# Chapter 1 : Introduction

### Background study

Sign language is a visual form of communication used by individuals who are deaf or hard of hearing. It relies on hand gestures, facial expressions, and body movements to convey meaning. In Nepal, **Nepali Sign Language (NSL)** serves as the primary communication method for the hearing-impaired community. However, due to limited awareness, educational resources, and assistive technologies, many individuals with hearing disabilities face challenges in interacting with the general population. This communication gap often leads to social isolation and reduced access to education, employment, and essential services.

In recent years, advancements in **artificial intelligence (AI)** and **computer vision** have enabled the development of systems that can recognize hand gestures and convert them into readable text. Such systems play a vital role in improving inclusive by enabling smoother communication between the deaf and hearing communities. While several studies have been conducted for American and Indian Sign Languages, very few research projects have focused on **Nepali Sign Language**, and even fewer offer real-time recognition. This project aims to fill that gap by designing a **Nepali Sign Language Detection System** that recognizes hand signs for individual letters and forms simple words by combining them.To detect hand gestures accurately, this project uses a **Convolutional Neural Network (CNN)**—a type of deep learning model well-suited for image classification tasks. CNNs can automatically extract important visual features from hand gesture images, making them effective for recognizing variations in hand shape, orientation, and lighting. The CNN is trained on a custom dataset of static hand signs corresponding to Nepali alphabets, and the training is conducted on **Kaggle**, which offers a cloud-based platform with GPU support for faster model development and testing.

A crucial part of the system is the **media pipeline**, which allows real-time gesture recognition. The pipeline involves capturing video input from a webcam, preprocessing each video frame to focus on the hand gesture, and feeding the frame into the CNN model. The output—recognized letters—is displayed instantly to the user and stored to form words like Kamala, Ghara, and Maga. Libraries like **OpenCV** are used for real-time video capture and image processing, while **TensorFlow/ Keras** are used to build and run the deep learning model.

## **1.2. Project Introduction**

Sign language is an expressive form of communication used by individuals who are deaf or hard of hearing. It allows them to convey thoughts, ideas, and emotions through manual gestures, facial expressions, and body movements. However, deaf individuals often encounter significant barriers in their daily interactions, primarily due to the difficulty of communicating with those who do not understand sign language. There are numerous sign languages, such as American Sign Language, Indian Sign Language, Nepalese Sign Language, and so on. The signs used in these sign languages are not all the same. In our country, Nepal, various organizations and schools have been assisting deaf people to learn Nepali sign language. Understanding the importance of connecting the deaf and hearing communities, multiple technologies aimed at improving communication using Nepalese Sign Language for persons who are deaf or hard of hearing were studied. It was found that there weren’t any specialized sources online regarding Nepalese Sign Language (NSL). As a result, we aimed to create a model that would translate the NSL into textual output. The user will provide input via webcam, and the model can detect the hand gesture and output the word/alphabet that the user provided as input to the model.

### 1.3. Purpose

The motivation for doing this project primarily is

* To develop a **real-time system** that recognizes **static hand gestures** of **Nepali Sign Language (NSL)** letters.
* To enable the system to **form simple words** by combining individually recognized letters.
* To implement a **media pipeline** that handles image input, preprocessing, and real-time gesture recognition.
* To support the **hearing-impaired community** by reducing communication barriers with the general public.

### 1.4. Problem Statement

* People with hearing impairments in Nepal face communication barriers due to limited support for Nepali Sign Language (NSL).
* Existing sign language systems mainly focus on other languages, lacking solutions for NSL.
* Many systems are not real-time or require complex hardware, reducing usability.
* There is a need for a simple, real-time system to detect NSL letters and form basic words.

### 1.5. Objective of the project

* To develop a real-time system that can recognize **static hand gestures** using a webcam.
* To design a **media pipeline** that captures, pre-processes, and feeds live video frames into the gesture recognition model.
* To enable the system to **form basic words** by combining a sequence of recognized letters (e.g., Kamala, Ghara, Maga).

