COURSE PROJECT REPORT

ON

REAL TIME SIGN LANGUAGE DETECTION

BY

NAME: Sana Firdous	**BITS ID:2022A7PS0193U**
NAME: Palak Sharma	**BITS ID:2022A7PS0201U**
NAME: Shraddha Sudhakaran	**BITS ID:2022A7PS0314U**

BITS Pilani, Dubai Campus
Dubai International Academic City, Dubai
UAE

1. INTRODUCTION

For millions of deaf and hard-of- hearing people around the world, sign language is their primary means of communication and allows them to find their voice and communicate in an often aural-centric world. But, there is still one important obstacle to that which is general ignorance and doesn't know about sign language itself. The inability to bridge this gap may result in the social alienation of and impediments in education and work for, and problems in day-to-day activities of people who use sign language. Understanding this urgent need, the current project proposes a solution that will close the gap by creating a real-time sign language gesture recognition system using the most recent advances in computer vision and artificial intelligence.

The intended outcome of this system is to take in real-time video footage of ASL finger spelling and output readable text in real-time. Equipped with hand-tracking technologies such as MediaPipe and deep learning models, the system is able to recognize hands movements reliably in less than ideal situations like low lighting conditions or video noise. In contrast to existing sign language recognizers, which are mostly focused on the interpretation of a single user or work in controlled environments, this project is able to recognize gestures performed by more than one signer at the same time, which makes it feasible for use in group environments.

More than a technical endeavor, the project is looking to creating more access and inclusion for the deaf communities at large. With a well-designed interface and solid performance, the system allows deaf people and nonsigners to better communicate with one another.

1.1 Motivation

The main goal of this project is to solve the communication barrier for the deaf in everyday life, schooling, employment, and social engagements. The general public's lack of knowledge in sign language reduces the accessibility and inclusion of sign language users.

1.2 Novelty

Our system, as opposed to existing signer recognition systems that aim at identifying single signers or are constrained by environmental conditions, features several novelties:

- Multi-Signer Real-Time Recognition
  The system can differentiate between gestures by multiple signers at the same time and produce a sentence for each. This is useful for communicating with a group of people or for those in a classroom setting.

- Robustness to Low-Quality Videos
  To enhance generalization and perform well in low light conditions, in cases of motion blur, or with low-resolution footage, our dataset comprises noisy, low-quality videos.

- Advanced Detection with Mediapipe
  The use of Mediapipe enhances person-detection in real-time which guarantees that only real signers are being tracked and processed, minimizing false positives while filtering background noise.

1.3 Objectives

To create a working system that can detect and translate finger spelling signs in American sign language in real time from a live video, create an accessible resource for those in the deaf community for daily communication purpose. It is to also highlight the need for sign language accessibility and show that technology can, indeed, lead to inclusion for all.

1.4 Contributions

- Dual-User Simultaneous Recognition
  Designed a system capable of recognizing both user's finger-spelled characters simultaneously, through the implementation of MediaPipe's multi-hand tracking as well as a separate prediction pipeline for each user.

- Queue-based Majority Voting for Stable Gesture Prediction
  Processed frames within queues to achieve a stable, noisy-free output that would as a result enhance the overall reliability during online usage.

- User-Defined ASL Landmark Dataset
  Combined custom landmarks datasets for several users, allowing for an increase in scale and greater variability for the model to generalize better on different hand shapes and sizes .

- Optimized DNN
  A light-weight fully connected deep neural network model has been designed and trained on 63 input landmarks that obtains a high classification accuracy amongst the ASL alphabet, including special tokens such as SPACE and BACKSPACE.

- User-Friendly GUI
  Created a user-friendly graphical user interface using Streamlit where a user can view live video, predictions, and updating sentences with the ability to start, stop, and clear the input without any coding knowledge.

- Real-Time Deployment
  Streamlit application was deployed in the field with low latency webcam input for real-time gesture to text in the classroom, accessibility, or communication scenarios.

2. Solution Approach

2.1 Block Diagram of the System

The overall architecture of the American Sign Language (ASL) finger spelling recognition system is designed to smoothly translate hand gestures in real time into structured text sentences. The entire process is shown in the block diagram below, which includes webcam video input capture, MediaPipe hand tracking and landmark extraction, feature preprocessing, and classification using a deep neural network model that has already been trained. A frame queue and majority voting mechanism are used to stabilize predictions after classification to guarantee accurate and seamless output. Sentences are then constructed in real time using the decoded labels, supporting both single-signer and dual-signer scenarios. A user-friendly graphical user interface (GUI) displays all results interactively and offers options for clearing or terminating the session. Even in noisy and multi-person settings, this architecture guarantees reliable, real-time ASL translation.

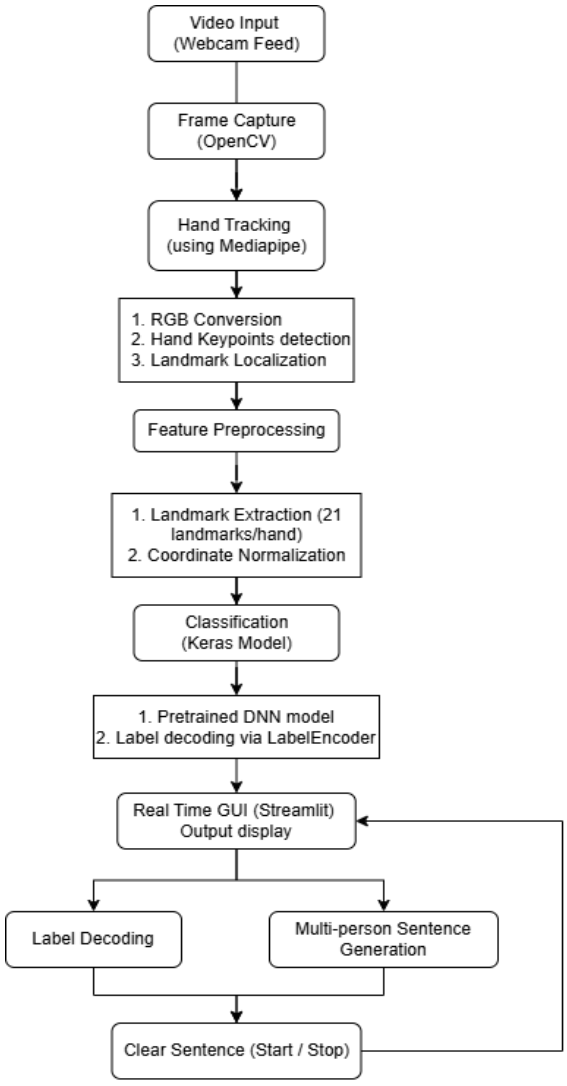

Figure 1: Real-Time ASL Translation System

2.2 Comprehensive Pseudocode

The complete system workflow used in the project to speed up the ASL finger spelling recognition process is described in the following pseudocode. The solution supports dual-user simultaneous recognition, applies stability filtering, and uses a trained deep learning model to translate gestures into text in real time. The pipeline includes video capture, landmark extraction, prediction using a pre-trained DNN Keras model, and two independent sentences—one for each detected hand. The code includes data collection, model training, evaluation, and GUI-based deployment using MediaPipe, TensorFlow/Keras, and Streamlit.

2.2.1 Data Collection and Labeling

To collect training data, the system uses a webcam interface and MediaPipe Hands module. Each hand gesture is labeled manually by the user during collection. A total of 63 features (21 landmarks with x, y, z coordinates) are recorded for each hand pose and appended with a label.

```
for i in range(21):
    columns += [f'x{i}', f'y{i}', f'z{i}']
columns.append('label')
```

Using OpenCV, the system captures frames, flips for mirror view, and processes them via MediaPipe. Upon pressing 's', the extracted landmark vector is saved to a CSV file with the associated label.

2.2.2 Dataset Merging

Data from multiple users are combined using Pandas:

```
df_you = pd.read_csv("landmarks_dataset.csv")
df_friend = pd.read_csv("landmarks_dataset_friend.csv")
df_combined = pd.concat([df_you, df_friend], ignore_index=True)
df_combined.to_csv("landmarks_dataset.csv", index=False)
```

2.2.3 Model Training

A deep neural network is built using TensorFlow/Keras. The dataset is label-encoded, split into training and test sets (80/20)

```
model = Sequential([
    Input(shape=(63,)),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(num_classes, activation='softmax')
])
```

The model is compiled and trained with Adam optimizer:

```
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

2.2.5 Single-User Real-Time Prediction

The system captures video frames, detects hand landmarks, reshapes input, and predicts using the trained model.

```
landmark_array = np.array(landmarks).reshape(1, -1)
prediction = model.predict(landmark_array)
predicted_label =
label_encoder.inverse_transform([np.argmax(prediction)])[0]
```

2.2.6 Dual-User Real-Time Prediction

MediaPipe is configured to detect two hands. Separate queues maintain predictions for each user. Prediction stability is ensured via majority voting (≥6 consistent frames).

```
if most_common == current_pred and count >= 6:
    if most_common == "SPACE":
        sentence += " "
    else:
        sentence += most_common
```

2.2.7 GUI-Based Dual Person ASL Sentence Builder

Streamlit integrates webcam feed, prediction display, and user interaction buttons:

Every 5 seconds, the system updates the sentence based on stable predictions.

In the final version, the system tracks two users independently with separate queues, timers, and prediction overlays. Both sentences are shown in real time.

```
cv2.putText(frame, f"Person 1: {predictions[0]}", (10, 50), ...)
cv2.putText(frame, f"Person 2: {predictions[1]}", (10, 100), ...)
sentence_box.markdown(f"### Person 1: {sentence_1}")
sentence_box.markdown(f"### Person 2: {sentence_2}")
```

2.3 Visual Output Snapshots

To begin to understand the effectiveness and interactivity of the proposed system, the following images display portions of the proposed solution. These snapshots were taken when the application was actively being executed.

A. Streamlit-Based UI for Real-Time Predictions

The Streamlit web application has a very simple and clean user interface where the live webcam feed is shown, the predicted letters, and a generated sentence. It allows to begin/end prediction, reset sentence, as well as to visualize direct feedback according to hand gestures.

Figure 2: Streamlit app displaying live webcam feed and detected character.

B. Dual-Hand Detection Capability (Optional Extension)

While the final version employs single-hand input, the architecture was tested also with twohand support by setting max_num_hands=2 in MediaPipe . This enables two individuals to sign at the same time and formulate distinct sentences, a capability that increases the robustness of the system.

Figure 3: Webcam image used for the prediction of two hands, where predictions on two different labels correspond to separate individuals.

C. Real-Time Sentence Generation

As users gesture letters and command words “SPACE”, “BACKSPACE” the sentence builder is instantly updated so that the communication is fluid.

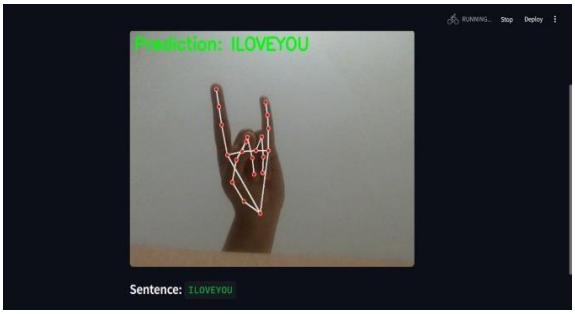

Figure 4: Predicted words and dynamically generated sentence displayed below the video.

3. Experimental Evaluation

3.1 Dataset and Preprocessing

A custom sign language capture system based on MediaPipe and OpenCV was used to selfrecord our dataset, which was centered on finger spelling gestures in American Sign Language (ASL). The collection pipeline extracted 63 features (21 points × 3 values) per sample from webcam footage by recording 21 hand landmarks per frame (x, y and z coordinates). Each piece of information was labeled, arranged, and stored in CSV format.

The system's ability to detect up to two hands simultaneously in real time makes it dependable in scenarios with multiple signers. Unlike traditional datasets, our approach ensures higher recognition accuracy in a range of scenarios by training on low-quality, multiperson videos.

3.2 Model Architecture

We made use of a deep learning architecture built on Long Short-Term Memory (LSTM) networks, which work especially well with sequential motion data. Among the architecture are:

- There are two LSTM layers:
  - 64 units first, with return_sequences=True
  - Second with 128 pieces
- Three dropout layers to avoid overfitting (rate = 0.3)
- 64 ReLU units in one dense layer
- Softmax-activated output layer for multiclass classification

3.3 Model Training and Testing

The self-recorded dataset was used to train the core DNN model, a deep learning classifier constructed with Keras (TensorFlow backend). The model predicts the matching ASL letter after receiving the hand landmark features.
For output that is easy to use, numeric predictions are mapped back to letter labels using label encoding (through scikit-learn).
To ensure real-time performance and practical usability, the system was tested in a live Streamlit app using standard webcam hardware.

3.3 Results and Analysis

The model performed well even in low-quality video and multi-signer situations, and it was able to classify ASL finger spelling gestures with high accuracy. To further increase the accuracy of real-time translation, frame stability filtering was used to lower prediction noise. This method proved to be more reliable and usable than conventional single-signer models, particularly in dynamic, real-world situations where lighting fluctuations and background distractions are frequent.

- Accuracy Over Epochs: After several epochs, the validation accuracy reaches a point above 95%, suggesting minimal overfitting and effective learning.

- Real-Time Performance: The system supports seamless, interactive translation for both single and multiple signers by maintaining real-time prediction speeds (processing each frame in less than 50 ms).
- Loss Graph: During the first 10 epochs, the loss graph displays a sharp decline in both train and validation loss; by epoch 50, there is a smooth convergence to zero. Given that validation loss closely follows training loss throughout, this suggests efficient learning without overfitting.

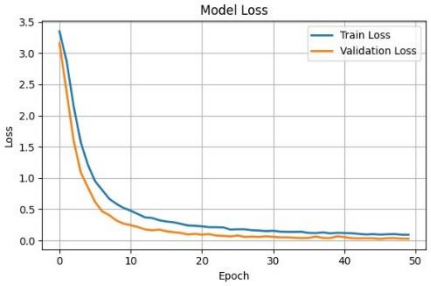

Figure 5: Model Loss Graph

- Accuracy Graph: The model's accuracy increases rapidly; after only 10 epochs, the validation accuracy reaches over 90%, and by the end of training, it has stabilized above 98%. Strong generalization is confirmed by the convergence of the train and validation accuracy curves.

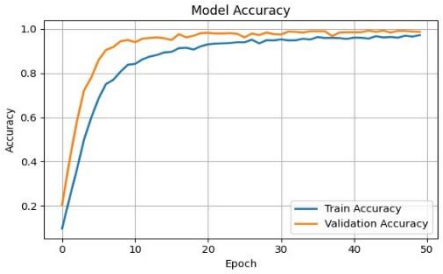

Figure 6: Model Accuracy Graph

- Confusion Matrix: With the majority of predictions on the diagonal, the confusion matrix shows excellent classification accuracy for all ASL gestures. Although there are occasionally misclassifications between signs that are visually similar, overall precision and recall are very good.

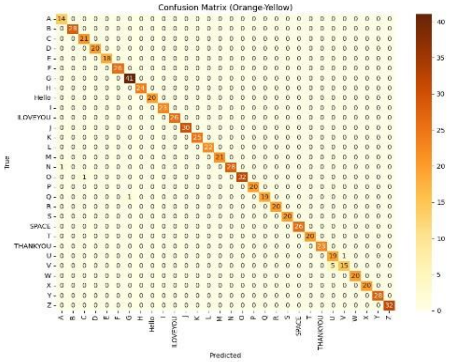

Figure 7: Confusion Matrix

4. Conclusion and Future Scope

The project was successful in implementing an ASL fingerspelling detection system in real time using computer vision and deep learning technologies. The high quality of the labels and consistency of the gestures, achieved by constructing our own dataset and training the model with our own script, made the system reliable. Mediapipe provided reliable landmark detection, offered strong real-time tracking of people, allowing the system to identify gestures from multiple signers at once. The ability of our solution to support multi-signers in combination with the frame stability filtering technology produced a more fluent and accurate real time translation than those obtained with single-signer systems.

We built our application to be used in the real world and accessible . It works well in normal webcams, and it has live visual output, making it practical to deploy in classrooms, public spaces, and among the general public for personal use. Its capacity to produce independent live sentences for each of the signers represents a solution to the challenges posed by group conversations, a capability that is typically not addressed by current solutions. The model was trained on low-quality and noisy video, which increased robustness and ensured reliable performance in the field under adverse or challenging conditions.

The possibilities for real-time sign language recognition are exciting and many. As machine learning and AI technologies develop, there are several directions in which to improve upon and extend this work:

- Support for Additional Sign Languages
