Software Requirements Specification

for

SilentSpeak

Prepared by Shitiz Rajvanshi

KIET Group of Institutions, Ghaziabad

27 March 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Sign language is an important mode of communication for people with hearing or speech impairments. However, it can be challenging for those who do not know sign language to communicate effectively. This is where sign language detection using hand gesture recognition through machine learning comes into play. This Model is designed to help those with hearing and speaking disabilities. It provides text to sign, image to sign, voice to sign and sign to text features. Sign language models can be used in educational settings to help people learn sign language. This can be especially helpful for people who are learning sign language as a second language or for parents who want to communicate with their deaf or hard of hearing child.it can make digital content more accessible for people who are deaf or hard of hearing. This can include videos, online courses, or other digital content that would otherwise be inaccessible to individuals who rely on sign language to communicate. can also be used in research to study the structure and use of sign language, as well as to analyze patterns in signed communication. This can lead to a better understanding of sign language and how it is used in different contexts. Sign language detection using machine learning is an exciting and rapidly evolving field of research and development. Machine learning algorithms can be trained to detect and interpret sign language gestures from video feeds or sensor data, enabling more accessible communication between sign language users and non-sign language users. Sign language detection using machine learning has the potential to greatly improve accessibility and inclusivity for individuals who are deaf or hard of hearing, as well as to advance research and education in the field of sign language. As machine learning algorithms continue to improve, we can expect to see even more sophisticated sign language detection models that can recognize a wider range of gestures and adapt to the unique communication needs of individual users.

1.2 Document Conventions

Some essential conventions for ensuring consistency and accuracy in the analysis of various expressions which can help to identify patterns and trends that may help others to understand to communicate.

Data collection protocols: the data of several expressions is collected at different forms such as text, images and voice.

1.3 Intended Audience and Reading Suggestions

The deaf and dumb model aims to provide a solution for people who are hearing or speech impaired. The problem faced by such individuals is the difficulty in communicating effectively with others, which often leads to social isolation and a lack of accessibility to many services. The model seeks to bridge this gap by providing a platform for users to communicate through text, images, and videos.

1.4 Product Scope

Computer vision and machine learning uses a camera to capture hand gestures to facilitate communication for people who are deaf or hard of hearing. The system translates the detected hand gestures into corresponding sign language phrases displayed on a screen, making communication more effective and empowering individuals with hearing impairments. This system represents a significant step forward n creating a more inclusive society.

2 Overall Description

1.1 Product Perspective

The project aims to develop a system for person with hearing and speaking disabilities. From a product perspective, a sign language detection model can be developed as a software tool or an application that utilizes machine learning algorithms to detect and interpret sign language gestures. Such a model could be designed to work with various input sources, such as video feeds or sensor data from wearable devices, and could potentially be integrated into existing software applications or devices. The main benefit of a sign language detection model is that it would enable more accessible communication between people who use sign language and those who don't. For example, it could be used to provide real-time translation of sign language gestures into spoken language or written text, or it could be used to generate captions or subtitles for video content. This could greatly improve accessibility and inclusivity for individuals who are deaf or hard of hearing. Developing a sign language detection model using machine learning involves collecting and labeling large datasets of sign language gestures, as well as training and fine-tuning machine learning algorithms to accurately recognize and interpret these gestures. Once trained, the model can be deployed in various applications, such as real-time translation, video captioning. In addition to improving accessibility, a sign language detection model could also have significant applications in education, healthcare, and entertainment. For example, it could be used to develop new learning tools for sign language education, or it could be integrated into telemedicine platforms to enable better communication between healthcare providers.

Product Functions

- Collect data on sign corresponding to their text.
- Training model using CNN for each of the feature for sign to text as well as for text to sign.
- Foer speech to sign firstly voice is converted using Python library and thereafter text to sign conversion is done.

- For image to sign firstly the text from the image would be extracted and thereafter converted into corresponding sign.
- All the four features would be integrated into the whole project and helps the user to understand and learn about sign language.

1.2 User Classes and Characteristics

Some potential user classes and their characteristics for our system that helps in eatablishing communication between everyone is:

- Deaf or hard of hearing individuals: This user class would be the primary beneficiaries of a sign language detection project. Characteristics of this user class include the need for visual communication, fluency in sign language, and a preference for communication in sign language over spoken language.

