**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**
THE SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
ICT K67

**IT3160E**

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

# SIGN LANGUAGE DETECTION

## PROJECT REPORT

### MEMBER LIST

| | | |
|---|---|---|
| Nguyễn Đình Dương | 20225966 | duong.nd225966@sis.hust.edu.vn |
| Nguyễn Mỹ Duyên | 20225967 | duyen.nm225967@sis.hust.edu.vn |
| Hồ Bảo Thư | 20226003 | duong.nd225966@sis.hust.edu.vn |
| Hà Việt Khánh | 20225979 | khanh.hv225979@sis.hust.edu.vn |
| Nguyễn Trọng Minh Phương | 20225992 | phuong.ntm5992@sis.hust.edu.vn |

**Teacher: Phạm Văn Hải**

**Hà Nội — 2024**

# CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# Abstract

Our team research an innovative gestural communication tool designed to address communication barriers, particularly for individuals with hearing impairments. The tool translates sign language into text or speech in real-time, offering efficient and accurate communication solutions. Incorporating advanced AI models like the Random Forest Classifier and MediaPipe Hands, the system ensures precise hand gesture detection and sign language interpretation. With anticipated accuracy rates of 90% to 95%, the tool sets a new standard for precision in sign language translation. Emphasizing intuitive design principles, the tool prioritizes usability and accessibility, fostering inclusivity and empowerment in communication. The initiative aims to create a more inclusive world by providing individuals with hearing impairments equal access to effective communication tools and opportunities for meaningful engagement in society.

# 1 Problem Detection

According to the World Health Organization (WHO), more than 5% of the global population, totaling approximately 466 million individuals, suffer from disabling hearing loss. Regrettably, only a minority among them possess proficiency in sign language—a visual-based natural language reliant on hand gestures, movements, facial expressions, and body language. Consequently, this situation gives rise to three primary challenges:

1. **Communication Barriers:** Individuals with hearing loss who lack proficiency in sign language often face significant challenges in communicating with others. This can lead to feelings of isolation, exclusion, and frustration in personal and professional contexts. Without effective communication channels, accessing information, expressing oneself, and engaging in social interactions become formidable tasks.
2. **Educational Disparities:** The lack of access to sign language or other communication accommodations in educational settings can result in educational disparities for individuals with hearing loss. This may manifest in difficulties understanding classroom lectures, participating in group discussions, accessing instructional materials, and receiving adequate support from educators. As a consequence, academic performance and educational outcomes may be adversely affected, potentially limiting opportunities for future advancement.
3. **Employment Discrimination:** In the workplace, individuals with hearing loss who do not have proficiency in sign language may encounter discriminatory practices and barriers to employment. Employers may overlook their qualifications or hesitate to hire them due to concerns about communication barriers or the perceived need for accommodations. As a result, individuals with hearing loss may face limited job opportunities, reduced career advancement prospects, and unequal treatment in the workplace, contributing to socioeconomic inequalities.

Addressing these challenges requires concerted efforts to promote accessibility, inclusion, and equal opportunities for individuals with hearing loss. This involves implementing policies and practices that support the use of sign language, providing appropriate accommodations in educational and workplace settings, raising awareness about the needs of individuals with hearing loss, and fostering a culture of inclusivity and acceptance.

# 2 Purpose

The purpose of this initiative is multifaceted, aiming to address the communication challenges faced by individuals with hearing impairments. Through tailored assistance and support mechanisms, our primary objective is to enhance the communicative abilities of those with hearing impairments, ensuring their messages are comprehensively understood by others. By bridging the gap between the communication methods of deaf individuals, whether through speech or writing, and the understanding of their intended audience, we aspire to facilitate seamless interaction in various contexts.

Moreover, this initiative seeks to empower individuals with hearing impairments, enabling them to articulate their thoughts, emotions, and ideas with clarity and precision. By providing the tools and resources necessary for effective expression, we endeavor to amplify their voices within their familial, educational, and societal spheres. Through advocacy and education, we strive to cultivate an inclusive environment where the unique perspectives and experiences of deaf and hearing-impaired individuals are acknowledged, respected, and integrated into broader social discourse.

In essence, the purpose of this initiative extends beyond mere facilitation of communication; it embodies a commitment to promoting equity, dignity, and empowerment for individuals with hearing impairments. By fostering understanding, acceptance, and appreciation of diverse modes of communication, we endeavor to create a more inclusive and accessible society for all.

# 3 Inputs and Outputs

## 3.1 Inputs

In this section, we outline the inputs and outputs of our system. Our inputs primarily consist of video clips and live video feeds containing hand gestures, captured through various devices such as cameras or smartphones. These inputs serve as the raw data that our system processes to extract meaningful information.

Our dataset comprises images of hand signs, publicly available on Kaggle. This dataset includes 26 different characters, with each character accompanied by approximately 3000 images. These characters represent the alphabet from A to Z in sign language. This diverse dataset enables us to train our models effectively, capturing the intricacies and variations present in real-world hand gestures.
Href: asl-alphabet data and Sign-language-gesture-images-dataset

## 3.2 Outputs

The main output of our system is the conversion of these hand gestures into either text or speech. Through sophisticated algorithms and machine learning techniques, we analyze the input videos to recognize and interpret the hand gestures performed. Subsequently, we generate corresponding textual representations or convert them into spoken language, depending on the application's requirements.

By transforming hand gestures into text or speech, our system enables individuals with

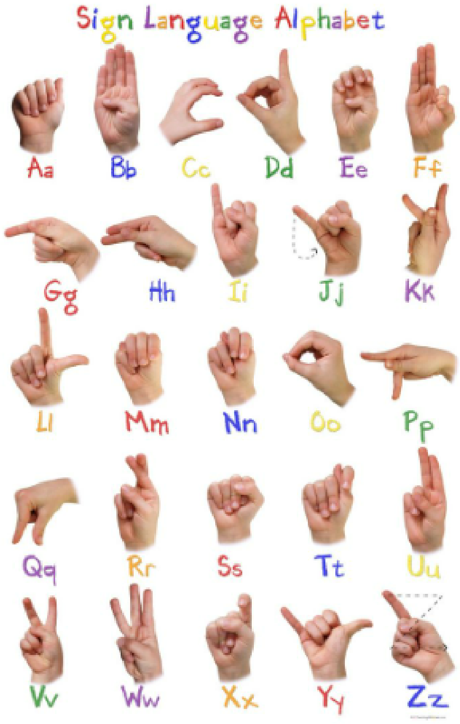

Figure 1: Input Data

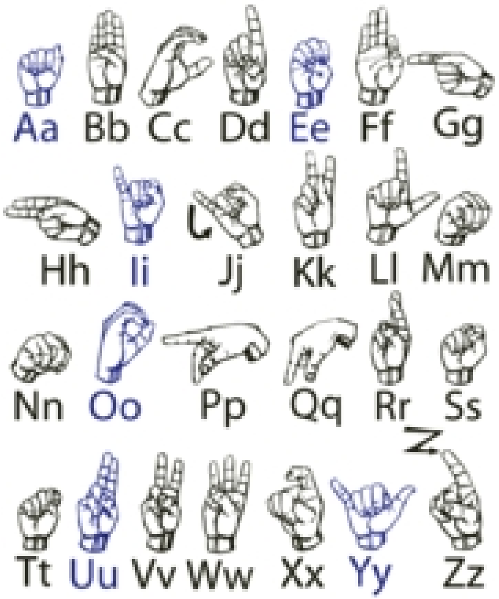

Figure 2: Hand sign language

hearing impairments to access and comprehend information conveyed through visual communication modalities. This facilitates greater inclusivity and accessibility in communication, empowering individuals to participate more fully in various social, educational, and professional settings.