### 1.6. Features of project

* Trained with custom dataset.
* Real-time hand gesture detection using webcam.
* Recognition of detected Sign.
* Translate to text and audio.
* CNN-based accurate gesture classification.

### 1.7.Assignment of roles and responsibilities

|  |  |
| --- | --- |
| Ashim Pokharel | Literature review, documentation(table of content,  Abstract), designing(Use case) and coding. |
| Jyoti Dangal | Features, objectives, Analysis, documentation, designing(Er diagram, system architecture) and coding(), testing(unit testing). |
| Renu Singh | Problem Statement, documentation, designing(DFD level 0 and 1 ) and coding(GUI), testing(integration testing). |

**1.8. Documentation organization**

|  |  |
| --- | --- |
| Chapter 1 | It describes about the introduction of the project and it is divided into 8-sub chapter.   1. Background study 2. Project introduction 3. Purpose and motivation 4. Problem statement 5. Project features 6. Objectives of project 7. Assignment of roles and responsibilities 8. Documentation organization |
| Chapter 2 | Literature review |
| Chapter 3 | Describes about the SDLC model and different  study done for the requirement analysis. |
| Chapter 4 | Describes about the diagrammatic working and relations of the projects. |
| Chapter 5 | Describes about the platform, software requirement and testing done during the process of development. |
| Chapter 6 | It describes about the conclusion, Future enhancement and limitations of the project. |

# Chapter 2: Literature review

### 2.1. Nepali Sign Language Recognition Using CNN

Shibesh Duwadi (2018) built a CNN-based system that recognized **37 alphabet signs and 10 numbers** using preprocessing methods such as grayscale conversion, threshold, and contour detection. After data augmentation, the system achieved an accuracy of **92.45 %**.

**Advantages**

* Effective recognition accuracy (~92 %) for static gestures.
* Extensive data preprocessing and augmentation improved performance.

**Limitations**

* Focused only on static alphabet and numbers, not word formation.
* Lacks real-time inference and live video pipeline.

### 2.2. Transfer Learning for Sign Language

A 2024 study focused on **American Sign Language (ASL)** applied **transfer learning** with models like VGG16, ResNet50, and DenseNet121. A customized CNN achieved **99.93% accuracy** .

**Advantages:**

* Very high classification accuracy using pretrained deep models.
* Demonstrates CNN-based models can perform well with proper tuning.

**Limitations:**

* Applies to ASL, not directly applicable to NSL due to linguistic differences.
* Does not address media pipeline or real-time recognition.

### 2.3. AI-Based ASL-to-Nepali Translation with Speech Output

A recent system uses CNN-based classification and OpenCV for sign detection and integrates with **text-to-speech (TTS)** to convert ASL to Nepali text and voice using models like ResNet50 and VGG16.

**Advantages:**

* Real-time gesture detection integrated with **voice output**.
* Major inclusion functionality for accessibility.

**Limitations:**

* Focused on ASL, not Nepali Sign Language.
* The heavy framework may not run on standard local devices without GPUs.

### 2.4. NSL Dataset Development (NSL23)

The **NSL23 dataset** consists of gesture videos from 14 volunteers performing **36 consonants** and **13 vowels**, captured under various lighting and background conditions. It’s the first publicly released dataset for NSL system development .

**Advantages:**

* Diverse and realistic dataset with real-world variety.
* Crucial for training robust NSL recognition models.

**Limitations:**

* Contains video rather than static images; may require more storage and preprocessing.

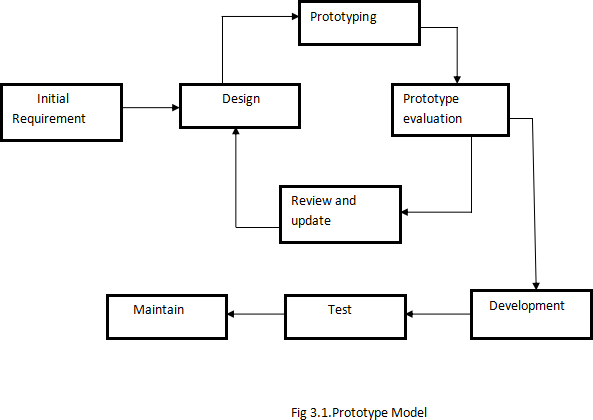
## **Chapter 3: System Analysis**

System Analysis is the process of studying a procedure in order to identify the goals and purpose and create the system and procedure that will achieve them in an efficient ways.it is the phase of the software development where a system requirement, identification, requirement Analysis and feasibility study are performed.

