**Sign Language Recognition Under Varying Lighting**

**Conditions: A Comparative Analysis of KNN, Random Forest, and SVM**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Udit Yadav EN21CS301821**

**Vanshika Maheshwari EN21CS301838**

**Varad Apte EN21CS301839**

Under the Guidance of

**Prof. Priyanka Dhasal**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDICAPS UNIVERSITY, INDORE- 453331**

**JAN - JUNE 2025**

**Sign Language Recognition Under Varying Lighting**

**Conditions: A Comparative Analysis of KNN, Random Forest, and SVM**

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING**

BY

**Udit Yadav EN21CS301821**

**Vanshika Maheshwari EN21CS301838**

**Varad Apte EN21CS301839**

Under the Guidance of

**Prof. Priyanka Dhasal**



**Department of Computer Science & Engineering**

**Faculty of Engineering**

**MEDICAPS UNIVERSITY, INDORE- 453331**

**JAN - JUNE 2025**

## Report Approval

The project work **“Sign Language Recognition Under Varying Lighting Conditions”** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner Name:

Designation

Affiliation

External Examiner Name:

Designation

Affiliation

## Declaration

We hereby declare that the project entitled “**Sign Language Recognition Under Varying Lighting Conditions”** submitted n partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science completed under the supervision of **Priyanka Dhasal, Professor in Computer Science, Faculty** of Engineering, Medicaps University Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Signature and name of the student(s) with date**

### Certificate

I Priyanka Dhasal certify that the project entitled **“Sign Language Recognition Under Varying Lighting Conditions”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Udit Yadav, Vanshika Maheshwari, Varad Apte** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Prof. Priyanka Dhasal  Computer Science and Engineering  Medicaps University, Indore | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  <Name of External Guide (If any)>  <Name of the Department>  Name of the Organization |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Ratnesh Litoriya

Head of the Department

Computer Science & Engineering

Medicaps University, Indore

## Acknowledgement

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) D. K. Patnaik,** Vice Chancellor, Medicaps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.)Pramod S. Nair,** Dean, Faculty of Engineering, Medicaps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Ratnesh Litoriya** for his continuous encouragement for the betterment of the project.

I express my heartfelt gratitude to my **Internal Guide**, **Prof. Priyanka Dhasal**, **Professor** Department of Computer Science, MU, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

**<Name of the Student(s)>**

B.Tech. IV Year

Department of Computer Science & Engineering

Faculty of Engineering

Medicaps University, Indore

## Abstract

Sign language recognition (SLR) helps bridge the gap in communication between hearing individuals and members of the deaf or hard-of-hearing community. Nonetheless, lighting conditions can drastically affect the accuracy of recognition. This paper compares three machine learning models—K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM)— for sign language recognition with varying lighting conditions. Experimental tests prove the strength and performance differences of these models, showing their strengths and weaknesses. Given good lighting conditions and a plain background, the overall accuracy results show that KNN shows the highest overall accuracy in all lighting conditions, with up to 95% in sunlight and 89.16% in darkness.

## Keywords

* Real Time gesture recognition
* MediaPipe
* Sign language recognition
* Traditional machine learning
* Computer Vision

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
|  |  |  |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Acknowledgement | v |
|  | Abstract | vi |
|  | Table of Contents | vii |
|  | List of figures | viii |
|  | List of tables | ix |
|  | Abbreviations | x |
|  | Notations & Symbols | xi |
| Chapter 1 | Introduction (Whichever is applicable) |  |
|  | 1.1 Introduction | 1 |
|  | 1.2 Literature Review | 1,2 |
|  | 1.3 Objectives | 2,3 |
|  | 1.4 Significance | 3, |
|  | 1.5 Research Design | 4-5 |
|  | 1.6 Source of Data | 5-6 |
|  | 1.7 Chapter Scheme | 7 |
| Chapter 2 | 2.1 Experimental Set-up | 8-9 |
|  | 2.2 Procedures Adopted | 9 |
|  | 2.3 Data Collection and Preprocessing |  |
| Chapter 3 | Training and Testing |  |
|  | 3.1 Training Phase | 10-11 |
|  | 3.2 Testing Phase | 11-14 |
| Chapter 4 | Result and Discussion | 15,16,17 |
| Chapter 5 | Summary and Conclusion | 18 |
| Chapter 6 | Future scope | 19 |
|  | Bibliography | 20,21 |

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** | **Title** |  | **Page No.** |
| Fig. 1 | ASL Alphabet Signs (Set 1) | 5 |  |
| Fig. 2 | ASL Alphabet Signs (Set 2) | 6 |  |
| Fig. 3 | ASL Alphabet Signs (Set 3) | 6 |  |
| Fig. 4 | ASL Alphabet Signs (Set 4) | 6 |  |
| Fig. 5 | ASL Sign for Alphabet A - KNN Model | 12 |  |
| Fig. 6 | ASL Sign for Alphabet B - KNN Model | 13 |  |
| Fig. 8 | ASL Sign for Alphabet C – SVM Model | 13 |  |
| Fig. 9 | ASL Sign for Alphabet D – RANDOM FOREST Model | 14 |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| Table 1 | Recognition Accuracy under Dark Lighting Conditions | 15 |
| Table 2 | Recognition Accuracy under Sunlight Conditions | 15 |
| Table 3 | Recognition Accuracy under Moderate Lighting Conditions | 16 |
| Table 4 | Overall Model Accuracy Comparison across Lighting Conditions | 16 |

**Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| SLR | Sign Language Recognition |
| ASL | American Sign Language |
| KNN | K-Nearest Neighbors |
| RF | Random Forest |
| SVM | Support Vector Machine |
| ML | Machine Learning |
| AI | Artificial Intelligence |
| GUI | Graphics User Interface |

**Notations & Symbols**

|  |  |
| --- | --- |
| **Symbol** | **Description** |
| **X\_train** | Training feature set (hand landmark coordinates used for training the model) |
| **X\_test** | Testing feature set (hand landmark coordinates used for evaluating the model) |
| **y\_train** | Training labels (corresponding alphabet labels for training data) |
| **y\_test** | Testing labels (corresponding alphabet labels for testing data) |
| **y\_pred** | Predicted labels generated by the model during testing |
| **accuracy** | Model performance metric, calculated as the percentage of correct predictions |
| **DATA\_DIR** | Directory path where the ASL dataset images are stored |
| **data** | List containing processed landmark features extracted from images |
| **labels** | List containing corresponding alphabet labels for each data sample |

**Chapter-1**

#### 1.1 Introduction

Human interaction depends heavily on effective communication. For people who are hard of hearing or deaf, though, communication is not always easy, as most people do not know sign language. American Sign Language (ASL) acts as a bridge between the hearing-impaired community and the rest of society, allowing a form of non-verbal communication. Despite its importance, sign language is not widely known among hearing individuals, often leading to communication challenges.This research seeks to analyze the performance of KNN, RF, and SVM classifiers in recognizing ASL in normal, dim, and bright light. Through their performance in these different settings, this research will establish the most resilient model for practical use. The results will help in creating more dependable SLR systems that operate suitably in diverse environments.

#### 1.2 Literature Review

Sign language recognition (SLR) plays a crucial role in bridging the communication gap between the hearing-impaired and hearing communities. With recent advances in machine learning and computer vision, various methods have been explored to improve recognition accuracy. However, most existing systems assume ideal environmental conditions, while real-world scenarios often include challenges such as varying lighting. This review examines existing studies that focus on SLR techniques, the use of tools like MediaPipe, and the impact of external factors such as lighting.

Pravin and Patil [1] developed a sign language training tool using machine learning techniques to help users learn static hand signs effectively. Their work primarily focused on model training for predefined signs but did not extensively address challenges caused by environmental variability such as low light or glare. Similarly, Krishnan [2] conducted a systematic literature review analyzing the robustness of SLR systems in low-light intensity environments. The study highlighted that while several models perform well in bright conditions, their effectiveness drops significantly under dim lighting, signaling the need for more resilient recognition systems.

Several studies have focused on optimizing machine learning models for alphabet sign recognition. Iqbal and Wahyudi [3] utilized the K-Nearest Neighbor (KNN) algorithm for alphabet sign language recognition and emphasized the importance of hyperparameter tuning to improve classification accuracy. Their findings support the use of traditional machine learning methods in low-resource scenarios. Fahmi and Saputra [8] compared KNN and Random Forest algorithms for recognizing Indonesian Sign Language in real time, concluding that KNN provided higher accuracy, but Random Forest offered faster prediction times, making model choice application specific.

Meanwhile, the use of MediaPipe, an efficient landmark detection framework, has become increasingly popular for hand gesture recognition. Kulkarni et al. [4] and Sharma and Rathi [6] employed MediaPipe for feature extraction in their SLR systems. Both studies demonstrated that MediaPipe significantly reduces preprocessing complexity and enhances system robustness. Their results, however, primarily considered uniform lighting, leaving gaps in understanding its performance under diverse environmental settings.

Real-time sign language recognition has also been explored extensively. Khan and Alzubaidi [5] proposed a real-time SLR system using computer vision and AI techniques, achieving promising results in controlled settings. However, scalability to outdoor or poorly lit environments was not fully validated. Similarly, Kumbhar and Patil [9] used OpenCV for gesture recognition and focused on basic gesture detection but highlighted issues of reduced detection rates in non-ideal lighting conditions.

In summary, while a variety of machine learning and computer vision-based approaches for SLR have been proposed, most studies focus on controlled environments with minimal variability in lighting. Very few have systematically evaluated the performance of different models under varied illumination conditions. Addressing this gap, the present research comparatively analyzes KNN, Random Forest, and SVM models for alphabet recognition across dark, moderate, and bright lighting environments, aiming to identify the most robust approach for real-world applications.

### 1.3 Objectives

The primary objectives of this research are:

1. **To develop a robust Sign Language Recognition (SLR) system** capable of recognizing static American Sign Language (ASL) gestures under diverse lighting conditions such as dim, moderate, and bright (sunlight) environments.

1. **To compare the performance of traditional machine learning classifiers** — KNearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) — for gesture recognition tasks based on extracted hand landmark features.

1. **To analyze the impact of environmental lighting** on the accuracy and reliability of different machine learning models when recognizing hand signs.
2. **To identify the most suitable machine learning model** for real-time sign language recognition applications without requiring complex hardware setups or extensive image preprocessing.

1. **To contribute towards enhancing communication technologies** for the deaf and hard-of-hearing community by proposing a system that performs efficiently across realworld scenarios.

#### 1.4 Significance of the Study

This research holds substantial significance in advancing the field of assistive technologies and human-computer interaction. The ability to accurately recognize sign language gestures under varying lighting conditions is critical for creating reliable, real-time communication tools for the deaf and hard-of-hearing communities.

By evaluating the performance of multiple Traditional machine learning models (KNN, Random Forest, and SVM), this study not only identifies the most robust algorithm for sign recognition but also provides insights into model behavior across different environmental settings. The findings contribute towards developing more adaptable and accessible SLR systems that can function effectively without dependence on controlled environments.