- Interpreters or translators: This user class would be responsible for using the sign language detection system to translate sign language into spoken language or written text for non-sign language users. Characteristics of this user class include fluency in both sign language and spoken language, as well as the ability to interpret sign language gestures accurately.

- Developers or technical experts: This user class would be responsible for developing and maintaining the sign language detection system. Characteristics of this user class include expertise in machine learning algorithms, software development, and computer vision.

- Educators: This user class would be interested in using the sign language detection system as a teaching tool for sign language education. Characteristics of this user class include experience in teaching sign language and an understanding of the learning needs of deaf or hard of hearing students.

- Healthcare providers: This user class would be interested in using the sign language detection system to communicate with deaf or hard of hearing patients. Characteristics of this user class include an understanding of the unique needs and communication challenges of deaf or hard of hearing patients.

1.3 Operating Environment

Operating environment for a system that analyzes expression behavior to convert text to sing, image to sign, voice to sign, sign to text:

1. Hardware Requirements:

- Mobile devices or telematics devices with camera, scanner, to collect data.

2. Software Requirements:

- Operating system: The system should be compatible with popular mobile operating systems such as Android and iOS, as well as desktop operating systems such as Windows and macOS.
- Database management system: The system should use a database management system (DBMS) to store and manage driver behavior data.
- Analytics software: The system should use machine learning algorithms and other data analysis tools to analyze driver behavior data and generate safety scores and performance metrics.
- Web application or mobile application: The system should have a user interface that allows drivers, fleet managers, insurance companies, and regulators to access safety scores and performance metrics.

3. Network Requirements:

- The system should be accessible through an internet connection.
- The system should have secure communication protocols in place to protect clear view of image to observe appropriate behavior data and user information.

4. Environmental Requirements:

- The system should be designed to work in a variety of environments .
- The system should be designed to handle a large volume of data and user requests to ensure scalability and availability.

1.4 Design and Implementation Constraints

Some potential design and implementation constraints for this system:

• Data collection: The first step is to collect a large and diverse dataset of sign language gestures. This dataset should include a range of sign languages and variations of individual signs. Additionally, it is important to label the data accurately to enable supervised learning.

• Preprocessing: Once the data is collected, it needs to be preprocessed to remove noise, augment the data, and prepare it for training. This step may involve techniques such as normalization, denoising, and data augmentation.

• Model training: The next step is to train a machine learning model to detect and interpret sign language gestures. The choice of algorithm will depend on the nature of the data and the task at hand. For example, CNNs are often used for image-based data, while RNNs are better suited for sequential data.

• Model evaluation: After the model is trained, it needs to be evaluated to ensure that it is accurate and reliable. This step may involve cross-validation techniques, testing on a held-out dataset, and measuring performance metrics such as accuracy, precision, and recall.

• Deployment: The final step is to deploy the model in a real-world application. This may involve integrating it into existing software or hardware platforms, optimizing it for specific devices or use cases, and ensuring that it is user-friendly and accessible

• Hardware limitations: Sign language detection models require significant computational resources to process and analyze video feeds or sensor data in real-time. However, some devices, such as smartphones or tablets, may have limited processing power or memory, which can constrain the implementation of the model.

• Algorithm complexity: Sign language detection models can be complex, requiring advanced machine learning algorithms such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs). These algorithms can be computationally intensive and require significant expertise in machine learning and computer vision.

• Privacy and ethical considerations: Sign language detection models may collect and analyze sensitive data, such as biometric information or user location. As such, the implementation of these models must consider privacy and ethical considerations, such as data encryption, secure storage, and user consent.

• Noise and variability: Sign language is a highly expressive language that includes a range of gestures and movements. However, this variability can also introduce noise and errors into the data, making it challenging to accurately interpret sign language gestures.

1.5 User Documentation

- To use the system, follow these steps:
    1. Open the web application or log in to the telematics device.
    2. The system will automatically start collecting data on your driving behavior, including speed, acceleration, braking, and other parameters.
    3. You can view your safety scores and performance metrics in real-time or at regular intervals.
    4. You can also receive alerts and notifications about unsafe driving behavior or other issues that may impact your safety scores or performance metrics.
- To access additional features or settings, consult the user manual or contact customer support for assistance.

