iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XII **Month of publication:** December 2024

DOI: https://doi.org/10.22214/ijraset.2024.65817

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 XII Dec 2024- Available at www.ijraset.com

Sign Language Detection Using Deep Learning

Pushpa R N[1], Deepika H D[2], Aishwarya Patil H M[3], Akash L Naik[4], Disha Shetty[5]
[1, 2, 3, 4, 5]*Department of Computer Science and Engineering, Jawaharlal Nehru New College of Engineering, Shimoga, India*

Abstract: ***Sign language recognition is an essential tool for bridging communication gaps between individuals with hearing or speech impairments and the broader community. This study introduces an advanced sign language recognition system leveraging computer vision and machine learning techniques. The system utilizes real-time hand tracking and gesture recognition to identify and classify hand gestures associated with common phrases such as "Hello," "I love you," and "Thank you." A two-step approach is implemented: first, a data collection module captures hand images using a robust preprocessing pipeline, ensuring uniformity in image size and quality; second, a classification module uses a trained deep learning model to accurately predict gestures in real-time. The framework integrates OpenCV for image processing, CVZone modules for hand detection, and TensorFlow for gesture classification. Extensive testing demonstrates the system's capability to process live video input, classify gestures accurately, and display corresponding labels seamlessly. This solution addresses challenges in gesture recognition, such as variable hand shapes and dynamic backgrounds, through efficient preprocessing and model training. By offering a scalable and efficient design, this work has the potential to contribute significantly to assistive technologies and accessible communication systems, paving the way for further advancements in human-computer interaction and inclusive technology.***

Keywords: ***Sign Language Recognition, Gesture Recognition, Hand Tracking, Computer Vision, Deep Learning, Real-Time Classification, Inclusive Communication, Human-Computer Interaction, OpenCV, TensorFlow, CVZone, Image Preprocessing, Accessibility Solutions, Machine Learning Applications.***

I. INTRODUCTION

Sign language is a vital mode of communication for individuals with hearing or speech impairments, enabling them to express themselves effectively. However, its reliance on visual gestures often limits seamless communication with people unfamiliar with sign language, creating barriers in everyday interactions. Addressing this challenge, advancements in technology have opened up possibilities for automated sign language recognition systems. This paper presents a comprehensive approach that integrates computer vision and deep learning techniques to recognize hand gestures in real-time. Using robust hand-tracking methods and gesture classification algorithms, the proposed system efficiently processes live video input to detect and classify gestures into meaningful phrases. The system leverages OpenCV for image processing, TensorFlow for gesture classification, and the CVZone library for hand tracking, ensuring accurate and efficient operation. By addressing key challenges such as varying hand shapes, orientations, and environmental conditions, this work aims to enhance accessibility and foster inclusivity. The development of such systems is a step toward bridging communication gaps and empowering individuals with hearing or speech impairments to interact more seamlessly with the world around them. Additionally, the system's capability to process live video feeds and provide instant feedback highlights its potential for real-world applications. By enhancing accessibility and promoting inclusivity, this work not only facilitates smoother interactions for individuals with hearing or speech impairments but also contributes to the development of assistive technologies that align with the broader goals of accessible communication and human-computer interaction advancements.

II. LITERATURE SURVEY

[1] A. Pathak, A. Kumar, P. Priyam, P. Gupta, and G. Chugh The paper discusses a real-time system for sign language detection using a pre-trained SSD MobileNet V2 model and CNN-based framework to process hand gestures via webcam input. The system captures, analyzes, and recognizes gestures like "Hello" and "Thank You" with 70-80% accuracy, highlighting accessibility and cost-effectiveness while addressing challenges like inconsistent lighting and overlapping gestures. [2] Maheshwari Chitampalli, Dnyaneshwari Takalkar, Gaytri Pillai, Pradnya Gaykar, Sanya Khubchandani This study proposes a sign language detection system utilizing CNNs to map hand movements to text or speech with 95% accuracy. Key processes include gesture segmentation, feature extraction, and real-time recognition, with plans to expand adaptability for diverse conditions and environments. [3] Monisha H. M., Manish B. S., Ranjini Ravi Iyer, Siddarth J. J.

©IJRASET: All Rights are Reserved

The research presents a real-time detection system combining hand tracking and deep learning through an FCNN framework. Mediapipe and TensorFlow enhance accuracy for gesture recognition, with applications in education and inclusivity, and future goals of incorporating depth-sensing and NLP for translation. [4] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh This paper outlines an Indian Sign Language recognition system using TensorFlow's Object Detection API, achieving 85.45% confidence on a small dataset of 650 images. It emphasizes cost-effectiveness, with potential for global language adaptation and sentence-level recognition in the future. [5] Dessai, S., & Naik, S. A literature review on Indian Sign Language systems compares vision-based and sensor-based methods, highlighting algorithms like CNN and SVM. Challenges include real-time processing and dataset limitations, with suggestions for integrating gestures with facial expressions to enhance accuracy. [6] Serai, D., Dokare, I., Salian, S., Ganorkar, P., & Suresh, A. The study proposes a sign language recognition system using CNNs and transfer learning with models like GoogleNet and AlexNet. Preprocessing techniques enhance accuracy, and future work aims to incorporate two-handed gestures and improved hardware capabilities. [7] Sreyasi Dutta, Adrija Bose, Sneha Dutta, Kunal Roy This paper explores the use of LSTM models and optical flow algorithms for action-based real-time sign language detection. The system achieves high accuracy and demonstrates potential in assistive devices and real-time interpretation for the hearing impaired. [8] Refat Khan Pathan, Munmun Biswas, Suraiya Yasmin, Mayeen Uddin Khandaker, Mohammad Salman, Ahmed A. F. Youssef A robust system for American Sign Language recognition integrates image and hand landmark fusion through a multi-headed CNN, achieving a test accuracy of 98.98%. The approach emphasizes real-world adaptability while minimizing computational resource requirements. [9] Ashok K. Sahoo, Gouri Sankar Mishra, Kiran Kumar Ravulakollu The paper surveys advancements in sign language recognition, highlighting methods for static and dynamic gesture classification. It addresses challenges like limited datasets and emphasizes real-world adaptability through neural networks and image processing techniques.

III.METHODOLOGY

A. Data Collection

In your project, the data collection step involves gathering images and videos of different sign language gestures. This dataset needs to be diverse and comprehensive to account for different hand positions, backgrounds, lighting conditions, and gestures. The dataset should be well-labeled, with each gesture mapped to a class. Quality is important, so images should have high resolution, consistent lighting, and clear hand visibility. If you have video data, it should be trimmed to focus on the relevant sections of hand movements.

B. Data Pre-processing

Project utilizes the Keras ImageDataGenerator class for data pre-processing, which includes the following steps:

1) Rescaling: Images are normalized by scaling pixel values to the range [0, 1], which is done by dividing pixel values by 255. This normalization helps the model converge faster during training.
2) Data Augmentation: The project uses transformations to increase the variability of the training data and prevent overfitting:
3) Shear Range: Applies a shear transformation, which helps simulate slight changes in hand orientation.
4) Zoom Range: Randomly zooms into parts of the image, making the model more resilient to variations in distance.
5) Horizontal Flip: Flips images horizontally, improving the model's robustness to mirrored gestures.
6) Validation Data: Only rescaling is applied to the validation set, ensuring it serves as an unbiased measure of the model's performance during training.

C. Feature Extraction

Feature extraction in your project is accomplished using a Convolutional Neural Network (CNN). The architecture is defined to automatically learn hierarchical features from the images. The steps involved are:

1) Model Architecture: A Sequential model is defined with several convolutional layers, max-pooling layers, and a dense layer. The final dense layer uses a softmax or sigmoid activation function, depending on whether the task is binary or multi-class classification.

2) Compilation: The model is compiled using:

- Loss Function: Binary cross-entropy or categorical cross-entropy depending on the classification type.
- Optimizer: RMSprop or Adam optimizer for efficient training.
- Metrics: Accuracy is used to monitor the model's performance.

3) Data Flow: The training and validation data generators are created using the flow_from_directory method. This method loads images in batches, applies the transformations defined in the ImageDataGenerator, and prepares them for training.
4) Model Training: The fit_generator() method is used to train the model with the generated data. This method allows the model to handle data that isn't entirely loaded into memory, which is crucial for large datasets.

D. Model Training and Saving

Training the CNN involves running the model on the training data for a specified number of epochs. The training process should be monitored to detect overfitting or underfitting, adjusting hyperparameters as needed. The trained model is then saved to a file (e.g., sign_language_model.h5) for later use in real-time detection.

E. Image Classification and Prediction

After the model is trained, it can be used for classifying new images or video frames:

1) Loading the Model: The trained model is loaded using load_model(), which ensures the model weights and architecture are accessible for prediction.
2) Pre-processing the Input Image: The input image is pre-processed to match the format the model expects (e.g., resizing to 224x224 pixels, normalizing pixel values).
3) Prediction: The model makes a prediction by outputting class probabilities. The class with the highest probability is taken as the final prediction.
4) Displaying Results: The predicted class and confidence score are displayed, providing feedback about the model's certainty.

F. Real-Time Sign Language Detection

1) Video Capture: Use OpenCV to capture live video from a camera.
2) Frame Pre-processing: Each frame is resized and normalized similarly to the pre-processed training images.
3) Model Inference: Each frame is fed to the trained CNN model, which classifies the sign language gesture.
4) Output Display: Overlay the predicted class and confidence score on the video feed for real-time feedback.
5) Performance Optimization: Implement multi-threading to handle video input and processing efficiently, ensuring that the frame rate is suitable for real-time performance.

G. Additional Considerations for Real-Time Processing

1) Frame Skipping: To maintain performance, skip frames selectively, processing only every nth frame.
2) Parallel Processing: Use Python's threading or multiprocessing modules to parallelize frame processing.
3) Optimized CNN Architecture: Consider using a lighter CNN architecture (e.g., MobileNet) for faster inference in real-time applications.

IV. RESULT

The system accurately identified predefined hand gestures in real-time, showcasing its reliability across different conditions, orientations, and hand sizes. These results indicate its effectiveness for use in assistive communication solutions. The following are the snapshots of result.

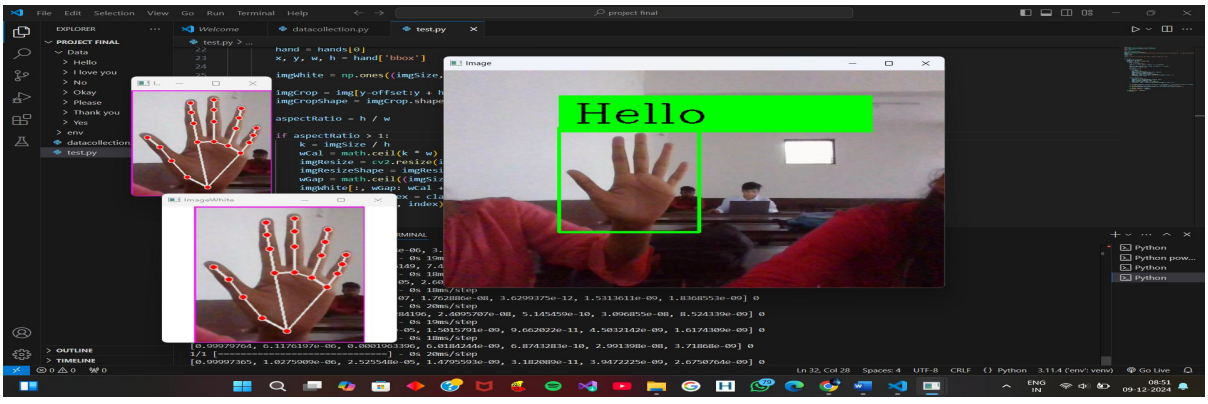


Fig.1 Hello

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 XII Dec 2024- Available at www.ijraset.com

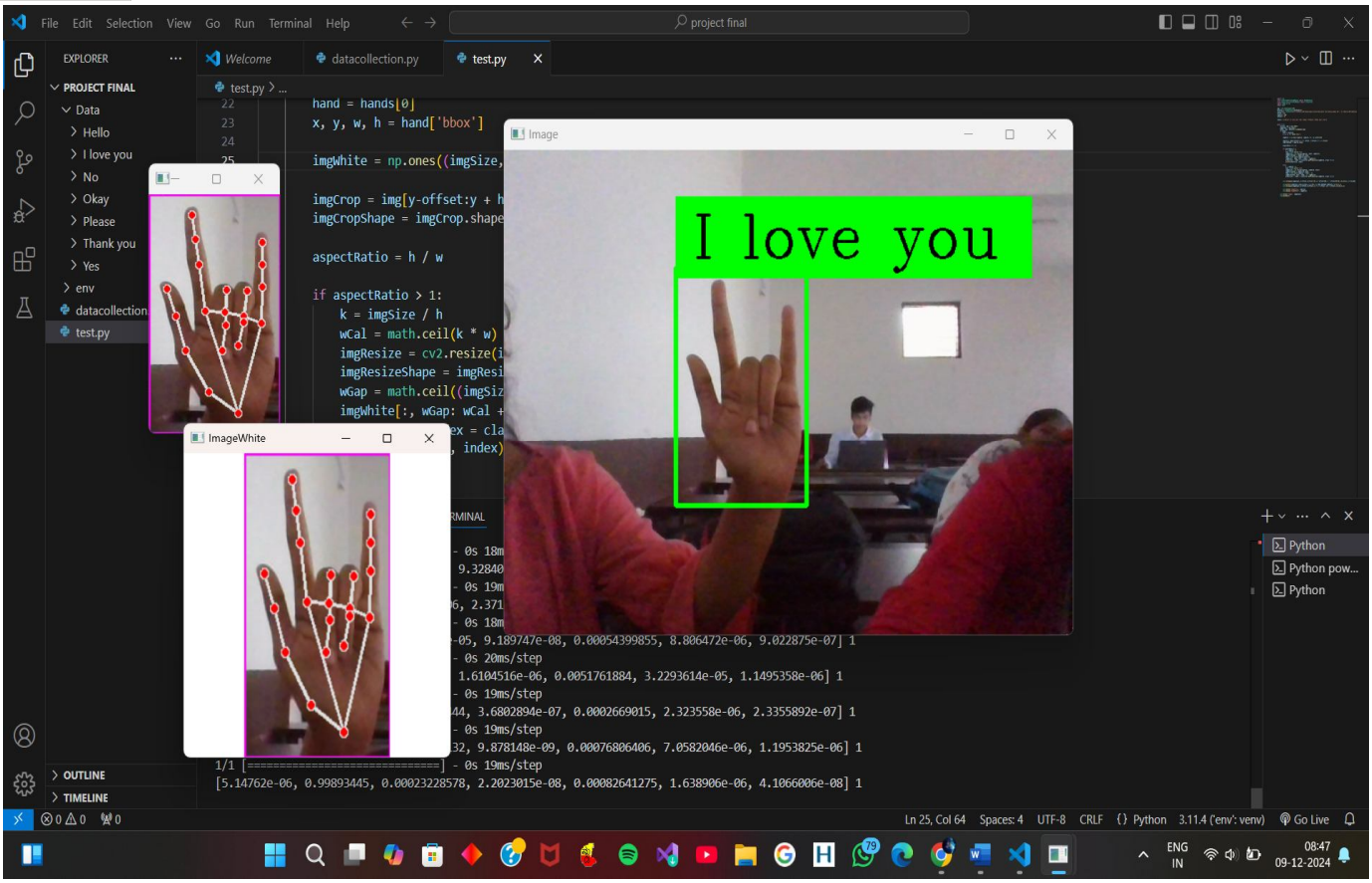


Fig. 2 I Love You

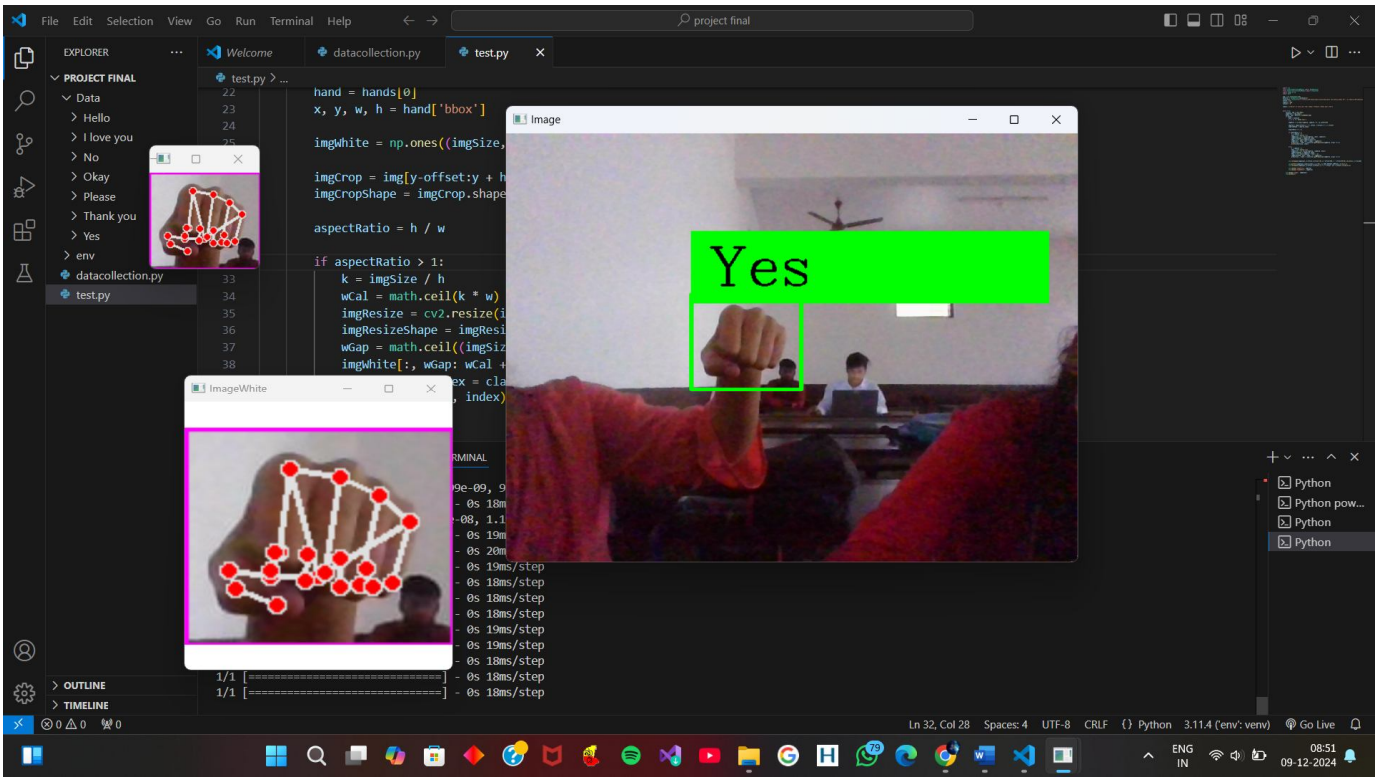


Fig.3 Yes

©IJRASET: All Rights are Reserved

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 XII Dec 2024- Available at www.ijraset.com

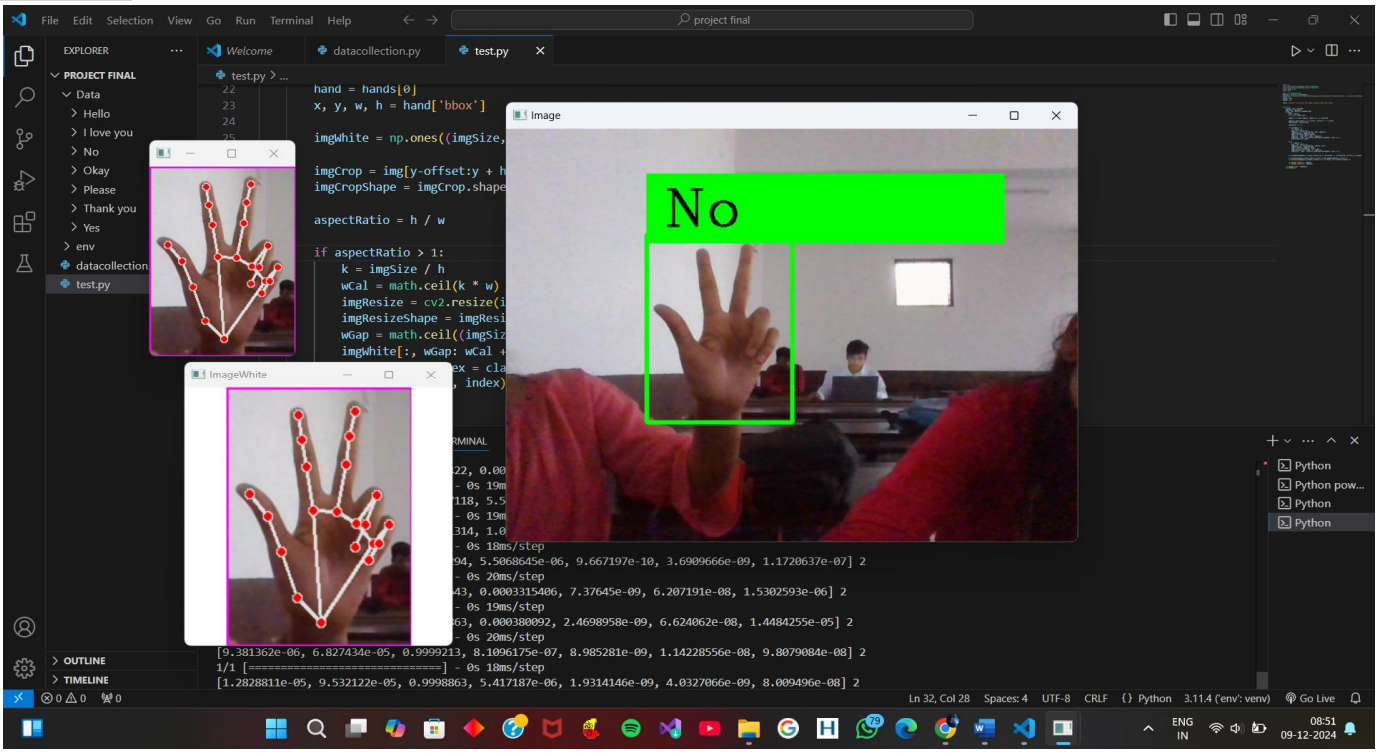


Fig.4 No

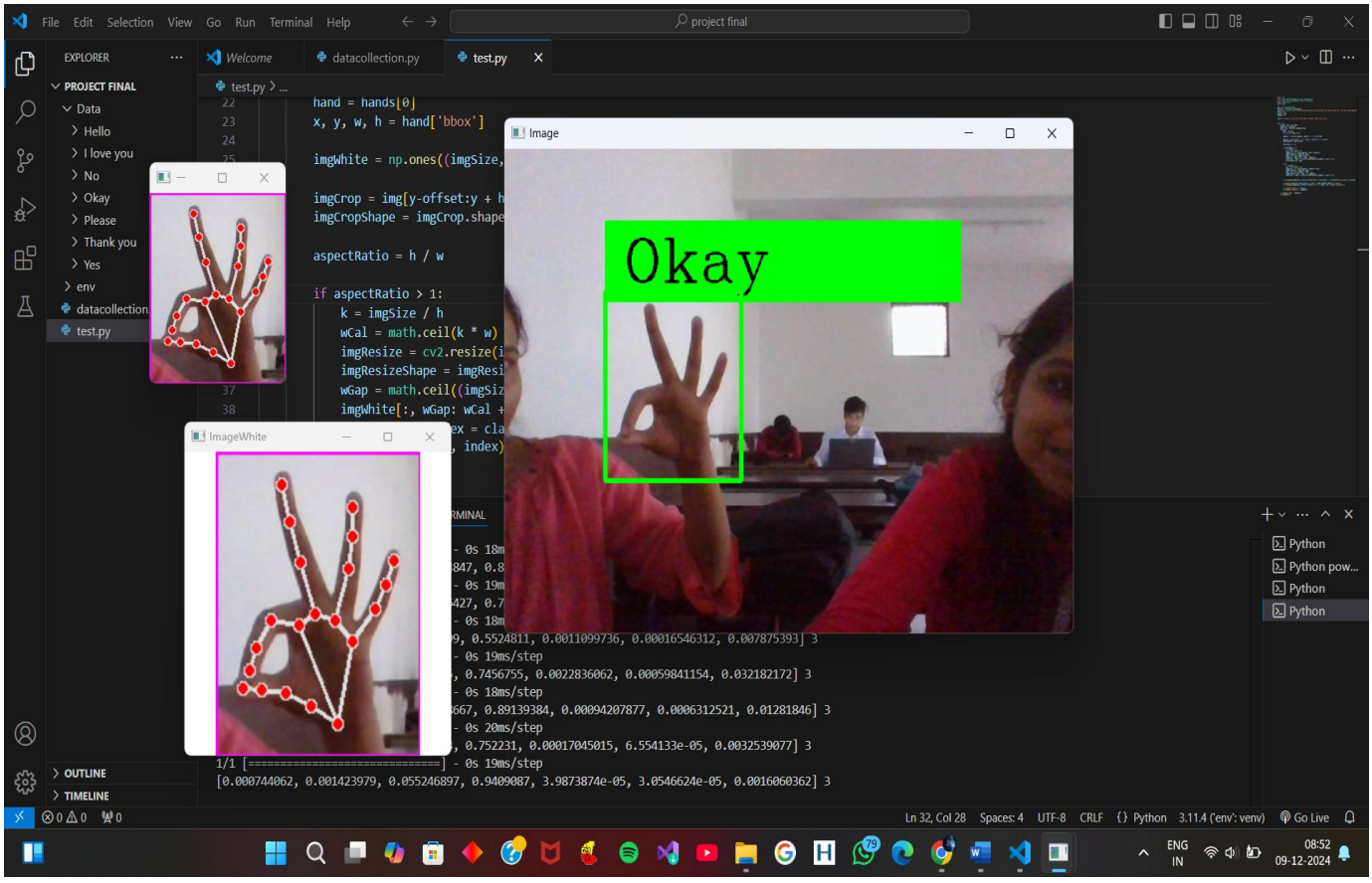


Fig.5 Okay

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 XII Dec 2024- Available at www.ijraset.com

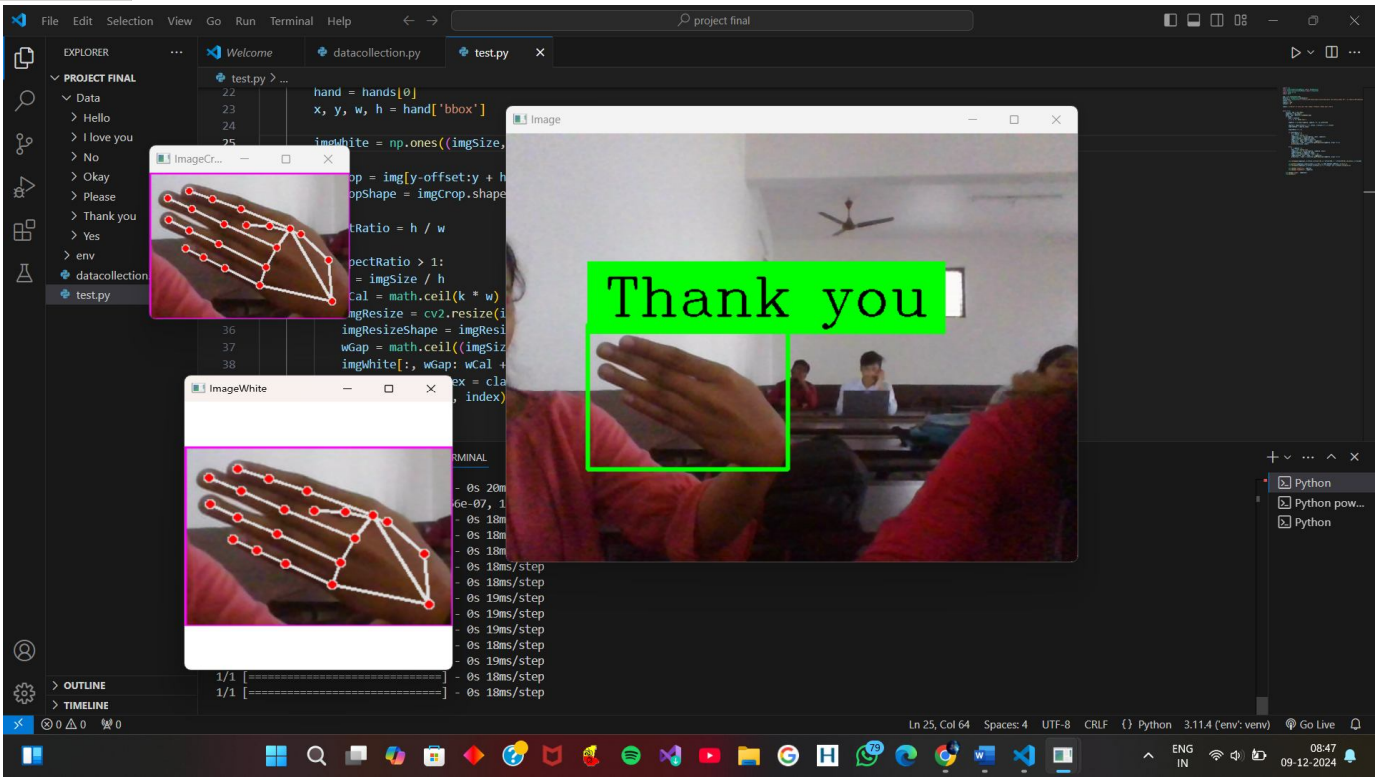


Fig.6 Thank you

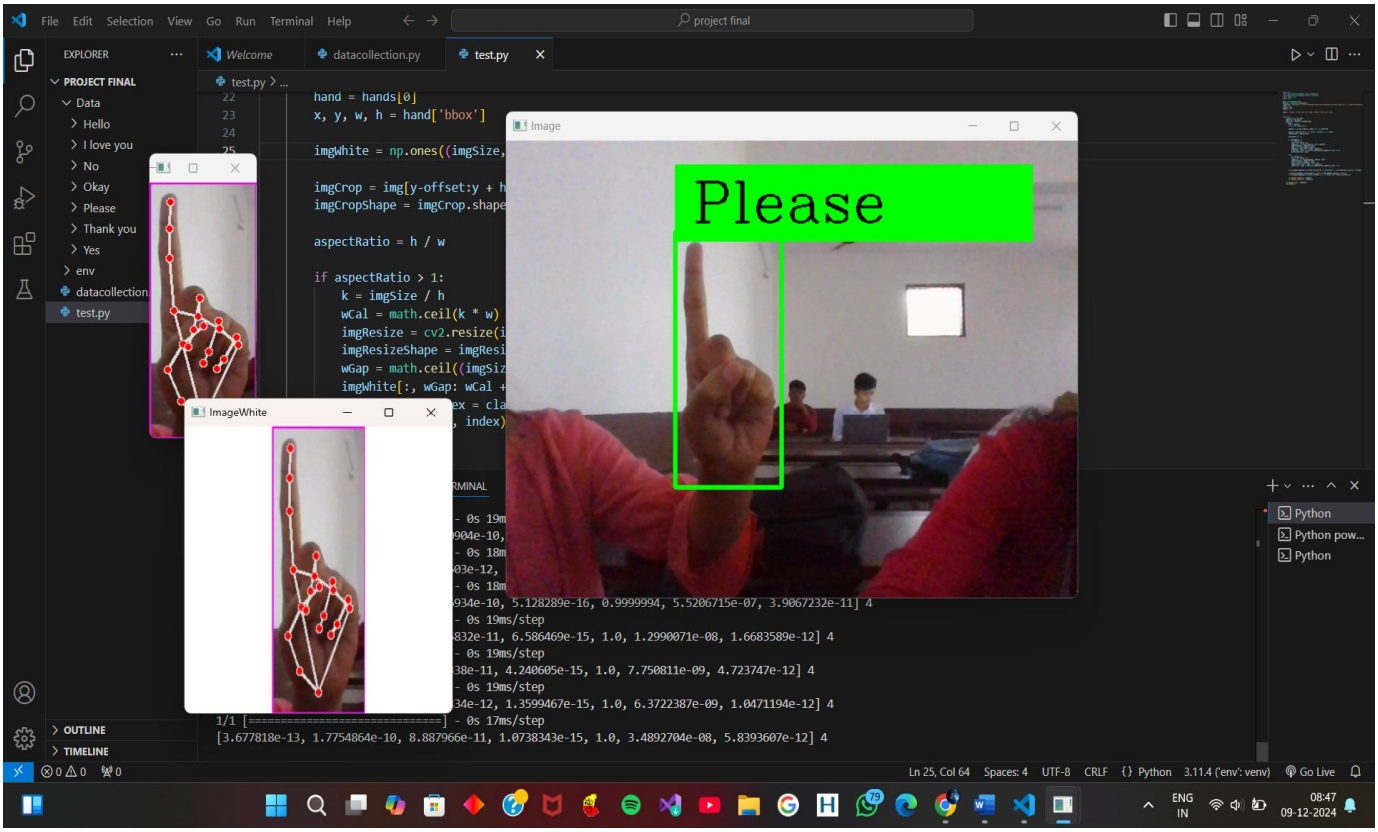


Fig.7 Please

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

V. CONCLUSIONS

The developed sign language recognition system provides an efficient and reliable solution to bridge the communication gap for individuals with hearing or speech impairments. By employing real-time hand tracking, preprocessing, and gesture classification using advanced computer vision and deep learning techniques, the system ensures accurate recognition of predefined gestures despite variations in hand size, orientation, and lighting. The preprocessing methodology standardizes input images to maintain consistency, while the robust classifier enables seamless and precise predictions. The integration of tools like OpenCV, TensorFlow, and CVZone ensures the system's adaptability, scalability, and ease of implementation for real-world applications. The results underscore the system's practicality for assistive communication technologies and its potential to enhance inclusivity. Moreover, this work lays a strong foundation for future improvements, such as expanding the range of recognizable gestures, incorporating voice output, and applying the technology in areas like education, healthcare, and smart devices, thereby contributing to advancements in accessibility and human-computer interaction.

VI. ACKNOWLEDGMENT

