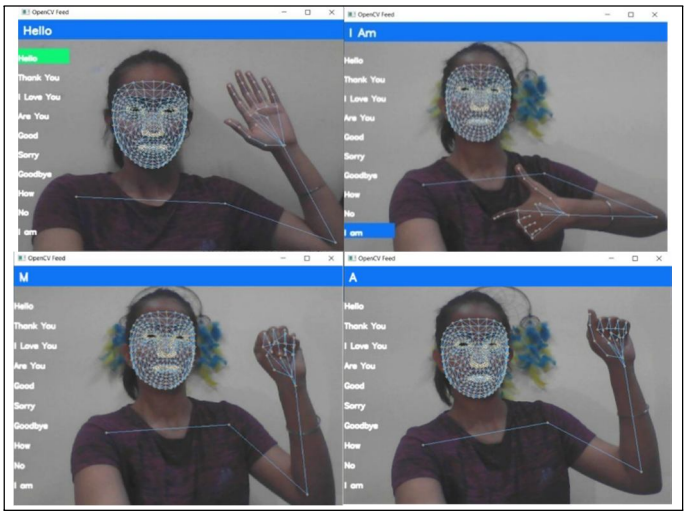


Fig 4: Real-Time Gesture Recognition

IX. CONCLUSION AND FUTURE SCOPE

To sum up, our study shows that a American Sign Language Recognition System based on deep learning techniques can be developed and validated effectively. Through painstaking testing on the validation data and under real-world condition, we have demonstrated the efficacy and practicality of our system in the changing situation.

The model's accuracy rate of above 85% on the testing dataset substantiates its efficiency by way of the correct classification of hand gestures. The training dynamics are valuable tools that provide us with the learning behavior of our model, thus showing how it can quickly converge during the training.

Real-time testing in addition to that proves the applicability of our system in real-world environments where it operates as a regular component of video feed inputs providing the user with an instant feedback at the crucial moments.

Our system can be designed in different methods and has a wide range of applications such as human-computer interaction, augmented reality, and assistive technology. Through the usage of the computer vision and deep learning, we have designed a multifunctional platform that can improve user experience and accessibility in different fields.

Moving forward, the research field would include a number of optimizations and explorations to be done. Alongside these modifications is the fine-tuning of the model architecture and training setup to attain accuracy at the highest efficiency level. Furthermore, the innovations in attention mechanisms and multi-modal fusion could make our system more advanced by allowing it to process more complex gestures and environments.

Not only this, the technology that could be distributed to a broad variety of areas such as gesture-based control systems, interactive interfaces and immersive experiences could contribute to the innovation and practical application. Partnerships with experts in areas such as robotics, healthcare, and education can be useful in making unique gesture recognition solutions for domain specific tasks.

In a nutshell, our research opens the way for further progress in gesture recognition technology, which has the ability to transform human-computer interaction and to shape the future of interactive systems. Our efforts will be anchored on continuous research and collaboration with the purpose of improving the gesture systems and making them a seamless part of our daily lives.

X. APPLICATIONS CONTRIBUTIONS

Unique challenges and opportunities are associated with the use of American Sign Language (ASL), especially for people who are deaf and use it to communicate. ASL is the main way that members of the deaf community talk to each other, using movements of the hands and face for expression. Therefore, there is a need for new ideas to help those who know ASL understand spoken language and vice versa.

B. Li et al. introduced a research paper titled “ASL Recognition Utilizing Deep Learning” , that suggests an initial stepping stone using advanced learning techniques to solve the challenge of recognizing and analyzing ASL. The content explains about the system which was developed that can recognise the ASL gestures from video records taken by Google Glass, a wearable device for augmented reality. An example of such is how the device uses deep learning algorithms to do a real-time ASL data analysis which offers a solution for transforming communication between people who know sign language and those who are strangers to this language.

Employment of such techniques in a real-world situation is seen as major advancement in the field of assistive technology and accessibility. A system that employs deep learning can automatically recognize ASL gestures and translate them into English text improving accessibility and communication among people who are deaf. Furthermore, the inclusion of Google Glass as a wearable device increases flexibility and portability allowing users to receive instant translations anywhere anytime across different environments thus enhancing communication.

Its key function is to immediately interpret ASL signs into English text and relay it through sound signals. This allows people with hearing disability to communicate effectively with those who do not understand sign language thus fostering unity among different language speakers as well. Moreover, the fact that these translations are done in real time also makes them useful while operating in changing environments because users can communicate freely without any hitches or delays caused by translation processes.