1.6 Assumptions and Dependencies

Assumptions:

- The system assumes that the data collected from GPS, accelerometer, and other sensors is accurate and reliable.
- The system assumes that the drivers are using the mobile or telematics devices consistently and as instructed.
- The system assumes that the drivers are not tampering with the mobile or telematics devices or the data they collect.
- The system assumes that the users have access to a reliable internet connection and a compatible device to access the system.

Dependencies:

- The system depends on mobile or telematics devices to collect driver behavior data.
- The system depends on a database management system to store and manage driver behavior data.
- The system depends on machine learning algorithms and other data analysis tools to analyze driver behavior data and generate safety scores and performance metrics.
- The system may depend on third-party software and services for specific functions such as secure communication protocols or geocoding services.
- The system may depend on regulations and industry standards that define the safety scores and performance metrics used by the system.

3. External Interface Requirements

3.1 User Interfaces

1. Web-Based Interface

- The web-based administrative interface allows administrators or other authorized users to access the system from a web browser and perform advanced functions such as data analysis and management.
- The administrative interface typically includes a dashboard that displays key performance indicators and trends for the entire fleet or selected drivers.
- The interface may also include tools for data visualization, analysis, and reporting,

as well as features for managing driver profiles, setting safety targets, and generating alerts and notifications.

- The interface also allows drivers to access the system and view their safety scores and performance metrics in real-time.
- This interface typically includes a dashboard that displays the driver's safety score and performance metrics, such as average speed, acceleration, and braking.

.

3.2 Hardware Interfaces

1. Telematics Device

- The system will require a telematics device to be installed in each vehicle to collect data on driving behavior.
- The telematics device should be compatible with the vehicle's make and model and should include sensors such as GPS, accelerometers, and gyroscopes.
- The telematics device should be able to communicate with the central system to transmit data in real-time or on a regular basis

.

2. Mobile Devices

- Drivers may use mobile devices such as smart phones or tablets to access the system and view their safety scores and performance metrics.
- The system should be compatible with popular mobile platforms such as iOS and Android.
- The mobile devices should be able to communicate with the central system over a wireless network, such as Wi-Fi or cellular data.

3. Web Server

- The system will require a web server to host the central system and allow authorized users to access the system from a web browser.
- The web server should meet the system requirements for software and hardware, including processing power, memory, and storage.
- The web server should be able to communicate with the telematics devices and other hardware components over a wireless network.

4. Sensors

- In addition to the sensors included in the telematics device, the system may require additional sensors to collect data on driving behavior or other factors that affect safety.
- The system may include sensors to measure vehicle speed, tire pressure, or weather conditions and cameras.
- The sensors should be compatible with the telematics device or other hardware components and should be able to communicate with the central system.

3.3 Software Interfaces

1. Database Management System

The system will require a database management system to store and manage the data collected from the telematics devices and other sensors.

2. Web Application Framework

The system will require a web application framework to build the central system that allows authorized users to access and interact with the system.

3. Programming Languages

- The system requires programming languages to develop the various software components of the system, including the telematics device software, the central system, and any additional software components. Programming languages include Python, Java, and JavaScript.
- The system will require APIs and communication protocols to allow different software components to communicate with each other.

4. System Features

4.1 Safety scoring:

The system should be able to generate a safety score for each driver based on their driving behavior, taking into account factors such as speed, acceleration, braking, lane changing, etc.

4.2 Driver feedback:

The system should provide feedback to the driver on their safety score and driving behavior, highlighting areas where they need to improve and providing suggestions for how to improve their driving skills.

4.3 Real-time monitoring:

The system should be able to monitor driving behavior in real-time, collecting data from the telematics device and other sensors.

4.4 Integration with other systems:

The system should be able to integrate with other systems or tools, such as fleet management systems or telematics platforms, to provide a comprehensive solution for driver safety and performance monitoring.

4.5 Data security:

The system should include robust security features to protect sensitive data, such as driver profiles and performance metrics, from unauthorized access or theft.