Furthermore, the use of lightweight tools like MediaPipe and traditional classifiers ensures that the system remains computationally efficient, making it practical for real-world applications such as mobile apps, embedded devices, and public service systems. This study ultimately aims to bridge the communication gap and foster inclusivity through affordable and reliable technological solutions.

#### 1.5 Research Design

The research design adopted for this study is **experimental and comparative** in nature. It involves the systematic collection of hand gesture images under varying lighting conditions, followed by feature extraction, model training, evaluation, and comparative analysis.

The design phases are as follows:

1. **Dataset Collection** 
   1. A dataset comprising approximately 8000 images of static American Sign Language (ASL) alphabets (A–L) was gathered.
   2. Images were captured under three different lighting conditions: **dim**, **moderate**, and **sunlight-exposed** environments.
2. **Data Preprocessing and Feature Extraction** 
   1. MediaPipe was used to extract 21 hand landmarks per hand (x and y coordinates), resulting in 84 features per sample.
   2. This approach ensured consistent and lightweight feature vectors for model training.
3. **Data Splitting** 
   1. The dataset was randomly divided into 80% training data and 20% testing data to maintain an unbiased evaluation setup.
4. **Model Implementation** 
   1. Three machine learning classifiers — K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) — were trained on the extracted features.
5. **Testing Procedure** 
   1. Real-time testing was conducted where participants performed static ASL gestures within a predefined camera frame.
   2. Accuracy and confidence scores were calculated separately for right and left hands, then averaged for final evaluation.
6. **Performance Analysis** 
   1. The models' performances were compared based on their accuracy under different lighting conditions.
   2. Detailed result tables and graphs were used to highlight strengths and weaknesses of each classifier.
7. **Conclusion Drawing** 
   1. Based on the results, the most robust model for real-time sign recognition was identified.
   2. Practical recommendations were made for future implementation in real-world applications.

#### 1.6 Source of Data

The primary dataset used in this research was downloaded from **Kaggle**, an online platform widely recognized for hosting machine learning datasets and competitions. The dataset is titled **"ASL (American Sign Language) Alphabet Dataset"** and contains approximately **8000 images** representing static hand gestures for alphabets **A to L**.

For this project, a subset of **1000 images** were randomly selected from the Dataset, ensuring a balanced representation of different lighting conditions.

The dataset includes images captured under three distinct lighting conditions:

* **Dim lighting** (low-light settings)
* **Moderate lighting** (standard indoor room lighting)
* **Bright lighting** (direct sunlight exposure)

In addition to the Kaggle dataset, **real-time testing data** was collected by recording three participants performing the ASL alphabet gestures using a standard webcam. This allowed evaluation of the trained models under practical, uncontrolled conditions involving different hand shapes and minor positional adjustments.

All images were preprocessed using **MediaPipe**, extracting **21 landmark points per hand** (x and y coordinates), resulting in **84 features per sample**.

**Dataset Source:**

* Kaggle Platform — *ASL (American Sign Language) Alphabet Dataset*



Fig. 1. ASL alphabet signs



Fig. 2. ASL alphabet signs

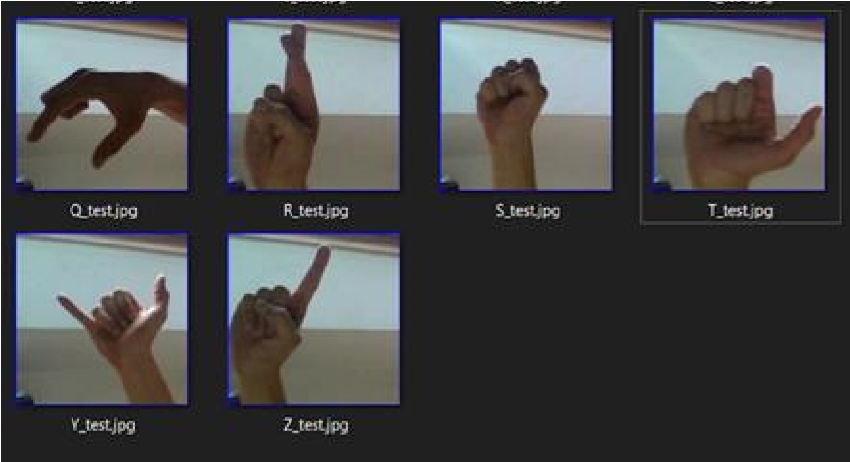


Fig. 3. ASL alphabet signs

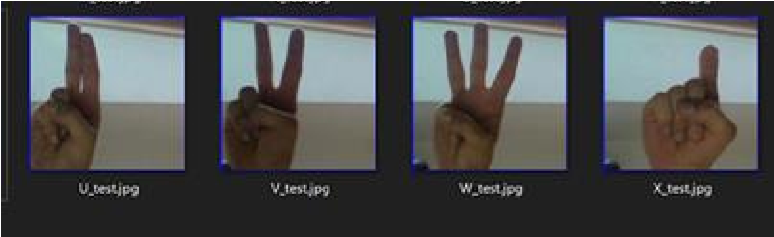


Fig. 4. ASL alphabet signs

### 1.7 Chapter Scheme

**Chapter 1: Introduction**

This chapter provides an overview of the project, highlighting the background, motivation, and relevance of Sign Language Recognition. It includes a brief literature review, outlines the specific objectives of the research, discusses the significance of the study, presents the research design adopted, mentions the data source, and describes the structure of the report.

**Chapter 2: Report on Present Investigation**

This chapter details the experimental setup and the systematic procedures followed during the study. It explains the methods adopted for data collection and preprocessing, including the extraction of hand landmarks using MediaPipe.

