Real Time Hand Sign Language Translation: Text and Speech Conversion

Yasaswini. M [1], Sanjay. S [2], Lokesh. U [3], Arun. M. A [4]

[1] Vel tech high tech Dr. Rangarajan Dr. Sakunthala engineering college Chennai
[2] Vel tech high tech Dr. Rangarajan Dr. Sakunthala engineering college Chennai
[3] Vel tech high tech Dr. Rangarajan Dr. Sakunthala engineering college Chennai
[4] Vel tech high tech Dr. Rangarajan Dr. Sakunthala engineering college Chennai

---***---

Abstract - The Sign Language conversion project presents a real-time system that can interpret sign language from a live webcam feed. Leveraging the power of the Media pipe library for landmark detection, the project extracts vital information from each frame, including hand landmarks. The detected landmark coordinates are then collected and stored in a CSV file for further analysis. Using machine learning techniques, a Random Forest Classifier is trained on this landmark data to classify different sign language patterns. During the webcam feed processing, the trained model predicts the sign language class and its probability in real-time. The results are overlaid on the video stream, providing users with immediate insights into the subject's sign language cues.

Key Words*: *Sign language recognition, Hand gesture recognition, Gesture-to-text conversion, Visual language processing.

I INTRODUCTION

Sign language uses manual and visual mode to convey what the person thinks, feels and experiences. For the local citizens of India, a foreign language like English would never become a major sign language because people normally learn to speak in their mother tongue which is not English. On the other hand, there are a variety of sign languages present throughout India. The different parts of India have little difference in signing but the grammar remains the same throughout the country. Hence, a sign language that is standardized and can be used by anyone who is deaf and mute and understood by the normal people is mandatory. Therefore, just like English language and is known by majority of the citizens, which makes communication efficient, Indian Sign Language has to be standardized.

II PROBLEM STATEMENT

In today's digital age, effective communication is a fundamental human right, yet many individuals in the deaf and hard of hearing community continue to face significant communication barriers. One of the most prominent of these challenges is the limited accessibility of sign language interpretation services, which are often expensive, not readily available, or subject to delays. Moreover, the global diversity of sign languages poses a considerable challenge in providing accurate interpretation for all users. To bridge this gap and promote inclusivity, there is a pressing need for innovative solutions that leverage technology, such as computer vision and natural language processing, to develop real-time and cost-effective sign language interpretation systems.

III NEED FOR THE SYSTEM

The Sign Language Conversion project aims to provide real-time sign language analysis by leveraging computer vision and machine learning techniques. It focuses on detecting hand landmarks from a live webcam feed using the Mediapipe library. A Random Forest Classifier is trained on this data to classify various sign language patterns, enabling the system to interpret and differentiate between different gestures, expressions, and postures. The project integrates seamlessly with a webcam feed, making it readily applicable in various real-world scenarios. It has potential applications in human-computer interaction, user behavior analysis. Overall, the scope of the Sign Language recognition project encompasses real-time sign language analysis, landmark detection, and machine learning-based classification, with the potential for broader applications and research in the field of non-verbal communication interpretation.

V SYSTEM DESIGN

User Interface (UI): The UI should be intuitive and user-friendly, designed to allow users to easily navigate through the app's features. It should include features such as image uploading, disease identification results display, treatment recommendations, and options for further actions.

Visual elements such as buttons, menus, and icons should be clear and accessible, catering to users with varying levels of technical expertise.

Image Processing: Upon uploading images of diseased plants, the app should preprocess the images to enhance clarity and remove any background noise. Image segmentation techniques can be applied to isolate diseased areas from healthy ones, improving the accuracy of disease detection.

Feature extraction methods may be employed to capture relevant visual attributes of plant diseases, such as texture, color, and shape, to be used as input for the machine learning model.

Machine Learning: A machine learning model, such as a convolutional neural network (CNN), can be trained on a dataset of labelled images of diseased plants to recognize patterns indicative of various diseases.