## 4 State of art

Researchers and developers have been actively engaged in creating and refining automatic sign language recognition systems, leveraging cutting-edge techniques in machine learning, computer vision, and natural language processing. These systems hold immense potential for facilitating real-time translation of sign language into spoken or written language, enhancing accessibility in various domains including education, communication, and entertainment. Furthermore, efforts are underway to ensure the accuracy, robustness, and inclusivity of these systems across diverse signing styles and environments. As such, the landscape of sign language recognition technology continues to evolve rapidly, driven by a commitment to accessibility and inclusivity for all. There are many sign language recognition technologies that have made breakthroughs recently.

### 4.1 Natural Language-Assisted Sign Language Recognition (1)

Sign languages are visual languages that convey information through the movements of the hands, body, head, mouth, and eyes, making them completely separate and distinct from natural languages.Due to the inherent constraints imposed by the combinations of visual elements, there is a considerable number of visually indistinguishable signs (VISigns) in sign languages. This characteristic poses limitations on the recognition capabilities of visual neural networks. To address this challenge, we introduce the Natural Language-Assisted Sign Language Recognition (NLA-SLR) framework. This framework leverages the semantic information embedded within glosses (sign labels).

Firstly, when VISigns exhibit similar semantic meanings, we employ a language-aware label smoothing approach. This method involves replacing hard labels with soft ones, akin to the well-established technique. For each training video, we adopt fastText to generate smoothed labels, thereby enhancing the model's ability to discern subtle semantic differences.

Conversely, in cases where VISigns convey distinct semantic meanings, we utilize the inter-modality mixup technique. This innovative approach involves independently integrating the vision feature with each gloss feature to create a blended representation. By seamlessly merging visual and semantic cues, this method enriches the model's understanding of the diverse semantic nuances inherent in sign language gestures.

The datasets had been used including MSASL, WLASL, and NMFs-CSL.In this study, S3D is opted as the backbone of VKNet. This decision is rooted in S3D's exceptional balance between accuracy and processing speed.

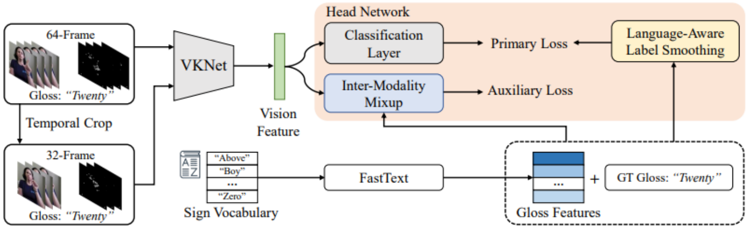

Figure 3: An overview of NLA-SLR 3 (1)

In data pre-processing, we Use HRNet trained on COCO-WholeBody, 63 key points (11 for upper body, 10 for mouth, and 42 for two hands)/frame. Besides, we also introduce a novel backbone, video-keypoint network, which not only models both RGB videos and human body keypoints but also derives knowledge from sign videos of different temporal re-ceptive field.

The network architectures of VKNet-32 and VKNet-64 are identical. Each of them employs a two-stream architecture comprising a video encoder and a keypoint encoder. Specifically, given a training sample belonging to the $b$-th class, label smoothing replaces the one-hot label with a soft label $y \in \mathbb{R}^N$.The head network follows the algorithm:

$$y[i] = \begin{cases} 1 - \epsilon & \text{if } i = b, \\ \epsilon . \dfrac{\exp\left(\frac{s[i]}{\tau}\right)}{\sum_{i=1, i \neq b}^{N} \exp\left(\frac{s[i]}{\tau}\right)} & \text{otherwise.} \end{cases}$$

where $\tau$ denotes a temperature parameter. The classification loss $L_{CLS}$ is a simple cross-entropy loss applied on the prediction and soft label $y$.

The accuracy of this algorithm, using both Lang-LS and sign mixup along with the VKNet can achieve the best performance: 61.05%/91.45% on the top-1 and top-5 accuracy, respectively.

## 4.2 Using reptile search algorithm with hybrid deep learning (2)

One of challenges with SLR comprises the variability in sign language through various cultures and individuals, the difficulty of certain signs, and require for realtime processing. This study introduces an Automated Sign Language Detection and Classification using Reptile Search Algorithm with Hybrid Deep Learning (SLDC-RSAHDL).

The presented SLDC-RSAHDL technique detects and classifies different types of signs using DL and metaheuristic optimizers. In the SLDC-RSAHDL technique, MobileNet feature extractor is utilized to produce feature vectors, and its hyperparameters can be adjusted by manta ray foraging optimization (MRFO) technique. The mathematical model written as:

$$\widehat{G} = \sum_{m}\sum_{i,j} \widehat{K}_{i,j,m}\, F_{k+1-1,l+j-1,m}$$

$$G_{k,l,n} = \sum_{m} \widehat{G}_{k,l,m} \cdot \overline{K}_{m,n}$$

The foraging chain has been developed if manta rays arrange head-to-tail. In each iteration, an optimum solution was utilized for updating every individual. The subsequent mathematical model can demonstrate it:

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r.(x_{best}^d(t) - x_{best}^d(t)) + \alpha.(x_{best}^d(t) - x_i^d(t)) & ,i = 1 \\ x_i^d(t) + r.(x_{i-1}^d(t) - x_i^d(t)) + \alpha.(x_{best}^d(t) - x_i^d(t)) & i = 2, 3, ..., N. \end{cases}$$

For sign language classification, the SLDC-RSAHDL technique applies HDL model, which incorporates the design of Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). At last, the RSA was exploited for the optimal hyperparameter selection of the HDL model, which resulted in an improved detection rate.

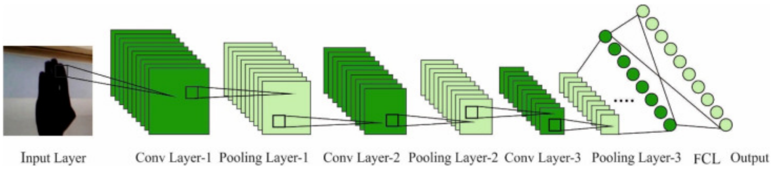

Figure 4: Sign language classification using optimal HDL model(2)

The RSA method creates a fitness function (FF) to make superior classifier result. It explains a positive integer to exemplify the good performance of candidate outcomes. During this effort, the minimizing of classifier error rate was supposed that FF is formulated in:

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{\textit{number of misclassified samples}}{\textit{Total number of samples}} * 100$$

Based on the evaluation metrics, it can be observed that the algorithm achieves a very high performance in classification. With an accuracy rate of 99.51%, along with precision and recall scores of 99.42% and 99.43% respectively, we can conclude that the algorithm performs well in classifying the samples. This is further illustrated by the high F-score value of 99.43%, indicating that the algorithm provides a good balance between precision and recall.

# 5 AI Models and Algorithms

## 5.1 Theoretical basis

### 5.1.1 Random Forest Classifier

Random forests are an Ensemble method, meaning they combine predictions from other models called decision trees. During the training phase, multiple trees are created using different random subsets of the data and features. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. Each decision tree is like an expert, providing its opinion on how to classify the data. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks).

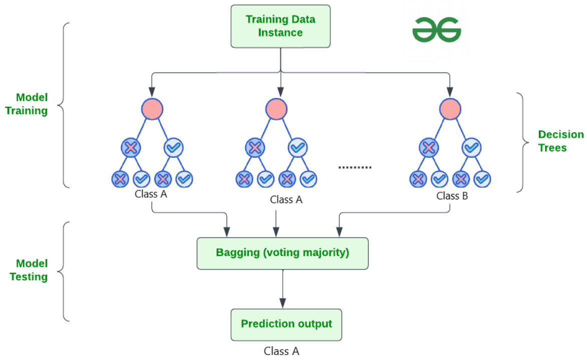

Figure 5: Random forest classifier (3)

The technique of bagging is a cornerstone of Random Forest's training strategy which involves creating multiple bootstrap samples from the original dataset, allowing instances to be sampled with replacement. This results in different subsets of data for each decision tree, introducing variability in the training process and making the model more robust.
Random forests find utility in a variety of industries, from predicting disease in healthcare to analyzing consumer behavior in business, thanks to their ability to manage complexity, reducing excessive input on, and because it provides reliable results.

### 5.1.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of artificial neural network specially designed for processing structured grid data, such as images or videos. It's particularly powerful in tasks involving visual imagery, like image recognition, classification, segmentation, and even generation.

Here's a breakdown of its components:

1. **Convolutional Layers**: These layers apply convolutional filters (also known as kernels) to the input data. These filters detect features in the input, such as edges, textures, or patterns, by sliding over the input data and performing element-wise multiplication and summation.

2. **Pooling Layers**: After convolutional layers, pooling layers are often applied to reduce the spatial dimensions of the feature maps produced by the convolutional layers. This helps in reducing the computational complexity and controlling overfitting.
3. **Activation Functions**: Non-linear activation functions, such as ReLU (Rectified Linear Unit), are applied after convolutional and pooling layers to introduce non-linearity into the network and enable it to learn complex patterns.
4. **Fully Connected Layers**: Following several convolutional and pooling layers, fully connected layers are often used to perform classification based on the features extracted by the earlier layers. These layers connect every neuron in one layer to every neuron in the next layer.
5. **Output Layer**: The output layer produces the final output of the network, which could be class probabilities in classification tasks or pixel values in image generation tasks.

CNNs are trained using labeled data through a process called backpropagation, where the network learns to adjust its parameters (weights and biases) to minimize the difference between its predictions and the actual labels. Training typically involves feeding forward input data through the network, comparing the output to the ground truth labels using a loss function, and then using an optimization algorithm like gradient descent to update the network parameters iteratively.

Overall, CNNs have revolutionized the field of computer vision and are widely used in various applications, including object detection, facial recognition, medical image analysis, and autonomous vehicles, among others.

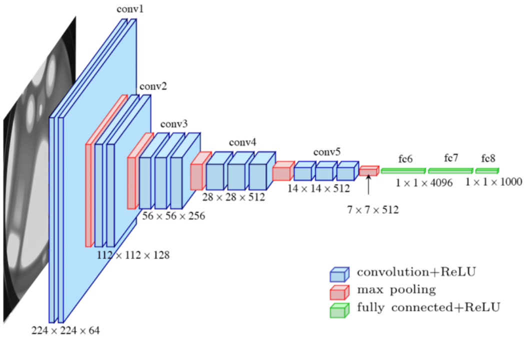

Figure 6: CNN's layers (4)

## 5.2 Models

In this section, we present two distinct AI models/algorithms, denoted as Type 1 and Type 2, each serving specific functions within our system.

### 5.2.1 Random Forest Classifier

Type 1 utilizes a combination of the Random Forest Classifier and MediaPipe Hands AI model, an AI model integrated into the MediaPipe library. (5)
MediaPipe Hands specializes in tracking within video frames with the ability to perceive the shape and motion of hands. It employs machine learning (ML) to infer the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions as below. The model was trained on approximately 30K real-world images, as well as several rendered synthetic hand models imposed over various backgrounds. Since running the palm detection model

is time consuming, when in video or live stream running mode, Hand Landmarker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames. Hand Landmarker only re-triggers the palm detection model if the hand landmarks model no longer identifies the presence of hands or fails to track the hands within the frame. This reduces the number of times Hand Landmarker triggers the palm detection model.

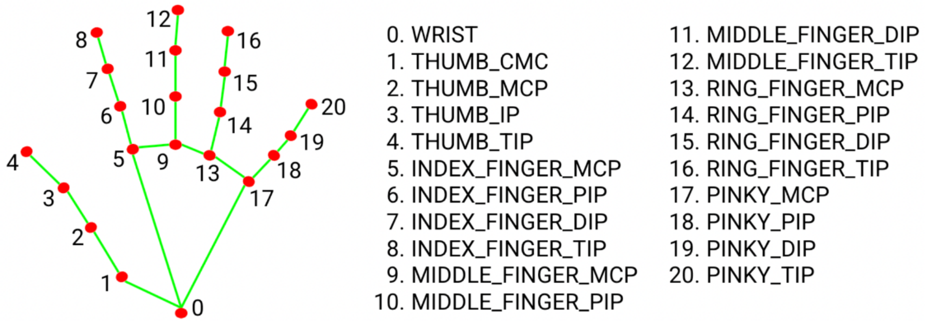

Figure 7: 21 hand landmarks (5)

Finally, the data is extracted and processed by the Random Forest Classifier to generate predictions.

The results clearly indicate that Type 1 excels in processing visual data by accurately detecting hand landmarks. This precision offers a distinct advantage, particularly in tasks requiring detailed analysis of hand motions.

