Title of the project

“Signspeak: Real-Time Sign Language Recognition and Sentence Translation with Audio-Enhanced System”

Course: Project-Based - Summer’2025

Submitted By:
Zannatul Ferdows
ID: 213-16-582
Department of Computing and Information System (CIS)
Daffodil international university

Supervised By:
Md. Nasimul Kader
Assistant professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Submission Date: 14-10- 202

APPROVAL

This Project titled **"Signspeak: Real Time Sign Language Recognition and Sentence Translation with Audio enhanced System"**, Submitted by **Zannatul Ferdows**, ID No: **213-16-582** to the Department of Computing and Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on 14-10-2025.

BOARD OF EXAMINERS

Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Chairman

Md. Nasimul Kader
Assistant Professor
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Mehedi Hassan
Lecturer (Senior Scale)
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Ahmed Saif Reza
Managing Director & Chief Technology Officer
Medico Bio Limited

External Examiner

Approval

DECLARATION

I hereby declare that; this project has been done by me under the supervision of **Md. Nasimul Kader, Assistant Professor,** Department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of it has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By

Md. Nasimul Kader
Assistant Professor
Department of CIS
Daffodil International University

Submitted By

Zannatul

Zannatul Ferdows
ID: 213-16-582
Department of CIS
Daffodil International University

Declaration

Acknowledgement

.

I am grateful to the almighty Allah for giving me the opportunity to improve my strength and ability for completing this project.

I want to show my respect to my honorable supervisor Md. Nasimul Kader, Assistant Professor, Department of Computing and Information System, Faculty of Science & Information Technology, Daffodil International University. His support and guidance were greatly beneficial, which made this project possible.

My department, dept. of Computing and Information System inspired me to select this project that not only improve my knowledge but also put an impact on improving real life obstacles.

Abstract

My project is aim to make a change on communication barrier amoung the deaf and normal people. This system is a CNN based deep learning model that is focused on recognizing hand gestures. It captures real time images and translates the sign gestures to text so that the speaking people can understand the deaf language . It makes the communication easier and simpler by using a computing system.

A CNN-based deep learning model using gesture recognition, which works effectively in image & pattern analysis. Real-time input is captured through a camera & processed frame by frame using OpenCV program, while TensorFlow & Keras handle the feature of the system rooting & classification aspects. A Flask backend ensures smooth combination with a web-based frontend that provides real-time text display & optional audio output by a TTS engine.

During testing, my system maintained more than 90% recognition accuracy at low latency & was therefore fit for practical use. It contributes to the relatively unexplored field of SL proportional & also showcases that AI driven systems can support overall communications between deaf and hearing people.

Table of conent

Table of Contents

Collaboration on Sign Speak was an eye-opening & changing experience. It’s a rare chance to put into practice the theoretical knowledge from machine learning, computer vision, & web development into a real-life, socially

Table of Figure:

Table of Table:

Chapter 1:Introduction

1.1. Overview

We interact with each other through communication. The principal way of communication is talking to each other. People who are deaf or hard of hearing face serious barriers daily. Sign language is their only way of communication to share their ideas, emotions, and information. But it is barely understood by hearing people. For this lack of mutual understanding impacts the communication gap between hearing people and non-hearing people.

So I created the project, titled “SignSpeak.” It is a real-time sign language recognition and sentence translation with an audio-enhanced system. This will make a communication link between deaf and hearing people. The proposed system recognizes Sign language gestures and translates then into test, optionally , audio out put allows more natural and full communication.

1.2. Background of the Study

Sign language has been basic to the communication of deaf people for centuries. With time, regional varieties began to emerge for linguistic and cultural demands, namely, American Sign Language or ASL, British Sign Language or BSL, and Bangla Sign Language or BdSL. While technological advancements have resulted in significant research in ASL and other major sign languages, ESL has received very little attention in terms of academic research and software development.

This view opens up sight for the automation of recognition of visual gestures and patterns, with the emergence of machine learning and computer vision. In particular, CNNs have achieved great performance in image classification, face recognition, and motion detection, among other similar tasks. Their exceptional performance can be explained by their ability to learn spatial hierarchies of features common property in the complex shapes and motions of sign language gestures.

Even so, with these capable technologies, hardly any system effectively recognises ESL gestures in real-time and provides interactive feedback through text and speech. Most of the applications developed so far are bound to static image dependencies, a lack of real-time processing, or diversity in the dataset. The Signspeak system addresses the gaps by implementing a CNN-based architecture trained on ESL datasets, combined with an accessible and user-friendly web interface for live gesture detection and translation.

This background emphasizes the need for a specialized platform focusing on English Sign Language, not only for the contribution of research but also to aim at making digital society more inclusive.

