FYP-PROPOSAL

Project Title: **In Progress...**

Project Description:

A Real-Time Sign Language Translation and Subtitling System for Video Conferencing Platforms

Department: **Department of Computer Science & AI**

Project Manager: **Dr. Muhammad Ilyas**

Project Advisor: **Dr. Fahad Maqbool (Internal)**

Project Team:

1. **Muhammad Hassan Shahbaz (BSCS51F22S007)**
2. **Behlol Abbas (BSCS51F22R038)**

Submission Date: **October 10, 2026**

Table of Contents

1. Abstract

Video conferencing has become a fundamental tool for communication, yet it presents significant barriers for Deaf and hard-of-hearing individuals. While platforms like Zoom and Microsoft Teams offer features such as "Signer View" to prioritize interpreters' video feeds, they do not provide direct sign language translation into speech or subtitles, creating a dependency on human interpreters who may not always be available. The project addresses this gap by developing a software application that integrates with major video conferencing platforms to enable real-time sign language recognition, translation to speech, and live subtitling. The system leverages state-of-the-art Human Pose Estimation models, such as PoseNet, to extract skeletal key points, followed by temporal deep learning models (e.g., LSTM or Transformer) trained on large-scale datasets like World Level American Sign Language (WLASL) and Multi-View Sign Language (Multi-VSL). The translated text is converted to speech and injected into the audio stream, while also being displayed as subtitles for all participants. Key objectives include: (1) collecting and preprocessing sign language datasets; (2) training machine learning models for gesture recognition; (3) developing integration APIs for platforms like Zoom and Google Meet; (4) implementing real-time translation using speech synthesis and subtitle generation; and (5) testing for accuracy, latency, and usability. The expected outcomes include a functional prototype that enhances digital accessibility and inclusivity, academic advancements in real-time continuous sign language recognition, and industrial/social benefits such as empowering Deaf individuals in professional, educational, and social settings without mandatory interpreters. This reduces communication inequalities, promotes inclusive remote collaboration, and opens markets for accessible software solutions.

2. Background and Justification

Recent years have seen growing interest in making technology more accessible. Major corporations have initiated research and development in this domain, as evidenced by the works we have studied:

The communication barriers faced by Deaf and hard-of-hearing individuals in virtual meetings remain a critical challenge, as standard video conferencing tools like Zoom and Google Meet primarily cater to spoken language users. Sign language, used by millions worldwide, is often overlooked, leading to exclusion in professional, educational, and social interactions.

Recent advancements highlight growing interest in accessibility. For instance, Microsoft Teams' Sign Language Mode prioritizes the signer's video and uses detection models to elevate them as the active speaker [1]. Similarly, Google's 2020 research introduced real-time sign language detection for video conferencing, employing PoseNet for key points extraction and optical flow for motion analysis, while using an ultrasonic audio tone to recognize signers as speakers [2]. This approach is efficient and privacy-preserving bu focuses solely on detection, not translation.

Academic progress includes the 2024 work on video-based sign language recognition (SLR) using ResNet and LSTM networks for high-accuracy isolated sign recognition [3]. The 2023 sign.mt application enables multilingual translation between signed and spoken languages [4], and edge-device optimizations for real-time SLR have been explored [5]. Large-scale datasets like WLASL provide extensive vocabulary for word-level recognition [6], while Multi-View Sign Language (Multi-VSL) offers robust, multi-view data to improve generalizability across camera angles and backgrounds [7].

Despite these developments, gaps persist: most systems offer detection without full real-time translation to voice, lack seamless multi-platform integration, and struggle with continuous signing in noisy environments. The project enhances this by providing a modular system that recognizes signs, translates them into natural voice output and subtitles, and supports multiple sign languages (initially ASL, extensible to others like BSL) with low latency. This justifies the project by addressing scalability and interoperability, leveraging open-source datasets and APIs for deployment as a browser extension or SDK to promote broader adoption.

2.1 Justification and Differentiation:

While the existing works provide excellent foundations, there is a clear gap between detecting a signer and understanding what they are signing. Our project aims to bridge this gap.

Key Differences:

1. **Translation vs. Detection**: Unlike Microsoft's and Google's solutions [1],[2] that stop at identifying that a person is signing, our system will recognize what they are signing and translate it into text and speech (Focusing on English Language).
2. **Speech Synthesis**: We will integrate a Text-to-Speech (TTS) engine to convert the translated text into audible speech, allowing non-signing participants to hear the translated content.
3. **Integrated Subtitling**: The translated text will be displayed as real-time subtitles within the meeting for all participants, providing a dual-mode communication aid (audio and visual).
4. **End-to-End Solution**: Our aims to be a comprehensive assistive tool that handles the entire pipeline from video input to translated speech and subtitles, reducing reliance on external human interpreters for basic communication.

3. Research Methodology (Still in Progress)

To accomplish our objectives, we will adopt the following methodology:

Phase 1: Literature Review and Technology Stack Selection

- Conducted in depth analysis of relevant SLR research [1–7].
- **Selected technologies**: Python, PyTorch/TensorFlow for model development, MediaPipe for real-time pose estimation [2], and TTS libraries (e.g., Google gTTS or Pyttsx3).

Phase 2: Data Acquisition and Preprocessing

- Download and preprocess datasets like WLASL [6] (for word level models) and Multi-VSL [7] (for multi-view robustness).

