Gestura: AI-Powered Mobile Application for Inclusive Communication

Raven Mott

CSCI 493 – Senior Project I

Personal Inspiration

My aunt is hard of hearing. My mother and her siblings took ASL classes to communicate with her.

This personal experience inspired me to create technology that makes ASL more accessible and prominent for everyone.

Current tools are limited, bulky, or fragmented - leaving communication barriers for deaf, hard of hearing, and multilingual users.

Related Work & Significance

Research into American Sign Language (ASL) recognition using deep learning and computer vision is an active field (Bantupalli & Xie, 2018). While some mobile applications explore real-time sign language translation (Firdousi & Team Semaphore, 2022), they often face limitations in accessibility, real-time performance, or comprehensive features for diverse user needs.

Gestura aims to advance beyond current offerings by integrating robust machine learning for dynamic sign interpretation within a user-friendly, responsive mobile interface. By emphasizing agile development practices (Miljkovic et al., 2024) and innovative data collection strategies crucial for AI integration (Roh, Heo & Whang, 2021), Gestura will provide a more accurate and accessible solution, making inclusive communication seamless and empowering for everyone.

The Problem To Solve

Limited Real-Time Translation

More portable, accessible solutions for ASL and multilingual communication in real-time.

Fragmented Solutions

Existing apps focus only on isolated gestures, not full conversations.

Technical Approach

01

ASL Recognition

CNN model with Firebase data contributions for gesture detection and learning.

02

Natural Language Processing

OpenAI integration for generating natural sentences from ASL gloss translation.

03

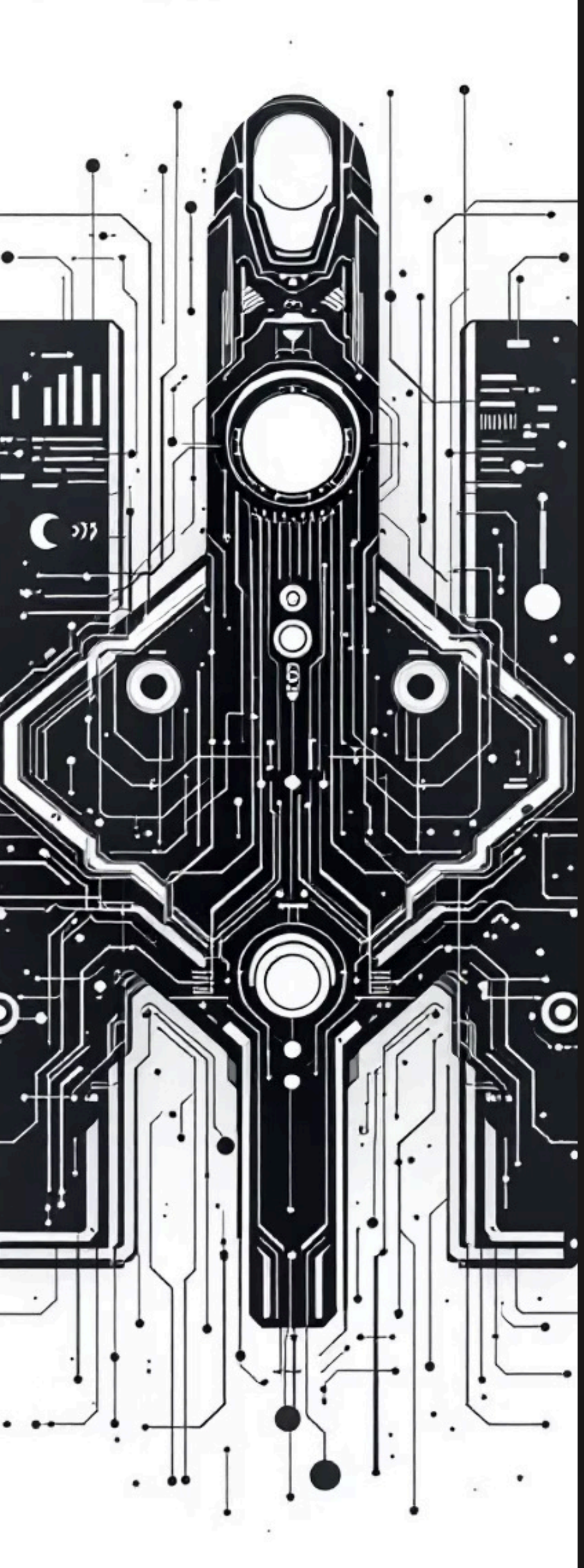
Mobile Application

Kotlin-based Android app for data capture, translation, and avatar visualization.