### 5.2.2 VGG16 Convolutional Neural Network (CNN)

Type 2 leverages the capabilities of the VGG16 Convolutional Neural Network (CNN), developed by the research team at the University of Oxford (7).It is characterized by its depth, consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. These layers are organized into blocks, with each block containing multiple convolutional layers followed by a max-pooling layer for downsampling.This architecture include the use of ReLU activation function and the final fully connected layer outputting probabilities for 1000 classes using softmax activation.

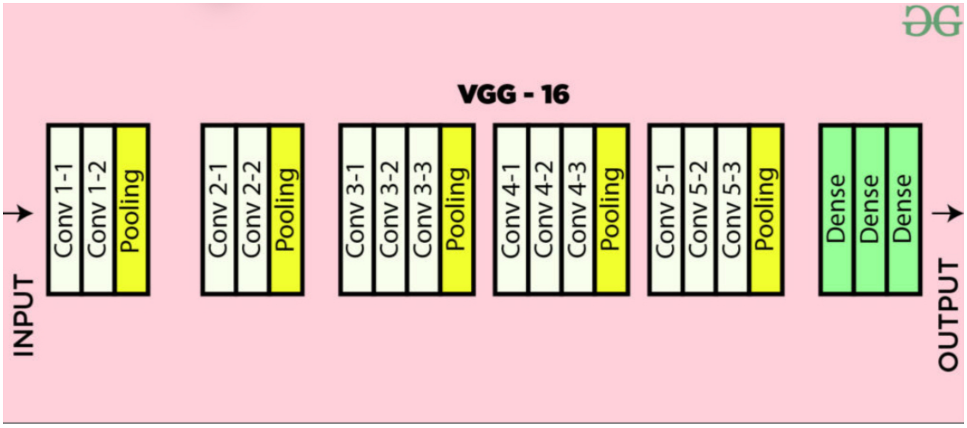

Figure 8: VGG-16 architecture Map (6)

VGG-16 is renowned for its simplicity and effectiveness, as well as its ability to achieve strong performance on various computer vision tasks, including image classification and object recognition. Through its convolutional layers, VGG16 learns hierarchical features, progressing from low-level features like edges and corners to high-level features such as complex shapes and objects. This hierarchical feature extraction enables VGG16 to comprehend the spatial structure of images comprehensively, facilitating accurate classification and analysis within our system.

# 6 Expected Results

Our anticipated outcomes are structured to achieve significant advancements in gestural communication technology:

1. Integration of both image and real-time video data streams to enrich gestural communication capabilities.
2. Substantial enhancement in the speed and efficiency of sign language translation processes.
3. Attainment of a high accuracy level ranging between 80% to 95% in the recognition and interpretation of hand gestures.
4. Development of a user-friendly interface, meticulously crafted with adherence to fundamental design principles, ensuring seamless usability and accessibility for all users.

These expected results underscore our commitment to delivering a sophisticated and effective gestural communication solution that addresses the diverse needs of our users while maintaining high standards of performance and usability.

## 6.1 Experiment

Our experiment focuses on rigorous testing and validation procedures to ensure the robustness and reliability of our gestural communication system. The experiment encompasses two key aspects:

### 6.1.1 Data Pipeline Integrity

The first component of our experiment is dedicated to verifying the integrity and functionality of the data pipeline. Through systematic checks and validation procedures, we meticulously examine each stage of the data pipeline to ensure that data integrity is maintained and that there are no instances of corruption. Upon detecting any anomalies or inconsistencies, corrective measures are promptly implemented to rectify the issues and restore the data pipeline to its intended state. The corrected data is then validated to ensure that it aligns with the expected format and quality standards.

### 6.1.2 Model Testing

The second component of our experiment involves comprehensive testing of our AI models to assess their performance and accuracy. This testing encompasses various aspects, including:

- **Invariance Test:** Evaluating the model's ability to maintain consistent performance across different conditions and environments.

- **Directional Expectation Test:** Assessing the model's capability to accurately interpret and respond to directional gestures.
- **Bias/Fairness Test:** Analyzing the model's behavior to identify and mitigate any biases or unfairness in its decision-making process.
- **Model Output Consistency:** Ensuring that the model consistently produces reliable and coherent output across different inputs and scenarios.

By conducting thorough testing and validation procedures, we aim to ascertain the effectiveness and reliability of our gestural communication system, thereby instilling confidence in its performance and capabilities.

## 6.2 Evaluation Metrics

In evaluating the performance of our gestural communication system, we employ a set of comprehensive metrics to assess its accuracy and effectiveness. These metrics provide valuable insights into the system's performance across various dimensions:

### 6.2.1 Accuracy

Accuracy measures the overall correctness of our system by calculating the proportion of correctly classified instances among all instances. It serves as a fundamental indicator of the system's ability to make accurate predictions and classifications.

### 6.2.2 Precision

Precision assesses the quality of positive predictions made by our system. It calculates the proportion of correctly predicted positive instances among all instances predicted as positive. Precision is crucial for evaluating the reliability and trustworthiness of positive predictions.

### 6.2.3 Recall

Recall measures the system's ability to capture all positive instances within the dataset. It calculates the proportion of correctly predicted positive instances among all actual positive instances. Recall is essential for evaluating the system's completeness and ability to identify all relevant instances.

### 6.2.4 F1-Score

The F1-score provides a balanced assessment of the system's performance by considering both precision and recall. It calculates the harmonic mean of precision and recall, offering a single metric to evaluate the model's performance comprehensively. The F1-score enables us to assess the trade-off between precision and recall, providing valuable insights into the overall effectiveness of our gestural communication system.

By employing these evaluation metrics, we aim to conduct a thorough and rigorous assessment of our system's performance, ensuring its reliability and effectiveness in real-world applications.

# 7 Results

## 7.1 Random Forest Classifier

With all the datasets, we split the train and test sets in a ratio of 80% for training and 20% for testing

