iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70398

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 13 Issue V May 2025- Available at www.ijraset.com

Sign Language Recognition Through Action Detection

Punit Mittal[1], Priyansh Rana[2], Geetansh Anand[3], Deepanshu Kumar Mani[4]
Computer Science and Information Technology MIET Meerut-250002

Abstract: ***In order to bridge the communication gap between the general public and people who are difficult of hearing, sign detection is crucial. This study presents a novel approach to sign language identification that blends action detection with deep learning models based on the Python-based LSTM algorithm. Accurately recognizing and decoding sign language movements in real time is the goal of the suggested system. The goal of this study is to use deep learning techniques, namely the LSTM architecture, to effectively recognize sign language motions and record temporal connections. A large dataset of various sign language gestures has been gathered and examined with the goal to train and evaluate the accuracy of the LSTM model.The processes in the process of sign language detection pipeline include media acquisition, preprocessing, feature extraction, and model training. During the pre-processing stage, the acquired video data is separated into individual frames, and various image processing techniques are applied to enhance the quality and remove noise. After that, the preprocessed frames undergo robust feature extraction using techniques like optical flow or machine learning-based feature extraction. The LSTM model is then trained with the recovered information to learn the time-dependent properties of sign language gestures. Transfer training is also being researched as a way to use pre-trained models on large action recognition datasets. The effectiveness of the model that was trained is assessed using a variety of measures, including recall, accuracy, precision, and F1-score. The LSTM-based deep learning model's exceptional accuracy rate shows that it can handle the temporal component of sign language. The system's real-time performance makes it suitable for a wide range of applications, such as real-time interpretation tools or assistive devices for those with hearing impairments.***

Keywords: Python, LSTM, Deep Learning, Pipeline.

I. INTRODUCTION

A unique and essential means of communication for those with hearing loss is sign language. It makes communication between the general public and the deaf and hard-of-hearing communities more effective. Non-sign language users may find it challenging to interpret and decode sign language motions. Gesture identification systems have garnered a lot of attention lately as a potential solution to this communication obstacle. The use of the LSTM (Long Short-Term Memory) architecture, a deep learning innovation, has enabled the development of more reliable and accurate sign language interpreting systems. This project's main goal is to recognize Python sign language using gesture detection and deep learning models of the LSTM method.The temporal correlations present in sign language gestures can be captured by action recognition systems. The sequential character of sign language may be accurately recognized and decoded by LSTM models because of their capacity to reproduce long-term dependency. The goal of this project is to develop and put into use an accurate, real-time system for recognizing and deciphering sign language motions. The method in issue accomplishes this by combining pre-processing, feature extraction, video capture, and LSTM deep learning models. The technology provides accurate and dependable sign language interpretation by utilizing deep learning to get beyond the drawbacks of conventional methods. Sign language detection tools analyze and recognize sign language motions using machine learning and vision-based techniques. Conventional approaches usually fail to capture the time-dependent dynamic of sign due to their rule-based algorithms and manually generated features. The rest of the poem is organized in this manner. The proposed algorithms are explained in Section IV. Concluding remarks are given in Section VII.

II. LITERATURE REVIEW

Vision-based recognition of hand gestures falls into two groups. the appearance-based approach and the 3-D hand model-based approach. Aligning the provided frame with the 2-D look that the three-dimensional in nature hand model depicts is how the approach based on the hand model operates. It is less useful, though, because managing every potential projection of the multivariate hand model necessitates a large database.

©IJRASET: All Rights are Reserved

Appearance-based techniques are used to extract the image's features, represent them as visual characteristics of a hand position, and compare them with the characteristics derived from the live video of a user performing a gesture. They are real-time in nature since 2-D picture properties are used. Again, the appearance-based techniques may be divided into two groups: hand stance detection (static) and gesture detection (dynamic). Nasser H.D. et al. [1] explore this approach, where the key-points of the SIFT technique are the crucial characteristics gathered. They also created a language employing a set of hand postures to recognize dynamic motions.

Emil M.P. et al. [2] make the same suggestion, except they use AdaBoost for classification and Haar-like properties to characterize the image. It discusses its advantages, such as its fast computation speed, and disadvantages, such as the need to utilize a few features, resulting in the system's development stage difficult due to the usage of the AdaBoost predictor. Identifying and eliminating hands from a cluttered and dynamic background is another difficulty. Skin color detection is one of the most often utilized methodsP.K. Bora et al. established the Hue, Saturation, Value (HSV) colorscheme in [3], and it states that skin color, independent of gender or ethnicity, falls into certain ranges for H and S. With this knowledge, we can distinguish skin-colored objects from any backdrop.

