# Final Project Report: Converting Spoken Language into Indian Sign Language Using ML and DL, Integrated with Flask

Prepared by: Mitravinda Resoju

## Executive Summary

This project aims to develop a system that converts spoken language into Indian Sign Language (ISL) using Machine Learning (ML) and Deep Learning (DL) techniques. The application is built with Flask to provide a web-based interface, allowing for interactive, real-time language translation into sign language, enhancing accessibility for the hearing-impaired community. Key features include speech recognition, gesture generation, and a responsive web interface to facilitate a seamless user experience.

## 1. Tools and Technologies Utilized

- Programming Language: Python  
- Libraries and Frameworks: TensorFlow, PyTorch, Flask, OpenCV  
- APIs and Services: Google Speech-to-Text API for speech recognition  
- Testing and Debugging: Pytest, Postman for API testing  
- Documentation and Design: Nimbus Platform for project tracking and documentation

## 2. Project Overview

The system leverages ML and DL algorithms to translate spoken language into ISL. Speech input is processed using a speech-to-text model, which then maps the text to ISL gestures. This project follows best practices in ML model training, data preprocessing, and API development, ensuring a reliable, user-friendly translation experience.

## 3. System Requirements

### Software Requirements

- Flask (latest version)  
- TensorFlow, PyTorch libraries for ML and DL models  
- Compatible IDEs: PyCharm, Visual Studio Code

### Hardware Requirements

- Minimum RAM: 8GB  
- Processor: Quad-core or better  
- Operating System: 64-bit (Linux, Windows)

## 4. Functional Requirements

- Speech Recognition: Accurately transcribes spoken language into text.  
- Gesture Mapping: Matches text input with ISL gestures.  
- ISL Display: Converts text to ISL gestures in real time on a web interface.  
- User Management: Tracks user interactions and provides personalized feedback.

## 5. User Interface Requirements

- Web Interface: Displays ISL gestures corresponding to spoken input.  
- Real-Time Feedback: Provides immediate translation of spoken input.  
- Error Handling: Clear, user-friendly messages for invalid or unclear input.

## 6. Inputs and Outputs

- Inputs: Spoken language captured via microphone.  
- Outputs: Visual display of ISL gestures on a web interface.

## 7. System Subcomponents

- Speech-to-Text Module: Handles speech recognition using an API.  
- ML Model for Gesture Recognition: Translates text to ISL gestures.  
- Web Interface: Built with Flask to display gestures in real time.

## 8. Potential for Other Applications

- Educational Tools: Enhances ISL learning for beginners.  
- Accessibility Software: Integrates with devices to assist the hearing-impaired community.  
- Customer Service Kiosks: Allows for easy communication in public spaces.

## 9. Test Case Design

Test Case ID  
Function  
Expected Outcome  
  
TC001: Speech Recognition  
Successfully transcribes spoken input into text  
  
TC002: Gesture Mapping  
Text accurately converted into corresponding ISL gesture  
  
TC003: Display Gesture  
ISL gesture displayed correctly on web interface  
  
TC004: Error Handling  
Clear error message for unsupported language input

## 10. Future Enhancements

- Expanded Language Support: Incorporate additional regional languages.  
- Real-Time Gesture Feedback: Optimize response time for smoother interactions.  
- Mobile Application: Develop a mobile version for on-the-go accessibility.

## 11. References

- Google Speech-to-Text API Documentation  
- Flask and Web Development Resources  
- Machine Learning Algorithms for Gesture Recognition  
- Indian Sign Language Resources and Documentation

## 12. Project Reflection

### Technical Challenges Encountered

- Model Accuracy: Balancing accuracy and real-time processing for gesture recognition.  
- Error Handling: Managing unrecognized input and ensuring graceful system response.  
- Data Mapping: Aligning Indian Sign Language gestures with spoken language data.

### Software Engineering Insights

Applying modular design and efficient error handling enhanced project reliability and user satisfaction. This project reinforced my knowledge in ML, DL, and Flask development.