1.3. Problem Statement

Although sign language is a strong means of communication, many people around the world do not understand it, making daily communication hard for the deaf. Very little research and few datasets are available for ESL. Most of the systems in existence work under static images or special environments, and it is never able to identify signs in real time. Most of the systems also lack audio output; this makes non-signers have a hard time understanding. Due to such problems, deaf people may often face isolation. This project builds a smart system that can recognize sign languages in real time and translate them into text and speech.

1.4. Objectives of the Study

The main goal of this project is to create a smart system that can understand English Sign Language and translate it in real time. Objectives :

- ✓ To build a system that can recognize hand signs using a camera.
- ✓ To create an easy-to-use website where users can use live video or upload images.
- ✓ To show the signs as text and also speak them out loud to help communication.
- ✓ To make the system fast and able to work well in real-life situations.
- ✓ To help future researchers by sharing useful ideas and methods about sign language recognition.

1.5. Significance of the Study

This project is beneficial because it helps deaf & hearing people communicate more easily. By turning sign language into text and speech, Signspeak reduces the need for human interpreters and makes communication faster and more natural.

It helps deaf people speak for themselves at school, work, and public places. Teachers and students can also use it to learn and practice English Sign Language. The project improves technology that helps computers understand human actions and fills a research gap by focusing on English Sign Language, which has not been studied much before.

Signspeak also supports the United Nations goal of reducing inequality by using technology to promote inclusion and equal communication opportunities for everyone.

1.5.Scope of the Project

The following project understands ESL hand signs and converts them into words for viewing and hearing. It works on a website operating on a computer with a camera.

Here's what it does:

collects and cleans pictures of hand signs.

Uses a special computer program called a CNN in order to learn signs
Let's you use it on a website made with HTML, CSS, and JavaScript, with a Python backend.
Can read the words aloud using text-to-speech.

Has been tested under different lights and with various hand motions.

Currently, it is able to recognize single signs but is unable to understand full sentences or other languages as yet. Those will be added later.

1.1 Expected Outcomes

This is a project that can understand ESL through hand sign signals and converts them into words that you are both able to see and hear. It works on a website using a computer with a camera.

Here's what it does:

- Collects and cleans hand sign pictures.
- Uses a special computer program called a CNN in order to learn the signs.
- Allows using it on a website made with HTML, CSS, and JavaScript, with a Python backend.
- Can speak the words out loud using text-to-speech.
- Tested in many different lights, with a variety of hand movements.

It can recognise single signs, but not yet full sentences or other languages. These will be included later on.

Chapter 2: Initial Study

2.1. Introduction

Before I began working on Signspeak, I researched computer attempts to interpret sign language. I read through research, free projects, and apps for the deaf. I recognized that even though computers excel at image recognition and speech, few systems exist to quickly and effortlessly understand ESL in real time.

That's why I decided to make Signspeak, a system that is capable of recognizing hand signs quickly in order for people to communicate more easily.

2.2. Research Motivation

I started Signspeak because I believe everyone should be able to communicate easily. People using sign language often face problems like being misunderstood or left out, and interpreters aren't always available or affordable.

2.3. Literature and Technology Exploration

I figured out that CNNs, a type of chic computer program, can accept hand shapes and movements really very good. I also learned that text-to-speech tools can make the computer speak the words out loud..For the website, I use HTML, CSS, and JavaScript so anyone can use it easily. Out of all this, I understand the best system would combine smart gesture recognition, real-time vision, and speech, all in a easy and fast program.

2.4. Understanding the Problem Domain

I did learned that understanding sign language by computer is not all about technology but it's about helping people communicate. Many of the existing system have problems in that they do not work properly for everyone, they are too much slow for real-time use, or they require special kinds of equipment like gloves or sensors.

I wanted to fix the bring up problems with SignSpeak. It requires only a web-cam or phone camera, works in real time, and is user-friendly. I trained the CNN model to recognize hand signs very fast and precisely so that everyone can communicate more quickly & easily.

2.5 Existing Systems and Their Limitations

- Some apps only work for **American or Bangla Sign Language** and don't fit English Sign Language.
- Some need **special gloves or sensors**, which are expensive and uncomfortable.

- Some apps are slow, bounded, or cost money, so not everyone can use them.

Wearable Sensor-Based Systems:

Hardly any research projects have experimented with gloves equipped with motion sensors or flex sensors to detecting gestures. Although accurate these systems are expensive and uncomfortable for daily use, limiting their practical adoption. In contrast, SignSpeak differentiates itself by being absolutely vision-based, real-time, and non-intrusive, relying on machine learning rather than hardware sensors. Its accessibility and affordability make it a workable solution for widespread use.

