# CHAPTER 1 INTRODUCTION

In our endeavor to promote inclusive communication, we have embarked on a Sign Language Detection Project. This innovative venture aims to bridge the communication gap between individuals proficient in sign language and those who may not be fluent in it. By harnessing the power of computer vision and machine learning, we have developed a real-time system capable of recognizing and translating sign language gestures into text. This technology holds immense potential for applications in education, healthcare, and various social contexts, ensuring that everyone can communicate effectively. Our commitment to inclusivity drives us to create solutions that foster better understanding and connection among diverse communities.

# 1.1 About the Project

The Sign Language Detection Project leverages innovative technology to facilitate seamless communication for individuals who use sign language. By employing computer vision techniques, we can accurately recognize and interpret a wide range of sign gestures. This real-time system employs machine learning algorithms to convert these gestures into written text, making communication more accessible for everyone. With applications spanning education, healthcare, and everyday interactions, this project is a significant step towards a more inclusive society. Through our efforts, we aim to empower individuals to express themselves freely and be understood by a broader audience.

# CHAPTER 2 RELATED WORK

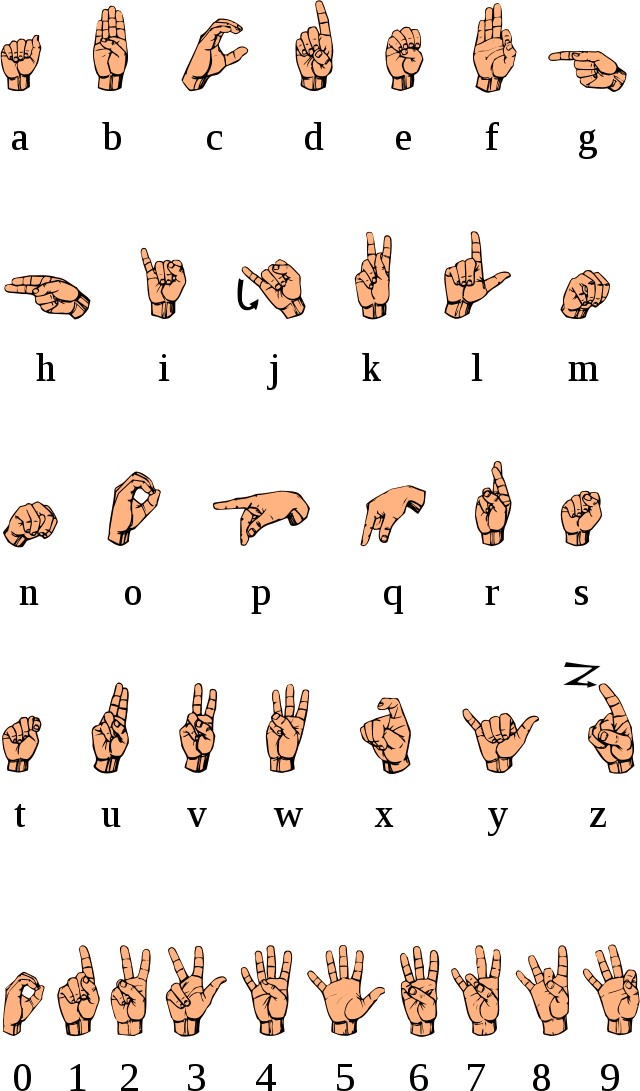
In recent years, significant strides have been made in the field of computer vision for sign language recognition, ushering in a new era of inclusive communication and accessibility. Notable research from 2014 by Yan, Ji, and Li highlighted the real-time capabilities of sign language recognition using depth sensors such as Kinect, which revolutionized gesture-based communication by effectively capturing and processing sign language gestures. Furthermore, the work of Malik and Zhang in the same year introduced the incorporation of wearable devices for American Sign Language recognition, offering portability and convenience. Their wearable computer-based video system has made it possible for users to engage in seamless sign language communication on the go.

A pivotal factor behind these transformative advancements has been the integration of machine learning, particularly deep learning techniques. Research by Som and Tiwari in 2018 demonstrated the efficacy of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks in real-time fingerspelling recognition, significantly enhancing the accuracy and efficiency of gesture interpretation. These collective achievements in the field of computer vision and machine learning have opened new possibilities for bridging communication gaps and promoting accessibility for the deaf and hard-of-hearing communities, marking a remarkable shift in sign language recognition technology.