### 7.1.1 System Infrastructure

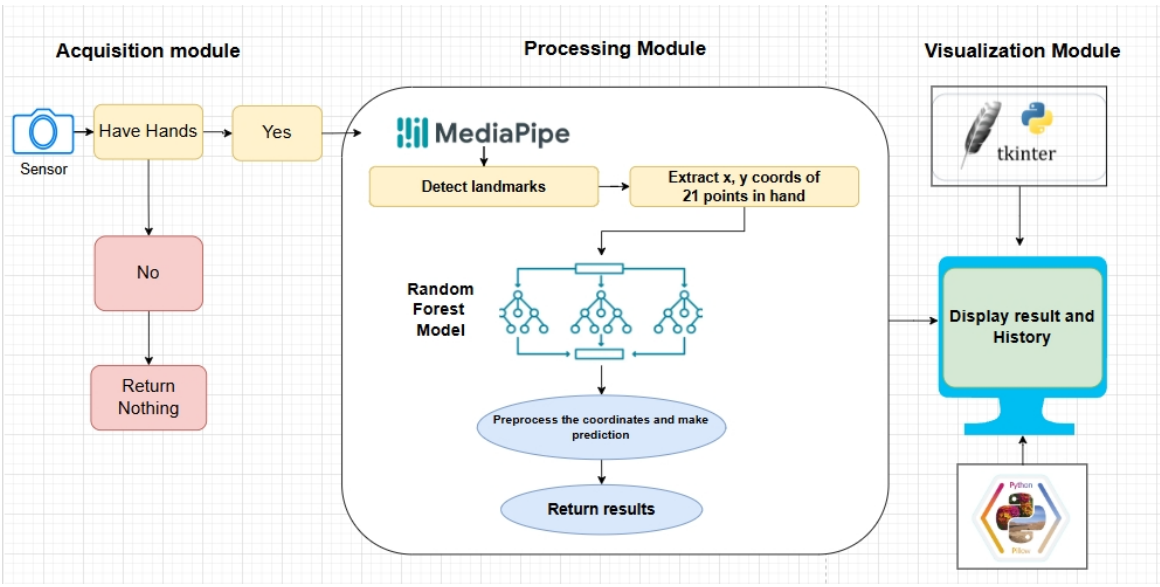

Figure 9: System Insfrastucture of Random Forest Classifier

**Acquisition Module**

- **Sensor (Camera)**: The camera captures video frames, acting as the primary sensor for the system.
- **Have Hands**:
  - This component checks whether there are hands detected in the frame.
  - If hands are detected (`Yes`), the system proceeds to the Processing Module.
  - If no hands are detected (`No`), the system returns nothing and waits for the next frame.

**Processing Module**

- **MediaPipe**:
  - **Detect Landmarks**: Uses MediaPipe to detect the landmarks of the hand. Specifically, it identifies 21 key points on the hand.
  - **Extract x, y coords of 21 points in hand**: Extracts the x and y coordinates of these 21 points for further processing.

- **Random Forest Model**:
  - This machine learning model takes the extracted coordinates as input.
  - **Preprocess the coordinates and make predictions**:
    - Preprocesses the coordinates to fit the input format required by the model.
    - The model then makes a prediction based on the input coordinates.
- **Return Results**: The predicted sign language character is returned as the result.

**Visualization Module Display result and History**:

- **Tkinter (Python GUI Library)**: Uses Tkinter to create a graphical user interface.
- **Pillow (PIL)**: Utilizes Pillow for image processing within the GUI.
- Displays the video feed with the detected hand landmarks and the predicted sign language character.
- Maintains and shows a history of previous predictions for reference.

### 7.1.2 Demo/interface

The following is a demo image that we have deployed, utilizing the rear camera. Afterwards, the system will automatically detect the hand and provide predictions.

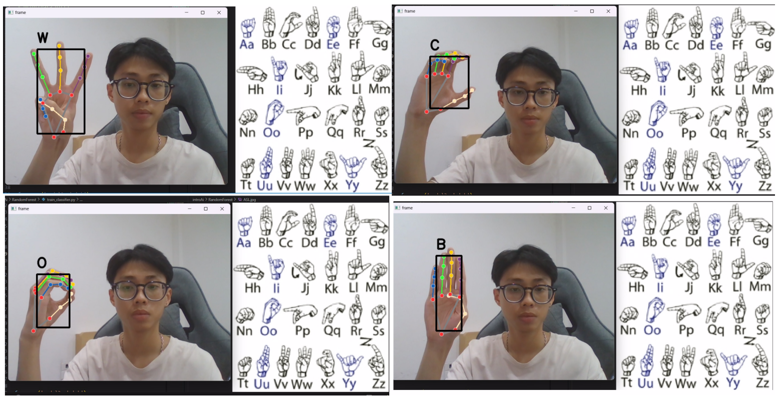

Figure 10: Interface of Random Forest Classifier Model

### 7.1.3 Evalution

**Evalution of First Dataset**

Here is link to our notebook using Random Forest Classifier model After trainning models, we evaluate the models with some aspect:

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| A | 0.97 | 0.99 | 0.98 | 200 |
| B | 1.00 | 1.00 | 1.00 | 200 |
| C | 0.99 | 1.00 | 1.00 | 200 |
| D | 0.99 | 0.97 | 0.98 | 200 |
| E | 1.00 | 1.00 | 1.00 | 200 |
| F | 0.99 | 1.00 | 0.99 | 200 |
| G | 0.97 | 0.99 | 0.98 | 200 |
| H | 0.99 | 0.99 | 0.99 | 200 |
| I | 0.99 | 0.97 | 0.98 | 200 |
| J | 0.99 | 0.97 | 0.98 | 200 |
| K | 1.00 | 0.98 | 0.99 | 200 |
| L | 0.99 | 0.99 | 0.99 | 200 |
| M | 0.92 | 0.93 | 0.93 | 200 |
| N | 0.96 | 0.94 | 0.95 | 200 |
| O | 0.99 | 0.99 | 0.99 | 200 |
| P | 1.00 | 0.98 | 0.99 | 200 |
| Q | 0.98 | 1.00 | 0.99 | 200 |
| R | 0.96 | 0.93 | 0.94 | 200 |
| S | 0.98 | 0.98 | 0.98 | 200 |
| T | 0.99 | 0.98 | 0.99 | 200 |
| U | 0.87 | 0.93 | 0.90 | 200 |
| V | 0.95 | 0.94 | 0.95 | 200 |
| W | 1.00 | 0.98 | 0.99 | 200 |
| X | 0.98 | 0.98 | 0.98 | 200 |
| Y | 1.00 | 1.00 | 1.00 | 200 |
| Z | 0.99 | 0.99 | 0.99 | 200 |
| **Accuracy** | | | 0.98 | 5200 |
| **Macro Avg** | 0.98 | 0.98 | 0.98 | 5200 |
| **Weighted Avg** | 0.98 | 0.98 | 0.98 | 5200 |

Table 1: Classification Report

**Accuracy:** 0.9785
**Support:** 5200

**Summary:**

- **Accuracy:** The overall accuracy of the model is 0.978, indicating that approximately 97.85% of the predictions made by the model are correct.
- **Precision, Recall, and F1-Score:**
  - Precision measures the proportion of true positives out of the total predicted positives.
  - Recall measures the proportion of true positives out of the actual positives.
  - F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both.

**Detailed Analysis:**

- **High Performance:** The model shows high precision, recall, and F1-scores across all classes, with most classes achieving scores close to or at 1.00. This indicates that the model is highly effective in distinguishing between different letters with minimal error.
- **Class-wise Performance:**
  - Classes B, C, E, F, H, K, L, O, P, Q, T, W, X, and Y exhibit perfect precision, recall, and F1-scores (1.00), suggesting these letters are consistently classified without errors.
  - Classes A, D, G, I, J, M, N, R, S, U, and V also show strong performance, with slight variations but still maintaining high precision, recall, and F1-scores.
  - The lowest performing class in terms of recall is 'U' with a recall of 0.93, but its precision remains high at 0.87, indicating some room for improvement in identifying this letter correctly.
  - Classes M, N, R, and U have slightly lower F1-scores compared to other letters, suggesting that while performance is still strong, these classes may have more misclassifications compared to others.