XI. APPENDICES

A. OpenCV

OpenCV, or Open Source Computer Vision Library, is a multipurpose tool which is licensed under a BSD rules, thus making it free for academic as well as commercial uses. OpenCV has interfaces in C++, Python, and Java and supports a lot of operating systems such as Windows, Linux, Mac OS, iOS, and Android. Built with computational efficiency and real-time applications in mind, OpenCV is developed in optimized C/C++, thereby relying on multi-core processing. Moreover, OpenCV has built-in OpenCL, making use of hardware acceleration across a variety of heterogeneous compute platforms. OpenCV has been widely adopted all over the world and it has a user community of over 47 thousand individuals and has been downloaded more than 14 million times. It is used in various domains, for example interactive art, inspection in mines, web-based map stitching, and advanced robotics.

B. TensorFlow

TensorFlow can be referred to as the open-source software library famous for data flow programming in any given process but specifically high-level neural networks like machine learning. The Google Brain team introduced TensorFlow to address the research and the production needs in Google, and it's a symbolic math library for efficient computation. Initially created for internal Google use, TensorFlow was released under the Apache 2.0 open-source license on November 9, 2015. Representing Google Brain's second-generation system, TensorFlow's version 1.0.0 debuted on February 11, 2017. While the reference implementation supports single devices, TensorFlow extends its capabilities to run across multiple CPUs and GPUs, with optional support for CUDA and SYCL extensions for general-purpose computing on graphics processing units. Available on various platforms including 64-bit Linux, macOS, Windows, Android, and iOS, TensorFlow's adaptable architecture facilitates computation deployment across a spectrum of devices, from desktops to server clusters to mobile and edge devices.

REFERENCES

[1] K. Cheng, “Top 10 & 25 American sign language signs for beginners – the most know top 10 & 25 ASL signs to learn first: Start ASL,” Start ASL Learn American Sign Language with our Complete 3-Level Course!, 29-Sep-2021. [Online]. Available: Top 10 & 25 American Sign Language Signs for Beginners – The Most Know Top 10 & 25 ASL Signs to Learn First.

[2] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B.B.Chaudhuri-https://ui.adsabs.harvard.edu/abs/2019ISenJ..19.7056M/abstract in IEEE Sensors Journal, vol. 19, no. 16, pp. 7056-7063, 15 Aug.15, 2019.

[3] “Real time sign language detection with tensorflow object detection and Python | Deep Learning SSD,” YouTube, 05-Nov-2020. [Online]. Available: https://www.youtube.com/watch?v=pDXdlXlaCco&t=1035s.

[4] V. Sharma, M. Jaiswal, A. Sharma, S. Saini and R. Tomar, "Dynamic Two Hand Gesture Recognition using CNN-LSTM based networks," 2021 IEEE International Symposium on Smart Electronic Systems (iSES), 2021,pp. 224- 229, doi: 10.11.09.

[5] K. Amrutha and P. Prabu, https://www.researchgate.net/publication/366239144_Indian_Sign_Language_to_Speech_Conversion__Using_Convolutional_Neural_Network 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), 2021, pp 1-6, doi:10.1109/ICITIIT51526.2021.9399594.| IEEE Conference Publication

[6] M. Wurangian, “American sign language alphabet recognition,” Medium, 15-Mar-2021. [Online]. Available: American Sign Language Alphabet Recognition | by Marshall Wurangian | MLearning.ai | Medium.

[7] A.Dennisan, https://arxiv.org/abs/1905.05487 rXiv, 10-Feb-2022. [Online].

[8] OpenCV (2022) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/OpenCV.

[9] P. Likhar, N. K. Bhagat and R. G N, "Deep Learning Methods for Indian Sign Language Recognition," 2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin), 2020, pp. 1-6.