- Use Open-CV and Media-Pipe for frame extraction, hand/pose landmark normalization, and data augmentation (e.g., rotation, scaling, brightness adjustments) to handle variations in lighting and angles.

Phase 3: Model Development and Training

- **Pose Estimation:** Extract 3D keypoints of body, face, and hands using MediaPipe Pose and Hands [2].
- **Feature Engineering:** Compute frame-to-frame optical flow for motion dynamics, normalizing for scale and framerate invariance [2].

Recognition Model: Train temporal models starting with LSTM for word level recognition on WLASL [6], progressing to Transformer architectures for sequential signs on Multi-VSL [7].
Translation and Synthesis: Use NLP models (e.g., BERT) for sentence formation from recognized glosses, followed by TTS for speech output.

Phase 4: System Integration and Application Development

- Develop a desktop application compatible with Zoom, Google Meet, etc.
- Capture video feeds (via virtual camera or screen share), perform real-time recognition, and output:
  - **Subtitles:** Inject text as overlays or display in windows.
  - **Speech:** Use virtual audio cables (e.g., VBAudio) to route synthesized speech to the conferencing microphone.
- Advantage WebRTC for browser based cross platform compatibility and SDK hooks for platform integration.

Phase 5: Testing and Evaluation

- **Model Evaluation:** Assess accuracy, precision, recall, and F1score on test sets (>90% target on benchmarks).
- **System Evaluation**: Perform unit tests, end-to-end simulations in mock meetings, and user trials with Deaf participants, iterating on feedback for edge cases like fast signing or occlusions.

Target latency: <100ms per gesture, using GPU accelerated environments (TensorFlow/PyTorch).

4. Project Scope

In-Scope

- Development of a real-time SLR model for a subset of vocabulary (100–200 common words/signs) in the prototype.
- Integration with at least one platform (e.g., Zoom) as proof of concept.
- Functionality: Real-time sign-to-text translation (English), text-to-speech conversion (English), and live subtitle display.
- Designed for single signers in well lit, clear background environments initially.

- Real-time detection and translation of common sign languages (ASL initially, extensible).
- Plugin/extension for Zoom, Google Meet, etc.
- Output in English audio and onscreen subtitles.
- Support for single signer video call scenarios.

Out-of-Scope

- Full complex grammar recognition, including non-manual markers (e.g., facial expressions) at production level.
- Simultaneous multi-signer support.
- Spoken to sign translation.
- Custom video conferencing platform development (focus on add on application).
- Robust handling of lowlight or cluttered environments in the prototype.
- Nonstandard/regional sign dialects beyond core vocabularies.
- Hardware specific optimizations (e.g., wearables).
- Standalone app; emphasis on integration module.

5. High level Project Plan

The project plan developed using MS Project, spans 6 months (October 2025 to March 2026) with key milestones. Below is a summarized Gantt-style overview:

Activity	Duration	Resources Assigned	Milestone/Submission Date
Literature Review & Requirements Gathering	2 weeks (Oct 1-14)	Project Manager, Team Lead	Requirements Document - Oct 14, 2025
Data Collection & Preprocessing	4 weeks (Oct 15-Nov 11)	All Group Members (4 people)	Dataset Ready - Nov 11, 2025
Model Training & Development	6 weeks (Nov 12-Dec 23)	Team Lead + 2 Members (ML expertise)	Prototype Model - Dec 23, 2025
Integration & API Development	4 weeks (Jan 1-28)	Project Manager + 2 Members (Software dev)	Integrated Module - Jan 28, 2026
Testing, Iteration & Documentation	4 weeks (Jan 29-Feb 25)	All Group Members	Final Testing Report - Feb 25, 2026
Final Presentation & Submission	2 weeks (Feb 26-Mar 11)	Project Manager	Complete Project - Mar 11, 2026

Resources include open-source tools (TensorFlow, OpenCV), university computing lab for training, and team of 4: Project Manager (planning), Team Lead (technical oversight), and two members (development/testing).

6. References

[1] Microsoft Support. "Use Sign Language Mode in Microsoft Teams Meetings." https://support.microsoft.com/enus/topic/usesignlanguagemodeinmicrosoftteamsmeetings8f88ed085a5e41dba190e0b30cab58ca

[2] Koller, O. (2020). "Developing Real-Time, Automatic Sign Language Detection for Video Conferencing." Google Research Blog. https://research.google/blog/developingrealtimeautomaticsignlanguagedetectionforvideoconferencing/

[3] (Note: Specific 2024 ResNet LSTM paper not cited in original; reference to general advancements in video based SLR.)

[4] Tze, M. et al. (2023). "sign.mt: Real-Time Multilingual Sign Language Translation Application." arXiv:2310.05064. https://arxiv.org/abs/2310.05064

[5] Yin, Y. et al. (2023). "Towards Real-Time Sign Language Recognition and Translation on Edge Devices." ACM Multimedia. https://yafengnju.github.io/YinPaper/MM2023Gan.pdf

[6] WLASL Dataset GitHub: https://github.com/dxli94/WLASL

[7] Dinh, L. et al. (2025). "Sign Language Recognition: A Large Scale Multi-View Dataset and Comprehensive Evaluation." WACV 2025. https://openaccess.thecvf.com/content/WACV2025/papers/Dinh_Sign_Language_Recognition_A_LargeScale_MultiView_Dataset_and_Comprehensive_Evaluation_WACV_2025_paper.pdf

Supervisor's Signature