5. Other Nonfunctional Requirements

5.1 Performance:

The system should be able to process and analyze large volumes of data quickly and accurately, with minimal latency or delay.

5.2 Reliability:

The system should be highly reliable and available, with minimal downtime or service disruptions.

5.3 Scalability:

The system should be scalable, able to handle increasing amounts of data and users as the system is adopted by more drivers and fleets.

5.4 Maintainability:

The system should be easy to maintain and update, with well-documented code and modular architecture.

5.5 Usability:

The system should be easy to use and navigate, with an intuitive user interface that requires minimal training for drivers and fleet managers.

5.6 Accessibility:

The system should be accessible to users with disabilities, such as those who use assistive technologies or have limited mobility.

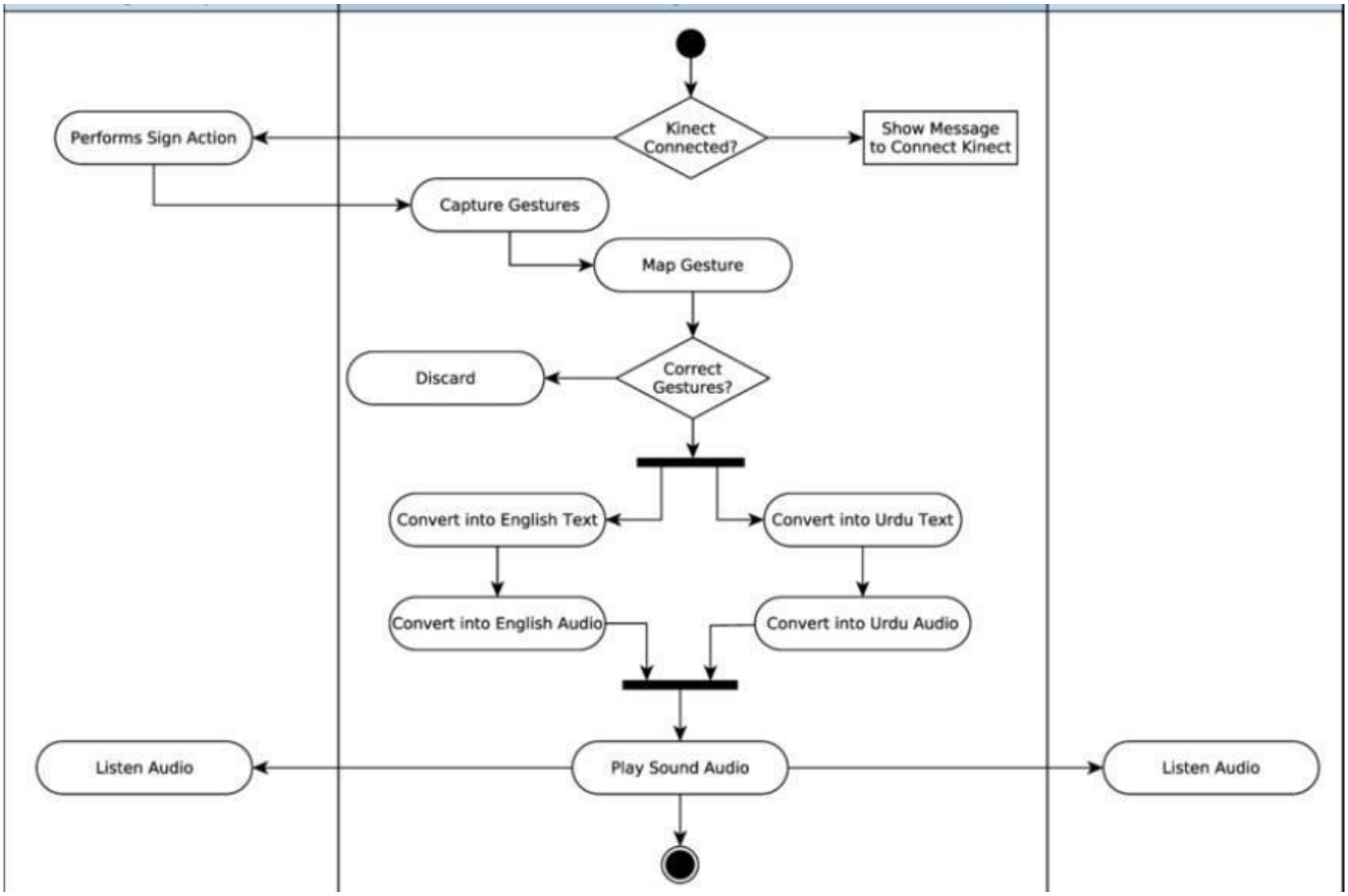
5.7 Security:

The system should have robust security features, such as encryption, access controls, and intrusion detection, to protect sensitive data and prevent unauthorized access or theft.

5.8 Compliance:

The system should comply with relevant regulations and standards, such as data protection

Appendix B: Analysis Models



Appendix A: Glossary

1. Telematics: A technology that allows for the transmission of data over long distances, typically used in the context of vehicles to track their location, speed, and other parameters.
2. Safety score: A numerical value that indicates a driver's safety performance, typically based on factors such as speed, acceleration, braking, and cornering.
3. Driver behavior analysis: The process of analyzing a driver's behavior behind the wheel, typically using data collected from telematics devices and other sensors.
4. Risk alert: An alert generated by the system when it detects potentially risky driving behavior, such as sudden braking or swerving.
5. Calibration: The process of adjusting or verifying the accuracy of a device or sensor, typically done to ensure that the data it provides is reliable and accurate.
6. Fleet management system: A software platform that allows fleet managers to track and manage their vehicles, drivers, and other assets.
7. Data analytics: The process of using statistical and computational techniques to analyze large datasets and extract meaningful insights from them.
8. User profile: A collection of data and settings that describe a specific user of the system, such as a driver or fleet manager.
9. Performance metrics: Measurements of a system's performance, typically based on factors such as speed, accuracy, and resource usage.
10. Access controls: Security features that limit access to certain functions or data within the system, typically based on user roles or permissions.

1.7 References

- L. Erhan et al., "Analyzing Objective and Subjective Data in Social Sciences: Implications for Smart Cities," in IEEE Access, vol. 7, pp. 19890-19906, 2019, doi: 10.1109/ACCESS.2019.2897217.
- T. Wang, Y. Chen, X. Yan, W. Li and D. Shi, "Assessment of Drivers' Comprehensive Driving Capability Under Man-Computer Cooperative Driving Conditions," in IEEE Access, vol. 8, pp. 152909-152923, 2020, doi: 10.1109/ACCESS.2020.3016834.
- Sivaramakrishnan R Guruvayur and Dr. Suchithra R, "A Detailed Study on Machine Learning Techniques for Data Mining" IEEE International Conference on Trends in Electronics and Informatics, 11-12 May 2017, IEEE Xplore - 22 February 2018, pp. 1187-1192.
- A. Kashevnik, I. Lashkov and A. Gurtov, "Methodology and Mobile Application for Driver Behavior Analysis and Accident Prevention," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 6, pp. 2427-2436, June 2020, doi: 10.1109/TITS.2019.2918328.
- Z. Deng et al., "A Probabilistic Model for Driving-Style-Recognition-Enabled Driver Steering Behaviors," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 3, pp. 1838-1851, March 2022, doi: 10.1109/TSMC.2020.3037229.
- G. Markkula, R. Romano, A. H. Jamson, L. Pariota, A. Bean and E. R. Boer, "Using Driver Control Models to Understand and Evaluate Behavioral Validity of Driving Simulators," in

IEEE Transactions on Human-Machine Systems, vol. 48, no. 6, pp. 592-603, Dec. 2018, doi: 10.1109/THMS.2018.2848998.

- A. AbouOuf, I. Sobh, M. Nasser, O. Alsaqa, O. Elezaby and J. F. W. Zaki, "Multimodel System for Driver Distraction Detection and Elimination," in IEEE Access, vol. 10, pp. 72458-72469, 2022, doi: 10.1109/ACCESS.2022.3188715.
- A. Ezzouhri, Z. Charouh, M. Ghogho and Z. Guennoun, "Robust Deep LearningBased Driver Distraction Detection and Classification," in IEEE Access, vol. 9, pp. 168080- 168092, 2021, doi: 10.1109/ACCESS.2021.3133797.