# CHAPTER 3 LIST OF MODULES

|  |  |
| --- | --- |
| **S.NO** | **MODULES** |
| **1.** | Image Collection |
| **2.** | Dataset Creation |
| **3.** | Training the Dataset |
| **4.** | Inference Classifier |

**Table 3.1** List of Modules



**Fig 3.1** American Standard Signs

# CHAPTER 4

**PROPOSED SYSTEM**

The proposed system aims to create a real-time sign language recognition system using computer vision and machine learning. The goal is to bridge the communication gap between sign language users and non-signers, making communication accessible and inclusive. This system recognizes and translates sign language gestures into text, facilitating easier interactions in various contexts, including education, healthcare, and daily life.

## Image Collection:

Image collection is a crucial part of our sign language recognition project. To build a reliable system, we need a diverse and comprehensive dataset of sign language gestures. The quality and variety of the collected images directly impact the system's performance in recognizing different signs and hand positions. Ensuring our image collection includes a wide range of sign language expressions, representing various sign languages and dialects, is essential. This diversity enables us to create a system that can cater to the needs of a more extensive user base, making it more inclusive and accessible. Additionally, rigorous image curation and data annotation processes are imperative to train our recognition algorithms effectively, making our project a valuable resource for the sign language community.

## Data Gathering:

Collecting data involves capturing images of sign language gestures using a camera. It's important to ensure the dataset includes a wide range of gestures, different hand shapes, and gestures performed by various individuals. Moreover, lighting conditions need to be considered to make the system robust to different environments. High-quality images are essential. They should be well-focused, well-lit, and have minimal background distractions. Images with different backgrounds and orientations are necessary to ensure the system's generalization. Additionally, to enhance the dataset's diversity, it's beneficial to capture both static and dynamic sign language gestures, as sign language involves both individual signs and the fluid transitions between them.

.

## Dataset Creation:

The creation of a sign language dataset is a fundamental step in developing a robust and accurate recognition system. The dataset serves as the training ground for the machine learning model to understand and interpret sign language gestures. It plays a pivotal role in ensuring that the model can effectively recognize and translate the intricate and diverse hand movements and expressions that constitute sign language. To build a comprehensive sign language dataset, a wide range of signs and expressions from various sign languages should be included. This diversity allows the model to be more inclusive and adaptable, catering to the needs of different sign language communities. Furthermore, capturing variations in lighting conditions, backgrounds, and signer demographics is essential to make the recognition system resilient and reliable in real-world scenarios. Collecting data from a diverse group of signers, including individuals with different proficiency levels, helps enhance the model's ability to handle natural language variations.

## Data Preprocessing:

Before adding images to the dataset, data preprocessing is conducted. This involves normalizing hand landmarks and padding them to match a consistent format. Normalization ensures that the model can manage variations in hand size, position, and orientation. By bringing all hand landmarks to a standardized scale and orientation, the model becomes less sensitive to individual differences in hand

anatomy, which is especially important when dealing with diverse user inputs. Furthermore, padding the hand landmarks to a consistent size allows for more efficient and streamlined processing by the neural network. This not only enhances computational efficiency but also simplifies the model architecture, making it more manageable and interpretable.

## Dataset Structure:

The dataset is structured into training and testing sets. The training set is used to teach the model, allowing it to learn and adapt to the underlying patterns and features in the data. This training process is crucial for the model's ability to make predictions and decisions based on the information it has acquired. In contrast, the testing set serves as a critical evaluation phase, assessing the model's accuracy and its ability to generalize beyond the training data. By evaluating the model on unseen data, we can gain insights into its performance and ensure that it can make reliable predictions on new, real-world data. This division of data into training and testing sets is a fundamental practice in machine learning and plays a pivotal role in building robust and effective models.

## Data Variety:

The dataset should cover a wide range of sign language gestures, including the alphabet, numerals, and common signs. It must also account for variations in signing speed, style, and hand positions. This diversity in data is essential to ensure that the model trained on the dataset can effectively recognize and interpret sign language across different contexts and individuals. It should encompass signs from various sign languages and dialects, as regional variations can significantly impact sign language communication. Furthermore, the dataset should include signs performed by individuals of different ages and backgrounds to account for the natural diversity present in the signing community.

## Data Annotation:

Each image in the dataset is annotated with the corresponding sign language symbol. Accurate labelling is essential for model training and assessment. These annotations not only provide critical context to the images but also enable the development and evaluation of sign language recognition and translation models. This meticulous labelling process involves expert sign language interpreters who ensure that the symbols are correctly identified and labelled, maintaining the dataset's integrity. Moreover, the precise annotation contributes to the dataset's accessibility and usability for researchers, making it a valuable resource for advancing the field of sign language recognition technology and promoting inclusivity in communication.