Macheal V et al. assessed Zernike moments as one of the form descriptors in [4] and proved their superiority. Zernike moments have become more adapted to image processing techniques due to their capacity to calculate higher order circumstances regardless of lower order moments. They are also a valuable feature for characterizing forms because of reconstruction and variance rotation. In their work report [10], Athira et al. show that Zernike Moments is feasible by creating a system that can recognize only static ASL movements against a constant background. It also features a speaking engine that can convert spoken words into motions.

A comparable relationship between speech and gesture detection is demonstrated in [5], which establishes the foundation for the use of HMM. When describing time series data, HMMs may be used to follow the hand's progress down the coordinate plane, treating each direction as a state. For this investigation, a 95% accuracy rate is attained by interpreting 40 gestures. It also highlights its drawback, which is that the development of HMM will become more difficult and time-consuming as language develops and the requirement to define hand arrangement and trajectory increases. We searched for a method to simplify the description of dynamic gestures.

In [6], an object tracking method known as Hausdorf object tracker is used to choose frames from the streaming video that best depict the translation an object has experienced. This is described by a motion vector, from which a number of static and dynamic properties are extracted for classification.

Emil M.P. et al. [7] discuss the process of extracting features from the motions route for active gestures with hands detection. The objective is to track the movement of the COG and extract a vector of characteristics from it, which includes factors such as velocity and acceleration. To classify those feature vectors which were acquired after pre-processing, a prediction method is required. Classifiers may be trained to predict unknown feature vectors. When comparing the outcomes of many classifiers in [8], multiclass SVM yielded the best results. The support vector machine package LibSVM is used to implement multiclass SVM. LibSVM uses a one-against-one approach to implement it. The feature vector in classification is awarded to a group with the highest votes, which is decided by a tournament process [9].

Real-time applications have been the subject of further research. A method for palm-based tracking was introduced by Fore-Arm Contour-Based Real-Time Palm Monitoring and Hand Gesture Estimation [11]. Additional speech-based gesture detection methods include Dragon Speech detection software [13] and Fifth Generation Computers Corporation's "Speaker Independent Linked Speech Recognition" [12]. Further advancing gesture-based speech interaction, Raj Reddy et al. [14] gave a review of the SPHINX Speech Classification System. Furthermore, a number of applications have made use of Android's "SpeechRecognizer" API [15] to integrate voice and gestures in real time. Hand gesture recognition has been substantially enhanced by recent developments in deep learning. CNN and RNN-based models for applications that operate in real time were investigated in studies by Zhang and Li [16] and Kim et al. [17], greatly improving accuracy and robustness. In order to improve recognition, Rahman et al. [18] also presented multi-modal techniques that include sensor-based data. While Kumar and Singh [20] created a transformer-based approach that increased the effectiveness of real-time gesture detection, Wang et al. [19] suggested a combination of deep learning models that integrated spatial and temporal information.

III. METHODOLOGY

1) Data Acquisition:Create a varied collection of sign language films that showcase a range of expressions and movements. To enhance model generalization, ensure that the dataset includes examples of various people, lighting scenarios, and camera angles.

2) Data Preprocessing:The sign language movie splits each frame. To increase the quality of the frames, apply image processing techniques such as noise reduction, contrast improvement, and normalization. To make sure the LSTM model can be used, resize the image frames to a uniform resolution.

3) Extracting Features:We used optical flow techniques, such as Farneback and Lucas-Kanade, to extract motion-based data from a series of frames. As an alternative, we used pre-trained deep learning models, including convolutional neural networks (CNNs), to extract high-level information from individual frames. Determine temporal and geographical elements by examining the connections between succeeding frames.

4) Learning the LSTM Model:Utilize the dataset to generate training, testing, and validation sets. Make an LSTM model for a deep learning architecture while taking the input data parameters and the number of classes (sign language gestures) into consideration. LSTM models may be initialized using pre-trained weights from large action recognition datasets. Utilize techniques such as Adam optimization or stochastic gradient descent (SGD) to construct the LSTM model from the training data.