System analysis starts with a preliminary analysis and later switches on the detailed one. During the preliminary analysis analyst takes a quick look at what is needed and whether that cost benefits.

#### **3.1. Requirement Specification**

The development of the system follows the prototype model.



#### Why Prototype is used?

Prototype model is used in this project due to following reason.

* For requirement uncertainty and changing needs.
* Easy to take feedback and continuous improvement until objective gets fulfilled.
* For testing the accuracy and reliability of the system.

#### **3.2. Requirement analysis**

Requirement analysis is the software engineering technique that is composed of the various tasks that determines the needs or conditions that are to be met for the new or altered product, taking into consideration the possible conflicting requirements of the various users.

#### **3.2.1. Functional Requirement**

Functional Requirement are those requirement that are used to illustrate the internal working nature of the system. Functional requirements define what a product must do and what its features and function are.

* **Gesture Recognition:**

The system should detect hand gestures in real time using a webcam and MediaPipe for extracting hand landmarks.

* **Sign Classification:**

The system should use a trained CNN model to classify static hand gestures representing Nepali Sign Language alphabets.

* **Word Formation:**

The system should recognize individual letters and combine them to form specific Nepali words.

* **Data Storage and Management:**

The system should store detected gesture results and logs of interactions, possibly using CSV files or local databases for review and evaluation.

* **Real-Time Feedback Interface:**

The system should provide immediate feedback by displaying detected letters and words in a GUI.

#### **3.2.2 Non functional Requirement**

Nonfunctional requirements describe the general properties of a system. They are also known as quality attributes. They are:

* **Performance & Scalability**:****

The system should instantly recognize hand gestures from webcam input without noticeable lag. It must handle increased gesture categories in the future as new signs and words are added to the model.

* **Availability:**  
  During live usage, especially in schools or public service environments, the system should remain operational without crashes. The result display must consistently reflect correct gesture-to-text translation.
* **Security:**  
  Only authorized personnel (e.g., admin or project team) should access the training dataset and system configurations. The system must handle unknown inputs safely using proper error messages instead of crashing or giving false outputs.
* **Usability:**  
  The user interface should be clean and visually simple, allowing even non-technical users such as teachers or public officers to operate the system without training. It should work on basic desktop setups used in Nepali schools or offices.
* **Maintainability & Portability:**  
  The system should allow easy debugging and updates (e.g., adding a new word or retraining the model).

**3.3. Feasibility Study**

Feasibility study is the process of feasibility analysis of the current as well as proposed system. A feasibility study is done to identify the deficiencies in the current system as well as proposed system. The main purpose of the project is to describe a proposed system that can not only be used for placing an order but such a system that is feasible and compatible to most of the existing system and upcoming new system.

#### **3.3.1.Technical feasibility**

The proposed system is technically feasible. The tools and technologies such as Python, TensorFlow, and OpenCV are freely available and widely used for similar applications. The required hardware like webcam and a standard laptop are easily accessible.

#### **3.3.2.Economical feasibility:**

The system is cost-effective since all used libraries (MediaPipe, OpenCV, etc.) and tools (Jupyter, VS Code) are open-source. No licensed or premium software is required, making it affordable even for personal or educational use.

#### **3.3.3 Operational feasibility**

The system's operation is simple: a user just needs to show their hand gestures in front of a webcam, and the system will recognize the sign and display the equivalent letter or word. This simplicity ensures smooth implementation and user adoption.

#### **3.3.4. Schedule feasibility**

The project was scheduled and managed within the academic semester timeline. All key tasks dataset collection, model training, testing, and interface building were completed according to plan using available team member efforts and lab resources.

## **Chapter 4: System Design**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves translating user requirements into a detailed blueprint that guides the implementation phase. The goal is to create a well-organized and efficient structure that meets the intended purpose while considering factors like scalability, maintainability, and performance.

**Objectives of System Design**

**1.Practicality**: We need a system that should be targeting the set of audiences(users) corresponding to which they are designing.

**2.Accuracy**: Above system design should be designed in such a way that it fulfills nearly all requirements around which it is designed be it functional o non-functional requirements.

3.**Completeness**: System design should meet all user requirements.

**4.Efficient**: The system design should be such that it should not overuse surpassing the cost of resources nor under use as it will by now, we know will result in low through put (output) and less response time(latency).

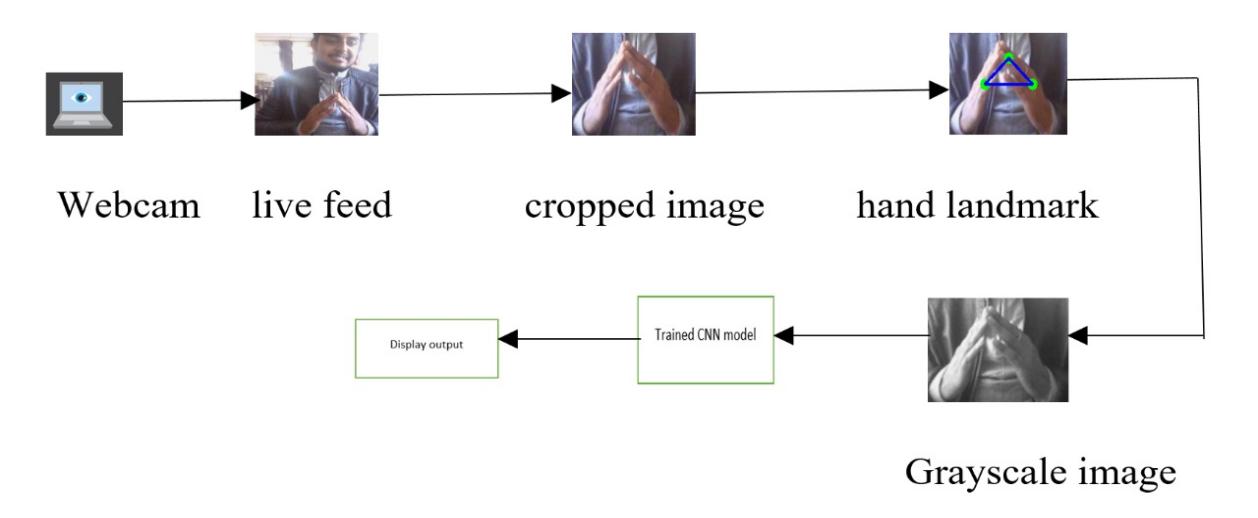


Fig 1. System architecture

**4.1. Data flow diagram(Level Zero)**

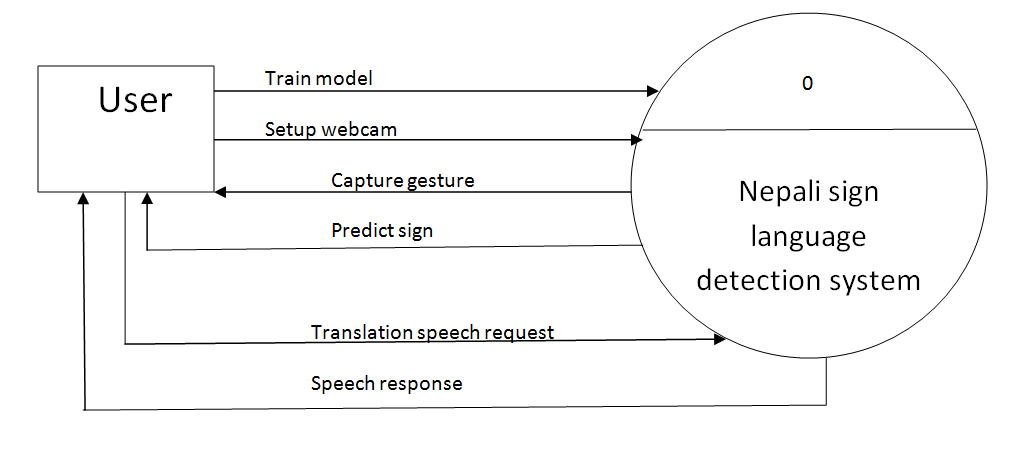


Fig 2. DFD(level 0)