## Model Training:

Model training is a pivotal stage in the development of our sign language recognition system. It involves teaching the machine learning model to understand and classify sign language gestures accurately. During this phase, a large dataset of sign language gestures is used to train the model, enabling it to learn the subtle nuances of various signs and their meanings. The model goes through numerous iterations, adjusting its internal parameters to improve accuracy and reduce errors. This iterative process is guided by the expertise of data scientists and linguists who ensure that the model captures the rich diversity of sign languages and the cultural context behind them. The success of our sign language recognition system relies heavily on the effectiveness of this training process in bridging the communication gap between the deaf and hearing communities.

## Data Preparation:

The model is trained using the pre-processed sign language dataset. The dataset

consists of images of sign language gestures, with each image labelled with the corresponding sign. These images have been carefully curated and processed to ensure high quality and consistency in both lighting and background conditions. Additionally, the dataset includes a diverse range of sign gestures to provide the model with a comprehensive understanding of sign language communication. To further enhance the model's performance, noise reduction techniques have been applied to the dataset, minimizing any unwanted artifacts or distractions. Data augmentation methods, such as rotation, scaling, and translation, have also been employed to increase the dataset's size and improve the model's robustness in recognizing signs from various angles and hand positions.

## Model Selection:

In the process of model selection, a critical decision is made to choose the most suitable machine learning algorithm that aligns with the objectives of the project. One of the algorithms that often proves effective is the Random Forest Classifier, known for its versatility and robust performance across various domains. The model selection phase also involves careful consideration of the hyperparameters and initialization values, ensuring that the algorithm is fine-tuned to achieve optimal results. Furthermore, the choice of the machine learning algorithm is contingent upon the nature of the data, the complexity of the problem, and the available resources.

## Training Process:

The machine learning model learns from the dataset, capturing patterns and relationships between images and their associated signs. The goal is to enable the model to make accurate predictions during real-time gesture detection. During the training process, the model undergoes multiple iterations, adjusting its internal parameters to minimize the difference between its predictions and the actual signs

in the dataset. This process, often referred to as "backpropagation," involves continuously fine-tuning the model's weights and biases. To enhance the model's ability to generalize, the dataset is typically divided into training, validation, and test sets. The training set is used for training the model, while the validation set helps in monitoring the model's performance and preventing overfitting, a common issue in machine learning.

## Model Evaluation:

The trained model's accuracy and performance are assessed using the testing dataset. This evaluation helps ensure the model's reliability and generalization capabilities. By comparing the model's predictions to the actual data in the testing dataset, we can gauge how well it performs on unseen data, which is a crucial measure of its practical utility. In addition to accuracy, various other metrics such as precision, recall, F1-score, and area under the ROC curve may also be considered, depending on the specific problem and its requirements. This comprehensive evaluation process is essential for identifying potential issues, improving the model's robustness, and making informed decisions about its deployment in real-world applications.

## Real-time Gesture Detection:

Real-time gesture detection is the core functionality of our sign language recognition system, serving as the bridge between the deaf and hard-of-hearing community and the hearing world. This innovative technology operates seamlessly, capturing the intricate movements and expressions of sign language users in real- time. As a result, it provides an invaluable tool for immediate translation of sign language gestures into text, facilitating effective communication and enhancing accessibility in various domains. Whether it's in educational settings, healthcare interactions, or everyday conversations, our system empowers individuals to

express themselves in their preferred mode of communication. By breaking down communication barriers, we aim to foster inclusivity and bridge the gap between the deaf and hearing communities.

## Video Input:

The system seamlessly captures video input from a camera, operating in real-time to ensure a continuous flow of data. This video stream is the lifeblood of our sign language recognition technology, as it provides the visual information necessary to detect and interpret sign language gestures with precision and accuracy. Through sophisticated algorithms and machine learning models, the system meticulously processes each frame of the video, recognizing not only the gestures themselves but also the subtleties and nuances that convey the rich and diverse vocabulary of sign language. Our commitment to this video input is not only about recognizing signs but also about understanding the context and emotions behind each gesture.

## Computer Vision Techniques:

Computer vision techniques are employed to identify and track hand landmarks within each frame, a crucial component in the realm of gesture recognition. This process entails the extraction and analysis of key features, allowing the system to comprehend and interpret intricate hand gestures. These techniques leverage advanced algorithms and image processing methods to detect specific points and contours on the hand, enabling real-time tracking and precise recognition of dynamic gestures. By harnessing the power of computer vision, this technology enhances human-computer interaction, enabling a wide range of applications, from sign language recognition to gesture-based control of devices, offering a seamless and intuitive user experience in various domains.

## Model Application:

The trained machine learning model is applied to the processed data, making real- time predictions about the sign being performed. This prediction is then translated into text or symbols, enabling seamless communication for individuals with hearing impairments. By harnessing advanced computer vision and pattern recognition techniques, the system can identify and interpret complex sign language gestures with a high degree of accuracy. This technology has the potential to bridge the communication gap between the deaf and hearing communities, offering a more inclusive and accessible environment. Additionally, the real-time nature of the predictions allows for immediate interaction, facilitating more natural and efficient conversations between sign language users and those who may not be proficient in sign language.

## Visualization:

The results of the real-time gesture detection are displayed on the screen, providing users with a seamless and interactive experience. Through innovative technology, the system can overlay the recognized sign onto the screen, creating a visual representation of the communicated message. Additionally, the corresponding text associated with the recognized sign is displayed, facilitating clear and efficient communication. This dual-mode approach not only enhances inclusivity but also ensures that the information is readily accessible to a wider audience. In this dynamic visualization process, the system leverages advanced graphics to highlight and emphasize the recognized gestures, making it easier for both the sender and receiver of the message to engage effectively.

# edCHAPTER 5 EVALUATION

Our evaluation process is a comprehensive assessment of our real-time sign language recognition system, with a focus on accuracy, generalization, real-time performance, user feedback, and inclusivity. By integrating these elements into our evaluation process, we strive to provide a well-rounded assessment of our system's capabilities and limitations, enabling us to continually refine and enhance our technology to better serve the diverse needs of the sign language community and contribute to a more inclusive and accessible communication environment.

## Model Accuracy:

We prioritize the accuracy of our machine learning model, ensuring it correctly identifies and classifies sign language gestures by comparing predictions with the ground truth data. High accuracy minimizes misunderstandings in real-time communication. Accurate predictions are crucial for facilitating effective communication and enhancing the user experience. Additionally, a highly accurate model helps in reducing the cognitive load on users, making it more accessible and user-friendly. We constantly strive to improve our model's accuracy through data refinement, algorithm optimization, and rigorous testing. Our commitment to accuracy is fundamental in ensuring that our technology is a reliable and empowering tool for the deaf and hard of hearing community.

## Model Robustness and Adaptability:

In addition to accuracy, it's crucial to assess the robustness and adaptability of our machine learning model when evaluating its performance. We should consider how well the model generalizes to various sign language styles, lighting conditions, and

backgrounds, ensuring it can maintain high accuracy in diverse real-world settings. Furthermore, evaluating the model's ability to adapt to new signs or gestures, as well as its efficiency in retraining, when necessary, can be pivotal in ensuring its long-term effectiveness in facilitating seamless communication for individuals who rely on sign language. This multifaceted evaluation approach goes beyond mere accuracy, addressing the model's real-world applicability and user experience.

## Generalization:

Our system's ability to handle diverse sign language gestures, irrespective of variations in signing speed, style, or hand positions, is critical. We rigorously evaluate its adaptability under various conditions. Sign language is a rich and dynamic form of communication, and we understand the importance of ensuring that our technology can effectively bridge communication gaps for a wide range of users. Whether it's a fluent, rapid signer or someone who signs with a distinct style, our system is designed to comprehend and respond accurately. Additionally, we recognize that hand positions and movements can vary significantly between sign languages and within different regional dialects. Our ongoing commitment to improving generalization means that our system is continually evolving to meet the diverse needs of the sign language community. It's not just about recognizing signs; it's about fostering inclusivity and accessibility for all.

## Usability and Accessibility:

In addition to assessing the system's generalization, it is equally essential to evaluate its usability and accessibility. An effective sign language recognition system should be user-friendly and accessible to a wide range of individuals, including those with varying levels of sign language proficiency. Therefore, our evaluation extends to user feedback and user testing, focusing on how intuitively the system can be operated and how well it serves its target users. This includes

considering the user interface design, the system's response time, and any potential barriers that might affect its accessibility for people with disabilities. By combining these usability and accessibility assessments with rigorous generalization testing, we aim to develop a comprehensive understanding of the system's overall performance and its suitability for diverse user populations.

## Real-time Performance:

Prompt gesture detection and translation are essential in our pursuit of seamless communication. We understand the critical need for near-instantaneous results, as this not only facilitates smooth and efficient interactions but also effectively mitigates any potential delays or lags that could compromise the overall user experience. Achieving real-time performance is paramount, as it ensures that users can communicate effortlessly and without interruption, bridging geographical and linguistic barriers with ease. To achieve this, we continuously optimize our systems and technologies, striving to push the boundaries of real-time responsiveness, ultimately enhancing the usability and accessibility of our services. Our commitment to delivering instantaneous results underscores our dedication to providing the highest quality communication tools for our users.

## Accuracy and Reliability:

In addition to real-time performance, the accuracy and reliability of prompt gesture detection and translation are paramount. Ensuring that the system correctly interprets and translates gestures is crucial for effective communication. Any errors or misinterpretations can lead to confusion and misunderstandings. Therefore, continuous monitoring, testing, and improvement of the algorithm's accuracy are essential. Reliability also plays a significant role, as users should be able to trust that the system will consistently provide accurate translations, fostering confidence

in the technology. Regular updates and quality control measures should be implemented to maintain and enhance both accuracy and reliability, aligning with the goal of delivering a seamless and dependable user experience.

## Inclusivity:

Inclusivity is at the core of our mission. Our goal is to enhance communication between sign language users and the broader community by bridging linguistic and cultural gaps. We are dedicated to creating a system that not only facilitates communication but also promotes mutual understanding and empathy. To achieve this, we rigorously assess the system's impact across various contexts, including education, healthcare, and daily life. By doing so, we ensure that our technology not only empowers sign language users but also enriches the lives of the broader community, fostering an environment of unity and inclusivity. We believe that meaningful and inclusive communication is the key to building a more equitable and compassionate world, and we are committed to making that vision a reality.

## User-Focused Design:

In developing our sign language communication system, we prioritize a user- centred approach. We engage with the sign language community, including deaf and hard-of-hearing individuals, interpreters, and educators, to gather feedback and ensure the system's design and features align with their needs and preferences. This iterative process allows us to continually refine the system to better serve the diverse and dynamic requirements of its users, ultimately driving a more inclusive and effective means of communication.

# CHAPTER 6 RESULTS

## VISUAL REPOSITORY:

This module captures sign language gestures via a camera feed. It records a diverse dataset, encompassing various hand positions and signs, ensuring high-quality images with proper lighting. These collected images lay the foundation for subsequent dataset creation and model training, making it a critical step in the sign language translation project.

**Fig 3.1** Business case 1

**Fig 6.1.1** Image Capture

During the image capture process, a high-resolution camera captures sign language gestures with precision, focusing on various hand positions and signs performed by a diverse group of individuals. Each image is meticulously reviewed to ensure optimal lighting, clarity, and quality. The collected images are then organized into a well-structured directory, making it easy to access and manage the dataset. This organized 'List of Images in Folder' serves as a valuable resource for researchers, facilitating efficient data retrieval and enabling seamless integration into the subsequent stages of the sign language translation project. The meticulous approach to image capture and dataset organization ensures that the foundation for this

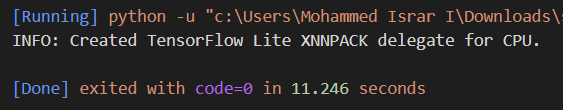
project is solid, setting the stage for effective model training and meaningful results.



**Fig 6.1.2** List of Images in Folder

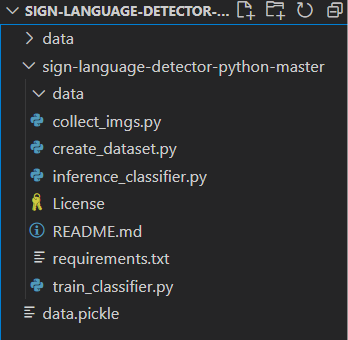
## CONSTRUCTING THE DATASET:

The "Dataset Creation" module is responsible for structuring the collected sign language gesture images into a well-organized dataset. It involves categorizing the images into different classes or labels, ensuring a balanced representation of gestures. Additionally, the module preprocesses the data by normalizing and padding the hand landmarks to a uniform format, making it suitable for machine learning.



**Fig 6.2.1** Dataset Created

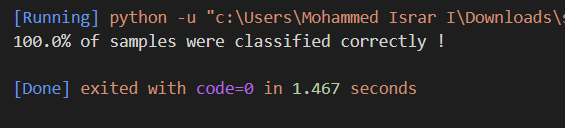
After the dataset creation process, the next crucial steps involve preparing the dataset for machine learning tasks. The dataset, once structured and categorized into different classes or labels, undergoes further processing to ensure it is in a format ready for model training. One common practice is normalizing the hand landmarks to ensure consistent scaling and orientation across all images. Additionally, padding may be applied to standardize the dimensions of the images, making them suitable for input into machine learning models. Once these preprocessing steps are complete, the dataset is often converted into a convenient storage format, such as a pickle file. This format is highly suitable for efficient data storage and retrieval, enabling researchers and developers to easily access and work with the dataset for various applications, including sign language recognition and gesture-based machine learning tasks.



**Fig 6.2.2** Dataset Converted into Pickle File

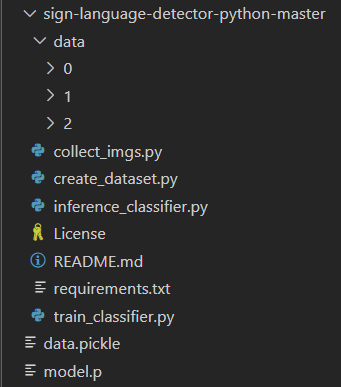
## MODEL TRAINING PHASE:

In this crucial phase, the dataset is fed into a machine learning algorithm, such as a Random Forest Classifier, for model creation. The algorithm learns patterns and relationships in sign language gestures, enabling accurate recognition. Training includes fine-tuning parameters to optimize performance and prepare the system for real-time inference. This module forms the heart of the sign language translation project, empowering the system to understand and interpret sign language gestures effectively.



**Fig 6.3.1** Classification of Data

Before training the machine learning model, it's essential to classify the sign language gesture data properly. This step involves labelling the data to distinguish different gestures and gestures' variations. Each sign is categorized into its respective class, ensuring that the model can differentiate between signs accurately. Classification is a critical aspect of preparing the dataset for training, as it lays the foundation for the model to learn and distinguish between different signs effectively.

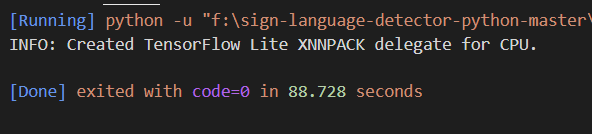


**Fig 6.3.2** Generation of Pickle Dataset

Once the data has been properly classified and labelled, it can be serialized into a "Pickle" dataset format. Pickle is a widely used data serialization library in Python that allows you to save complex data structures, such as labelled sign language gesture data, in a binary format. This serialized dataset is efficient for storing and loading data, making it readily accessible for model training and real-time inference. The Pickle dataset contains all the necessary information, including the sign labels and their corresponding image data, enabling seamless integration with the machine learning algorithm. This step streamlines the training process and facilitates the model's ability to recognize sign language gestures accurately.

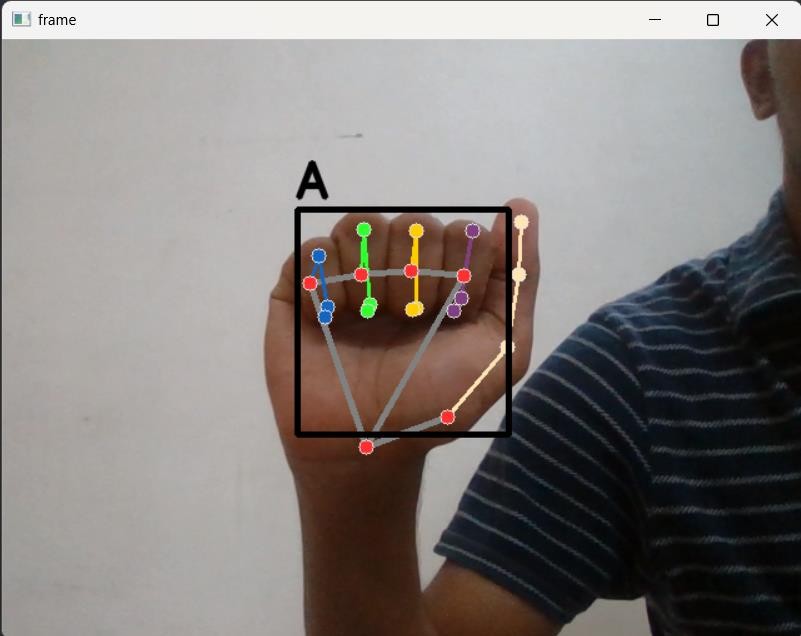
## GESTURE RECOGNITION:

This module utilizes machine learning algorithms to train a model on the collected sign language gesture data. It learns and internalizes patterns, allowing it to recognize and classify gestures accurately. The trained model is a crucial component for real-time gesture translation and enhances communication accessibility.



**Fig 6.4.1** Execution of Trained Dataset

Once the machine learning model is trained on the collected sign language gesture data, it becomes an indispensable tool for real-time communication accessibility. The execution of this trained dataset involves a two-fold process: sign detection and gesture recognition. Sign detection is the initial step where the system captures the user's hand movements or facial expressions using cameras and sensors. These captured signs are then analysed by the trained model in real-time. Through a sophisticated combination of computer vision and pattern recognition techniques, the system identifies the signs being performed and translates them into corresponding words or phrases. This seamless integration of sign detection and gesture recognition enables individuals with hearing impairments to express themselves effectively, breaking down communication barriers and enhancing inclusivity in various settings.