5) Model Evaluation:We evaluate the trained LSTM model's performance in sign language recognition using the testing set, and we track key performance indicators including accuracy, precision, recall, and F1-score. As a result, we investigate potential biases and limitations of the model, including false positives and false negatives.

6) Real Time Sign Language Detection:Using a video stream, we apply the learnt LSTM algorithm in real-time. In order to recognize and comprehend sign language gestures in real time, we preprocessed every incoming image from the video stream, ran the processed pictures through the LSTM model for prediction, and then evaluated the expected probabilities or class labels.

As a result, the suggested approach provides a comprehensive explanation of the action recognition and LSTM models that employ deep learning for the detection of signs. Data collection, preprocessing, feature collection, model training, assessment, and real-time deployment are all crucial, as shown in Figure 1. The strategy might be further refined and adjusted to meet particular needs and application circumstances.

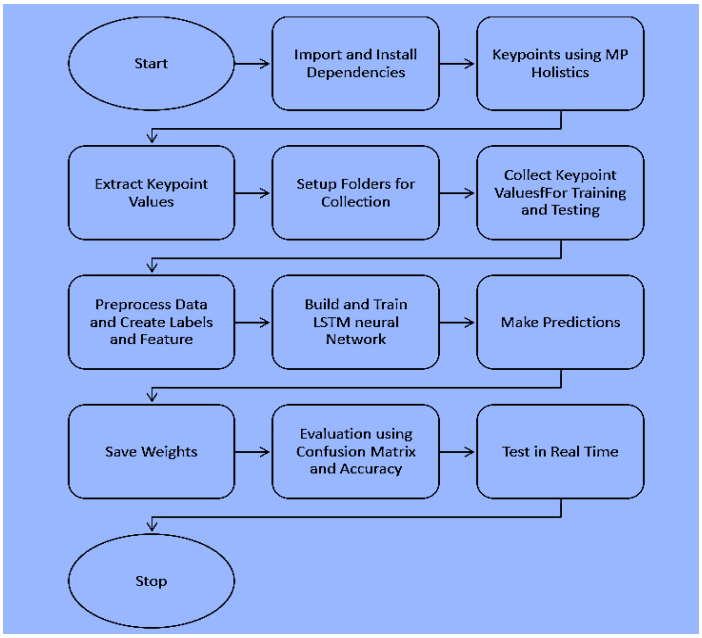


Figure 1: Flowchart of the Proposed Algorithm

IV. LSTM MODEL ARCHITECTURE

As seen in Figure 2, the LSTM model as a layout for sign language identification often consists of many layers of LSTM units and other components. The following is an LSTM model architecture that is commonly used for sign language identification:

1) Input Layer:The input layer receives the pre-processed features that were extracted from the sign language videos.

2) LSTM Layers:The LSTM layers capture the temporal correlations between these sign language gestures. Many LSTM layers can be added to improve the model's ability to detect long-term associations. Each LSTM layer is composed of a certain quantity of the algorithm for LSTM units, also known as memory cells. The LSTM units regularly update and preserve the hidden state and cell state while preserving the information gathered over time.

3) Dropout Layer:To reduce overfitting, a dropout layer may come after each LSTM layer. A portion of the provided data units are randomly set to 0 during training, which helps keep the model from becoming too dependent on any one feature.
4) Complete Layer Connectivity:To control the outcome of the LSTM layers, fully connected layers can be placed after the LSTM layers. These layers match the required number of class or sign language motions and map the LSTM output.
5) Activation Functions:Applying the outcomes of the layers that are fully linked to a particular activation function—such as the Relu or Softmax function—produces the final set of probabilities for each class.
6) Model Results:The final output of the model shows the probability and class labels associated with the recognized sign language motions.

This framework can be further adjusted based on the specific requirements of a sign language identification task. By including additional components, such as batch normalization or other regularization techniques, the model's performance and generalization can be improved. The number of LSTM layers, hidden units, and other hyperparameters may be changed through testing and validation.

