Development Document

# Project: Sign Language Translator - SignSense

#### Team: Team Yoda

#### Date: January 24, 2024

## **I. Technical Stack**

**Programming Languages:**

* **Python** (primary language for backend development and model training)
* **Dart** (for frontend development and user interface interactions)
* **Frameworks and Libraries:**
* **TensorFlow:**

TensorFlow is an open-source machine learning framework developed by Google that offers flexibility, scalability, and a comprehensive ecosystem, making it a popular choice for building and training machine learning models across diverse applications and platforms.

It will be used to accurately interpret and translate sign language gestures into meaningful digital interactions, fostering inclusivity and accessibility in technology.

* **Keras:**

While Keras is integrated into TensorFlow, it can also be used as a standalone high-level neural networks API. It's user-friendly and facilitates rapid prototyping of neural network architectures.

Convolutional Neural Networks are well-suited for image recognition tasks, making them an ideal choice for detecting hand gestures in sign language. CNNs can learn hierarchical features and spatial relationships in images, enabling accurate identification of key elements in sign language signs.

**Backend:**

* **Firebase**(Firebase is a robust backend-as-a-service (BaaS) platform developed by Google)
* **TensorFlow** (deep learning framework for model development)
* **NLTK** (natural language processing library)
* **Pandas and NumPy** (data manipulation and analysis libraries)

**Frontend:**

* **Flutter**(Used for open-source UI software development)

**Cloud Platform**:

* Google Cloud Platform (potential use for cloud storage, compute resources, and deployment)

**II. AI Model Architecture**

**Chosen Model:**

The Convolutional Neural Network (CNN) architecture crafted for sign language identification focuses on capturing the distinctive visual elements inherent in sign gestures. Starting with convolutional layers, the network extracts spatial features crucial for discerning hand shapes, movements, and facial expressions unique to sign language.

Pooling layers streamline this information, preserving essential details. Fully connected layers then amalgamate these learned features, mapping them to specific sign gestures. The network's design is tailored to the intricacies of sign language, ensuring a nuanced understanding of the visual cues that convey meaning, ultimately enabling accurate and efficient identification of sign language expressions.

**III. Key Functionalities**

Our model is designed for sign language detection using Convolutional Neural Networks (CNNs):

**Architecture Overview:**

**1. Input Data:**

The model takes as input sequences of images or frames representing sign language gestures. These images typically capture the hand movements, positions, and relevant facial expressions involved in signing.

**2. Convolutional Layers:**

The initial layers of the CNN are convolutional layers. These layers use filters to convolve over input images, extracting spatial features and patterns associated with different signs. The depth and number of filters can be adjusted to capture both simple and complex features.

**3.Training Process:**

The model is trained using labeled datasets of sign language gestures. During training, the model adjusts its weights based on the error between predicted and actual gestures using optimization techniques like stochastic gradient descent.

**4. Real-time Inference:**

Once trained, the model can be deployed for real-time inference, allowing it to interpret and predict sign language gestures in live video streams or recorded videos.

**User Interface:**

* Provide a user-friendly interface for:
* Viewing analysis results
* Helps users learn about sign language

**Sample Links:**

* **Github Repository Link:** https://github.com/ultralytics