USHA PRAVIN GANDHI COLLEGE OF ARTS, SCIENCE AND COMMERCE

SIGN LANGUAGE RECOGNITION SYSTEM

PROJECT BY

JAINIL SHAH

UNDER THE GUIDENCE OF

DR. MANISHA DIVATE

INTRODUCTION

The exchange of ideas, emotions and information is an integral part of human interaction.
However for individuals, with hearing disorders sign language and spoken language can create a barrier to communication making it challenging to connect or converse.
That's where Sign Language Recognition systems come into play as a solution. Our project focuses on developing an accurate and efficient sign language detection system

OBJECTIVES

Enhancing Communication

Enable seamless communication for individuals with hearing impairments.

Break Communication Barriers

Remove obstacles in communication between deaf and hearing individuals.

Encourage Independence

Empower deaf individuals to express themselves and make choices independently.

Dataset Collection

An extensive and diverse dataset of sign language gestures will be assembled, serving as the goal.

Deep Learning Model Crafting

With TensorFlow, the goal is to create a robust and adaptable model capable of accurately identifying various sign language.

Real-time Gesture Recognition

With the ability to interpret gestures in real time, interactions become smoother and more spontaneous

User-friendly Interface Development

Prioritizing user experience, this goal aims to create an interface that is intuitive and easy to use.

TECHNOLOGIES IN USE

Python

Programming Language

The project primarily relies on Python as the programming language for its flexibility, extensive libraries, and compatibility with machine learning frameworks.

OpenCV

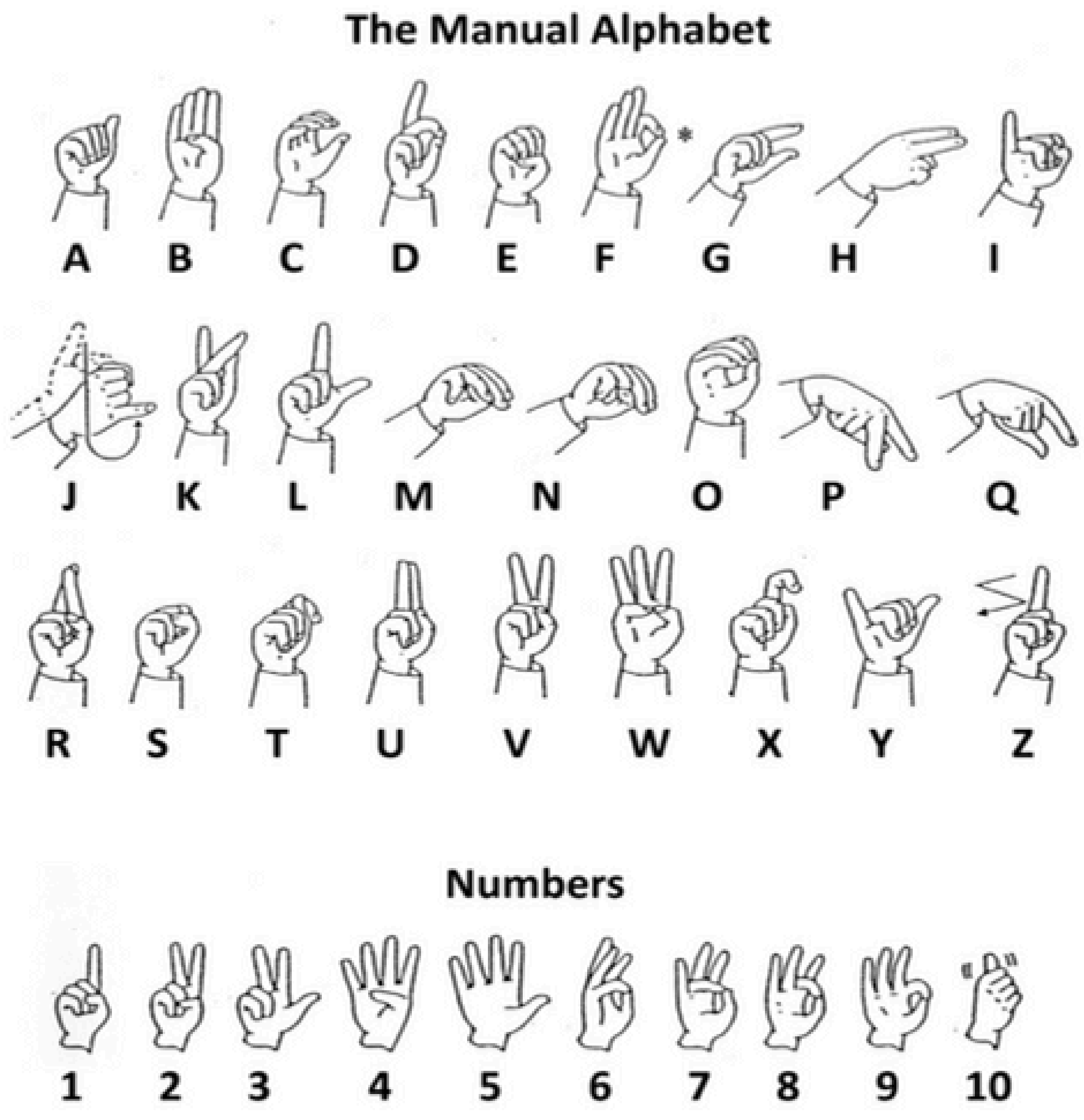
Computer Vision Library

Open Source Computer Vision Library (OpenCV) is used for image and video processing, making it an indispensable component for capturing and processing sign language gestures.