**Fig 6.4.2** Sign Detection

By continuously processing and updating its knowledge base, the inference classifier becomes more adept at understanding the nuanced expressions and variations within sign language. It not only identifies the basic sign gestures but also adapts to the unique signing styles of different individuals, making it a versatile tool for a wide range of users. Its capacity to evolve and learn ensures that it remains relevant and effective as it encounters new signs and user-specific variations. Moreover, the inference classifier can be integrated into various communication devices and applications, making it a valuable resource for promoting inclusive and accessible communication for the Deaf and Hard of Hearing communities.

# CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENT**

The Sign Language Translation Project represents a significant leap in leveraging computer vision and machine learning to bridge the communication gap for sign language users. By enabling real-time gesture recognition and translation, this project enhances accessibility and contributes to a deeper understanding of sign language. Its potential applications in education, healthcare, and everyday interactions make it a promising solution.

Looking ahead, there are opportunities for further improvement and expansion. To enhance accuracy assessment, the project can develop a pixel-level evaluation method tailored to sign language recognition. This method would effectively compare detected signs to ground truth values.

The challenge of recognizing multiple signs within a single frame, especially when sign sizes vary, needs careful consideration. Future enhancements should focus on refining the model to handle complex signing scenarios accurately.

Moreover, accommodating regional or language-specific sign variations and expanding the system's vocabulary can enhance its inclusivity and usability. Ongoing research and development in these areas will refine the project's performance, ultimately benefiting sign language users and their interactions with the wider community.

## APPENDICES APPENDIX 1 SAMPLE SCRIPT

**Code:**

**Image Collection:**

import os import cv2

DATA\_DIR = './data'

if not os.path.exists(DATA\_DIR): os.makedirs(DATA\_DIR)

number\_of\_classes = 36 # Update the number of classes to 36 (0 to 35) dataset\_size = 100

cap = cv2.VideoCapture(0) # Use camera index 0 (the default camera) for j in range(number\_of\_classes):

if not os.path.exists(os.path.join(DATA\_DIR, str(j))): os.makedirs(os.path.join(DATA\_DIR, str(j)))

if j < 26:

class\_label = chr(ord('A') + j) # Letters from 'A' to 'Z' else:

class\_label = str(j - 26) # Numbers from 0 to 9 print('Collecting data for class {}'.format(class\_label))

done = False while True:

ret, frame = cap.read() if not ret:

continue # Skip frames without valid data cv2.putText(frame, 'Ready? Press "Q" ! :)', (100, 50),

cv2.FONT\_HERSHEY\_SIMPLEX, 1.3, (0, 255, 0), 3, cv2.LINE\_AA)

cv2.imshow('frame', frame)

if cv2.waitKey(25) == ord('q'): break

counter = 0

while counter < dataset\_size: ret, frame = cap.read()

if not ret:

continue # Skip frames without valid data cv2.imshow('frame', frame) cv2.waitKey(25)

cv2.imwrite(os.path.join(DATA\_DIR, str(j), '{}.jpg'.format(counter)), frame) counter += 1

cap.release() cv2.destroyAllWindows()

**Create Dataset:**

import os import pickle

import mediapipe as mp import cv2

mp\_hands = mp.solutions.hands mp\_drawing = mp.solutions.drawing\_utils

hands = mp\_hands.Hands(static\_image\_mode=True, min\_detection\_confidence=0.3)

DATA\_DIR = 'F:/sign-language-detector-python-master/data' data = []

labels = []

for dir\_ in os.listdir(DATA\_DIR):

dir\_path = os.path.join(DATA\_DIR, dir\_)

if os.path.isdir(dir\_path): # Check if it's a directory for img\_path in os.listdir(dir\_path):

data\_aux = []

x\_ = []

y\_ = []

img = cv2.imread(os.path.join(dir\_path, img\_path)) img\_rgb = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

results = hands.process(img\_rgb)

if results.multi\_hand\_landmarks:

for hand\_landmarks in results.multi\_hand\_landmarks: for i in range(len(hand\_landmarks.landmark)):

x = hand\_landmarks.landmark[i].x y = hand\_landmarks.landmark[i].y

x\_.append(x) y\_.append(y)

for i in range(len(hand\_landmarks.landmark)): x = hand\_landmarks.landmark[i].x

y = hand\_landmarks.landmark[i].y data\_aux.append(x - min(x\_)) data\_aux.append(y - min(y\_))

data.append(data\_aux) labels.append(dir\_)

f = open('data.pickle', 'wb') pickle.dump({'data': data, 'labels': labels}, f) f.close()

**Train Classifier:**

import pickle import numpy as np

from sklearn.ensemble import RandomForestClassifier from sklearn.model\_selection import train\_test\_split from sklearn.metrics import accuracy\_score

data\_dict = pickle.load(open('./data.pickle', 'rb')) data = data\_dict['data']

labels = data\_dict['labels']