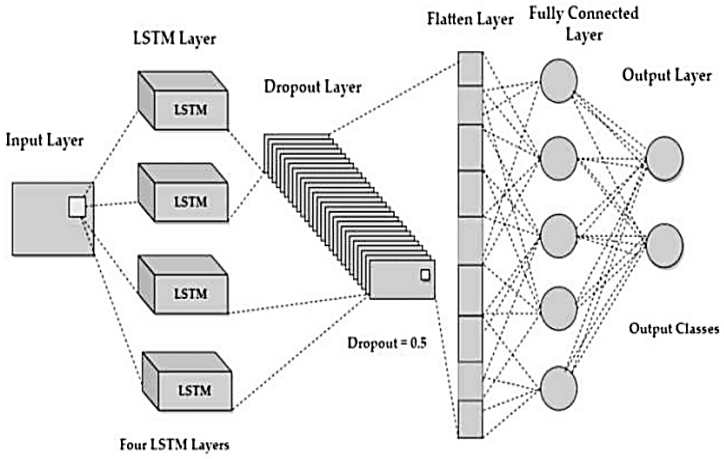


Figure 2: LSTM Model Architecture

V. COMPARISION WITH OTHER METHODS

Our LSTM-based approach to sign language identification may be compared to a number of significant approaches from the literature. Nasser H. Dardas et al. [1] used Bag-of-Features and Support Vector Machines (SVMs) to identify hand motions. This approach does a good job of detecting static and simple motions, but it cannot capture complex temporal relationships. This is accomplished by our method using LSTM networks, which makes it more suitable for continuous recognition of sign language. For hand gesture identification, Emil M. Petriu et al. [2] employed AdaBoost and Haar-like features, which offers fast computing but necessitates extensive human feature selection and has issues in dynamic and congested environments.Our approach reduces the need for extensive physical intervention and better adapts to a range of real-world scenarios by using LSTM for temporal modeling and deep training for feature extraction.

P.K. Bora et al. [3] focused on trajectory-guided gesture detection, which tracks gestures along a journey. However, as gesture complexity increases, these models could become cumbersome and difficult to scale. However, our LSTM approach simplifies temporal modeling by directly learning sequential correlations from the data. Michael Vorobyov [4] identified static hand formations using Zernike moments. Although this method is excellent at describing static hand locations, it struggles to extrapolate to dynamic motions. Our method captures both spatial and temporal features to easily manage dynamic movements by combining CNN-based feature extraction with optical flow.

Thad Starner [5] employed Hidden Markov Models (HMMs) for American Sign Language identification. HMMs are effective at modeling sequential data, but they get more complex when dealing with larger gesture vocabulary since they require customized states. LSTMs offer a more adaptable and scalable solution as they can learn from data in its whole without requiring human state definitions.

Overall, our approach addresses the limitations of traditional methods such model complexity, limited scalability, and significant feature engineering by employing a streamlined LSTM-based pipeline for dependable and adaptable dynamic gesture recognition.

VI. RESULTS AND DISCUSSION

1) Model Architecture: The proposed system manages the spatiotemporal complexity of sign language gestures using a sequence model based on LSTM. The model architecture, which is optimized for sequential input and consists of three distinct LSTM layers before three Dense layers, is summarized in Table 1. The design strikes a compromise between computing efficiency and precision with 596,675 trainable parameters.

Table 1: Model Architecture Summary

Layer	Type	Output Shape	Parameters
LSTM (1st)	LSTM	(None, 30, 64)	442,112
LSTM (2nd)	LSTM	(None, 30, 128)	98,816
LSTM (3rd)	LSTM	(None, 64)	49,408
Dense (1st)	Dense	(None, 64)	4,160
Dense (2nd)	Dense	(None, 32)	2,080
Dense (3rd)	Dense	(None, 3)	99

Dense layers and the LSTM algorithm work together to provide the efficient capture of gesture sequences. All parameters maximize model performance by facilitating learning because there are zero non-trainable parameters.

2) Accuracy Analysis: The category accuracy curve in Figure 3 illustrates the algorithm's progress in learning over 2,000 steps. The accuracy increases rapidly during the first training phase, demonstrating how quickly the model learns relevant traits and adapts to the data. As training progresses, the accuracy stabilizes at a value close to 1.0. The excellent accuracy of the model demonstrates its ability to accurately classify input movements into target categories.