04

Cloud Integration

Model retraining capabilities with optional Meta Glasses integration.



Key Technologies and Tools

AWS “GenASL”

AWS GenASL is an open reference solution from AWS that turns **speech or text into ASL avatar videos**.

OpenAI

Integrated for its advanced Natural Language Processing (NLP) capabilities, transforming ASL gloss translations into fluid, natural-sounding sentences for enhanced communication.

Android Studio

The primary integrated development environment (IDE) for building the native Gestura mobile application, ensuring a robust and user-friendly interface.

Google Firebase

Utilized for real-time data storage, authentication, and cloud functions, supporting dynamic content updates and user management for the application.

Visual Studio Code

Employed for backend development, script management, and data processing tasks, facilitating model training and integration workflows.

Meta Glasses Integration

Explored as a future integration point to provide an immersive, hands-free ASL translation and learning experience through augmented reality.

Development Timeline

1 Aug 18 – Sep 26: Foundation & Core Skills

Deliverables: Comprehensive understanding of Kotlin and Android development; demonstrable mini-projects.

- **Technical Milestones:** Proficient in Android Studio, Kotlin syntax, UI/UX principles, and basic API interaction.
- **Detailed Tasks:** Complete CodePath AND102 coursework, implement several small Android applications focusing on UI layouts, data binding, and asynchronous operations.
- **Measurable Outcomes:** Successful completion of CodePath AND102 with an average project score of 90%+, and a functional personal project demonstrating core skills.

2 Sep 29 – Oct 24: ASL Recognition Prototype Development

Deliverables: Initial ASL gesture recognition prototype; integrated Firebase for data storage.

- **Technical Milestones:** Functional CNN model (e.g., MobileNetV2 based) capable of classifying 15 core ASL handshapes/motions. Firebase Realtime Database configured for gesture-label mapping.
- **Detailed Tasks:**
  a. Curate and augment initial ASL datasets for training (e.g., from publicly available sources like WLASL).
  b. Design and train a CNN model using TensorFlow/PyTorch for gesture classification.
  c. Set up Firebase project; implement initial data schemas for gesture storage and user profiles.
  d. Develop a basic Android module to capture camera frames and feed them to the CNN, displaying raw classification output.
- **Measurable Outcomes:** CNN model achieving >85% accuracy on a held-out test set for the initial 15 gestures; successful Firebase data writes/reads.

3 Oct 27 – Dec 1: Android Integration & Guided Capture

Deliverables: Alpha version of Gestura Android application with guided ASL capture and enhanced recognition.

- **Technical Milestones:** CNN model integrated into Android app via TensorFlow Lite; robust, user-friendly guided ASL capture interface.
- **Detailed Tasks:**
  a. Integrate the trained CNN model into the Kotlin Android application using TensorFlow Lite.
  b. Implement the "guided capture" feature, providing visual cues for users to contribute new gesture data.
  c. Optimize on-device inference for real-time performance.
  d. Conduct internal alpha testing to identify bugs in recognition flow and data submission.
- **Measurable Outcomes:** On-device gesture recognition accuracy within 5% of server-side model; successful user contribution of at least 50 new gesture variations; completion of midpoint testing with <5 critical bugs.

4 Jan 21 – Feb 28: NLP Integration & Cloud Retraining

Deliverables: End-to-end ASL gloss to natural language sentence generation; cloud-based model retraining pipeline.

- **Technical Milestones:** OpenAI API integrated for semantic sentence construction from ASL glosses; automated cloud functions for CNN model retraining.
- **Detailed Tasks:**
  a. Develop a module to translate sequences of recognized ASL gestures into an ASL gloss format.
  b. Integrate with OpenAI's API (e.g., GPT-3.5/4) to convert ASL glosses into grammatically correct English sentences.
  a. Implement basic multilingual translation support (e.g., using Google Translate API) for the generated sentences.
- **Measurable Outcomes:** Generated sentences are coherent and contextually appropriate >70% of the time; automated model retraining pipeline successfully triggered weekly; support for 2 additional languages.

5 Mar 3 – Mar 20: Implementation of lesser Features

Deliverables: Asl avatar implementation, language translation

Technical Milestones:

- First working version of ASL avatar integrated into the app
- Functional language translation module (text-to-text or text-to-speech)
- Initial internal testing of both features

Detailed Tasks:

- Implement ASL avatar API integration and validate animations
- Connect translation module to app UI (input/output pipeline)
- Conduct usability tests to ensure translations are accurate and responsive
- Debug early issues and refine avatar presentation quality
- Document feature functionality for later final documentation

Measurable Outcomes:

- Successful demonstration of ASL avatar within the app
- Accurate translation with >90% success rate in test cases
- Stable operation of both modules during typical use

6 Mar 20 - Presentation : Meta Glasses Integration & Finalization

Deliverables: Beta version with optional Meta Glasses integration; polished application for presentation.

- **Technical Milestones:** Proof-of-concept integration with Meta Glasses SDK for gesture input; final application optimization and hardening.
- **Detailed Tasks:**
  a. Explore Meta Glasses SDK for potential gesture data input or display integration. Implement a minimal feature (e.g., displaying translated text on glasses).
  b. Conduct final end-to-end testing, including stress testing and edge case scenarios.
  c. Address all identified bugs and performance bottlenecks.
  d. Prepare comprehensive documentation and create presentation materials for the final demo.
- **Measurable Outcomes:** Functional demonstration of Meta Glasses integration; application stability >99% during prolonged use; completion of all planned features for the final release.

Core Features

Core Translation

Gesture Recognition & Text Translation

Real-time recognition of American Sign Language gestures captured by your device's camera, instantly translating them into on-screen text.

Text-to-Speech for Recognized Signs

Converts recognized ASL signs and their translated text into spoken audio, enhancing accessibility and communication.

Animated ASL Avatar

A lifelike animated avatar translates spoken or typed language into ASL signs, providing reverse translation for comprehensive communication.

Multilingual Translation

Supports English, Spanish, Japanese, and more, with modular language packs for flexible expansion and diverse communication needs.

User Experience

User Authentication

Secure login options including Firebase email, Google, and Apple, ensuring personalized and protected user access.

Dark Mode & High-Contrast

System-aware dark mode with an optional high-contrast setting for optimal viewing comfort and accessibility in various environments.

Bottom Nav

For optimal App navigation

Training & Learning

Guided Gesture Training

Offers guided uploads to the cloud with on-device pre-checks for framing and lighting, facilitating accurate user contributions.

Contributions & Accuracy Dashboard

Provides users with a personal dashboard to track valid contribution counts and individual validation pass-rates, fostering engagement and improvement.

Technical Features

Model Update & Rollback

A dedicated button to sync with the latest AI model, displaying version information, changelog, and a rollback option for stability.

Offline Mode

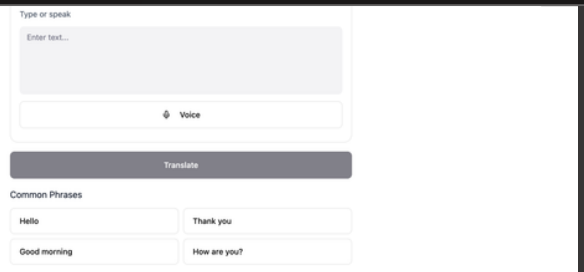
Enables functionality without an internet connection through cached AI models and queued translations/uploads, ensuring uninterrupted use.

Gestura Application Wireframe

This wireframe illustrates the core user interface of the Gestura mobile application, focusing on real-time sign language recognition and translation for an intuitive user experience.

Error uploading image.

https://www.figma.com/design/5H6ROAYBu9HovzsvvUVtWZ/Gestura?node-id=0-1&t=bxGaaTkxCkFL2fAH-1



Figma

ASL Translation App

Created with Figma

Empowering Inclusive Communication

Personal Mission

Driven by personal experience with my aunt and family's ASL journey.

Integrated Platform

Combining ASL recognition, multilingual translation, and avatar animation in one app.

Accessibility Goal

Making ASL more visible, accessible, and practical in everyday life for everyone.

References

1. Bantupalli, K., & Xie, Y. (2018). American Sign Language Recognition using Deep Learning and Computer Vision. 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141
2. Y. Roh, G. Heo and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 1 April 2021, doi: 10.1109/TKDE.2019.2946162
3. Miljkovic, K., Zivkovic, M., Bacanin, D., Simic, N. (2024). Agile Multi-tier Android Application Development With Firebase: Authentication, Authorization, and Profile Management. Paper presented at Sinteza 2024 - International Conference on Information Technology, Computer Science, and Data Science, doi:10.15308/Sinteza-2024-106-112
4. M. Firdousi and Team Semaphore, Synaera: Android application for sign language to text translation in real time. GitHub repository, 2022. [Online]. Available: https://github.com/MehrinFirdousi/Synaera-TeamSemaphore
5. Koishigawa, N. O. (2019). Machine learning projects for mobile applications. Packt Publishing.