TensorFlow

Deep Learning Framework

This deep learning frameworks is essential for training and deploying machine learning models. In this project, TensorFlow will be employed for gesture recognition and classification



AMERICAN SIGN LANGUAGE (ASL)

ASL, or American Sign Language, is a complete, natural language used by Deaf and hard-of-hearing people in the United States of America and parts of Canada. ASL is a complete and organized visual language that is expressed by employing both manual and nonmanual features. Besides North America, dialects of ASL and ASL-based gestures or even ASL as a whole are used in many countries around the world, including much of West Africa and parts of Southeast Asia. ASL is also widely learned as a second language

WHY ASL?

When it comes to training sign language recognition models, choosing the right sign language is crucial. American Sign Language (ASL) stands out as a preferred choice for several reasons, making it an ideal language for training machine learning models in the field of sign language recognition.

• **Simplicity of Single-Hand Gestures:** ASL predominantly uses single-hand gestures, simplifying the training process.

• **Global Recognition and Usage:** ASL is recognized and used globally, making it a valuable language for international applications. ASL is also used in the Philippines, Puerto Rico, Dominican Republic, Canada, Mexico, much of West Africa and parts of South Asia.

• **High Accuracy in Gesture Detection:** Since most of ASL gestures being single handed it makes it easier for the model to study the gestures efficiently resulting in a highly accurate model.

Incorporating ASL in sign language recognition models not only simplifies the training process but also ensures the applicability, accuracy, and cultural sensitivity of the technology.

TRAINING OF THE MODEL

featuring Teachable Machine

Step 1: Gather and Prepare Your Dataset

- Collect a dataset of images relevant to your classification task.
- Organize the images into different classes or categories.
- Ensure images are of consistent size and quality for accurate training.

Step 2: Access Teachable Machine

- Visit the Teachable Machine website to access the platform.

Step 3: Create a New Project

- Click on "Get Started" to create a new project.

Step 4: Upload and Label Your Images

- Upload your prepared dataset to Teachable Machine.
- Label each class/category accurately.
- Ensure a balanced representation of classes for effective training.

Step 5: Train Your Model

- Click on the "Train Model" button to start training.
- Teachable Machine will use your dataset to train a CNN model in the background.
- Wait for the training process to complete. This may take some time depending on the size of your dataset and complexity of your classes.

TRAINING OF THE MODEL

featuring Teachable Machine

Step 6: Test and Evaluate Your Model

- After training, use the "Test" tab to test your model's predictions in real-time.
- Evaluate the model's accuracy and adjust your dataset or training parameters if necessary.

Step 7: Export Your Model

- Once you are satisfied with your model's performance, click on the "Model" tab.
- Choose the export option based on your needs (TensorFlow.js, Keras, or MobileNet).
- Download the exported model files.

Step 8: Integrate Your Model

- Integrate your trained model into your desired application or platform using appropriate frameworks and libraries.
- Use the exported model for real-time predictions on new data.

SYSTEM REQUIREMENTS

Functional Requirements

The functional requirements describe what the Sign Language Detection System must be capable of doing. These include:

• **Gesture Recognition:** The system shall accurately recognize and classify a wide range of sign language gestures, including individual signs, words, and phrases.

• **Real-time Processing:** The system shall process sign language gestures in real-time, with minimal latency between gesture recognition and translation.

• **Translation to Text:** The system shall translate recognized sign language gestures into readable text, maintaining accuracy and fluency.

• **User Interface:** The system shall feature an intuitive and user-friendly interface, allowing users to interact with ease and input gestures

SYSTEM REQUIREMENTS

Non-functional Requirements

Non-functional requirements specify the characteristics and constraints of the system, including performance, reliability, and usability. These consists of:

• **Accuracy:** The system shall achieve a high level of accuracy in gesture recognition and translation to ensure effective communication.

• **Real-time Performance:** The system shall process gestures with low latency, providing a seamless user experience.

• **Compatibility:** The system shall be compatible with standard hardware, including personal computers, smartphones, and tablets.

• **Accessibility:** The user interface must be accessible to individuals with varying levels of technical proficiency and physical abilities.

UNDERSTANDING THE SYSTEM

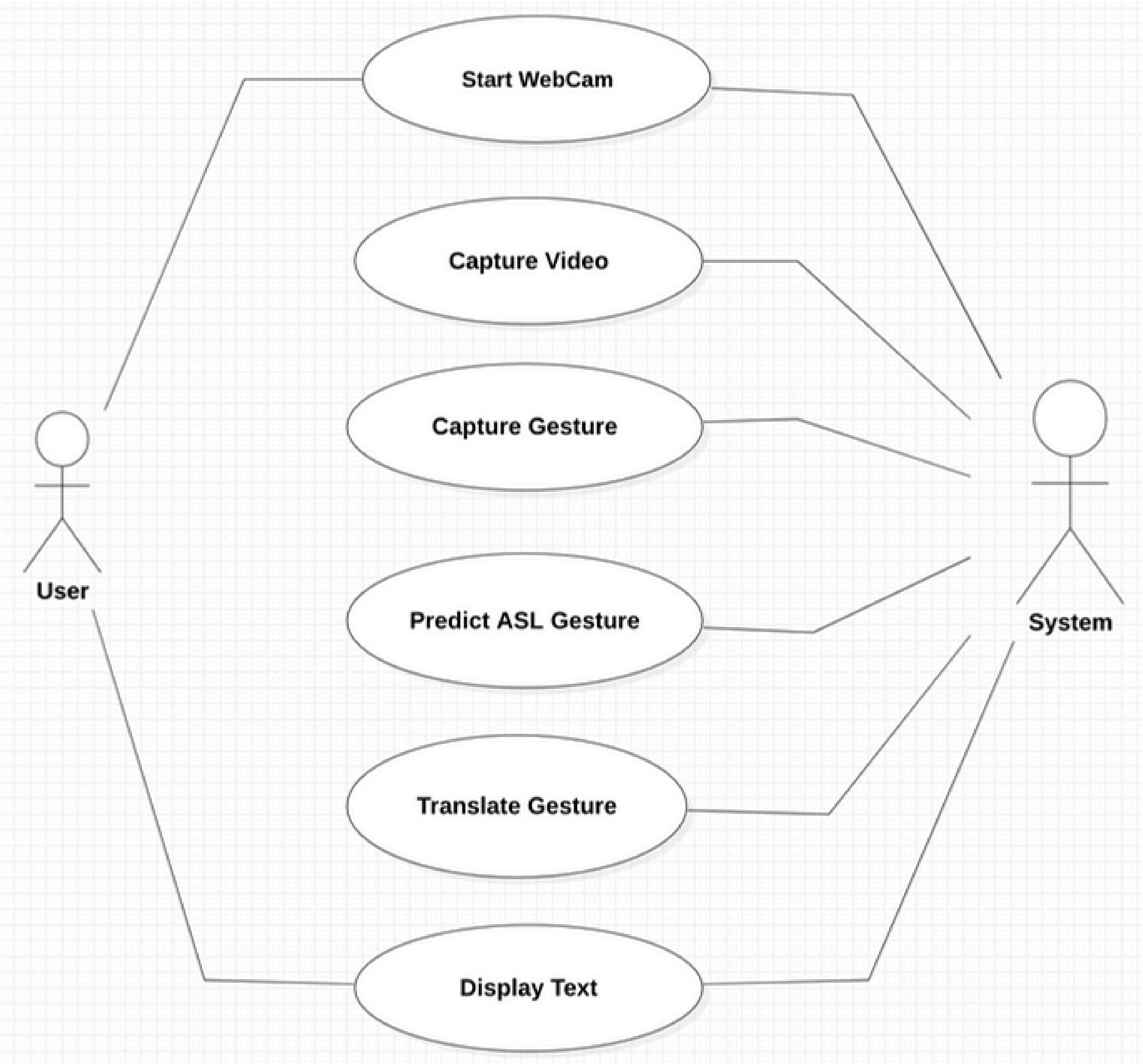
Via Models

We consider that it is important to showcase our system with appropriate set of figures that gives proper justice to our Sign Language Recognition System. It is a crucial part for any system to have graphical repsentation of itself, for the people who are intrested in educating themselves with the fundamentals of that system or even its domain.

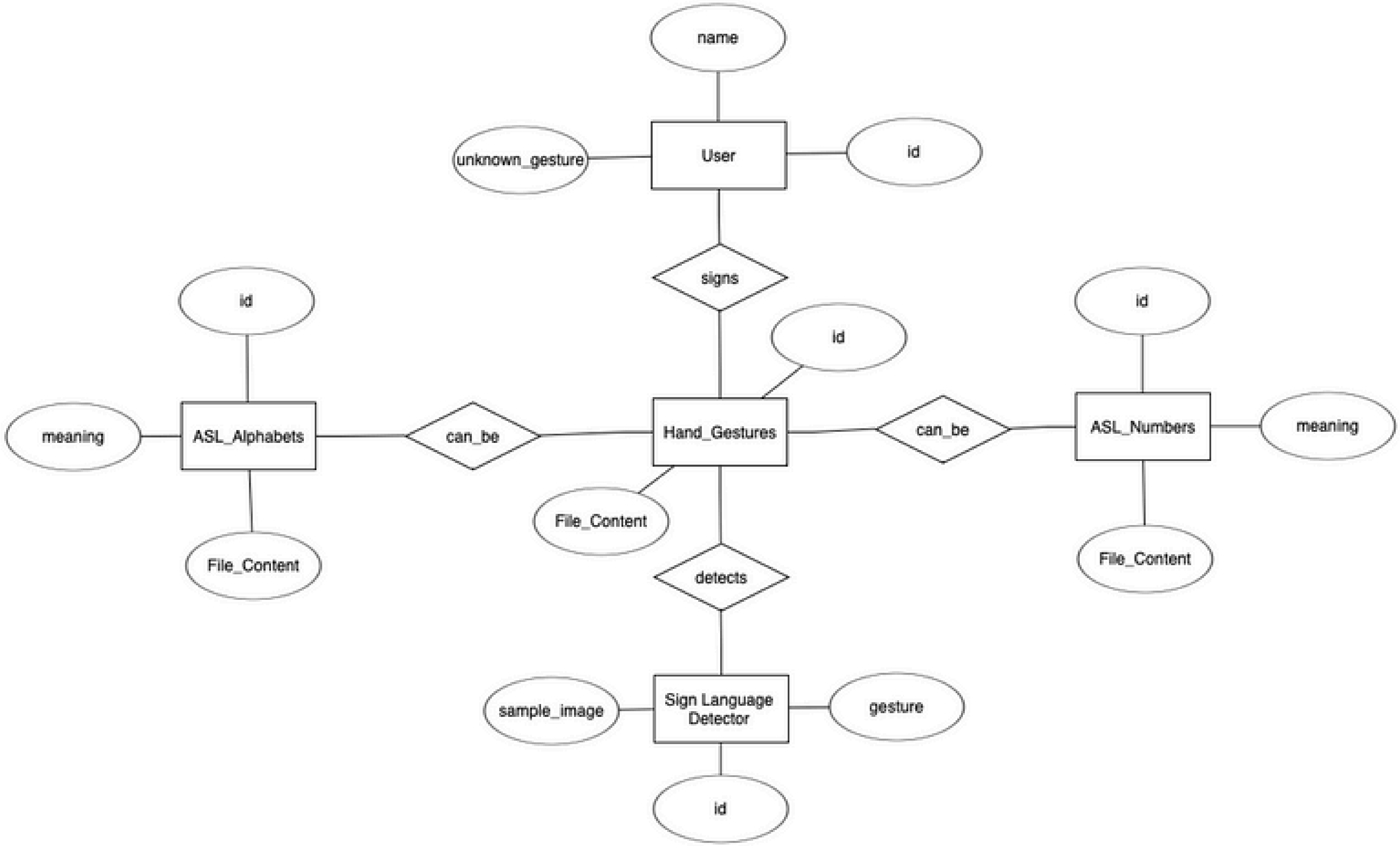
Keeping all of this in mind we have leaned over the following set of diagrams to present our system in depth:

- **Use Case Diagram:** It visualize the interactions between actors and a system.
- **ER Diagram:** It illustrate the relationships between entities in a database.
- **Sequence Diagram:** It show interactions between objects in a sequential order
- **Data Flow Diagram:** It represent the flow of data within a system
    - Level 1
    - Level 2
    - Level 3

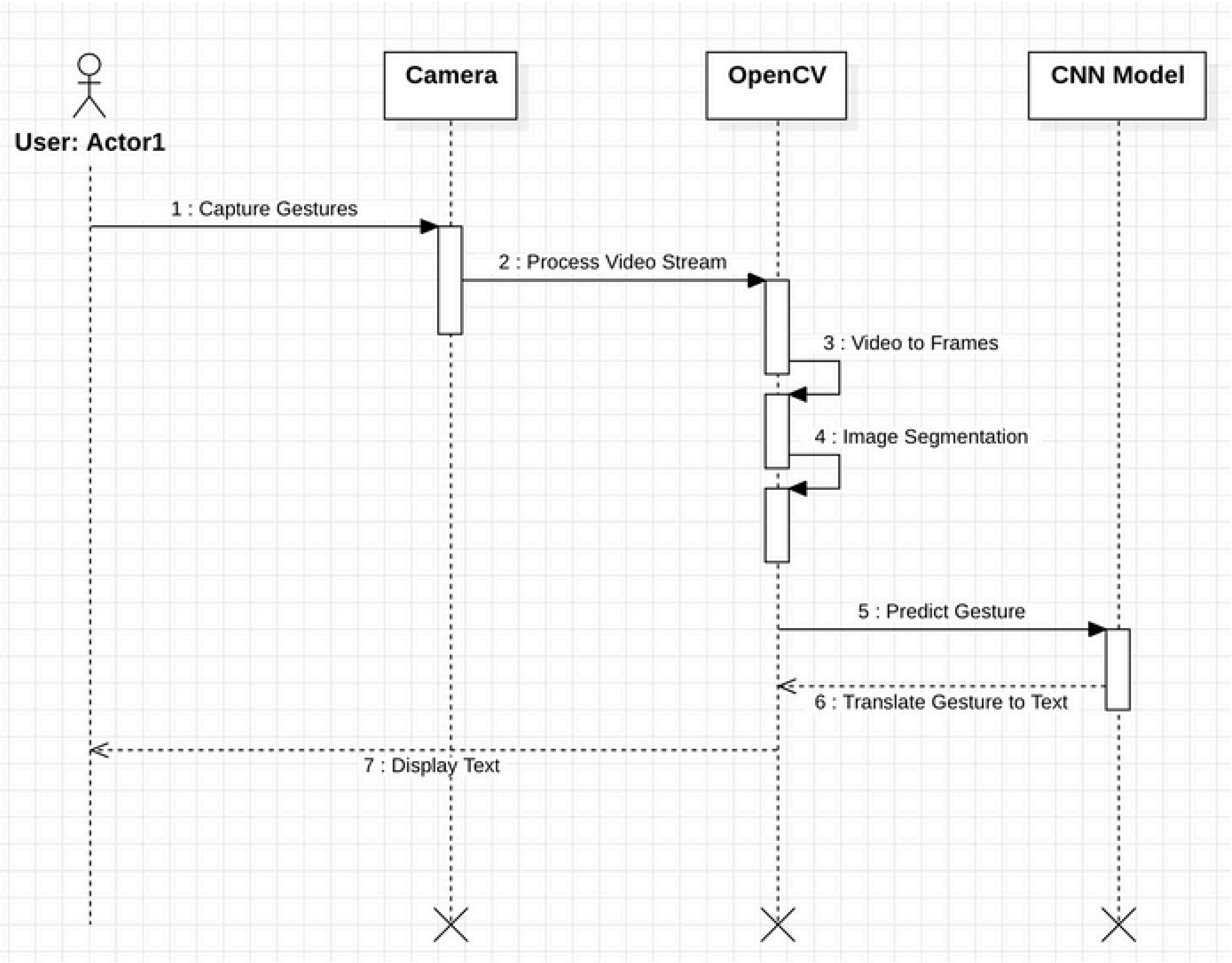
USE CASE DIAGRAM



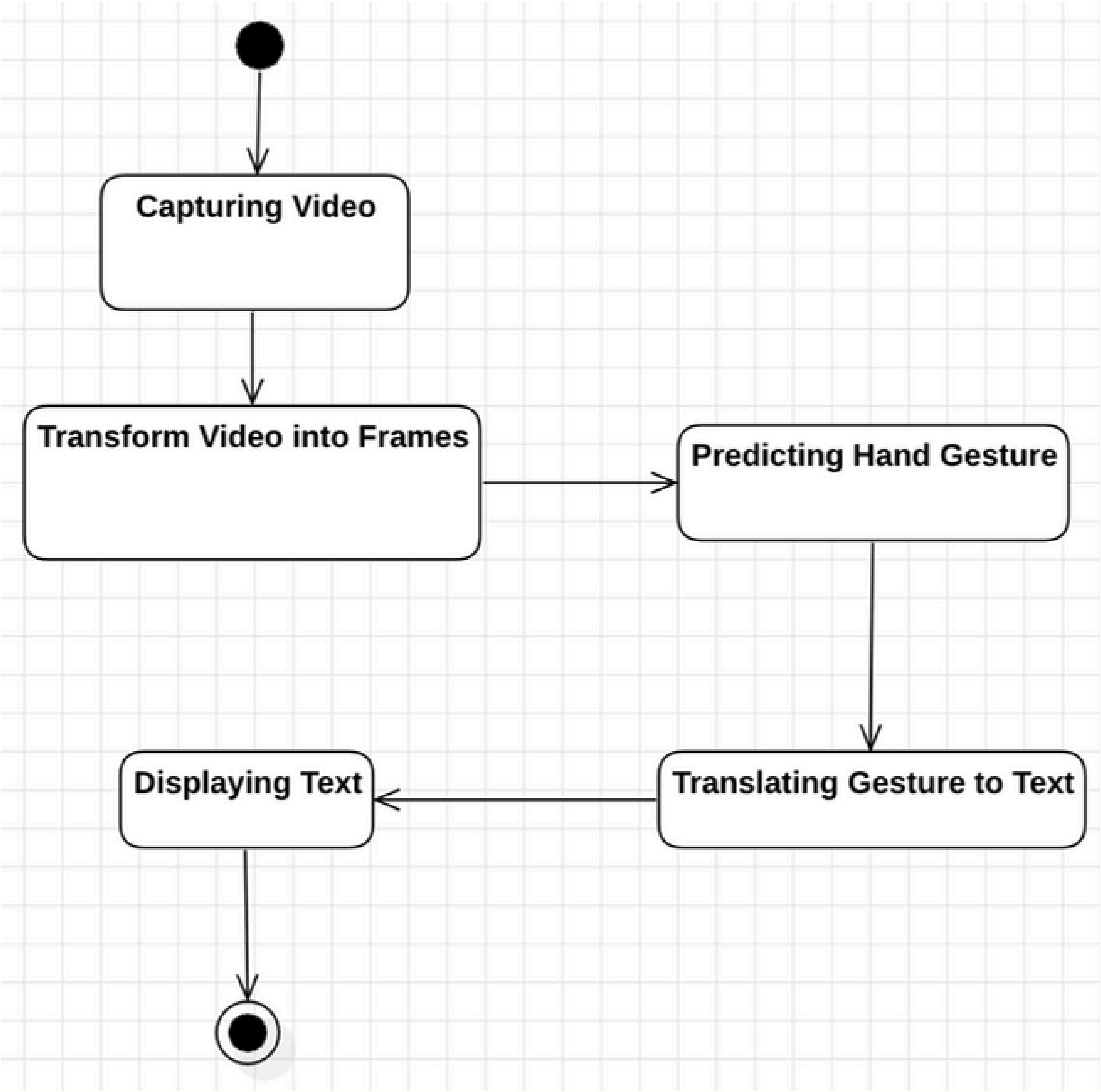
ER DIAGRAM


name
unknown_gesture
User
id
signs
id
id
id
meaning
ASL_Alphabets
can_be
Hand_Gestures
can_be
ASL_Numbers
meaning
File_Content
File_Content
File_Content
detects
sample_image
Sign Language Detector
gesture
id

SEQUENCE DIAGRAM

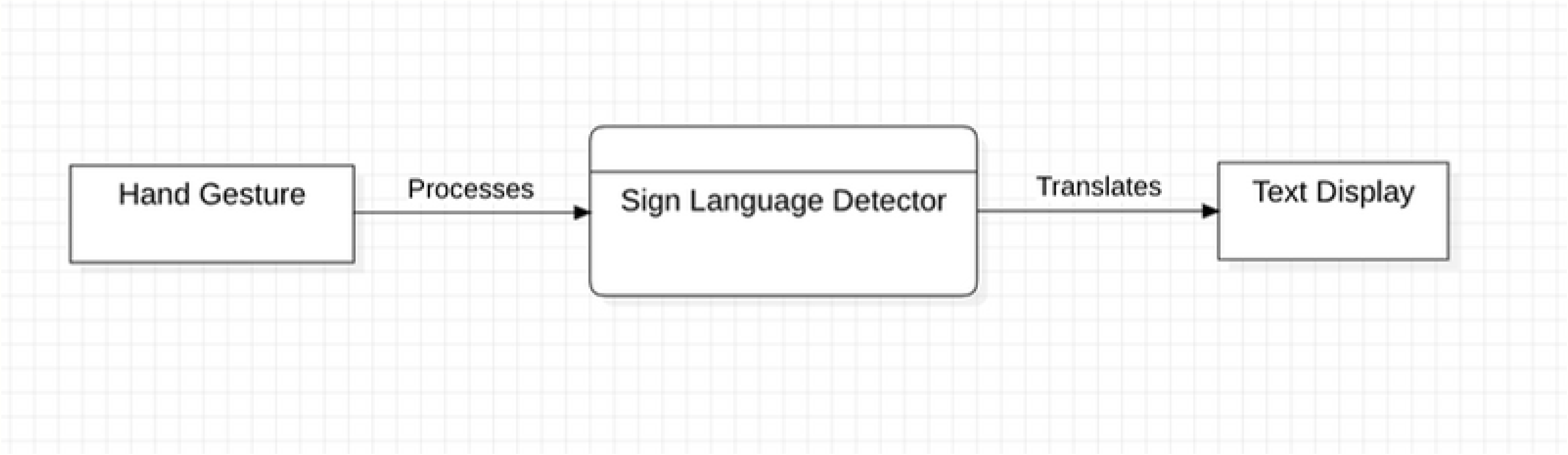


STATEFLOW DIAGRAM



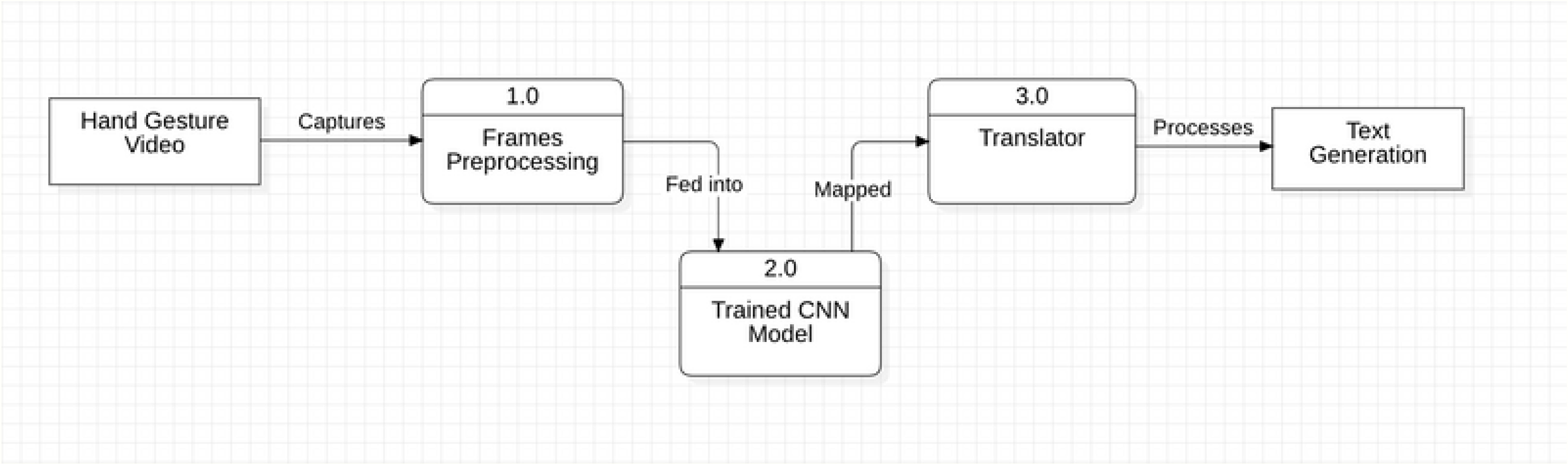
DATA FLOW DIAGRAM

Level 0



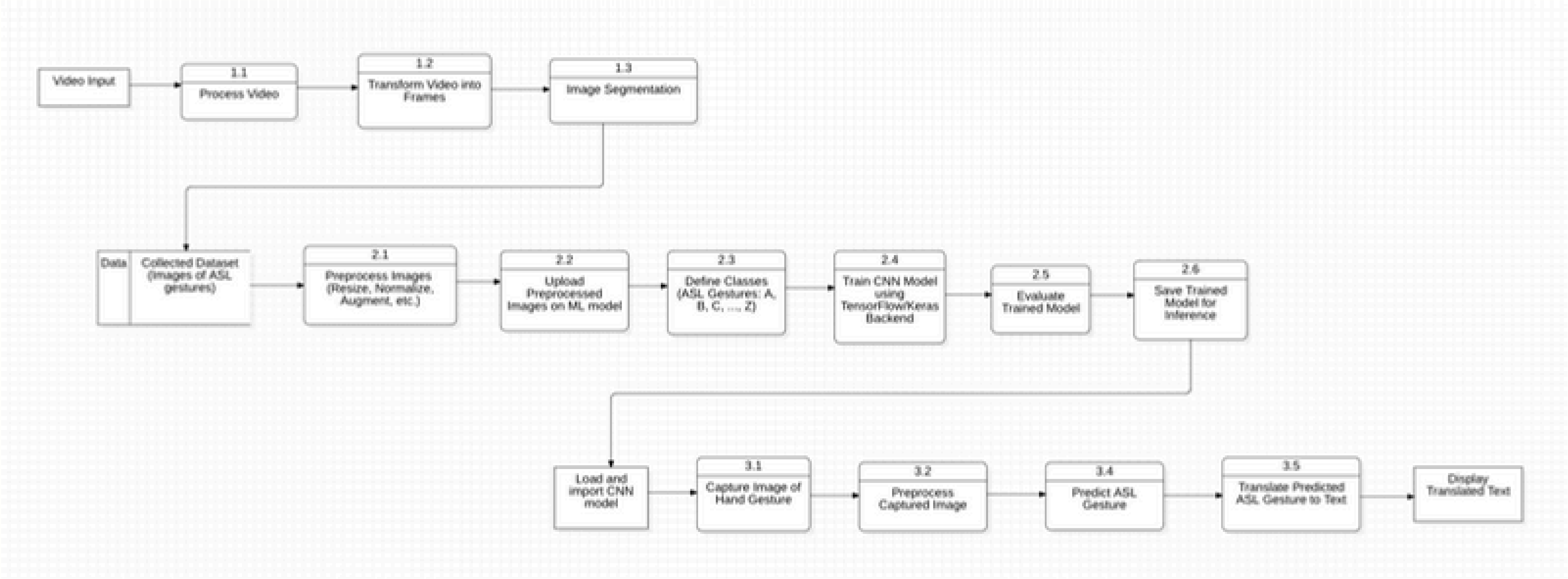
DATA FLOW DIAGRAM

Level 1

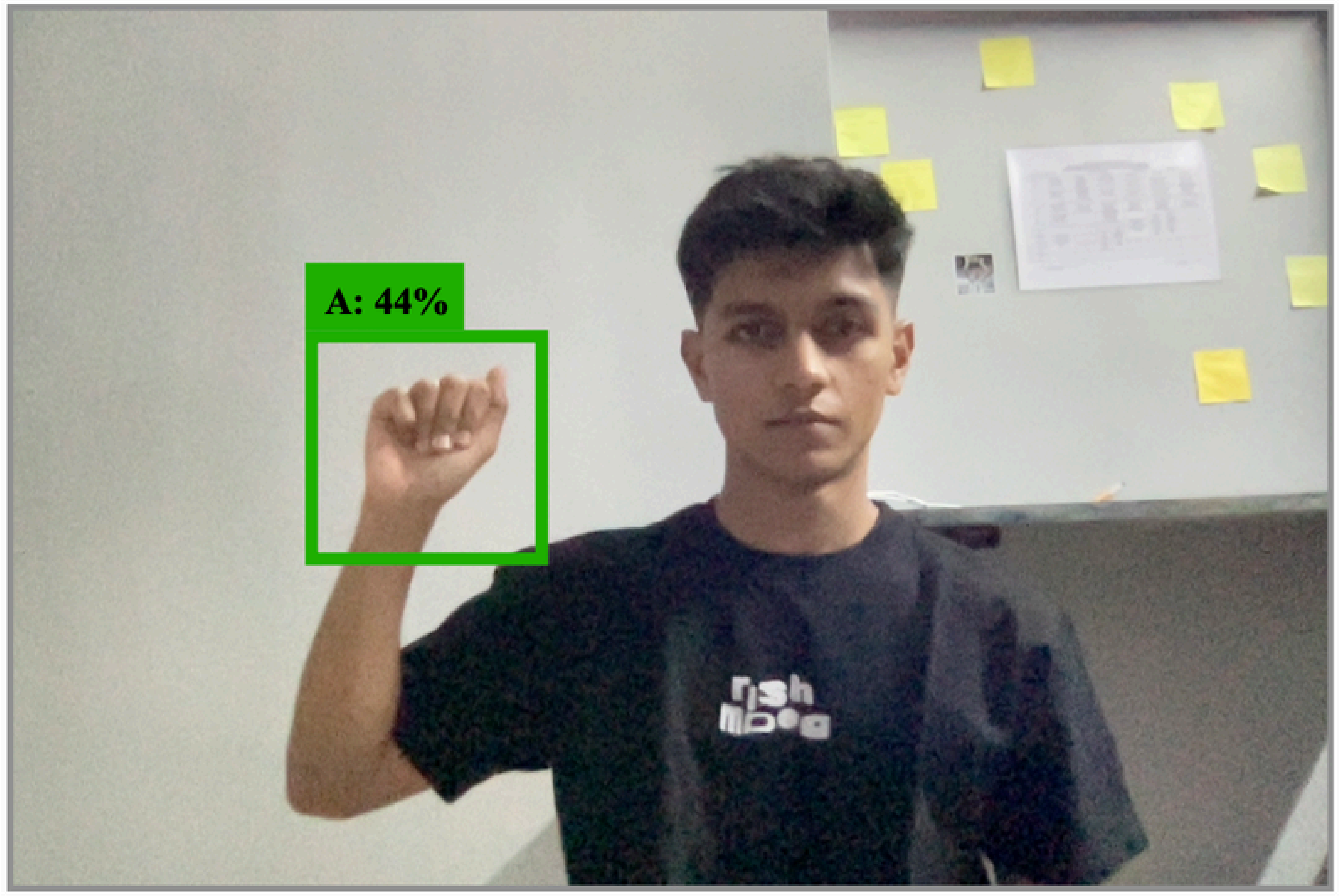


DATA FLOW DIAGRAM

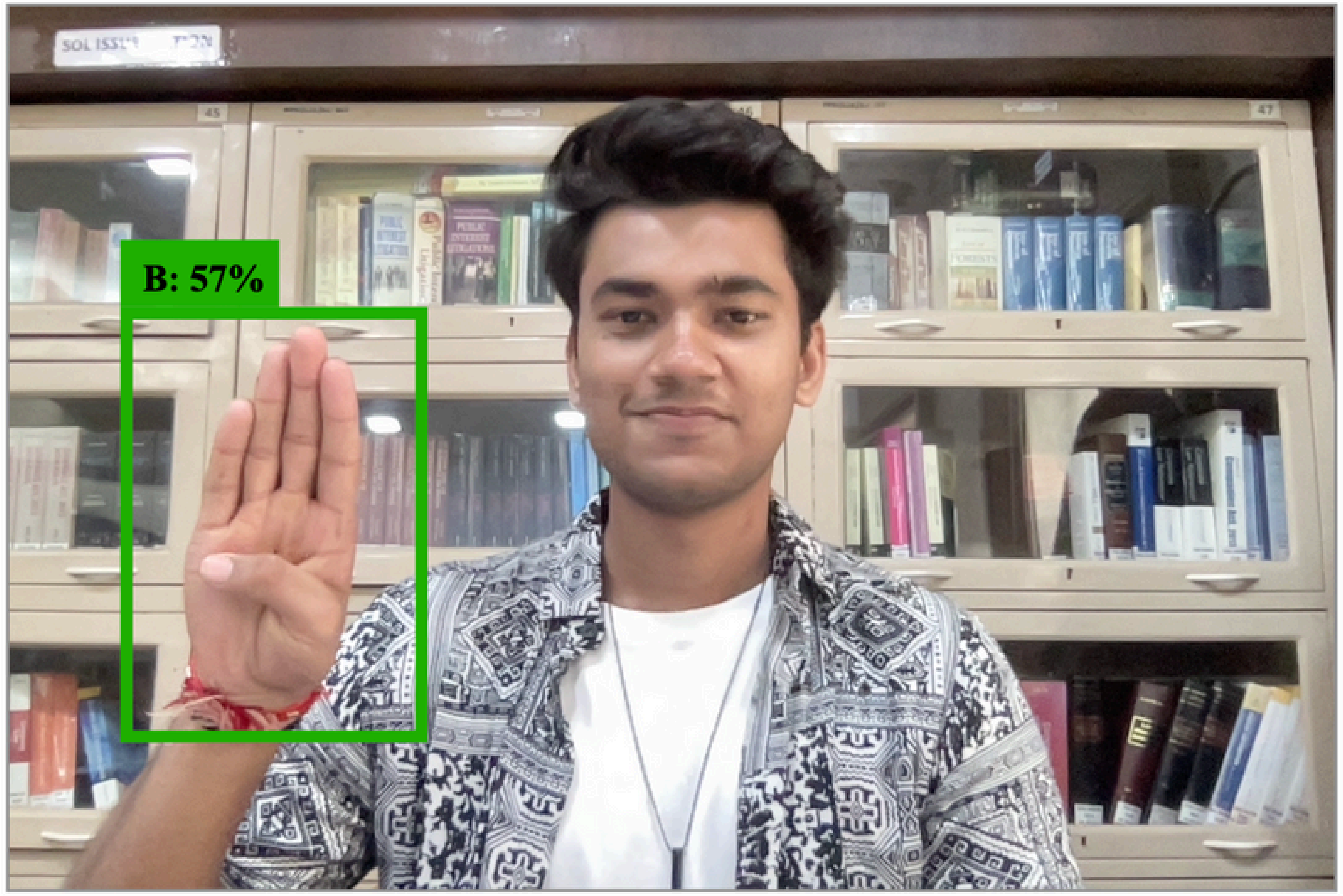
Level 2



USER INTERFACE



UI Detecting ASL 'A'



UI Detecting ASL 'B'

LIMITATIONS

- **Accuracy Fluctuations:**
  - Similar-looking gestures may lead to inaccuracies in recognition, impacting the system's reliability in distinguishing between subtle differences in sign language gestures.
- **Hardware Requirements:**
  - The system's accuracy and performance rely on fast computer processors and high-quality webcams, potentially excluding users with older or less powerful hardware.
- **Gesture Positioning and Alignment:**
  - Accurate positioning and alignment of gestures are crucial for successful recognition, necessitating users to align their gestures precisely as per the model's training data, which can be challenging in real-world scenarios.
- **Hand Feature Variability:**
  - Variations in hand features among users may introduce confusion and inconsistencies in gesture recognition, as the system may struggle to adapt to diverse hand shapes and sizes.
- **Frame Boundary Dependency:**
  - If the area being detected moves out of the webcam frame, the program terminates, limiting its robustness in dynamic environments.

FUTURE WORK

- **Advanced Gesture Recognition**
  - Further research into deep learning and multi-modal sensor fusion techniques to enhance accuracy and robustness in recognizing sign language gestures.
- **User-Centric Design**
  - Incorporate user feedback to develop intuitive interfaces and features, ensuring accessibility for users with varying levels of technological proficiency.
- **Multi-Modal Integration**
  - Integrate audio and haptic feedback to enrich communication experiences, particularly in immersive virtual or augmented reality environments.
- **Global Accessibility**
  - Expand language support to include diverse sign languages and regional variations, promoting inclusivity on a global scale.
- **Continuous Improvement**
  - Implement mechanisms for ongoing evaluation and refinement based on usage data and user feedback to optimize system performance and functionality.

THANK YOU