**Chapter 3: Training and Testing**

This chapter describes the training phase where traditional machine learning models—Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—were developed. It also covers the testing phase, including real-time gesture recognition experiments conducted under varying lighting conditions and the evaluation methodology.

**Chapter 4: Result and Discussion**

This chapter presents the experimental results obtained from model testing. It provides an analytical discussion comparing the performance of different models under different lighting conditions, based on accuracy and confidence scores.

**Chapter 5: Summary and Conclusion**

This chapter summarizes the overall project work, key findings, and achievements. It highlights how the objectives were fulfilled and presents the final conclusions derived from the research.

**Chapter 6: Future Scope**

This chapter discusses the potential areas for future improvement and expansion of the project. It suggests enhancements such as increasing the gesture vocabulary, handling dynamic gestures, improving robustness under different environments, and optimizing the system for real-time applications on mobile or embedded devices.

# Chapter 2:

**Report on Present Investigation**

#### 2.1 Experimental Setup

The experimental setup for this study was designed to evaluate the performance of machine learning models for American Sign Language (ASL) recognition under different lighting conditions.

* **Hardware Used:** 
  + Laptop with Intel i5 Processor, 8GB RAM o Integrated Webcam (for real-time testing)
* **Software Tools:** 
  + Python 3.9 o OpenCV for image handling o MediaPipe for hand landmark detection o Scikit-learn for machine learning model development o Pickle to save the data set and models
* **Dataset:** 
  + Downloaded from **Kaggle** — *ASL (American Sign Language) Alphabet Dataset* o About 1000 images, covering static gestures from A to L under varying lighting conditions.
* **Preprocessing:** 
  + Hand landmarks (21 points per hand) extracted using **MediaPipe**.
  + Each point's x and y coordinates were used, resulting in **84 features** per sample.
* **Train-Test Split:** 
  + 80% of the data was used for training. o 20% of the data was reserved for testing model performance.

#### 2.2 Procedures Adopted

The project followed a structured procedure as described below:

1. **Data Collection:** 
   1. Downloaded Kaggle dataset of ASL alphabets (A–L) images.
   2. Real-time hand gesture images captured from three different participants.
2. **Feature Extraction:** 
   1. MediaPipe was employed to detect 21 hand landmarks from images and videos.
   2. Extracted x and y coordinates of landmarks and concatenated them to form the input features.
3. **Model Selection:** 
   1. Three machine learning models were selected for comparative study:
      1. K-Nearest Neighbors (KNN)
      2. Random Forest (RF)
      3. Support Vector Machine (SVM)
4. **Model Training:** 
   1. Models were trained on the extracted landmark features.
5. **Testing Procedure:** 
   1. During testing, participants positioned their hand inside a predefined virtual box.
   2. Real-time recognition was performed, and results were collected across three lighting conditions: dark, moderate, and bright.
6. **Accuracy Measurement:** 
   1. Accuracy was computed separately for right and left hands.
   2. Final accuracy was the average of right-hand and left-hand accuracies.

#### 2.3 Data Collection and Preprocessing

**Data Collection:**

* The dataset used was downloaded from **Kaggle**, consisting of static hand gesture images representing ASL alphabets **A to L**.
* From the complete dataset of approximately **8000 images**, a subset of **1000 images** was randomly selected, ensuring coverage across different lighting conditions

**Preprocessing:**

* Each image was processed using **MediaPipe** to extract **21 hand landmarks per hand**, resulting in **42 features** (x and y coordinates) for a single hand or **84 features** when considering both hands.
* The extracted coordinates were **normalized**:
* After normalization:
  + If the number of features was **less than 84**, zero-padding was applied.
  + If the number of features was **greater than 84**, extra values were truncated.
* The final normalized feature vectors and their corresponding alphabet labels were stored using the pickle module to create a structured dataset for machine learning model training.

# Chapter 3:

#### 3.1 Training Phase

1. **Random Forest**
   * The dataset was loaded from a pickle file, containing feature vectors and labels.
   * Data was converted into NumPy arrays for easier handling and processing.
   * The dataset was split into training and testing sets using an 80:20 ratio with stratified sampling to maintain label distribution.
   * A RandomForestClassifier from sklearn.ensemble was used to build the classification model.
   * The model was trained (fitted) using the training data (x\_train, y\_train).
   * After training, the model was evaluated by predicting on the test set and calculating the accuracy using accuracy score.
   * The model achieved a high accuracy percentage, showing good performance.
   * Finally, the trained model was saved using Python’s pickle module for future use without retraining.
2. **SVM** 
   * The dataset was loaded from a pickle file, containing feature vectors and labels.
   * Data was converted into NumPy arrays for easier handling and processing.
   * The labels were encoded into numerical values using LabelEncoder for compatibility with the model.
   * The features were normalized using StandardScaler to bring all values into a similar range.
   * The dataset was split into training and testing sets using a 60:40 ratio.
   * An SVC (Support Vector Classifier) from sklearn.svm was used to build the classification model.
   * The model was trained (fitted) using the training data (X\_train, y\_train).
   * After training, the model was evaluated by predicting on the test set and calculating the accuracy using accuracy\_score.
   * Finally, the trained model, scaler, and label encoder were saved using joblib for future use without retraining.
3. **KNN** 
   * The dataset was loaded from a pickle file, containing feature vectors and corresponding labels.
   * Feature data and labels were extracted into NumPy arrays for efficient data handling and processing.
   * The labels were encoded into numerical form using LabelEncoder to make them compatible with the machine learning model.
   * The dataset was split into training and testing sets with an 80:20 ratio using train\_test\_split.
   * A KNeighborsClassifier (KNN) from sklearn.neighbors was selected for building the classification model.
   * The model was configured with 5 neighbors and 'euclidean' distance as the metric for finding nearest neighbors.
   * The model was trained (fitted) using the training data (X\_train, y\_train).
   * After training, the model was evaluated by predicting on the test set and calculating the accuracy using accuracy\_score.
   * The KNN model achieved a good accuracy score, indicating reliable recognition performance.
   * Finally, the trained KNN model and label encoder were saved using pickle for future use without retraining.