  The system is currently adapted for ASL finger spelling. Future versions can even be developed to support the recognition of different sign languages, including BSL, ISL and others, in order to support global inclusion and accessibility.
- Educational and Training Tools
  This technology could thus be adapted as a sign language learning tool for deaf and hearing people and even interpreters, helping thus sign language literacy and awareness more widely.
- Advanced Natural Language Processing (NLP)
  The use of NLP would help this system, and others like it, to formulate better translations for complicated sentences, idioms and other such words/phrases that require more than just finger spelling to translate.
- Enhanced Gesture and Context Recognition
  The development of computer vision, motion tracking and wearable sensors will allow the systems of the future to identify more complex gestures, facial expressions and body language signals, thus making the translation even better.

With these future scopes, real-time sign language recognition systems can become even more inclusive, accurate, and integral to daily communication, truly empowering the deaf and hardof-hearing community and fostering a more connected society.

References:

[1] S. Ashwath and A. S. M, "Neural Network-based Real-Time Recognition of American Sign Language FingerSpelled Gestures: Bridging Communication Gaps," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp. 170-174, doi: 10.1109/ICSSAS57918.2023.10331682.

[2] C. Raghavachari and G. Shanmugha Sundaram, "Deep Learning Framework for Fingerspelling System using CNN," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 469-473, doi: 10.1109/ICCSP48568.2020.9182155.

[3] M. Chitampalli, D. Takalkar, G. Pillai, P. Gaykar, and S. Khubchandani, "Real Time Sign Language Detection," *International Research Journal of Modernization in Engineering, Technology and Science*, vol. 5, no. 4, pp. 2983–2987, Apr. 2023, doi: 10.56726/IRJMETS36648.

[4] A. Howal, A. Golapkar, Y. Khan, S. Bokade, S. Varma and M. V. Vyawahare, "Sign Language Finger-Spelling Recognition System Using Deep Convolutional Neural Network," 2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 2023, pp. 1-6, doi: 10.1109/ICNTE56631.2023.10146675.

[5] S. Bele, A. Shinde, K. Sharma, and A. Shinde, "Sign Language Recognition System Using Machine Learning," *International Journal of Innovative Research in Management, Pharmacy and Sciences*, vol. 11, no. 6, pp. 1–5, Nov.–Dec. 2023.

[6] M. Manoharan and P. Roy, "A comprehensive review of sign language recognition: Different types, modalities, and datasets," *arXiv preprint arXiv:2204.03328*, 2022. doi: 10.48550/arXiv.2204.03328

[7] T. Kim and B. Kim, "Techniques for detecting the start and end points of sign language utterances to enhance recognition performance in mobile environments," *Applied Sciences*, vol. 14, no. 21, p. 9199, 2024. doi: 10.3390/app14209199

[8] B. Alsharif, E. Alalwany, A. Ibrahim, I. Mahgoub and M. Ilyas, "Real-time American sign language interpretation using deep learning and keypoint tracking," *Sensors*, vol. 25, no. 7, p. 2138, 2025. doi: 10.3390/s25072138

[9] N. S. Dinh *et al.*, "Sign language recognition: A large-scale multi-view dataset and comprehensive evaluation," in *Proc. 2025 IEEE/CVF Winter Conf. Applications of Computer Vision (WACV)*, Tucson, AZ, USA, 2025, pp. 7887–7897. doi: 10.1109/WACV61041.2025.00766

[10] M. Al-Qurishi, T. Khalid and R. Souissi, "Deep learning for sign language recognition: Current techniques, benchmarks, and open issues," *IEEE Access*, vol. 9, pp. 126917–126951, 2021. doi: 10.1109/ACCESS.2021.3110912

[11] A. Bhavana, K. Shalini Reddy, Madhu and D. Praveen Kumar, "Deep Neural Network based Sign Language Detection," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1474-1479, doi: 10.1109/ICECA55336.2022.10009360.

[12] S. Sharma and P. Mahato, "Real-Time American Sign Language Recognition Using CNN," *International Journal of Research Publication and Reviews*, vol. 2, no. 5, pp. 377–380, May 2021. [Online]. Available: https://ijrpr.com/uploads/V2ISSUE5/IJRPR462.pdf