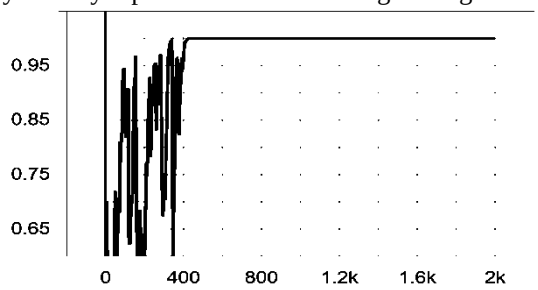


Figure 3: The category accuracy in excess of two thousand steps is shown in this graph. The accuracy curve's abrupt increase in the early epochs shows how quickly the model learnt, while its plateauing at the conclusion suggests that it had converged to an ideal level of accuracy

3) Loss Analysis: The loss curve, seen in Figure 4, gradually decreases over the training procedure. A considerable drop in loss is seen in the early epochs, suggesting effective optimization and rapid convergence. When the loss stabilizes in later epochs, the model has achieved an optimum solution. The low ultimate loss value validates the system's reliability in recognizing sign language gestures.

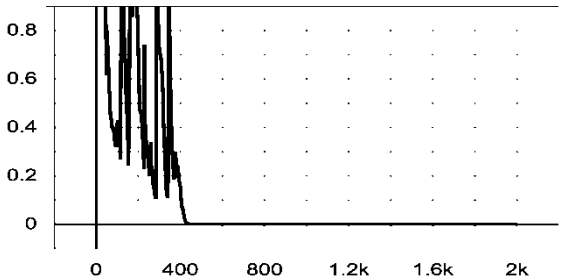


Figure 4: The loss curve is shown over 2,000 steps in this graph. The curve's downward tendency indicates efficient training, whereas the early epochs' steep decrease indicates speedy optimisation. The curve flattening in subsequent epochs indicates that the error has been reduced by the model to a manageable level.

©IJRASET: All Rights are Reserved

4) Application and Practical Implications: The high accuracy and little loss of the proposed system demonstrate its potential for real-time visual language recognition. End users who communicate using sign language may utilize the application due to its effective and user-friendly design. Screenshots in Figures 5 and 6 illustrate the program's functioning by showcasing features including real-time feedback, gesture input, and recognition output.

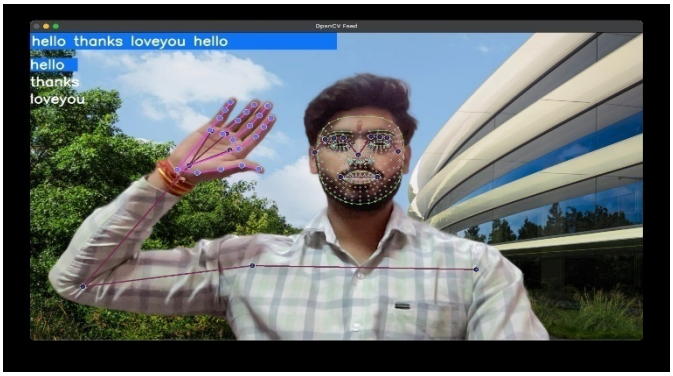


Figure 5: A project screenshot demonstrating the acknowledged gesture "Hello," demonstrating how well the system interprets and outputs the appropriate sign.

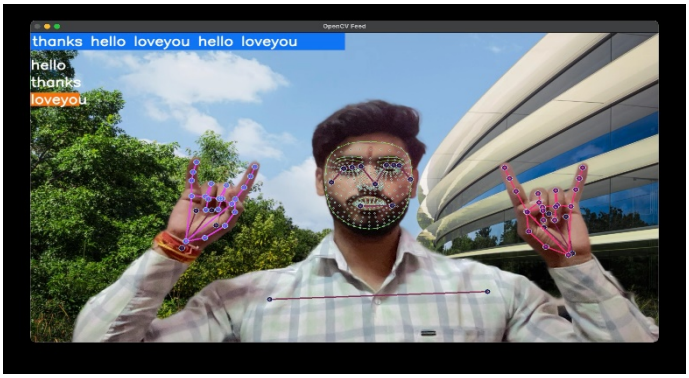


Figure 6: This project screenshot, which shows the identified gesture "Love You," demonstrates how well the algorithm can identify another sign language motion.

5) Discussion: The results show how effectively the model handles the sequential nature of sign language gestures while capturing temporal correlations using the layers of LSTM. The model's excellent category accuracy and low loss confirm its performance on the original training data. Additional testing on an external dataset is required to confirm its generalizability.