#### 3.2 Testing Phase

After the completion of the training phase, real-time testing was conducted to evaluate the effectiveness and robustness of the trained models. For each model — **Random Forest**, **KNearest Neighbors (KNN)**, and **Support Vector Machine (SVM)** — a **separate GUI** was designed to ensure uniformity in testing conditions and to allow real-time gesture recognition and feedback.

The system displayed a predefined frame on the screen, guiding users to properly position their hands for accurate detection. A gesture was considered **successfully recognized** if the corresponding output appeared with a **confidence score greater than 0.5**.

**Testing Procedure:**

* **Participants:** Three individuals were selected to perform the static hand gestures.
* **Gestures:** Each participant performed hand signs for alphabets **A to L** separately using their **right hand** and **left hand**.
* **Trials and Averaging:** 
  + For each alphabet, all three participants attempted the gesture individually. o The system captured the recognition result (confidence score and predicted letter) for each attempt.
  + The **average confidence score** across the three participants was calculated separately for the right hand and the left hand.
  + This average was considered the **final recorded value** for that particular alphabet under the given lighting condition.

**Lighting Conditions:**

* **Dark Room:** Low ambient lighting environment to test recognition in challenging conditions.
* **Moderate Lighting:** Normal indoor lighting simulating everyday usage.
* **Under Sunlight:** Bright natural light to simulate outdoor scenarios.

**Real-Time Output:**

* The GUI displayed the recognized alphabet, and its associated confidence score live on the screen.
* The system updated the predictions instantly based on the hand gesture placed within the camera frame.
* Users were instructed to keep their hands static once correctly positioned to allow stable and accurate recognition.

By evaluating each model across multiple users and different lighting conditions, the testing phase provided a comprehensive understanding of the system’s reliability, model performance, and practical usability.