**4.2. Dataflow diagram(Level one)**

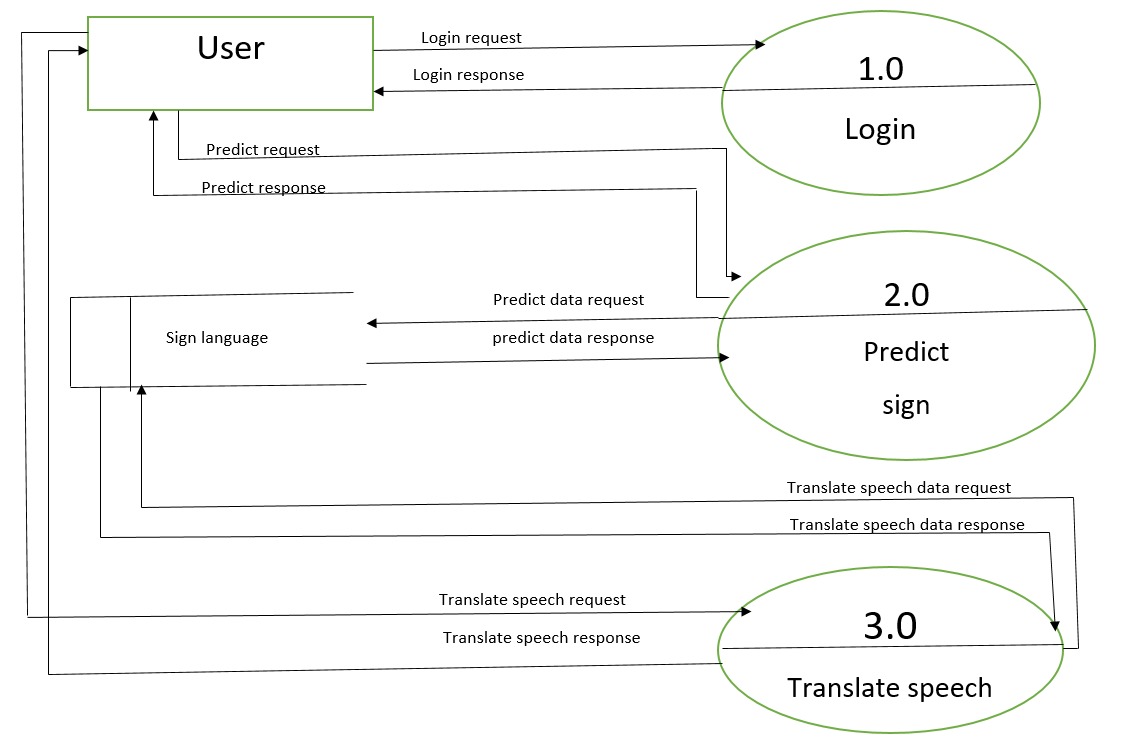


Fig 3. Dfd(level 1)

## **4.3. Use case Diagram**

## WhatsApp Image 2025-07-25 at 15.32.22_480f9d81

## Fig 4. Use case diagram

## 

## **Use case Table**

## **4.4. Methodology**

1. Model Training

- A Convolutional Neural Network (CNN) is trained on labeled images of different sign language gestures.

- The trained model is saved as gesture\_model.h5.

1. System Initialization

- The project loads the trained CNN model (gesture\_model.h5).

- The class labels (e.g., 'dhanyabaad', 'ghar', 'ma', 'namaskaar') are defined in the code.