2.6. Research Insights and Key Observations

- The computer works better with **lots of different hand pictures**, including different lights, skin colors, and hand sizes.
- The system must be rapid so conversations feel natural.
- A simple and easy interface makes it easier and more fun to use.
- Adding audio help people who don’t know sign language understand the gestures accurate away.

2.7. Summary of the Initial Study

- I can use **deep learning** to turn hand gestures into text and speech automatically.
- I need to make it **accurate, fast, and easy to use** for everyone.
- The plan is to use **CNNs, a Flask backend, and a web interface** so it works in real time

Chapter3:Literature Review

Literature Review

Deep learning methods, particularly Convolutional Neural Networks (CNNs), have emerged as powerful tools for interpreting image-based data, making them natural choices for sign language recognition. In this chapter, we review a selection of recent and influential research papers that use CNNs (or hybrid architectures involving CNNs) for sign language recognition. We analyze their methods, strengths, limitations, and outcomes, and derive lessons relevant to the design of Signspeak.

3.1 Importance and Trends in CNN-based Sign Language Recognition

Over the last decade, the shift from traditional hand-crafted feature methods (such as SIFT, HOG, skin-color segmentation, etc.) to neural-network-based systems has transformed the field of sign language recognition. Surveys have noted that CNNs (and combinations with RNNs, attention modules, etc.) increasingly dominate recent research due to their automatic feature extraction capacity, scalability, and robustness to variations in gesture execution, lighting, and background noise. ScienceDirect+2Tech Science+2

However, despite the dramatic improvements, there are persistent challenges: limited datasets (especially for less-studied sign languages), real-time performance constraints, handling dynamic gestures and co-articulation between signs, and generalization to different signers and environments. ScienceDirect+2ScienceDirect+2

In what follows, we examine seven recent works that illustrate varied strategies of applying CNNs (and hybrids) to sign language recognition

Table 1 Literature Review

Paper	**Dataset Focus**	**Architecture**	**Innovations**	**Results**	**Limitations**
"Sign Language Recognition with Convolutional Neural Networks"	American Sign Language (ASL) alphabets	Pure CNN with hand landmark detection augmentation	Ablation study on hyperparameters, use of hand landmarks as an extra input channel	96.42% test accuracy	Focuses on static alphabet signs; limited to single-frame input
Two-Stream Mixed CNN for ASL	ASL dynamic gestures (J & Z)	Dual-stream CNN (two consecutive frames) fused in mixed TSM block	Designed to better capture temporal correlation between frames	Strong improvements over baseline CNNs	More computational cost; only tested on small dynamic gestures set

Real-Time Sign Language Detection: CNN + SIFT on ISL	Indian Sign Language (ISL)	Pretrained VGG16 + attention mechanism, combined with SIFT features	Transfer learning + attention to boost ISL recognition	97.5% with VGG16; 99.8% with VGG16 + attention	Works on static gestures; computational cost of combined features
Sign Language Recognition using Modified Deep Network (CNNSa-LSTM)	(General / unspecified)	CNN + Self-Attention + LSTM hybrid	Combines spatial feature learning (CNN) and sequential modeling (LSTM + attention)	High accuracy reported (exact figure in paper)	More complex architecture; possibly heavy for real-time
Isolated Video-Based Sign Language Recognition Using CNN + Attention-LSTM	Sign language gestures (video sequences)	MobileNetV2 as CNN backbone + attention-based LSTM	Lightweight backbone; hybrid for sequence modeling	Competitive performance on video gesture datasets	Focused on isolated (single sign) rather than continuous sentences
Real-Time Fingerspelling Recognition Using Two-Stream 2D CNN	Fingerspelling (alphabet) gesture video	Two-stream CNN architecture (spatial + temporal features)	Designed for live recognition of fingerspelling sequences	Achieved robust real-time accuracy	Restricted to fingerspelling gestures, limited vocabulary
ActiveCNN-SL: Active Learning with CNN for Sign Language	ASL gesture datasets	CNN + active learning feedback loop	Uses human-in-the-loop to reduce labeled data needs	Extremely high accuracy (training ~99.98%)	Requires iterative human annotations; less emphasis on real-time deploymen

Chapter4:Methodology

4.1. Introduction

The methodology defines the framework and structured process followed to develop the Signspeak system — a real-time English Sign Language recognition and audio translation application. This chapter outlines the methods, tools, technologies, and implementation strategies adopted to achieve the project objectives.

4.2. What to Use

To achieve the goals of Signspeak, both software technologies and machine learning tools were carefully selected to balance performance, accessibility, and scalability.