**Conclusion:**

The classification report indicates that the sign-language model is highly accurate and reliable for recognizing the 26 letters of the alphabet. The high precision, recall, and F1-scores across the board demonstrate that the model has been effectively trained and is capable of making precise predictions with minimal errors. The slight variations in performance for certain letters, such as M, N, R, and U, could be areas for further refinement to achieve even higher accuracy. Overall, the model's performance is exemplary and suitable for practical applications in sign language recognition.

## 7.2 Convolutional Neural Network (CNN)

### 7.2.1 System Infrastructure

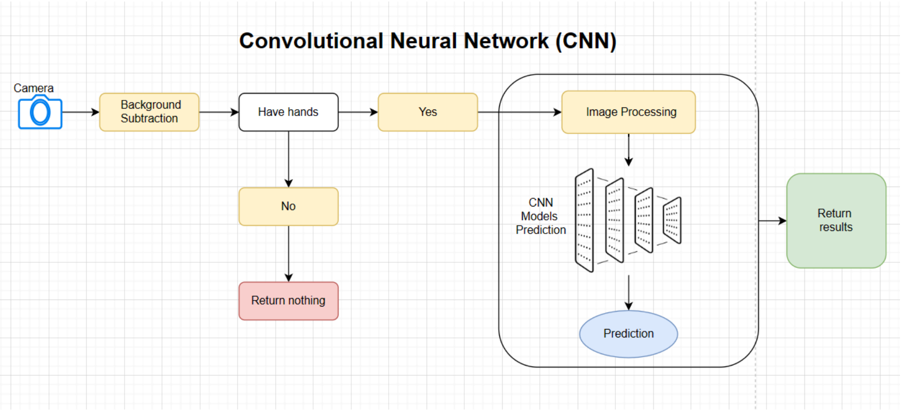

Figure 11: System Infrastructure of Type 2

### 7.2.2 Demo/interface

The following is a demo image that we have deployed, utilizing the rear camera. The system will make a prediction about the character based on the image of the hand placed within the predefined frame on the frame.

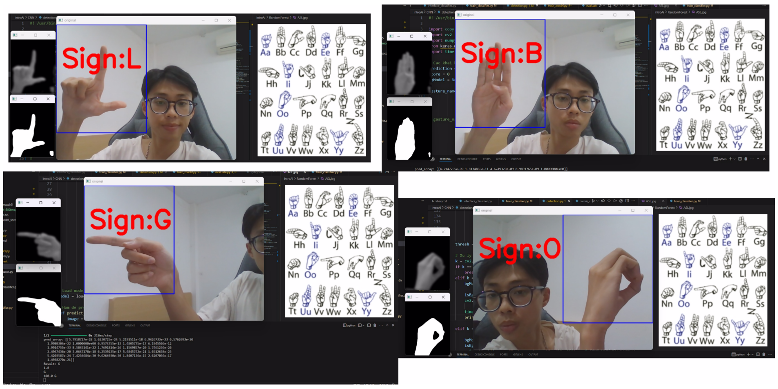

Figure 12: Interface of CNN Model

### 7.2.3 Evalution

**Training**

Here is link to our notebook using VGG16 model

| Epoch | Step | Time | Accuracy | Loss | Validation Acc | Validation Loss |
|---|---|---|---|---|---|---|
| 1 | 325/325 | 58s | 0.1676 | 2.8026 | 0.7058 | 1.0057 |
| 2 | 325/325 | 27s | 0.6766 | 0.9301 | 0.8135 | 0.7095 |
| 3 | 325/325 | 27s | 0.8409 | 0.4691 | 0.9254 | 0.5320 |
| 4 | 325/325 | 27s | 0.9075 | 0.2679 | 0.9127 | 0.7108 |
| 5 | 325/325 | 27s | 0.9113 | 0.2724 | 0.9515 | 0.6055 |
| 6 | 325/325 | 27s | 0.9403 | 0.1831 | 0.9435 | 0.7721 |
| ... | ... | ... | ... | ... | ... | ... |
| 24 | 325/325 | 27s | 0.9822 | 0.0591 | 0.9704 | 1.1321 |
| 25 | 325/325 | 27s | 0.9907 | 0.0367 | 0.9465 | 1.2900 |
| 26 | 325/325 | 27s | 0.9814 | 0.0697 | 0.9546 | 1.2333 |
| 27 | 325/325 | 27s | 0.9771 | 0.0760 | 0.9608 | 1.0604 |

Table 2: Training Progress

**Classification Report**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| A | 0.99 | 0.95 | 0.97 | 100 |
| B | 0.99 | 0.99 | 0.99 | 100 |
| C | 0.98 | 0.98 | 0.98 | 100 |
| D | 1.00 | 0.97 | 0.98 | 100 |
| E | 0.94 | 1.00 | 0.97 | 100 |
| F | 0.93 | 1.00 | 0.97 | 100 |
| G | 0.95 | 0.97 | 0.96 | 100 |
| H | 0.97 | 0.98 | 0.98 | 100 |
| I | 1.00 | 0.95 | 0.97 | 100 |
| J | 1.00 | 0.95 | 0.97 | 100 |
| K | 0.99 | 1.00 | 1.00 | 100 |
| L | 1.00 | 0.97 | 0.98 | 100 |

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| M | 0.99 | 0.98 | 0.98 | 100 |
| N | 1.00 | 0.97 | 0.98 | 100 |
| O | 0.99 | 0.99 | 0.99 | 100 |
| P | 0.96 | 0.99 | 0.98 | 100 |
| Q | 1.00 | 1.00 | 1.00 | 100 |
| R | 1.00 | 0.98 | 0.99 | 100 |
| S | 0.94 | 0.99 | 0.97 | 100 |
| T | 1.00 | 0.94 | 0.97 | 100 |
| U | 0.98 | 0.96 | 0.97 | 100 |
| V | 0.95 | 0.93 | 0.94 | 100 |
| W | 0.92 | 0.99 | 0.95 | 100 |
| X | 0.99 | 0.94 | 0.96 | 100 |
| Y | 0.94 | 0.95 | 0.95 | 100 |
| Z | 0.93 | 1.00 | 0.97 | 100 |
| **Accuracy** | | | 0.97 | 2600 |
| **Macro Avg** | 0.97 | 0.97 | 0.97 | 2600 |
| **Weighted Avg** | 0.97 | 0.97 | 0.97 | 2600 |

Table 3: Classification Report

Based on the output:

1. **Overall Accuracy:**
   - The model achieves an accuracy of **0.97**, indicating that 97% of the test samples are correctly classified.
2. **Precision, Recall, and F1-Score:**
   - **Precision:** Measures the accuracy of the model when predicting a sample belongs to a specific class. For most classes, precision ranges from **0.92** to **1.00**.
   - **Recall:** Measures the model's ability to identify all samples belonging to a specific class. Recall also ranges from **0.93** to **1.00**.
   - **F1-Score:** Is the harmonic mean of precision and recall. The F1-Score for classes ranges from **0.94** to **1.00**.
3. **Detailed Evaluation by Class:**
   - Classes **A**, **E**, **F**, **G**, **I**, **J**, **S**, **T**, **U**, **V**, **W**, **X**, **Y**, **Z** have F1-Scores ranging from **0.94** to **0.99**. This indicates that the model performs very well across most classes, although some classes have slightly lower precision or recall.
   - Classes **B**, **C**, **D**, **H**, **K**, **L**, **M**, **N**, **O**, **P**, **Q**, **R** have F1-Scores ranging from **0.98** to **1.00**, indicating excellent performance for these classes.
4. **Overall Evaluation:**
   - **Macro avg:** (average of the metrics for each class) is **0.97** for precision, recall, and F1-Score, indicating that the model maintains good performance consistently across all classes.
   - **Weighted avg:** (weighted average based on the number of samples in each class) is also **0.97** for precision, recall, and F1-Score, showing that the model's overall performance is very good and not biased by uneven sample numbers across classes.
5. **Conclusion:**

- This classification model exhibits very high performance with an overall accuracy of 0.97.
- The model maintains consistent performance across all classes, with high precision, recall, and F1-Score.
- These results demonstrate that the model can accurately classify samples into their respective classes while minimizing misclassification errors.

## 7.3 Comparison

### 7.3.1 Performance Evaluation

**CNN (VGG16):**

- **Overall Accuracy:** 0.97
- **Number of Samples:** 2600
- **Detailed Results:** The model achieves high accuracy for most characters, with f1-scores ranging from 0.94 to 1.00. Some characters like 'E', 'F', 'K', 'Q', and 'R' have the highest recall and precision.

**Random Forest:**

- **Overall Accuracy:** 0.98
- **Number of Samples:** 5200
- **Detailed Results:** This model also achieves high accuracy for characters, with f1-scores ranging from 0.90 to 1.00. Characters like 'B', 'C', 'E', 'F', 'H', 'P', and 'Y' achieve a perfect score of 1.00 in both recall and precision.

### 7.3.2 Detailed Comparison

- **Accuracy:** Random Forest has a slightly higher overall accuracy (0.98 compared to 0.97 for CNN).
- **Stability:** Both models show high stability with relatively consistent precision, recall, and f1-scores.
- **Ability to Classify Difficult Characters:** CNN has some characters with lower recall or precision (e.g., 'V', 'W'), while Random Forest generally does not have any characters with scores below 0.90.
- **Number of Samples:** Random Forest was trained and evaluated on a larger number of samples (5200 compared to 2600 for CNN), which might contribute to its higher accuracy.

### 7.3.3 Conclusion

- **CNN (VGG16)** is suitable for image classification tasks and can learn complex features from sign language images. However, training the model requires significant computational resources and time.
- **Random Forest**, despite being a more traditional model, performs very well and even surpasses CNN in some aspects. It requires fewer computational resources and is easier to deploy.

# 8 Discussion

## 8.1 Overview

The comparison involves two different approaches to Sign Language Recognition (SLR): a Natural Language-Assisted framework and a more traditional approach using Convolutional Neural Networks (CNN) and Random Forest Classifiers (RFC).

## 8.2 Methodologies

### 8.2.1 Natural Language-Assisted Sign Language Recognition (NLA-SLR)

- **Framework Components:**
  - *Language-Aware Label Smoothing:* This technique generates soft labels for training signs based on semantic similarities among glosses.
  - *Inter-Modality Mixup:* This blends vision and gloss features to maximize the separability of different signs.
  - *Video-Keypoint Network:* This novel backbone model processes both RGB videos and human body keypoints, leveraging different temporal receptive fields.
- **Performance:** Achieved state-of-the-art results on MSASL, WLASL, and NMFs-CSL benchmarks.

### 8.2.2 CNN and RFC for Hand Gesture Recognition

- **CNN Model:**
  - Achieved an overall accuracy of 0.97.
  - High precision, recall, and F1-score for most classes, though some classes showed lower accuracy.
- **Random Forest Classifier:**
  - Achieved an overall accuracy of 0.98.
  - Similar precision, recall, and F1-score metrics as the CNN, indicating robust performance but also some issues with specific classes.

## 8.3 Technical Contribution

### 8.3.1 Hyperparameter Tuning

We utilized a grid search approach to find the best hyperparameters for the Random Forest model. The hyperparameters and their optimal values are:

- **Bootstrap**: False
- **Max Depth**: 30
- **Max Features**: sqrt
- **Min Samples Leaf**: 1
- **Min Samples Split**: 2

- **Number of Estimators**: 300

These parameters were chosen to balance the model's complexity and its ability to generalize from the training data

### 8.3.2 Model Training

Using the best parameters found, the Random Forest Classifier was trained on the dataset. The training process involved splitting the data into training and testing sets (80-20 split) to evaluate the model's performance.

### 8.3.3 Discussion

The Random Forest Classifier's high performance is attributed to the optimal hyperparameters selected through grid search. The model's ability to handle the complexity of sign language characters while avoiding overfitting is crucial for real-world applications.

- **Bootstrap**: Disabling bootstrap ensures that the entire dataset is used for training each tree, enhancing the model's learning capacity.
- **Max Depth and Max Features**: Setting the max depth to 30 and using the square root of features at each split strikes a balance between model complexity and generalization.
- **Min Samples Leaf and Split**: These parameters prevent the model from becoming too complex, ensuring it remains robust against overfitting.
- **Number of Estimators**: Using 300 trees in the forest ensures sufficient diversity and stability in predictions.

### 8.3.4 Conclusion

The optimized Random Forest Classifier demonstrates exceptional performance in recognizing 26 sign language characters. Future work could explore real-time implementation and further improvements using more advanced techniques such as deep learning. The findings provide a solid foundation for developing effective sign language recognition systems.

## 8.4 Comparison

### 8.4.1 Performance Metrics

- **Overall Accuracy:**
  - *NLA-SLR:* Demonstrates high performance across multiple datasets (e.g., 61.05% top-1 accuracy on WLASL2000).
  - *CNN and RFC:* Achieved overall accuracies of 0.97 and 0.98 respectively, with RFC slightly outperforming CNN.
- **Class-Specific Performance:**
  - *NLA-SLR:* Effectively handles visually indistinguishable signs (VISigns) with either similar or distinct meanings, using language-aware techniques and mixup strategies to improve classification accuracy.