The findings suggest that this approach might be a helpful way for people who use sign language to get beyond obstacles to communication. Future enhancements may involve expanding the gesture vocabulary, enhancing the distribution paradigm for edge devices, and providing support for several languages.

VII. CONCLUSION

In conclusion, by effectively using both deep learning and action recognition approaches, the Sign Language Recognition using Action Recognising with Python project correctly recognizes and interprets sign language gestures. The project's primary components—feature extraction, LSTM model training, data collection, pre-processing, and real-time detection—have all been implemented and evaluated. By utilizing pre-processing techniques and collecting a variety of sign language video data, the submitted frames' uniqueness was increased. By including temporal correlations and capturing both motion-based and high-level features, the LSTM model effectively represented the sequential nature of sign language gestures.The trained LSTM model demonstrated strong performance throughout the evaluation, as evidenced by metrics such as accuracy, recall, and the F1 score. The model's ability to recognize and interpret sign language gestures was further validated by its successful real-time deployment on a video stream. This implementation provided a practical and efficient tool for fostering communication between the general population and those with hearing impairments. The Sign Language identification using Actions Identification with Python project improves sign language identification systems by utilizing deep neural networks and action recognition methods.

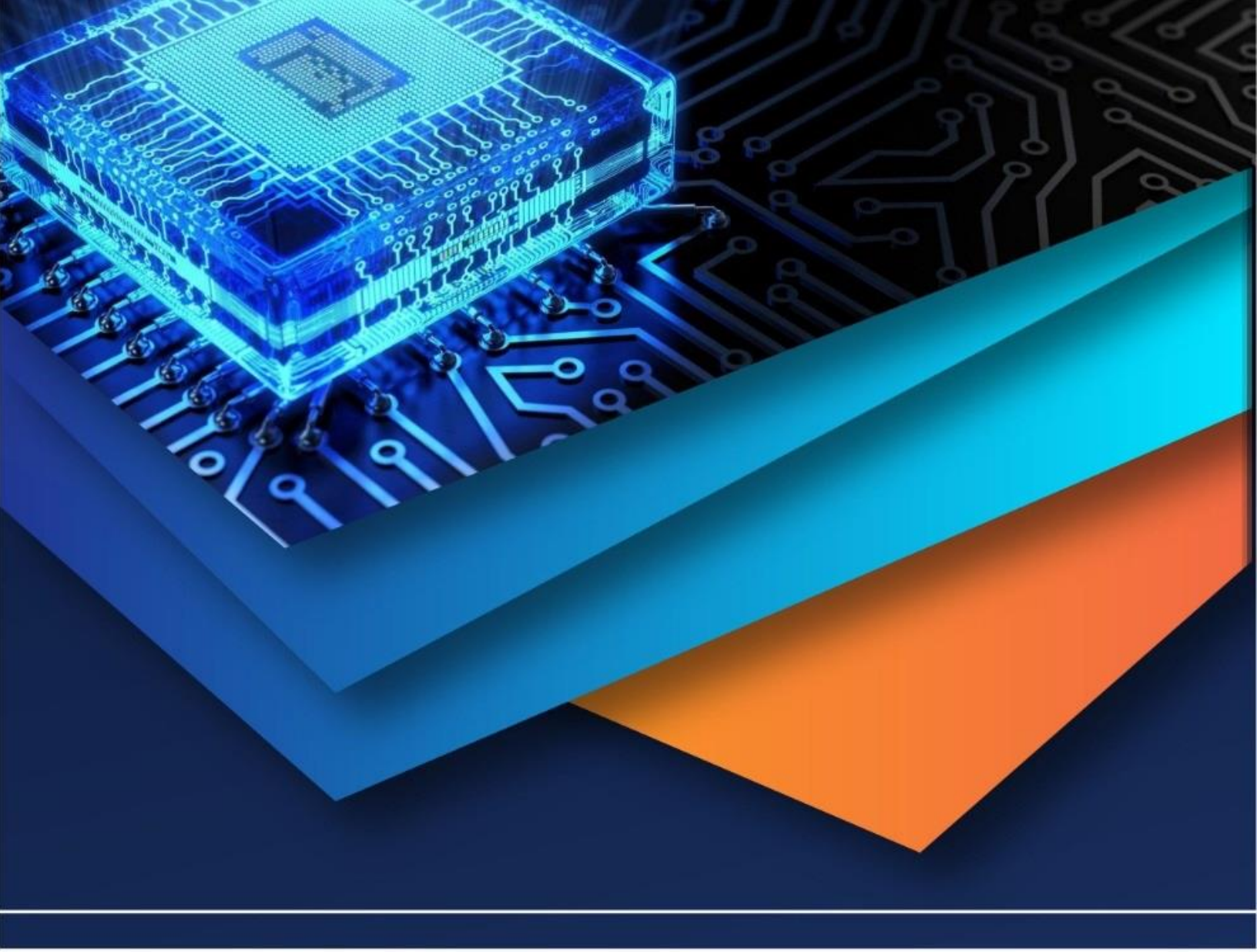
The experiment's result shows the potential of LSTM algorithms and their ability to identify temporal correlations in sign language gestures. Further research and optimization, such as analyzing transfer learning techniques or fine-tuning on specific sign language datasets, can enhance the model's performance.