Fig. 5. Asl sign for Alphabet A for KNN Model in Moderate lighting

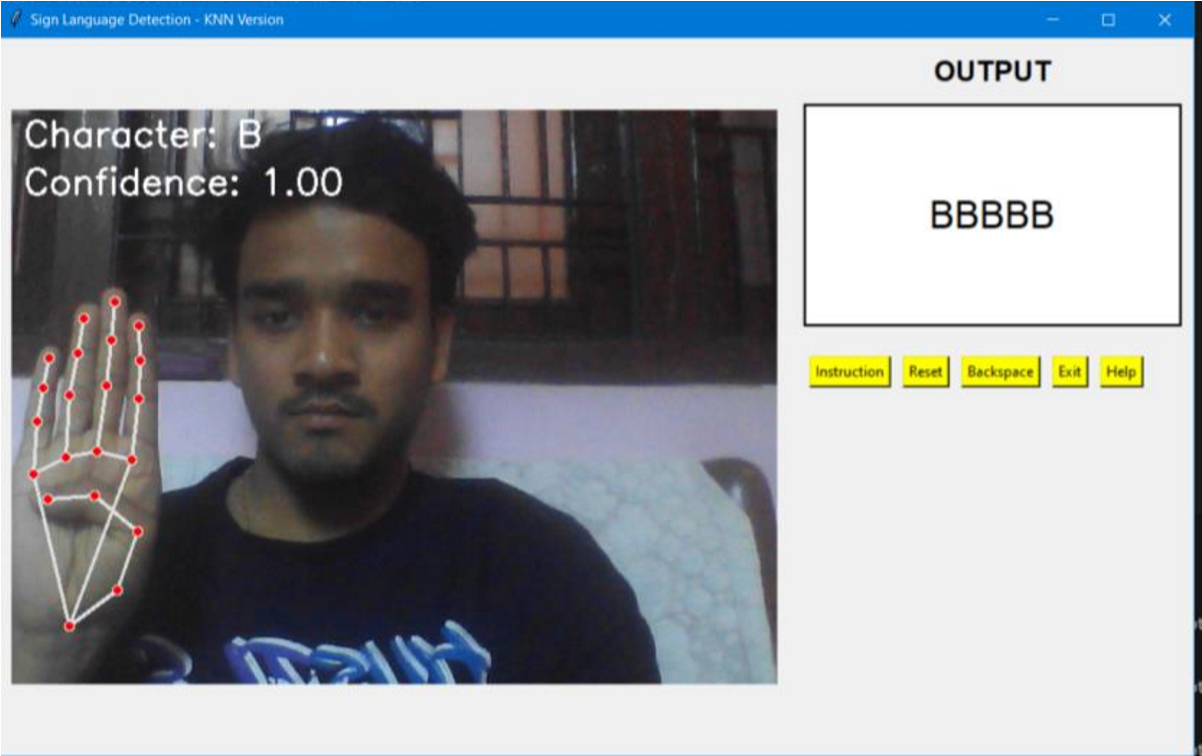


Fig. 6. Asl sign for Alphabet B for KNN Model in Moderate lighting

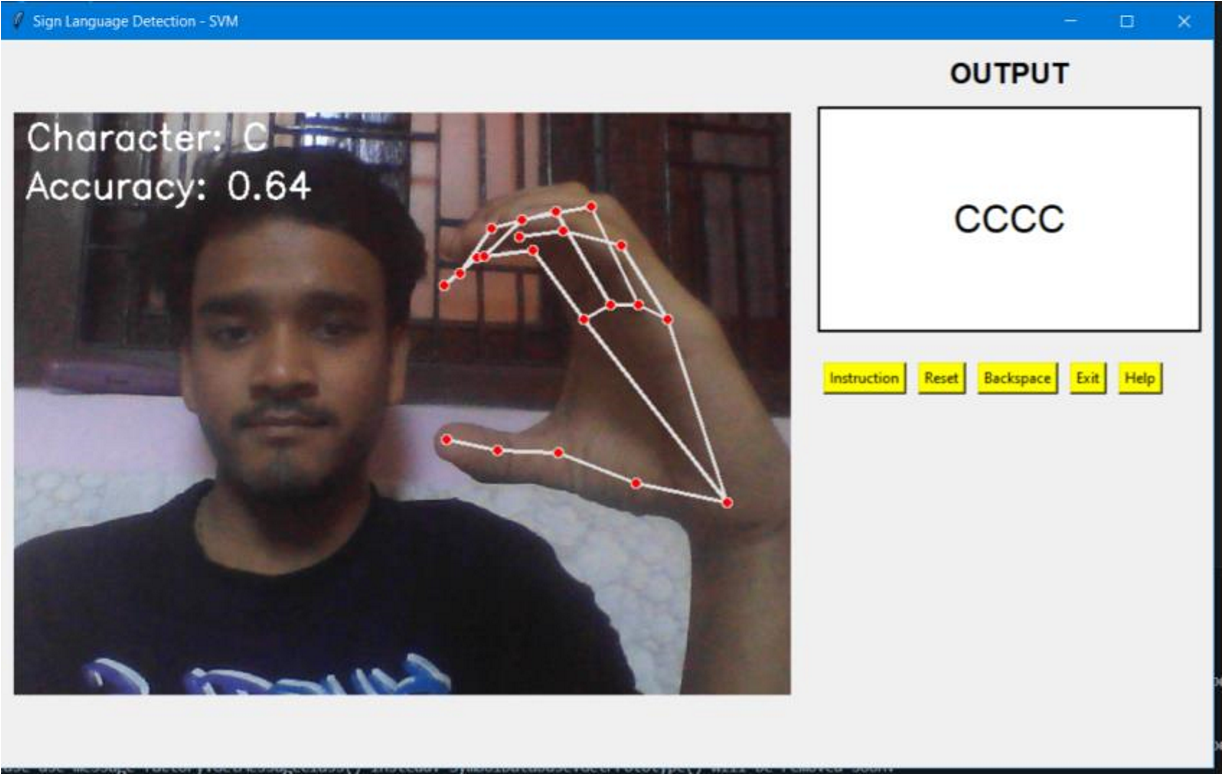


Fig. 7. Asl sign for Alphabet C for SVM Model in Moderate lighting



Fig 8. Asl sign for Alphabet D for Random Forest Model in Moderate lighting

# Chapter 4:

### RESULTS AND DISCUSSION

The results show performance metrics for three machine learning models (Random Forest, KNN, and SVM) tested on American Sign Language alphabet recognition under three different lighting conditions: dark room, moderate lighting, and under sunlight.

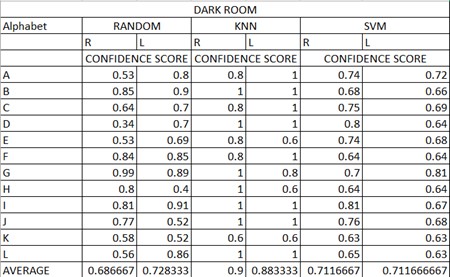


Table 1. Result table under Dark condition

The KNN model showed the highest average confidence with a score of 0.9000 for the right hand and 0.8833 for the left hand, demonstrating high consistency and reliability in detecting signs under dark conditions. Compared to that, the SVM model had moderate performance at an average of 0.7117 for both the right and left hand. Random Forest performed the lowest with an average confidence score of 0.6867 for the right hand and 0.7283 for the left hand.

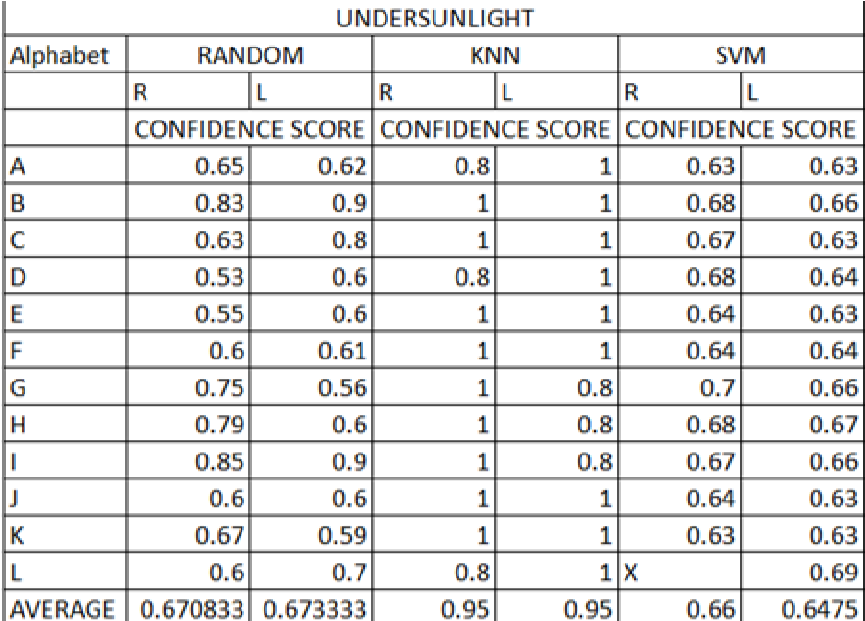


Table 2. Result table under sunlight condition.

The KNN classifier demonstrated the highest robustness to lighting, achieving an average confidence score of 0.95 for both hands. In contrast, the Random Forest model showed moderate recognition performance, with average scores of 0.6708 (R) and 0.6733 (L). The SVM model had slightly lower averages of 0.66 (R) and 0.6475 (L). Notably, the SVM model failed to recognize the gesture corresponding to the letter L (left hand), marked as ‘X’ in the table, indicating gesture not-recognise

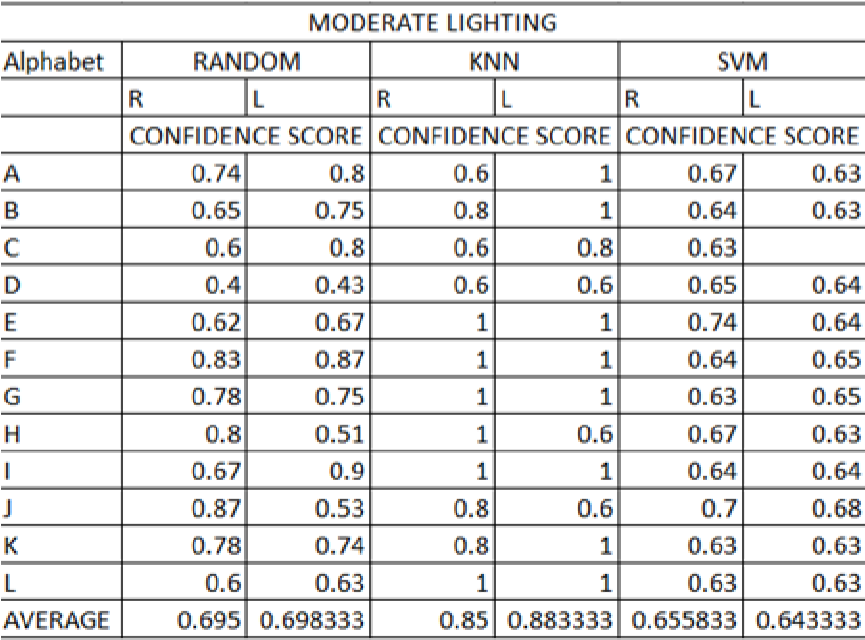


Table 1. Result table under Moderate condition

The KNN classifier once more performed better than the others with average confidence values of 0.85 (R) and 0.8833 (L), portraying stable and consistent recognition under this lighting configuration. Random Forest had moderate performance with values of 0.695 (R) and 0.6983 (L), depicting acceptable but weaker recognition strength than that of KNN. SVM, although operational, had the lowest averages: 0.6558 (R) and 0.6433 (L), reflecting weaker robustness under such illumination.

V. OVERALL MODEL ACCURACY

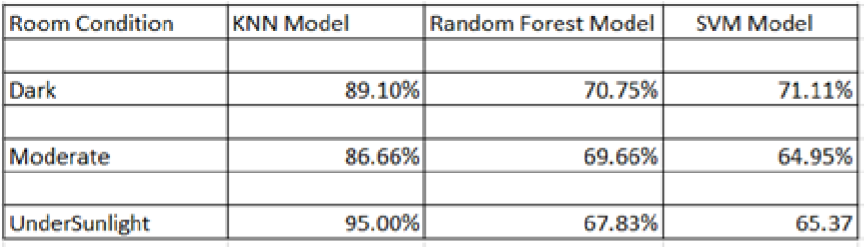


Table 4. Result table under Moderate condition

In UnderSunlight, the KNN model achieved the highest performance with an accuracy of

95.00%, followed by SVM at 65.37% and Random Forest at 67.83%.

Under Dark conditions, KNN again led with 89.10%, whereas SVM (71.11%) slightly outperformed Random Forest (70.75%).

In Moderate lighting, KNN showed robust accuracy of 86.66%, while Random Forest and SVM recorded 69.66% and 64.95%, respectively.

# Chapter 5:

### SUMMARY AND CONCLUSION

This study aimed to develop a system for recognizing static alphabet gestures from sign language inputs. The system was tested with varying conditions of environmental light—dim, moderate, and under sunlight—simulating consistency and performance in practical scenarios. For training and testing, a combined total of 1,000 images of each alphabet gesture was utilized across broad variations in order to imitate real-time applications.

According to the findings, KNN performed better than both SVM and Random Forest models under the three lighting environments with higher stability and responsiveness towards environmental changes. The best performance was recorded during sunlight with an accuracy of 95.00%, while the worst was achieved during moderate illumination for SVM with an accuracy of 64.95%. From these findings, it is inferred that KNN is the best model among those evaluated for use in real-time recognition of static hand signs in the absence of aids or exterior enhancement

# Chapter 6:

### FUTURE SCOPE

The current system successfully recognizes static ASL alphabet gestures (A–L) using traditional machine learning models under varying lighting conditions. However, several areas for future enhancement and expansion have been identified:

* **Expansion of Gesture Vocabulary:**

Future work can include recognizing the complete ASL alphabet (A–Z), numbers, and commonly used static words or phrases to broaden the system's applicability.