3. Webcam Activation

- The webcam is started to capture live video frames.

4.Region of Interest (ROI) Selection

- For each frame, a specific region (ROI) is defined where the user should show their hand gesture.

1. Image Preprocessing

- The ROI is cropped from the frame.

- The cropped image is resized to the input size expected by the model (e.g., 128x128 pixels).

- The image is normalized (pixel values scaled between 0 and 1).

- The image is reshaped to match the model’s input format.

6. Gesture Prediction

- The preprocessed ROI is passed to the CNN model.

- The model predicts the probabilities for each gesture class.

- The class with the highest probability is selected as the predicted gesture.

## **Chapter 5: System Development and Implementation**

## **5.1. Programming platform**

* Python version: >=3.10.11
* Django==5.2.4
* djangorestframework==3.16.0
* numpy==2.1.3
* OpenCV-python
* mediapipe
* tensorflow==2.19.0
* Pillow==11.3.0
* django-cors-headers==4.7.0

#### **5.2. Operating Environment**

* Operating System: Microsoft Windows 11 Home (10.0.26100)
* IDE: VS Code (1.98.2)

#### **5.3. Hardware specification**

* Webcam: Driver Provider: Microsoft, Driver Version: 10.0.26100.3470 RAM :16GB
* SSD 512 GB

#### **5.4. Testing and Debugging**

The aim of testing and debugging process was to determine all defects in our project. The program was subjected to set the tests inputs and various observations were made based on these observations, it will be decided whether the program behaves as expected or not. Some testing operations that we performed in our code as shown in table below as:

**CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT**

#### **Conclusion**

The Nepali Sign Language Detection System successfully demonstrates a practical solution for bridging communication gaps between the hearing-impaired community and the general population. By using computer vision and deep learning—specifically TensorFlow, OpenCV, and MediaPipe—the system accurately recognizes static hand gestures representing both individual Nepali alphabets and a set of commonly used words in real time.

#### **Limitations**

* Only a small set of Nepali words and static alphabets are supported.
* Dynamic gestures (moving signs) are not recognized.
* Performance may drop in poor lighting or cluttered b
* Model accuracy may vary with different hand shapes, sizes, or skin tones.

#### **6.3. Future Enhancement.**

In future, we can overcome the limitations and software by the modification.

* Support for **dynamic gesture recognition** (continuous motion-based signing).
* Add **multilingual support**, such as English, Hindi, and Nepali translations.
* Use **transfer learning** (e.g., Mobile-net, ResNet) to improve model performance.
* Provide **gesture correction tips** for invalid or unclear hand signs.

#### **6.4. References**

* Duwadi, S. (2018). Nepali Sign Language Recognition Using CNN. Deerwalk Institute of Technology. Retrieved from [https://projects.deerwalk.edu.np](https://projects.deerwalk.edu.np" \t "_new)
* Adhikari, R., & Chaudhary, N. (2024). Transfer Learning for ASL Detection using Deep CNN Models. International Journal of Emerging Technologies. Retrieved from [https://www.mdpi.com](https://www.mdpi.com" \t "_new)
* Bhandari, A., et al. (2023). AI-Based ASL to Nepali Speech Translation System. Nepal Journal of Artificial Intelligence Research. Retrieved from [https://www.nepjol.info/index.php/injet/article/view/66704](https://www.nepjol.info/index.php/injet/article/view/66704" \t "_new)
* Poudel, B. (2022). Nepali Sign Language Recognition and Translation using Deep Learning [GitHub repository]. Retrieved from [https://github.com/Birat-Poudel/Nepali-Sign-Language-Recognition-and-Translation](https://github.com/Birat-Poudel/Nepali-Sign-Language-Recognition-and-Translation" \t "_new)
* NSL23 Dataset (2023). Nepali Sign Language Dataset for Consonants and Vowels. Open Source Research Group.
* ArXiv. org. (2022). Comparative Study of CNN and SVM for Hand Gesture Recognition. Retrieved from [https://arxiv.org](https://arxiv.org" \t "_new)

窗体底端