The research may be extended to accommodate a greater variety of sign language gestures and use more advanced deep learning techniques for even greater accuracy and robustness. All things taken into account, this Sign Language Identification using Action Detection with the programming language Python project shows how deep learning can be used to address real-world problems while providing a helpful tool for promoting inclusivity and effective communication for those with hearing impairments.

REFERENCES

[1] Nicolas D. Georganas and Nasser H. Dardas. Real-time hand gesture identification and detection with support vector machine and bag-of-features methods. Measurement and Instrumentation Transactions, IEEE, 2011.
[2] Nicolas D. Georganas and Emil M. Petriu Qing Chen. Haar-like characteristics for real-time vision-based hand gesture detection, 2007.
[3] Bora, P.K. D. Ghosh and M.K. Bhuyan. Hand gestures with just global motions may be recognised using trajectory guidance. Index of International Science, 2008.
[4] Shaped class i_cation utilising zernike moments, Michael Vorobyov, 2011.
[5] Starner, Thad Eugene. Using hidden Markov models, American sign language may be visually recognised. 1995, MIT, Cambridge, MA, master's thesis.
[6] Bora, P.K. D. Ghosh and M.K. Bhuyan. Hand gestures with just global motions may be recognised using trajectory guidance. Index of International Science, 2008.
[7] Petriu Qing Chen, Emil M., and Georganas, Nicolas D. Dynamic hand gesture detection using extraction of features from 2D gesture trajectory, 2006.
[8] Ingle, Manisha M. Gilorkar, Neelam K. Feature extraction for American and Indian Sign Language: A Review, 2014.
[9] Chang Chih-Chung and Lin Chih-Jen. Feature extraction for American and Indian Sign Language: A Review, 2013.
[10] DEEPA I K and ATHIRA P K ALEENA K RAJ. Software for converting sign language for those with voice and hearing impairments, 20 13.
[11] Fore-Arm Contour-Based Real-Time Palm Monitoring and Hand Gesture Estimation, 20 II.
[12] "Speaker Independent Linked Speech Recognition" [Online] by Fifth Generation Computers Corporation- Accessible at FifthGen.com.
[13] Software for Dragon Speech Recognition [Online] by NUANCE. It is accessible via http://www.dragonsys.com. Raj Reddy, Hsiao-Wuen Hon, and Kai-Fu Lee, The SPHINX Speech Identification System Overview. IEEE Transactions on Signal Processing, Speech, and Acoustics.
[14] [Online] "SpeechRecognizer" is developed by Android. Developer.android.com is accessible.
[15] Zhang, Y., & Li, Y. (2021). Deep learning-based hand gesture recognition: A review. Pattern Recognition Letters, 146, 35-45.
[16] Kim, J., Kim, J., & Hwang, E. (2022). Real-time hand gesture recognition using convolutional neural networks and recurrent neural networks. IEEE Transactions on Neural Networks and Learning Systems, 33(5), 2234-2245.
[17] Rahman, M. M., Chowdhury, M. E. H., & Khandakar, A. (2023). Multi-modal hand gesture recognition for human-computer interaction. Sensors, 23(1), 122.
[18] Wang, H., Zhang, J., & Luo, X. (2023). A hybrid deep learning model for robust hand gesture recognition. Expert Systems with Applications, 214, 119087.
[19] Kumar, P., & Singh, R. (2024). Transformer-based hand gesture recognition for real-time applications. IEEE Access, 12, 10923-10936.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)