* **Incorporating Dynamic Gestures:**

The system can be extended to support **dynamic gestures** (e.g., full sentences, actions) by integrating sequence-based models like **LSTM** or **GRU**, allowing real-time continuous sign language recognition.

* **Improved Lighting Adaptability:**

Future improvements could involve adaptive preprocessing techniques to automatically adjust for extreme lighting variations (e.g., shadow removal, brightness normalization) without user intervention.

* **Deployment on Mobile and Edge Devices:**

By optimizing model size and computational requirements, the system can be made lightweight enough to run on **mobile devices** or **embedded systems**, enhancing accessibility for real-world use.

* **Deep Learning Integration:**

Incorporating deep learning-based models like **Convolutional Neural Networks (CNNs)** or **Transformer architectures** can further improve recognition accuracy and robustness, especially in uncontrolled environments.

* **User-Centric Enhancements:**

Adding features such as voice output for recognized gestures, multi-language support, and interactive feedback could make the system more user-friendly, especially for educational or assistive technology applications.

* **Multi-Hand and Multi-Person Detection:**

Extending the system to recognize **gestures from both hands simultaneously** or **gestures performed by multiple users** could improve its application scope in collaborative environments.

### BIBLIOGRAPHY

1. MediaPipe Hands Solution, Google Developers, [Online].
2. Available: <https://google.github.io/mediapipe/solutions/hands.html>
3. OpenCV Library Documentation, [Online].
4. Available: <https://docs.opencv.org/>
5. Python Software Foundation, Python Language Reference, version 3, [Online]. Available: <https://www.python.org/doc/>
6. Scikit-learn: Machine Learning in Python, Pedregosa et al., Journal of Machine Learning Research 12, pp. 2825–2830, 2011. [Online].

Available: <https://scikit-learn.org/stable/>

1. Pillow (PIL Fork) - Python Imaging Library Documentation, [Online]. Available: <https://pillow.readthedocs.io/en/stable/>
2. Joblib Documentation – Efficient Python Serialization, [Online]. Available: <https://joblib.readthedocs.io/en/latest/>
3. Tkinter Documentation – Python Standard Library, [Online]. Available: <https://docs.python.org/3/library/tkinter.html>
4. A. S. Pravin and R. Patil, “Sign Language Training Tool Using Machine Learning Techniques,” *International Journal of Research Publication and Reviews*, vol. 4, no. 6, pp. 4811–4814, 2023. [Online]. Available:<https://ijrpr.com/uploads/V4ISSUE6/IJRPR14556.pdf>
5. [2] L. Krishnan, “A Systematic Literature Review on the Robustness of Sign Language Recognition in Low Light Intensity Environments,” *TechRxiv*, 2023. [Online].

Available: [https://www.techrxiv.org/users/712252/articles/696680/master/file/data/wma180001%2 0Journal/wma180001%20Journal.pdf](https://www.techrxiv.org/users/712252/articles/696680/master/file/data/wma180001%20Journal/wma180001%20Journal.pdf)

1. [3] M. Iqbal and A. Wahyudi, “Alphabet Sign Language Recognition Using K-Nearest Neighbor Optimization,” in *Proc. Int. Conf. Intelligent Systems and Computing (ICISC)*, 2020. [Online]. Available: [https://www.researchgate.net/publication/338578140\_Alphabet\_Sign\_Language\_Recog nition\_Using\_K-Nearest\_Neighbor\_Optimization](https://www.researchgate.net/publication/338578140_Alphabet_Sign_Language_Recognition_Using_K-Nearest_Neighbor_Optimization)
2. [4] S. G. Kulkarni, V. A. Deshmukh, and R. V. Padole, “Gesture-Based Sign Language Recognition System Using MediaPipe,” 2023. [Online]. Available: [https://www.researchgate.net/publication/371937989\_Gesture\_based\_Sign\_Language\_ Recognition\_system\_using\_Mediapipe](https://www.researchgate.net/publication/371937989_Gesture_based_Sign_Language_Recognition_system_using_Mediapipe)
3. [5] M. Z. Khan and H. Alzubaidi, “Real-Time Sign Language Recognition Using Computer Vision and AI,” *Kean University Research*, 2023. [Online]. Available:

[https://researchers.kean.edu/en/publications/real-time-sign-language-recognition-usingcomputer-vision-and-ai](https://researchers.kean.edu/en/publications/real-time-sign-language-recognition-using-computer-vision-and-ai)

1. [6] A. Sharma and V. Rathi, “American Sign Language Recognition for Alphabets Using MediaPipe,” *Procedia Computer Science*, vol. 199, pp. 123–130, 2022. [Online].

Available:<https://www.sciencedirect.com/science/article/pii/S1877050922021378>

1. [7] A. S. Alrashidi et al., “A Survey on Sign Language Literature,” *Journal of King Saud University - Computer and Information Sciences*, 2023. [Online]. Available:<https://www.sciencedirect.com/science/article/pii/S2666827023000579>
2. [8] D. Z. Fahmi and I. R. Saputra, “The Comparison of K-Nearest Neighbors and

Random Forest Algorithm to Recognize Indonesian Sign Language in Real-Time,” *Scientific Journal of Informatics*, vol. 10, no. 1, pp. 96–103, 2023. [Online]. Available:<https://journal.unnes.ac.id/nju/sji/article/download/48475/pdf>

1. [9] R. D. Kumbhar and V. S. Patil, “Gesture Recognition Using OpenCV for Deaf and Dumb,” *International Journal of Novel Research and Development*, vol. 8, no. 5, pp.

594–598, 2023. [Online]. Available:<https://www.ijnrd.org/papers/IJNRD2305632.pdf>