The model should be optimized for accuracy, speed, and resource efficiency to perform inference tasks efficiently on mobile devices. Transfer learning techniques can be utilized to leverage pre-trained models and adapt them to the specific domain of plant disease detection, reducing the need for large-scale training datasets.

Database: The app should maintain a database to store information about plant diseases, including images, descriptions, symptoms, treatment options, and expert recommendations. The database should be scalable and secure, allowing for efficient retrieval and updating of information as new data becomes available. Data integrity and consistency should be ensured through proper data management practices, including regular backups and version control.

Analysing Accuracy: The accuracy of the plant disease detection system can be evaluated through rigorous testing and validation processes. This involves assessing the model's performance on a diverse range of plant images, including different species, disease severities, and environmental conditions. Metrics such as precision, recall, F1-score, and confusion matrix can be used to quantitatively measure the model's accuracy and performance. Continuous monitoring and feedback from users can help identify areas for improvement and refine the system's accuracy over time.

Security and Privacy: User authentication mechanisms, such as username/password or biometric authentication, should be implemented to restrict access to sensitive features and data. Data encryption techniques should be employed to secure the transmission and storage of user data, including uploaded images and personal information.

Privacy policies should be clearly communicated to users, outlining how their data will be collected, used, and protected, and providing options for controlling their privacy settings. Compliance with relevant regulations and standards, such as GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act), should be ensured to safeguard user privacy and data security.

VI ALGORITHM:

Introduction to Machine Learning:

Machine Learning (ML) is a transformative subset of artificial intelligence (AI) that empowers computers to learn from data and improve their performance on tasks without explicit programming. At its core, ML involves algorithms and statistical models that recognize patterns, relationships, and trends within datasets, enabling the automation of predictions, classifications, and decision-making processes. Three primary categories of ML include supervised learning, which relies on labeled data for tasks like classification and regression; unsupervised learning, which discovers patterns and structures in unlabeled data through clustering and dimensionality reduction; and reinforcement learning, where agents learn optimal actions by interacting with an environment.

Data Collection and Storage– This component contains a variety of data sources, including databases, data lakes, and APIs. It also includes gathering information from multiple sources and storing it in a centralized area for processing.

Data Preprocessing– This component includes data cleansing, feature engineering, and data normalization. Data preparation is essential for increasing data quality and assuring data suitability for analysis

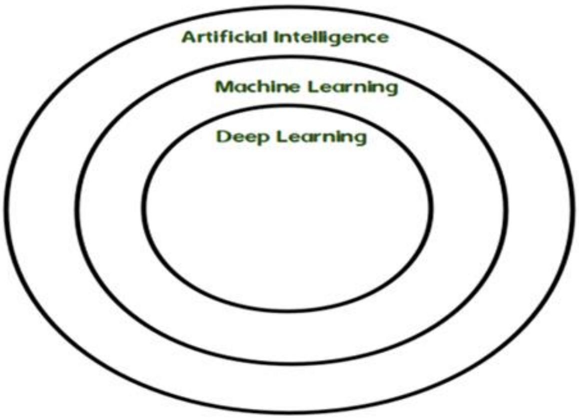
Model Training and Tuning– In this phase, you choose the best algorithm, train the model, and fine-tune the hyperparameters. The objective is to create a model that predicts outcomes correctly and generalizes well to new inputs**.**

Model Deployment and Monitoring– This component refers to deploying the model to a production environment and continuously monitoring its performance. This assists in identifying any problems and ensuring that the model is working as planned.

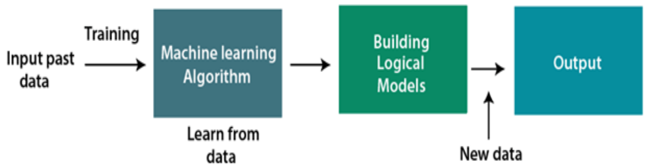
International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 08 | Aug - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

User Interface– The interface through which users interact to get the model's predictions is included in this component. A dashboard, a mobile app, or a web application might be used.

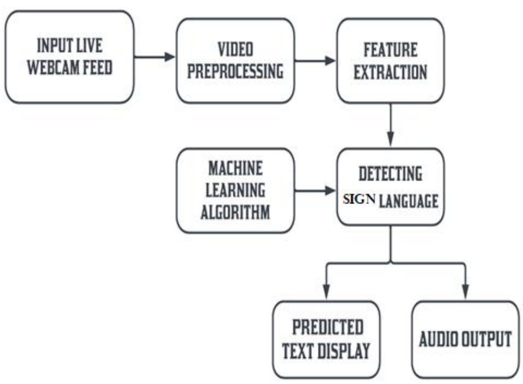
Iteration and feedback– Gather user input and apply it to enhance the model's performance. To increase the model's accuracy, the feedback may be used to retrain the model.



Working:
The Machine Learning process starts with inputting training data into the selected algorithm. Training data being known or unknown data to develop the final Machine Learning algorithm. The type of training data input does impact the algorithm, and that concept will be covered further momentarily. New input data is fed into the machine learning algorithm to test whether the algorithm works correctly. The prediction and results are then checked against each other.



VII. ARCHITECTURE DIAGRAM



VIII. RESULT

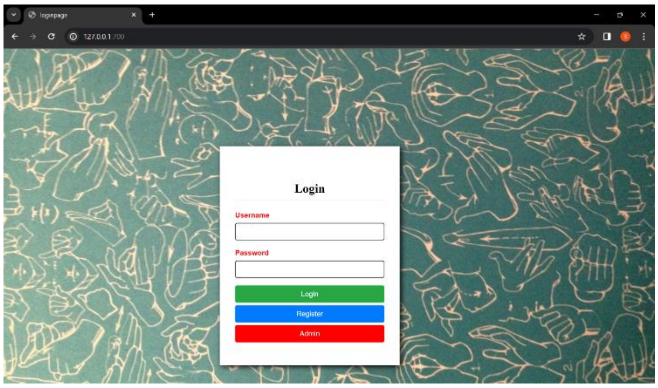


Fig 8.1: User Login Page

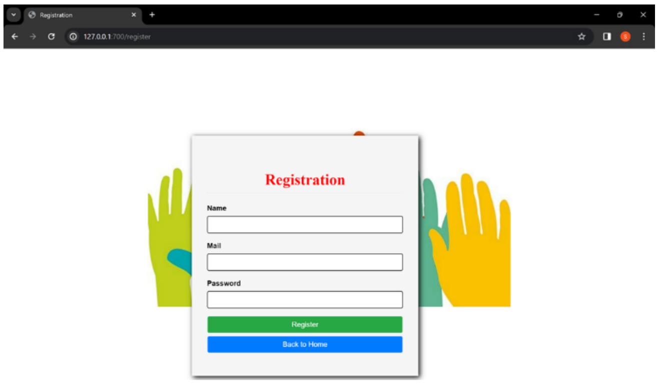


Fig 8.2: New User Registration Page

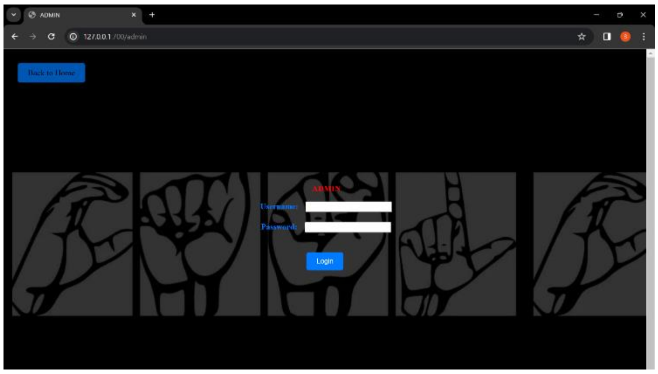


Fig 8.3: Admin Login Page

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 08 | Aug - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Fig 8.4: Home Page

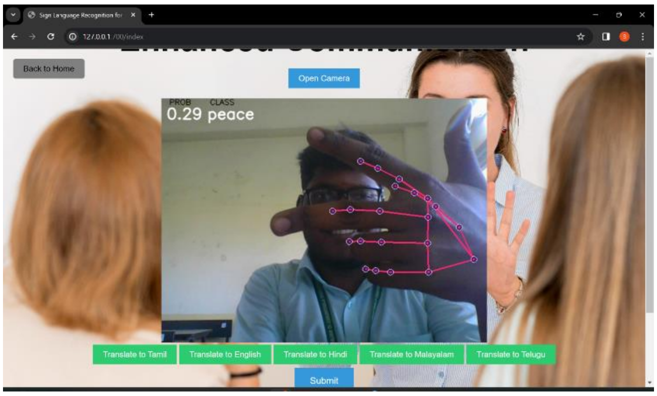


Fig 8.5: Sign Language Detection

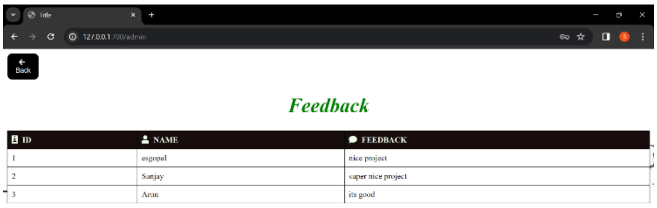


Fig 8.6: User Feedback Viewing Page

VIII. CONCLUSION

In conclusion, the Sign Language Conversion project successfully develops an automated and real-time system for interpreting and classifying sign language cues from live webcam feeds. Through the integration of computer vision and machine learning, the system detects hand landmarks, providing a comprehensive view of non-verbal communication. The use of a Random Forest Classifier ensures accurate and objective sign language classification, making the system reliable and consistent. The user-friendly frontend enhances the interactive experience, displaying real-time analysis results and empowering users with instantaneous feedback. With applications in human-computer interaction, user behavior analysis, the project represents a significant advancement in non-verbal communication analysis and offers valuable insights for future research and development in this domain.

IX. REFERENCES

[1]Y. Saleh and G. F. Issa, ‘‘Arabic sign language recognition through deep neural networks fine-tuning,’’ Int. J. Online Biomed. Eng., vol. 16, no. 5, pp. 71–83, 2020.

[2] X. Jiang, M. Lu, and S.-H. Wang, ‘‘An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language,’’ Multimedia Tools Appl., vol. 79, nos. 21–22, pp. 15697–15715, Jun. 2020.

[3] O. Sevli and N. Kemaloğlu, ‘‘Turkish sign language digits classification with CNN using different optimizers,’’ Int. Adv. Researches Eng. J., vol. 4, no. 3, pp. 200–207, Dec. 2020

[3] A. Tyagi and S. Bansal, ‘‘Feature extraction technique for vision-based Indian sign language recognition system: A review,’’ in Computational Methods and Data Engineering. Singapore: Springer, 2021, pp. 39–53.

[4] M. Mukushev, ‘‘Evaluation of manual and non-manual components for sign language recognition,’’ in Proc. 12th Lang. Resour. Eval. Conf., Eur. Lang. Resour. Assoc. (ELRA), 2020, pp. 1–6.

[5] A. Tunga, S. V. Nuthalapati, and J. Wachs, ‘‘Pose-based sign language recognition using GCN and BERT,’’ in Proc. IEEE Winter Conf. Appl. Comput. Vis. Workshops (WACVW), Jan. 2021, pp. 31–40.

[6] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, ‘‘Neural sign language translation,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7784–7793.

[7]S. Aly and W. Aly, ''DeepArSLR: A novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition,'' IEEE Access, vol. 8, pp. 83199–83212,

2020.

[8] D. Li, X. Yu, C. Xu, L. Petersson, and H. Li, ''Transferring cross-domain knowledge for video sign language recognition,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6205–6214.

[9] R. Rastgoo, K. Kiani, and S. Escalera, ''Video-based isolated hand sign language recognition using a deep cascaded model,'' Multimedia Tools Appl., vol. 79, nos. 31–32, pp. 22965–22987, Aug. 2020.

[10] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, ''Skeleton aware multimodal sign language recognition,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2021, pp. 3413–3423.