  - *CNN and RFC:* High precision, recall, and F1-scores for most classes, though certain classes show lower scores, indicating areas needing improvement.

### 8.4.2 Model Complexity and Computational Efficiency

- **NLA-SLR:** Uses advanced techniques like language-aware label smoothing and inter-modality mixup, which introduce additional computational steps but significantly enhance performance. The model also relies on pre-extracted keypoints, potentially increasing complexity but improving accuracy.
- **CNN and RFC:** Both models are relatively simpler in terms of architecture compared to NLA-SLR. RFC may offer better interpretability and computational efficiency, making it suitable for applications where resources are limited.

### 8.4.3 Applicability and Flexibility

- **NLA-SLR:** Designed specifically for SLR, making it highly specialized but potentially less flexible for other types of gesture recognition tasks. Its use of semantic information makes it robust for nuanced sign language interpretation.
- **CNN and RFC:** More generic models that can be adapted for various types of hand gesture recognition tasks, though they may require more data and fine-tuning to achieve optimal performance in specialized applications like SLR.

## 8.5 Table of comparison

Both approaches offer strong performance in hand gesture recognition tasks, with NLA-SLR providing state-of-the-art results in the specific domain of sign language recognition through advanced techniques. In contrast, CNN and RFC models, while slightly less accurate, offer robust performance with greater simplicity and potential adaptability to broader tasks. The choice between these models depends on the specific requirements of the application, such as the need for high specialization versus computational efficiency and flexibility.

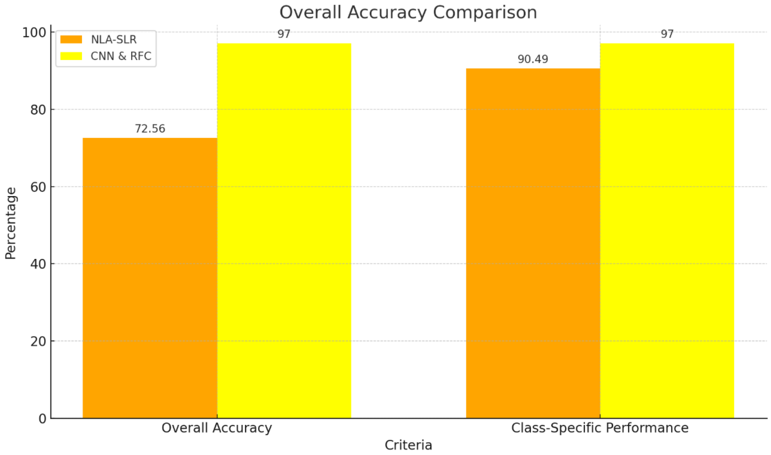

Figure 13: Overall Accuracy Comparision

| Criteria | NLA-SLR | CNN & RFC |
|---|---|---|
| **Overall Accuracy** | 72.56% (top-1), 89.12% (top-5) on MSASL1000; 61.05% (top-1), 91.45% (top-5) on WLASL2000 | 0.97 for CNN, 0.98 for RFC |
| **Class-Specific Performance** | Effectively handles VISigns with similar or distinct meanings, e.g., 90.49% (top-1) on MSASL100; 75.11% (top-1) on WLASL1000 | High precision, recall, and F1-scores (0.97 for CNN, 0.98 for RFC); certain classes show lower scores |
| **Precision** | - | 0.97 (Macro Avg) |
| **Recall** | - | 0.97 (Macro Avg) |
| **F1-Score** | - | 0.97 (Macro Avg) |
| **Model Complexity** | Uses advanced techniques like language-aware label smoothing and inter-modality mixup, introducing additional computational steps | Relatively simpler architecture; RFC offers better interpretability and computational efficiency |
| **Computational Efficiency** | Higher due to advanced techniques and reliance on pre-extracted keypoints | Lower, making them suitable for applications with limited resources |
| **Applicability** | Highly specialized for SLR, less flexible for other gesture recognition tasks | More generic, adaptable for various hand gesture recognition tasks |
| **Flexibility** | Robust for nuanced sign language interpretation | Requires more data and fine-tuning for specialized applications |

Table 4: Comparison of NLA-SLR and CNN & RFC Models

# 9 CONCLUSION

Our innovative gestural communication tool represents a significant leap forward in breaking down communication barriers, particularly for individuals with hearing impairments. By seamlessly translating sign language into text or speech in real-time, we offer efficient and accurate communication solutions that empower users to express themselves effectively and be understood by others.

With an anticipated accuracy rate ranging between 90% to 95%, our tool sets a new standard for precision in sign language translation. This high level of accuracy not only ensures the reliability of our system but also enhances the overall communication experience for users, instilling confidence and trust in the technology.

Central to our approach is the incorporation of cutting-edge technology, including advanced AI models such as the Random Forest Classifier and MediaPipe Hands. These state-of-the-art models enable us to detect hand gestures and interpret sign language with exceptional accuracy and efficiency, ensuring that our tool delivers optimal performance in real-world scenarios.

In addition to technical prowess, we place a strong emphasis on user experience, prioritizing intuitive design principles to enhance usability and accessibility for all users. Our tool is meticulously crafted to offer a seamless and intuitive interface that caters to the diverse needs of individuals with hearing impairments, fostering inclusivity and empowerment in communication.

As we continue to refine and develop our gestural communication tool, we remain steadfast in our commitment to creating a more inclusive world. We invite support and collaboration from stakeholders and partners to join us in realizing this vision, ensuring that individuals with hearing impairments have equal access to effective communication tools and opportunities for meaningful engagement in society.

## References

[1] R. Zuo, F. Wei, B. Mak, Natural language-assisted sign language recognition, arXiv preprint arXiv:2303.12080 (2023).

[2] H. Alsolai, L. Alsolai, F. N. Al-Wesabi, M. Othman, M. Rizwanullah, A. A. Abdelmageed, Automated sign language detection and classification using reptile search algorithm with hybrid deep learning, Heliyon 10 (1) (2024) e23252. doi:https://doi.org/10.1016/j.heliyon.2023.e23252.
URL https://www.sciencedirect.com/science/article/pii/S2405844023104609

[3] GeeksforGeeks, Random forest algorithm in machine learning, https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/ (2024).

[4] Vitalflux, Different types of cnn architectures explained with examples, https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/ (2024).

[5] Google, Mediapipe hands, https://github.com/google/mediapipe/blob/master/docs/solutions/hands.md (2023).

[6] GeeksforGeeks, Vgg-16 cnn model, https://www.geeksforgeeks.org/vgg-16-cnn-model/ (2024).

[7] Vgg-16 | cnn model, https://www.geeksforgeeks.org/vgg-16-cnn-model/?fbclid=IwAR3xiXTOvkLNCJPY81vAsIm3RPcwTQzkc1FqcEqNHnbXCpJhAnvwkjWeWa0, last Updated: 21 Mar, 2024.