[10] Scikit Learn - Documentation Scikit-learn

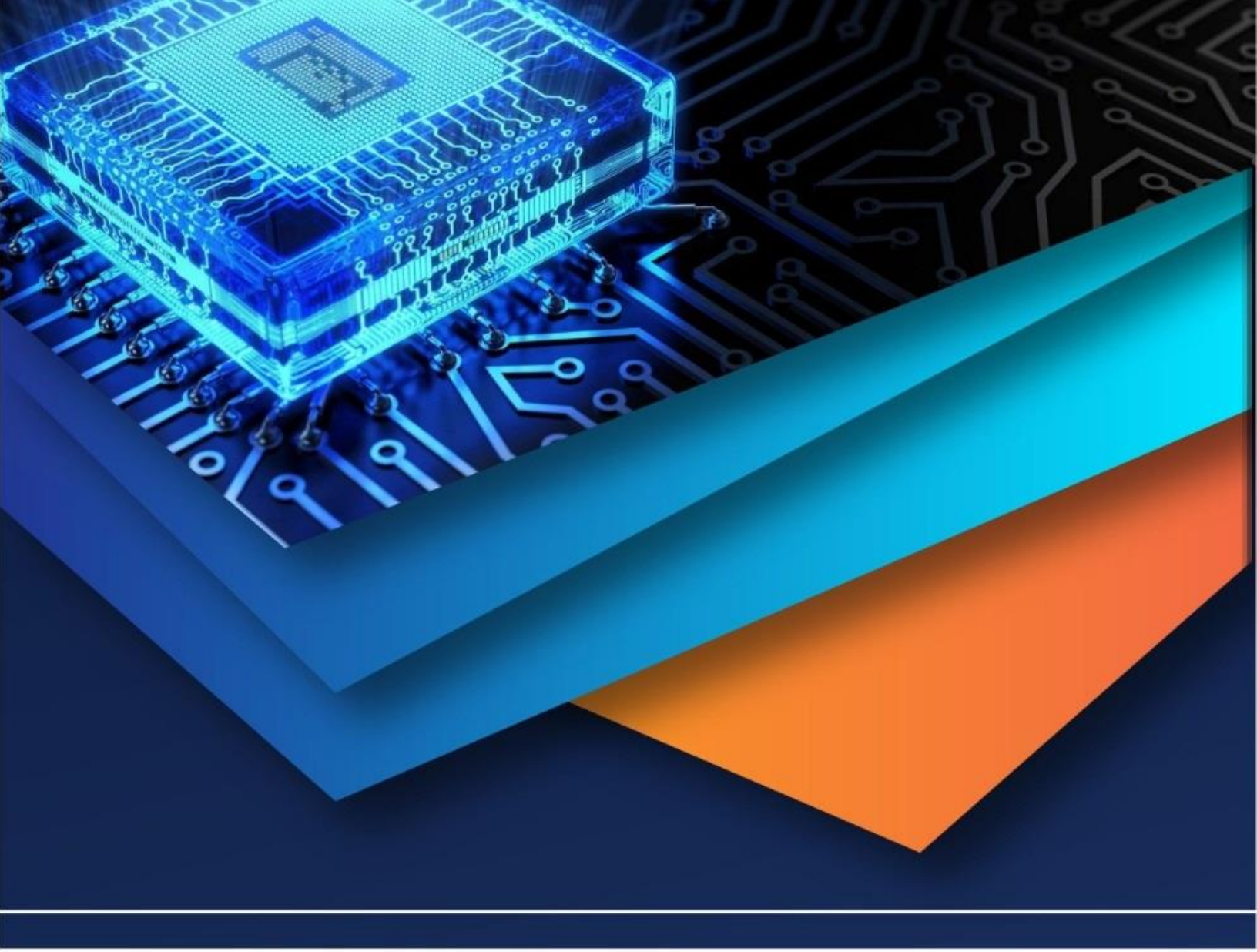
[11] K.Bantupalli and Y. Xie, https://ieeexplore.ieee.org/document/8622141 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.

[12] Shivashankara, Ss, and S. Srinath.https://www.researchgate.net/publication/326972551_American_Sign_Language_Recognition_System_An_Optimal_Approach International Journal of Image, Graphics and Signal Processing 11, no. 8 (2018): 18.
[13] N.K. Bhagat, Y.Vishnusai and G.N. Rathna, https://www.researchgate.net/publication/__343263135_Indian_Sign_Language___Communicator_Using_Convolutional_Neural_Network 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019, pp. 1-8, doi: 10.1109/DICTA47822.2019.8945850.
[14] Q. Wu, Y. Liu, Q. Li, S. Jin and F. Li, "The application of deep learning in computer vision," 2017 Chinese Automation Congress (CAC),Jinan, 2017, pp. 6522-6527.
[15] D. G. Lowe, https://www.scirp.org/reference/referencespapers?referenceid=986635, vol. 13, no. 2, pp. 111122, 1981.
[16] S. Agrawal, A. Chakraborty, and C.M. Rajalakshmi, " Real-Time Hand Gesture Recognition System Using MediaPipe andLSTM", https://ijrpr.com/uploads/V3ISSUE4/IJRPR3693.pdf.
[17] Dr. Mohammed Ahmed , Gunjan Agarwal , Ishan Ahmed , Tamim Ahmad and Sudarshan Tarmale, https://ijarsct.co.in/Paper17693.pdf, "Sign Language Detection using ML Technologies"
[18] Y. Wu, B. Zheng and Y. Zhao, "Dynamic Gesture Recognition Based on LSTM-CNN," 2018 Chinese Automation Congress (CAC), 2018,pp. 2446-2450, doi: 10.1109/CAC.2018.8623035.

BIOGRAPHIES

Vanshika is a final year student in CSE program at the Bhagwan Mahaveer College of Engineering & Management. She has keen interest in Java Programming, DBMS, Data Science, Networking, AI and ML.

Ankur Jain is an Assistant Professor in the Bhagwan Mahaveer College of Engg. And Management. He has completed his M.Tech in Digital Communication from UTU, Dehradun. He is curently pursuing PhD in field of Machine Learning and Deep Learning.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY