MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

SIGN LANGUAGE TRANSLATOR

Roll No.	Name	SAP ID
B116	Adit Khandelwal	70022200119
B119	Suhani Pugalia	70022200075
B134	Vishwa Bhanushali	70022200409

ABSTRACT

This project will focus on hand gesture recognition in real time using computer vision and machine learning. It captures images of hands by webcam, processes them, isolates the hand region, resizes it, and a CNN model will classify the gestures into predefined categories; thus, the system will interpret hand gestures in real time. These hand gestures are read precisely and continuously; so the corresponding characters that come up on the screen make them helpful for communication aids or control interfaces.

INTRODUCTION

There is a huge communication gap within the sign language community existing between sign language users and nonusers. This project aims to develop a hand sign language translator that can identify individual letters of the alphabet, which will be the first step in further translating sign language into readable text.

Using the OpenCV computer vision library, fully open source and flexible to capture and process hand gestures in real time, our system utilises machine learning models and image-processing techniques to identify particular shapes of hands that represent letters in the American Sign Language alphabet, thus being able to recognize letters letter by letter, then words and phrases.

The technical approach, design of the system, and implementation details based on the image segmentation, feature extraction, and classification technique for hand gesture analysis are provided in this report. It also deals with such challenges as background noise and lighting variation and diversified hand shapes and the measures used to increase the precision of the model and the confidence of the results. Although the system is only able to work with letters, it will eventually build up to a fully functional sign language translation system.

BACKGROUND WORK

Sr. No.	Title	Problem Statement	Approach/Method	Results	Future Work
1	**Vision-Based Hand Posture Detection and Recognition for Sign Language** (Bilal et al., 2011)	Detecting and recognizing static hand postures, crucial for sign language (SL), which face challenges due to occlusions and varying lighting conditions.	Discusses three main approaches: 3D hand modelling, appearance-based methods, and hand shape analysis. Reviews existing techniques for hand detection, including skin-colour methods and marker-based tracking.	Highlights limitations of skin-colour approaches and benefits of combining methods for improved accuracy.	Suggests improving real-time capabilities and robustness under varying conditions.
2	**Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and SVM Techniques** (Dardas & Georganas, 2011)	Challenges in real-time gesture detection due to cluttered backgrounds and variations in skin colour.	Utilises a combination of skin detection, face subtraction, contour comparison, SIFT feature extraction, bag-of-features representation, and multiclass SVM for recognition.	Achieved high recognition rates and real-time performance, proving effective for multiple gestures.	Future work could focus on enhancing scalability and robustness under more diverse lighting conditions.
3	**Real-Time Finger Tracking and Contour Detection for**	Improving the interaction between humans and computers using hand gestures for	Implements hand detection with AdaBoost using Haar-like features, HSV model for segmentation, and	Achieved 92% accuracy with Convex Hull and 70% with AdaBoost;	Recommended exploring integration with more gestures and

	Gesture Recognition Using OpenCV (Gurav & Kadbe, 2015)	tasks like controlling robots or devices.	Convex Hull algorithm for contour and fingertip detection.	provided real-time detection.	improvements for complex backgrounds.
4	**Large Lexicon Detection of Sign Language** (Cooper & Bowden, 2007)	Developing a scalable system for large lexicon sign language recognition without the need for hand tracking.	Uses a non-tracking approach with viseme classifiers and Markov chains for word-level recognition. Skin segmentation and boosting for viseme detection are employed.	Achieved classification rates up to 74.3% for 164-sign vocabulary, comparable to tracking-based methods.	Future work includes enhancing robustness and expanding to larger vocabularies with fewer training samples.
5	**Recognition of Sign Language Using Image Processing** (Arora & Roy, 2018)	Communication barriers for the deaf and mute, as most people are not fluent in sign language.	Image processing with OpenCV: Captures hand gestures, converts images to HSV color space, calculates histograms, and uses Bhattacharyya Distance Metric to match gestures to a stored ASL alphabet database.	Accurately recognizes ASL letters with constraints (e.g., static gestures, black background, no accessories).	Improve flexibility with edge detection and content-based image retrieval to handle more backgrounds and gestures.
6	**Sign Language Translator Application Using OpenCV** (Triyono et al., 2018)	Limited accessibility of sign language recognition tools and challenges in various lighting and background settings.	Android-based app with OpenCV: Uses fingertip coordinates for open gestures and Hu Moments for closed gestures. Images are converted to binary after detecting hand and background color, followed by feature extraction.	Achieves 95% accuracy for static hand gestures under controlled indoor lighting and plain backgrounds; poor performance outdoors or in crowded settings.	Develop dynamic sign language recognition using motion templates in OpenCV to handle complex gestures and different lighting.

7	**Sign Language Recognition Using Image-Based Hand Gesture Recognition Techniques** (Nikam & Ambekar, 2016)	Communication challenges for the deaf-mute community and limited flexibility in existing sign language systems	Image-based gesture recognition with OpenCV: Converts images to HSV, uses convex hull and contour detection for hand segmentation, and extracts hand features (finger positioning) for gesture identification.	Efficiently recognizes hand gestures for both English and Marathi alphabets, leveraging contour and convex hull algorithms.	Expand to dynamic gestures and incorporate voice output for real-time translation to assist the speech-impaired .
8	**Hand Sign Recognition using CNN**	Develop a model to recognize hand gestures and signs for sign language conversion to aid communication for individuals who are deaf or have speech disabilities.	Used Convolutional Neural Network (CNN) with Python and OpenCV for training a model to recognize gestures. Background subtraction and HSV color scheme were applied for segmentation.	The model successfully recognized American Sign Language gestures and displayed corresponding text, achieving at least 75% accuracy.	Improve recognition in non-plain backgrounds, integrate real-time input enhancements, and increase dataset diversity for better accuracy.
9	**Hand Gesture Recognition using OpenCV**	This study addresses the communication challenges for deaf and hard-of-hearing individuals, aiming to create a system that translates Indian Sign Language (ISL) gestures into text for real-time communication with non-signing individuals.	The paper addresses the communication barriers faced by individuals who are deaf or hard of hearing, particularly in India where a significant number of people suffer from auditory impairments. These individuals often use sign language to communicate, but most of the general population is not familiar with it, leading to a reliance on	The model achieved: • 48% accuracy with 50 images, • 53% with 100 images, • 72.5% with 200 images, • 86.4% with 300	Plans include extending to other sign languages (like ASL), training on larger datasets, and enhancing expression recognition for richer communication.

			interpreters which can be expensive and inconvenient. The study aims to develop a sign language detection system that can bridge this gap by recognizing hand gestures and translating them into text for better communication.	images, showing improved accuracy with larger datasets.	
10	**Sign Language Translator Application Using OpenCV**	Develop a mobile app that translates static sign language gestures into text, aiming to bridge communication gaps for the deaf.	The app uses OpenCV to detect hand gestures by distinguishing fingertip coordinates or Hu Moments. It then employs SVM to classify gestures into alphabetical letters.	The fingertip method achieved 95% accuracy indoors, while Hu Moments reached only 40%. Outdoor performance was limited due to background complexity.	Future improvements include adding dynamic gesture recognition using Motion Templates to enhance translation accuracy in varied environments.

PROBLEM STATEMENT

This can be applied to human-computer interaction, assistive technology and sign language. While the system is also designed to be quick, precise and reliable, it is very challenging to achieve these metrics in the presence of varying light conditions, hand-orientation position, and also small variances between gestures. The aim of this project is to provide a solution to the requirement net result of such a system to recognize and interpret hand gestures via machine learning model and a camera feed in real time to allow touchless communication and control.

PROPOSED SYSTEM

This is an individual letter-based sign language translator system for the American Sign Language (ASL) alphabet. In this section, we will describe and outline the main components as well as functions of the proposed system on how hand signs are read, classified, and eventually output in terms of text.

1. System Architecture

The basic architecture of the system mainly involves these four core modules responsible for carrying out some particular functionalities for effective recognition of the gesture.

- Input Module (Image Capture): The input module employs a camera that captures video frames in real-time of the hand of the user. This module can be seen as the first step that will feed the visual input for further processing.
- Preprocessing Module: The images captured in this stage are prepared for analysis. Some of the key preprocessing steps include resizing, converting the images to grayscale, and background subtraction that minimises noise and maximises the accuracy of the feature extraction process.
- Feature Extraction: After the data preprocessing, the system discovers some specific features that constitute a hand gesture contour, edges, and even features which are crucial for differentiation from another letter gesture. They form a base for correct classification.
- Classification Model: Now a machine learning model, labelled with images of the ASL letters is being trained, and upon being classified by the process,

extracts the features by the process, and after having been matched with one among known ASL hand shapes, guarantees correct recognition.

- Output Module: The final module makes the recognized letter appears in text on the screen so that users can read real-time each letter that is spelled out to eventually form words or phrases

2. Modules and Techniques

Each of the modules uses specific techniques and algorithms towards achieving accurate gesture recognition;

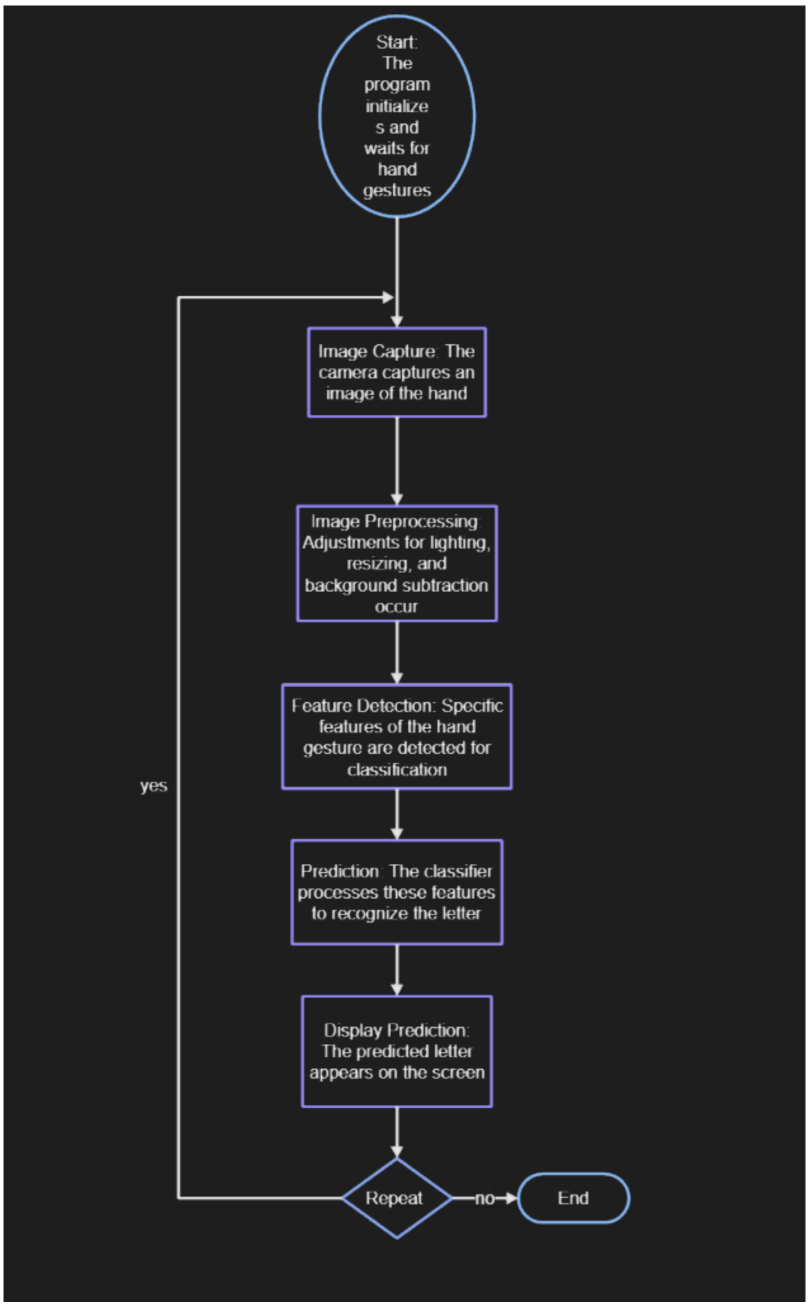
- Image Preprocessing: Techniques which include conversion to grayscale with the use of Gaussian blur enhance the reduction of the background noise through thresholding while ensuring good hand segmentation.
- Feature extraction: It uses contour or edge detection to find shape features of hands. Since all the letter gestures differ in some respect and characteristics, it supports effective classification.
- Classification Model: A CNN, or other machine learning classification model, is trained on diverse ASL letter data with the goal of letter-based recognition using the features seen. This model can help distinguish between very similar, slightly different gestures, giving a consistent letter recognition model.

3. System Requirements

The system is quite lightweight and requires minimal computational hardware and software resources in that:

- Hardware: webcam standard.
- Software: OpenCV library for processing images, TensorFlow is a machine learning library used for classification model; display module where text should be output.

- Performance Requirements:Real-time recognition at low latency allows for immediate feedback while in use.



WORKING

Data Collection:

- A webcam is used to capture hand images in real time. A cvzone library uses a HandDetector for single-hand detection in every frame.
- When a hand is detected, the coordinates of a bounding box around the hand are computed adding extra offset for optimal cropping.
- The cropped hand image was resized and centred onto a 300x300 white background image with uniform input for the classifier.

Gesture Classification:

- It sends the processed hand image to a pre-trained CNN model trained on labelled gesture images and saved as keras_model.h5.
- Every tracked movement is related to some symbol (for example "A," "B," "C", and further). The result of such a model is prediction according to the hand movement within a certain frame.

Consistency Check and Text Recognition:

- For accuracy, it checks if the same gesture is kept for some specified amount of time. It checks if the gesture has stayed the same for some time of 3 seconds; if so, then the gesture is recorded and added to the text display as a valid character.
- This accepted text shows it on the screen that makes it easier for users to see their translated gestures in real time.

Display Output:

- Camera feeds come with a gesture recognition overlay such that the user can see his hand gesture and its interpretation by the system.

Exit Mechanism:

- The system captures images until the user presses the ESC key, at which point resources are released, and the windows are closed.

RESULTS AND OUTCOME

The hand sign language translator effectively demonstrated its ability to recognize individual ASL letters, providing real-time feedback and translating gestures into text.

1. Recognition Accuracy

The translator successfully recognized a broad range of ASL letters. Clear, well-defined gestures were consistently identified with minimal errors, especially for letters with distinct hand shapes. Some letters with similar shapes posed slight challenges, but overall recognition was satisfactory for most users.

2. Speed and Response Time

The system gave instant feedback, and the users were able to report a fluid interaction as the translator showed the recognized letters almost right after every gesture. This prompt response led to an intuitive interaction that was natural and even suitable for spelling words letter by letter.

3. Performance in Different Environments

The system was tested in different environments to check its ability to adapt:

<u>Lighting:</u> The performance of the translator was maximum in moderate lighting where shapes of hands are clearly visible. In extremely bright and dim lighting, the performance is slightly variant, suggesting that consistent lighting is key to good results.

Complexity of background: In simple backgrounds where the hand can be easily isolated from the background, the system works well. In the case of busy or cluttered backgrounds, it became more difficult to achieve an accurate recognition; however, adjustments in preprocessing helped enhance the performance.

4. User Experience and Usability

The overall general good experience was testified to by user feedback with regard to the translator's use. The system of real-time text enabled fluid spelling of words; most found it intuitive. Some gestures have to be practised to use correctly, especially if the shape is complex enough like letters.

5. Observed Limitations

Some limitations came out during the testing:

- Light Sensitivity with complex background: Changed at times recognition conditions so need to have stable recognition.
- Limited recognition: Translator might recognize a letter and at times one would write an entire word letter by letter so it takes more time on making larger words or even phrases.

CONCLUSION & FUTURE SCOPE

The computer vision, machine learning and also real time hand gesture recognition system proposed is really implemented. Using a CNN model and a consistency check, it is now accurately recognizing predefined hand gestures as the characters in the video stream. The system could be useful in assistive technology as a communication supplement for individuals that are speech or hearing impaired. The potential near futures of the project range from gesture vocabulary expansion, advancing the model to yield results at higher accuracy levels given variations in gesture conditions, to expanding its capability and modality in scope.

REFERENCES

1. Bilal, Sara & Akmeliawati, Rini & Salami, Momoh & Shafie, A.A.. (2011). Vision-based hand posture detection and recognition for Sign Language — A study. 10.1109/ICOM.2011.5937178.
2. R. M. Gurav and P. K. Kadbe, "Real time finger tracking and contour detection for gesture recognition using OpenCV," 2015 International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, 2015, pp. 974-977, doi: 10.1109/IIC.2015.7150886.
3. Triyono, Liliek & Pratisto, Eko & Bawono, S & Aji, Fendi & Yudhanto, Yudho & Raharjo, Bangun. (2018). Sign Language Translator Application Using OpenCV. IOP Conference Series: Materials Science and Engineering. 333. 012109. 10.1088/1757-899X/333/1/012109.
4. A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786.