A Software requirements specification
on

Sign-Language Recognition System

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Computer Science and Engineering from Rajasthan Technical University, Kota

MENTOR:

Mr. Pratipal Singh

(Dept. of Computer Science & Engineering)

COORDINATOR:

Ms. Anjana Sangwan

(Dept. of Computer Science & Engineering)

SUBMITTED BY:

Aman Jain	**19ESKCS032**
Anshika Mittal	**19ESKCS038**
Arunabh Jain	**19ESKCS046**
Aryan Sharma	**19ESKCS047**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN
Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2022-23

Table of Contents

Introduction

1.1 Methodology

Rational Unified Process:

The Rational Unified Process brings together elements from all of the generic process models, supports iteration and illustrates good practice in specification and design. The RUP is normally described from three perspectives:

- A ***dynamic perspective*** that shows the phases of the model over time.
- A ***static perspective*** that shows the process activities that are enacted.
- A ***practice perspective*** that suggests good practices to be used during the process.

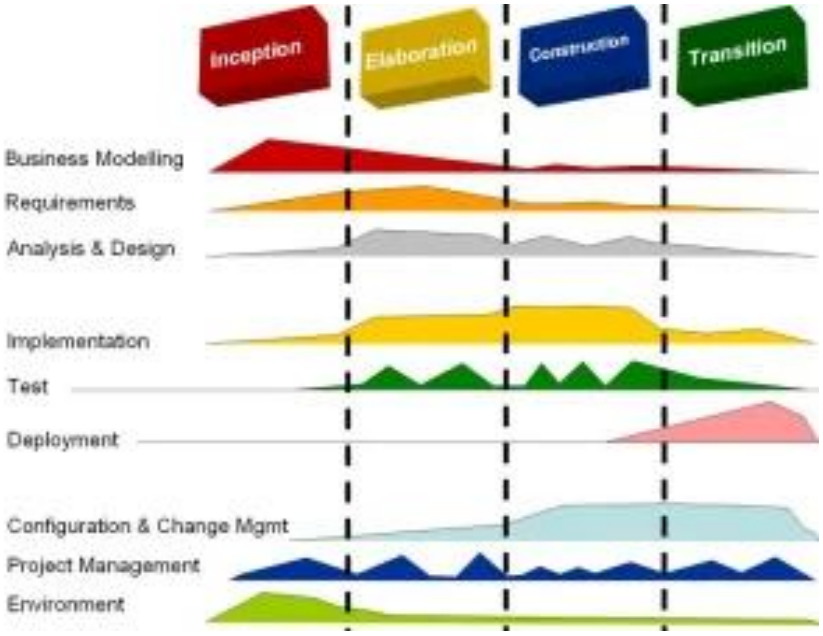

Fig 1.1: Phases of RUP

The different phases in RUP are:

- **Inception**

  The goal of the inception phase is to establish a business case for the system. Identifying all external entities that will interact with the system and defining these interactions. This information is used to assess the contribution of a system to business.

- **Elaboration**

  The goals of the elaboration phase are to develop an understanding of the problem

domain, establish an architectural framework, develop project plan and identify key project risks.

- **Construction**

  This phase is concerned with system design, programming and testing. Parts of the system are developed in parallel and integrated during this phase.

- **Transition**

  This is the final phase of RUP and is concerned with moving the system from the development com-munity to the user community and making it work in real environment.

1.2 Purpose

There have been several advancements in technology and a lot of research has been done to help the people who are deaf and dumb. Deaf and hard-of-hearing persons, as well as others who are unable to communicate verbally, utilize sign language to communicate within their communities and with others.

Aiding the cause, Deep learning, and computer vision can be used to make an impact on this cause. This can be very helpful for the deaf and dumb people in communicating with others as knowing sign language is not something that is common to all, moreover, this can be extended to creating automatic editors, where the person can easily write by just their hand gestures.

1.3 Scope

- We can develop a model for ISL word and sentence level recognition. This will require a system that can detect changes with respect to the temporal space.
- We can develop a complete product that will help the speech and hearing-impaired people, and thereby reduce the communication gap.

1.4 Definitions, Acronyms and Abbreviations

- **Neural Network**

  A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain

operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input.So, the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

Areas of Application:

Speech Recognition, Character Recognition, Human Face Recognition etc.

- **Convolutional Neural Netwrok**

Convolutional neural networks (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. For example, Facebook uses CNN for automatic tagging algorithms, Amazon - for generating product recommendations and Google - for search through among users' photos.

CNN	Convolutional Neural Network
ASL	American Sign Language
ReLU	Rectified Linear Unit
RGB Image	Red Green Blue Image
ROI	Regional of Interest

Table 1: Abbreviations

1.5 Tools Used

1.5.1 Application architecture – Node.js, Express.js, Next.js

- **Python**

  Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs.

- **Jupyter Notebook**

  The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

- **TensorFlow**

  TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

  Features: TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell Julia, MATLAB, R, Scala, Rust, and Crystal.

- **OpenCV**

  OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel [2]). The library is cross-platform and free for use under the open-source BSD license.

  OpenCV's application areas include:

  2D and 3D feature toolkits, Egomotion estimation, Facial recognition system, Gesture recognition, Human–computer interaction (HCI), Mobile robotics, Motion understanding, Object identification, Segmentation and recognition.

- **Keras**

  Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the exception deep neural network model.

1.5.2 Development Tool - FLASK

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine.

1.5.3 Design tool - Creately

Creately is a SaaS visual collaboration tool with diagramming and design capabilities designed by Cinergix. Creately has two versions: an online cloud edition and a downloadable offline edition for desktop which is compatible with Windows, Mac and Linux.

1.6 Technologies to be used

- HTML : Front End Web Development
- Git, Github : Version Control System
- CSS : Front End Web Development
- FLASK : Backend Development
- JavaScript : Programming Language for Website
- Python : Programming Language

1.7 Overview

Existing System:

- Less features.
- User registration required

Drawbacks:

- Remote area users and people who doesn't have knowledge of internet cannot use the system
- Free to use.

Our Plan:

- Making a web app for user.
- Web camera interaction.
- Instant language conversion

Overall Description

2.1 Product Perspective

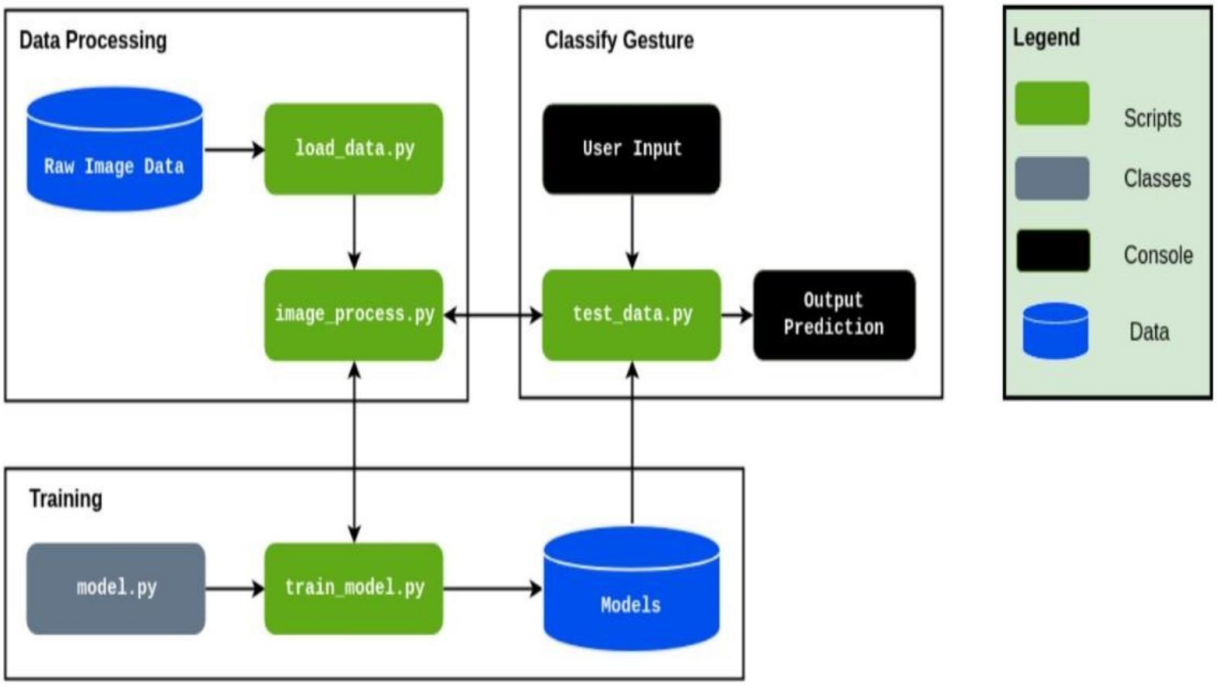

Fig 2.1: Product Perspective

Sign language is the only tool of communication for the person who is not able speak and hear anything. Sign language is a boon for the physically challenged people to express their thoughts and emotion.

In this work, a novel scheme of sign language recognition has been proposed for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output.

Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people.

2.2 Software Interface

- **Client on Internet**

  Web Browser, Operating System (any)

- **Client on Intranet**

  Web Browser, Operating System (any)
- **Web Server**

  Flask Server, Operating System (any)
- **Development End**

  Flask, Python, HTML, CSS, JavaScript, OS(Windows)

2.3 Hardware Interface

Minimum Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Google Chrome	Intel Pentium III or AMD -800 MHz	1 GB	100 MB

Server Side			
	Processor	**RAM**	**Disk Space**
Flask	Intel Pentium III or AMD -800 MHz	1 GB	3.5 GB

Recommended Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Google Chrome	Intel i3 Processor or AMD Ryzen 3	4 GB	1 GB

Server Side			
	Processor	**RAM**	**Disk Space**
Flask	Intel i3 Processor or AMD Ryzen 3	4 GB	3.5 GB

2.4 Communication Interface

- Clients (customer) on the Internet will be using HTTP/HTTPS protocol.
- clients (system user) on the Internet will be using HTTP/HTTPS protocol.

2.5 Constraints

- GUI is only in English.
- Login and password are used for the identification of users.
- Only registered patients and doctors will be authorized to use the services.
- Limited to HTTP/HTTPS.

2.6 E-R Diagram

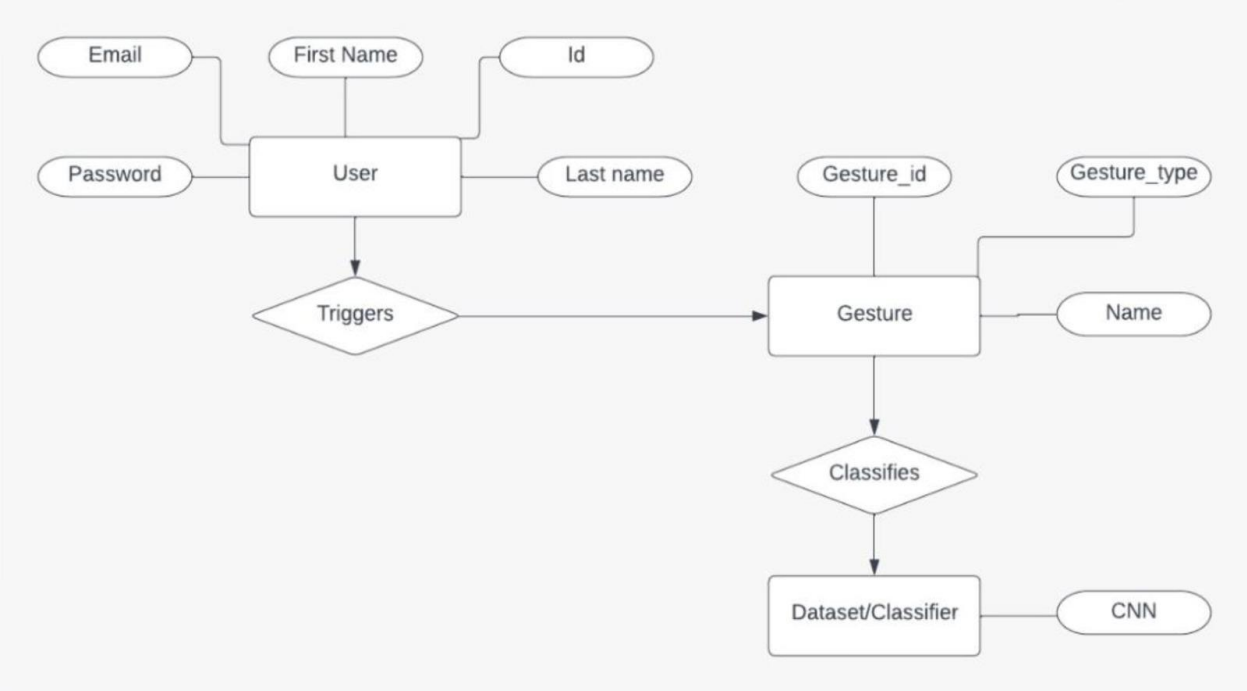

Fig 2.2: E-R Diagram

2.7 Architecture Diagram

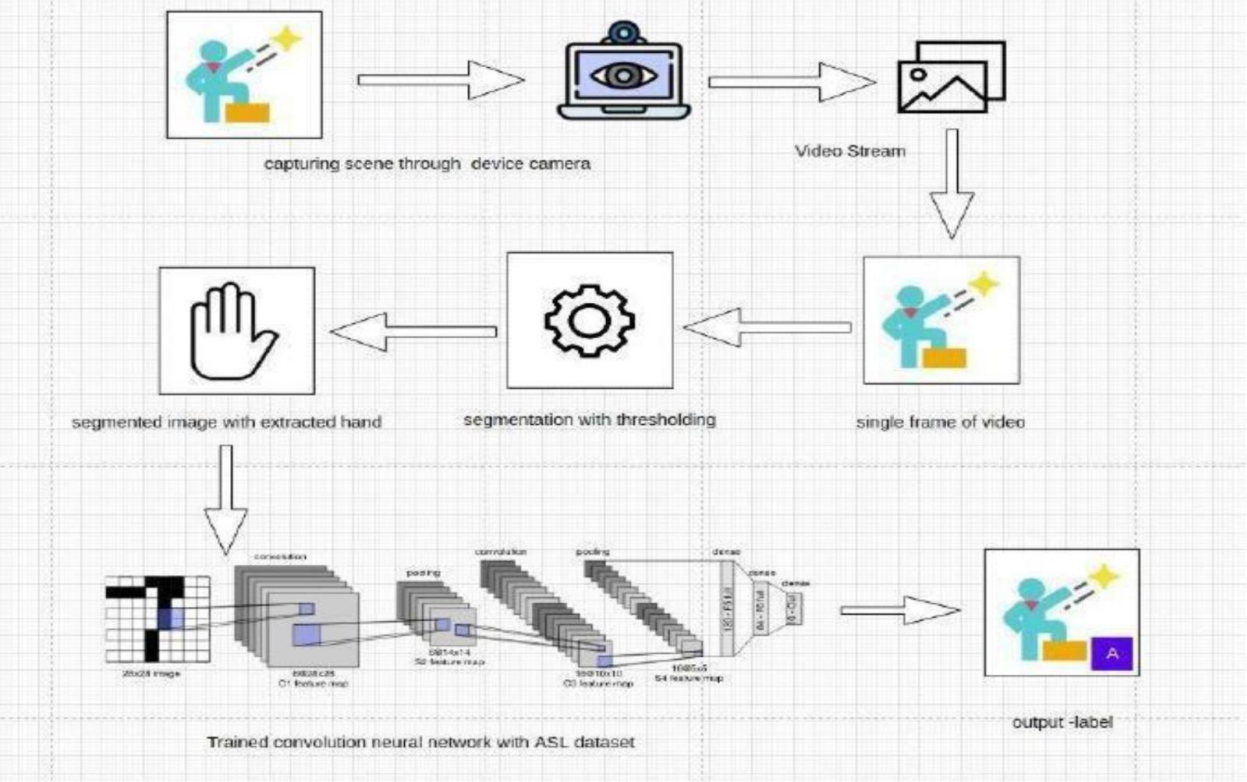

Fig 2.3: Architecture Diagram

Specific Requirements

3.1 Use Case Reports

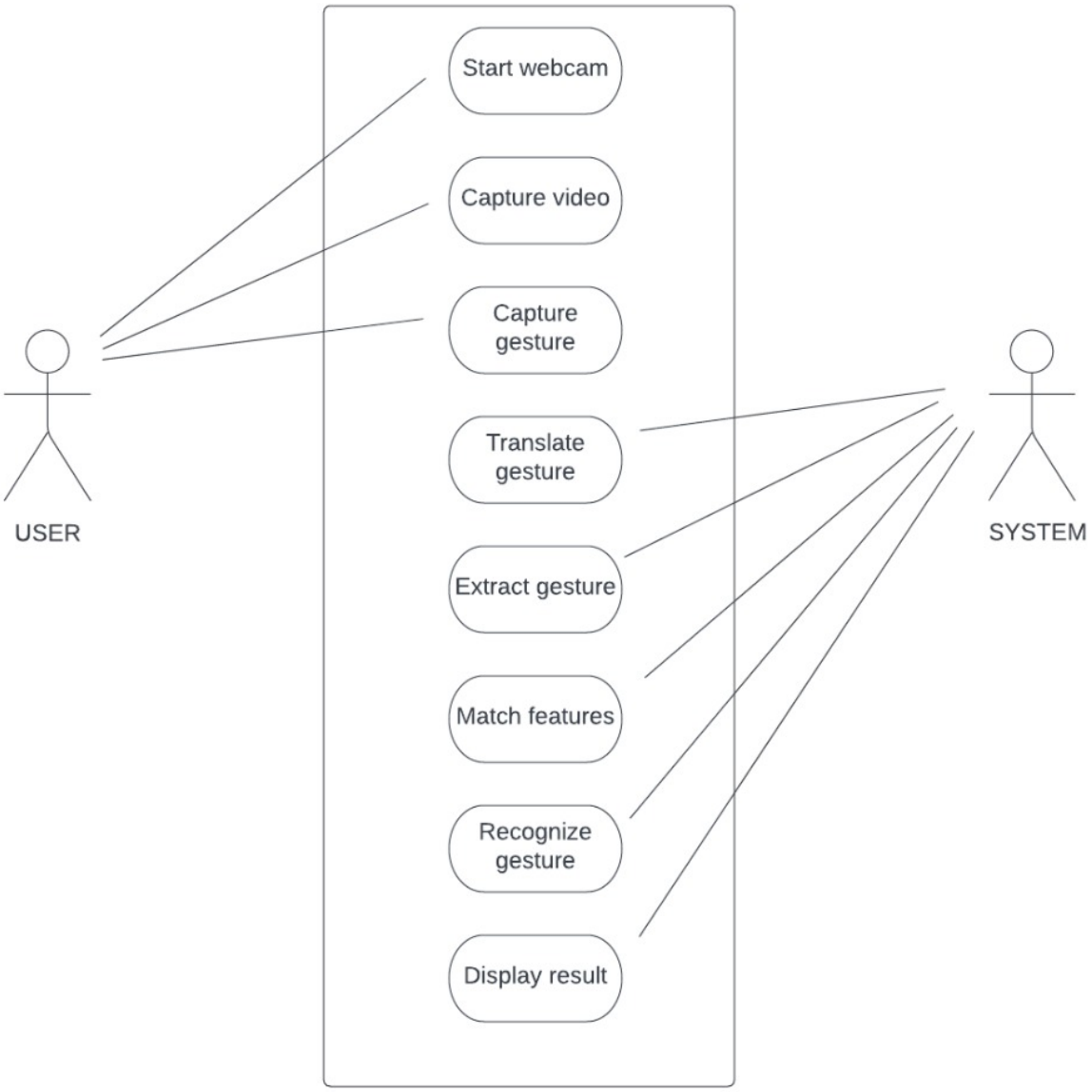

Fig 3.1: Use case diagram

Use Case	Description
Use-Case Name	Sign Language Recognition System
Participating actors	User, System
Flow of Events	Start the system(u) Capturing video(s) Capturing gesture(s) Translate gesture(s) Extract feature(s) Match feature(s) Recognizing gesture(s) Display result
Entry Condition	Run the code
Exit Condition	Displaying the label
Quality Requirements	Cam pixels clarity, good light condition

3.2 Activity Diagrams

Initially user is made to fill all mandatory fields filled in registration form. Once the user clicks submit, the username is verified. If the username is already present, then the user is again taken back, so that he can change the username. If the username is not present then it checks for password and remaining mandatory fields. If any of the mandatory field is left empty or filled incorrect, then the user is informed to enter the correct values. Once all these verifications are succeeded, then the registration is done.

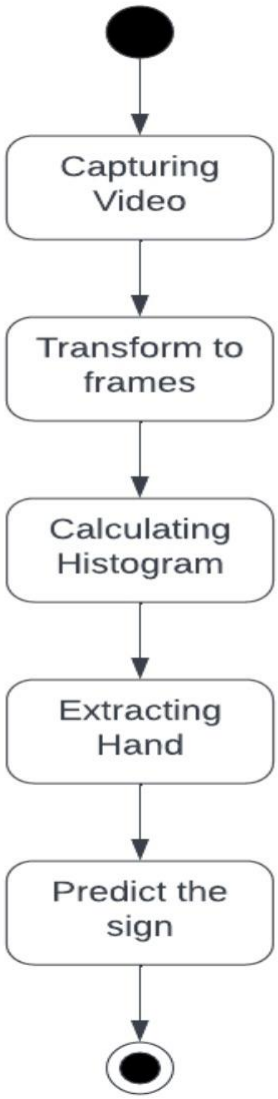

Fig 3.6: Activity Diagram

3.2 Sequence Diagrams

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them. However, an organization's business staff can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact. Besides documenting an organization's current affairs, a business-level sequence diagram can be used as a requirements document to communicate requirements for a future system implementation. During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement. When that occurs, use cases are often refined into one or more sequence diagrams.