We express our sincere gratitude to our guide, Mrs. Pushpa R.N for her suggestions and support during every stage of this work. We also convey our deep sense of gratitude to Dr. Jalesh Kumar, Head of the department. we would like to express our sincere gratitude to all those who have contributed to this paper submission to IJRASET. we would like to thank co-authors for their valuable insights and contributions to this work. we would also like to extend our appreciation to the reviewers and editors for their time and effort in reviewing and improving this manuscript. Finally, we are grateful to my institution for providing me with the necessary resources to carry out this research.

REFERENCES

[1] A. Pathak, A. Kumar, P. Priyam, P. Gupta, and G. Chugh, "Real-Time Sign Language Detection," International Journal for Modern Trends in Science and Technology, vol. 8, no. 01, pp. 32-37, 2022, doi: 10.46501/IJMTST0801006.

[2] Maheshwari Chitampalli, Dnyaneshwari Takalkar, Gaytri Pillai, Pradnya Gaykar, Sanya Khubchandani. Real-Time Sign Language Detection. International Research Journal of Modernization in Engineering Technology and Science, 2023, 5(04): 2983-2986. DOI: 10.56726/IRJMETS36648.

[3] Monisha H. M., Manish B. S., Ranjini Ravi Iyer, Siddarth J. J. Sign Language Detection and Classification Using Hand Tracking and Deep Learning in Real-Time. International Research Journal of Engineering and Technology (IRJET), 2023, 10(11): 875-881. DOI: 10.56726/IRJET2381.

[4] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh, "Sign Language Recognition System using TensorFlow Object Detection API," Preprint, International Conference on Advanced Network Technologies and Intelligent Computing (ANTIC-2021), 2021, doi: 10.1007/978-3-030-96040-7_48.

[5] Dessai, S., & Naik, S. (2022). Literature Review on Indian Sign Language Recognition System. International Research Journal of Engineering and Technology (IRJET), Volume 9, Issue 7. Available at: www.irjet.net.

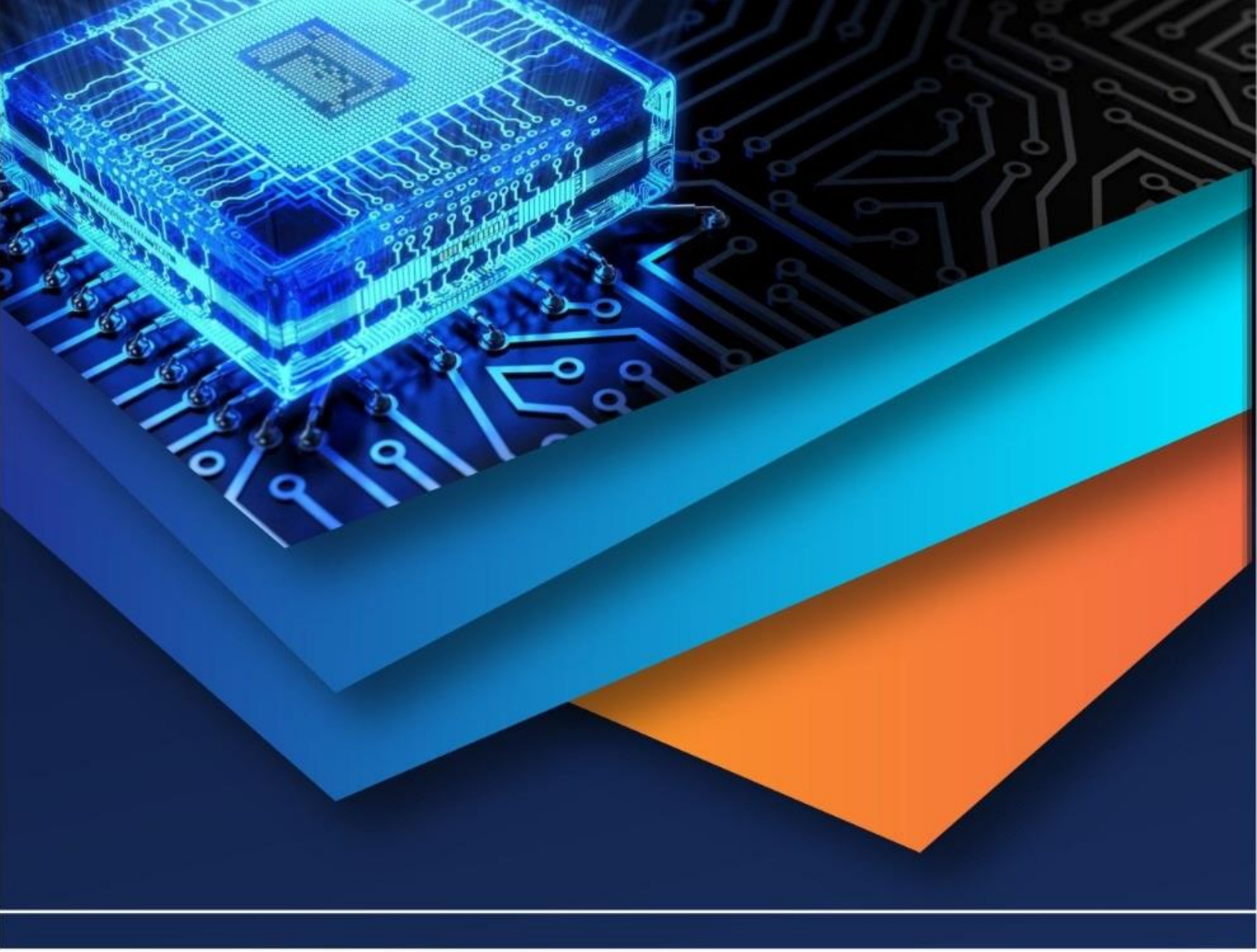
[6] Serai, D., Dokare, I., Salian, S., Ganorkar, P., & Suresh, A. (2017). Proposed System for Sign Language Recognition. 2017 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC). IEEE.

[7] Sreyasi Dutta, Adrija Bose, Sneha Dutta, Kunal Roy. "Sign Language Detection Using Action Recognition with Python." International Journal of Engineering Applied Sciences and Technology, Vol. 8, Issue 01, ISSN No. 2455-2143, Pages 61-67, Published Online May 2023.

[8] Refat Khan Pathan, Munmun Biswas, Suraiya Yasmin, Mayeen Uddin Khandaker, Mohammad Salman, Ahmed A. F. Youssef. "Sign Language Recognition Using the Fusion of Image and Hand Landmarks Through Multi-Headed Convolutional Neural Network." Scientific Reports, Vol. 13, Article 16975, 2023. DOI: 10.1038/s41598-023-43852-x. "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[9] Ashok K. Sahoo, Gouri Sankar Mishra, Kiran Kumar Ravulakollu. "Sign Language Recognition: State of the Art." ARPN Journal of Engineering and Applied Sciences, Vol. 9, No. 2, February 2014.

©IJRASET: All Rights are Reserved



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)