# Ensure all data points have the same shape by padding or truncating them max\_data\_length = max(len(data\_point) for data\_point in data)

for i in range(len(data)):

data[i] = data[i] + [0] \* (max\_data\_length - len(data[i]))

data = np.array(data) labels = np.array(labels)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(data, labels, test\_size=0.2, shuffle=True, stratify=labels)

model = RandomForestClassifier() model.fit(x\_train, y\_train)

y\_predict = model.predict(x\_test)

score = accuracy\_score(y\_predict, y\_test)

print('{}% of samples were classified correctly !'.format(score \* 100)) f = open('model.p', 'wb')

pickle.dump({'model': model}, f) f.close()

**Inference Classifier:**

import pickle import cv2

import mediapipe as mp import numpy as np

model\_dict = pickle.load(open('./model.p', 'rb')) model = model\_dict['model']

cap = cv2.VideoCapture(0) if not cap.isOpened():

print("Error: Camera not found or not accessible.")

exit()

mp\_hands = mp.solutions.hands mp\_drawing = mp.solutions.drawing\_utils

mp\_drawing\_styles = mp.solutions.drawing\_styles

hands = mp\_hands.Hands(static\_image\_mode=True, min\_detection\_confidence=0.3)

labels\_dict = {

0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'I', 9: 'J',

10: 'K', 11: 'L', 12: 'M', 13: 'N', 14: 'O', 15: 'P', 16: 'Q', 17: 'R', 18: 'S',

19: 'T', 20: 'U', 21: 'V', 22: 'W', 23: 'X', 24: 'Y', 25: 'Z',

26: '0', 27: '1', 28: '2', 29: '3', 30: '4', 31: '5', 32: '6', 33: '7', 34: '8', 35: '9'

}

while True: data\_aux = [] x\_ = []

y\_ = []

ret, frame = cap.read()

if not ret: # Check if a valid frame was retrieved continue

H, W, \_ = frame.shape

frame\_rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB) results = hands.process(frame\_rgb)

if results.multi\_hand\_landmarks:

for hand\_landmarks in results.multi\_hand\_landmarks: mp\_drawing.draw\_landmarks(

frame, # image to draw hand\_landmarks, # model output

mp\_hands.HAND\_CONNECTIONS, # hand connections mp\_drawing\_styles.get\_default\_hand\_landmarks\_style(), mp\_drawing\_styles.get\_default\_hand\_connections\_style())

for hand\_landmarks in results.multi\_hand\_landmarks: for i in range(len(hand\_landmarks.landmark)):

x = hand\_landmarks.landmark[i].x y = hand\_landmarks.landmark[i].y

x\_.append(x) y\_.append(y)

for i in range(len(hand\_landmarks.landmark)): x = hand\_landmarks.landmark[i].x

y = hand\_landmarks.landmark[i].y data\_aux.append(x - min(x\_)) data\_aux.append(y - min(y\_))

# Zero-pad data\_aux to match the expected 42 features

data\_aux = data\_aux + [0] \* (42 - len(data\_aux))

x1 = int(min(x\_) \* W) – 10 y1 = int(min(y\_) \* H) - 10 x2 = int (max(x\_) \* W) – 10 y2 = int (max(y\_) \* H) – 10

prediction = model.predict([np.asarray(data\_aux)]) predicted\_character = labels\_dict[int(prediction[0])]

cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 0), 4) cv2.putText(frame, predicted\_character, (x1, y1 - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 1.3, (0, 0, 0), 3, cv2.LINE\_AA)

cv2.imshow('frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'): break

cap.release() cv2.destroyAllWindows()

# REFERENCES

* Kodandaram, Satwik Ram & Kumar, N. & Gl, Sunil. (2021). Sign Language Recognition. 10.13140/RG.2.2.29061.47845.

https://[www.researchgate.net/publication/353141966\_Sign\_Language\_Re](http://www.researchgate.net/publication/353141966_Sign_Language_Re) cognit ion

* Kang, Byeongkeun & Tripathi, Subarna & Nguyen, Truong. (2015). Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. 136-140.

10.1109/ACPR.2015.7486481.

https://[www.researchgate.net/publication/305233784\_Real-](http://www.researchgate.net/publication/305233784_Real-) time\_sign\_language\_fingerspelling\_recognition\_using\_convolutiona l\_neural\_networks\_from\_depth\_map

* Hansen, L.K. & Salamon, Peter. (1990). Neural Network Ensembles. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 12. 993 - 1001. 10.1109/34.58871.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=58871&isnum ber=2141