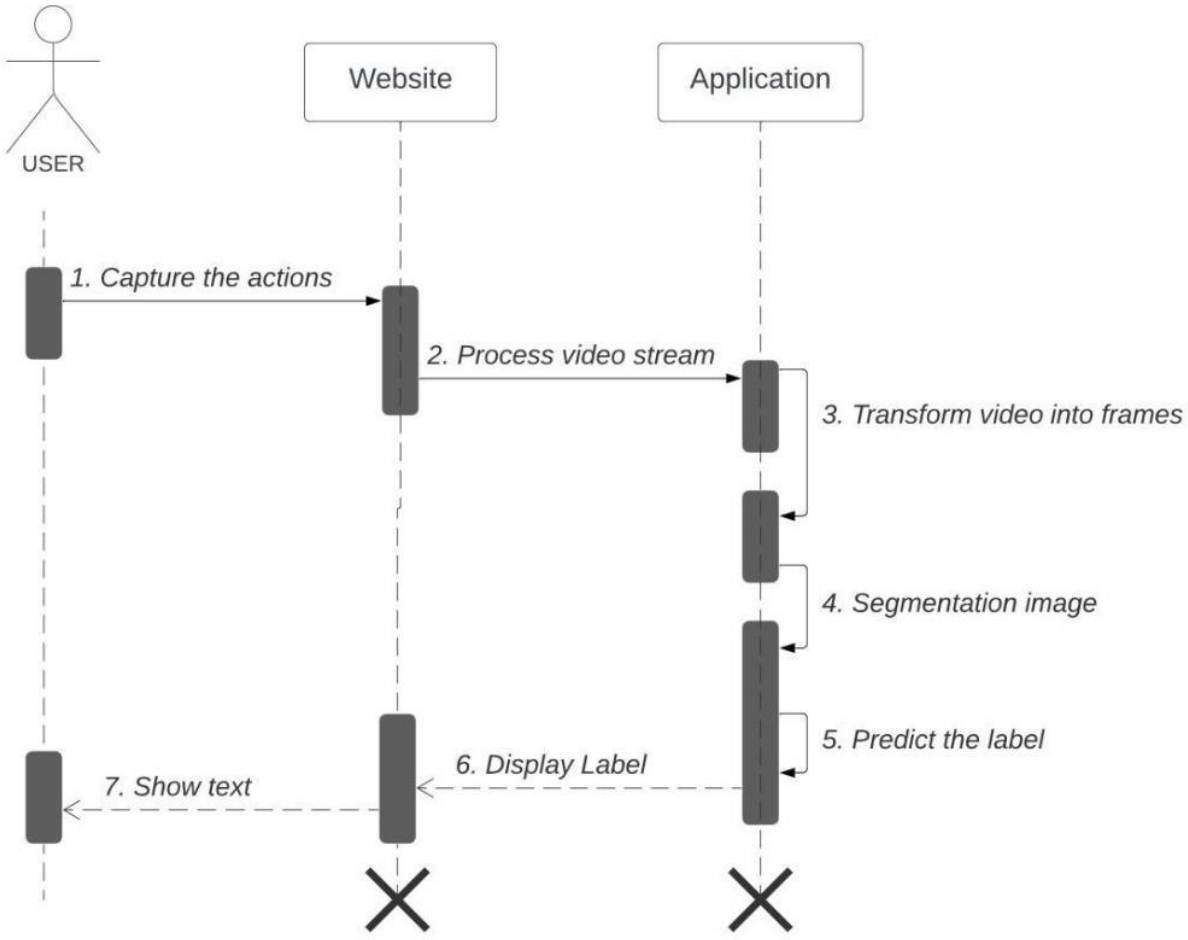

Fig 3.11: Sequence Diagram

4. References

[1] Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.

[2] Software Engineering, Seventh Edition, Ian Sommerville.

[3] http://cs231n.stanford.edu/reports/2016/pdfs/214_Report.pdf

[4] http://www.iosrjen.org/Papers/vol3_issue2%20(part-2)/H03224551.pdf

[5] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.734.8389&rep=rep1&type=pdf

[6] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.734.8389&rep=rep1&type=pdf

[7] https://ieeexplore.ieee.org/document/7507939

[8] https://ieeexplore.ieee.org/document/7916786

[9] https://www.sciencedirect.com/science/article/pii/S1877050917320720

[10] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[11] https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/