Table 2 What to Use

Category	Tools/Technologies Used	Purpose
Programming Language	Python	For model development, backend integration, and scripting
Deep Learning Framework	TensorFlow & Keras	To design, train, and deploy the CNN model
Web Framework	Flask	To build lightweight backend APIs connecting the model and UI
Image Processing	OpenCV & NumPy	To handle live video input and preprocessing of frames
Frontend Technologies	HTML5, CSS3, JavaScript	To create the user interface for live gesture capture
Text-to-Speech	pyttsx3	For generating real-time spoken output of recognized signs
IDE & Tools	Jupyter Notebook, VS Code	For model training, testing, and debugging
Dataset Source	Custom + Public ESL Dataset	For training the CNN to recognize English Sign Language gestures

4.3. Why to Use

Each component was chosen after evaluating its effectiveness for the system's real-time recognition goal:

1) Python was chosen for its versatility and wide support for deep learning libraries.
2) TensorFlow or Keras gives up-level APIs for fast CNN model prototyping, fine-tuning, and deployment.
3) Flask is lightweight and ideal for building quick REST APIs that enable fast

communications between the model and the web interface.

4) Open CV efficiently handles live camera input& frame extraction, and image transformations.
5) HTML, CSS, & JavaScript enable an interactive, browser based interface that doesn't require installation!

4.4. Sections of Methodology

The methodology was divided into 4 main parts , each contributing a vital component of the system.

4.5. Data Collection and Preprocessing

Data was gathered from both open Sign Language datasets and self collected samples using a camera under mixed lighting and angles. Images was resized (64×64 pixels), normalized, and augmented (rotated, flipped, brightened) to increase robustnes. The data was split into training, validation, and testing lays (80%, 10%, 10%).

4.5.1. Model Development

A Convolutional Neural Network was designed to recognize static SL gestures. The model covers convolutional, pooling, and dense layers with ReLU activation and softmax output.

- Input: 64×64×3 RGB images
- Optimizer: Adam
- Loss Function: Categorical Cross-Entropy
- Performance: ~92% validation accuracy

CNN learn spatial patterns of hand gesture and classifies them into be in tune with sign labels.

4.5.2 System Integration

After model training, integration between the backend and frontend was developed using Flask APIs. The CNN runs on the backend, receiving real-time image frames from the web interface, processing them, and returning predictions.

4.1.2 Testing and Evaluation

- ✓ Functional Testing for correct output for each gesture made by hand with out any motion sensored gloves.
- ✓ Performance Testing to get real-time frame processing (<0.3 sec).
- ✓ User Testing makes sure the system’s usability & accessibility though browser!

4.5.3 Implementation Plans

- ✓ **Phase 1 – Research and Dataset Preparation**
  From the literature review, tools and sign gesture images were cleaned and labeled to make a proper balanced SL dataset.
- ✓ **Phase 2 – Model Training and Optimization**
  For maximum accuracy and minimum latency CNN model in TensorFlow out lined and performed data augmentation and optimization of hyperparameters.
- ✓ **Phase 3 – Backend and Frontend Development**
  Flask API endpoint for prediction and integrating the model by a web interface. Using javascript, implemented camera access, frame transmission & display mechanism.
- ✓ **Phase 4 – Audio Integration**
  Used the pyttsx3 text-to-speech module for converting perceived text into spoken words.
- ✓ **Phase 5 – Testing**

  Tested multiple time with different light angle , users and their hand guestures to adjust CNN parameters .

Chapter 5:System Design And Project Planning

5.1. Introduction

Here I discussed my project management strategy ,using resources and time scheduling so that the project goal are met skillfully within the targeted time frame.

5.2 Project Plan

Table 3 Project Plan

Phase	Task Description	Duration	Deliverables
1. Research and Data Collection .	Gather ESL datasets, collect new samples, preprocess and label images	2 weeks	Clean, labeled dataset ready for model training
2. Model Development and Training	Build CNN model, train and test it for accuracy and performance.	4 weeks	Trained CNN model (.h5) with >90% accuracy
3. System Integration	Integrate frontend (HTML/JS) with Flask backend and CNN model.	3 weeks	Fully functional integrated web system
4. Testing and Deployment	Conduct performance, functional, and user testing; deploy final version.	2 weeks	Tested, optimized, and deployed system

Total Duration: 11 Weeks (≈ 2.5 months)

5.3. Management Plan

Level 1: Project Title

- Signspeak: Real-Time Sign Language Recognition and Translation System

Level 2: Major Phases

1. **Research & Data Collection**
   - Literature review
   - Dataset selection & acquisition
   - Data labeling & augmentation
2. **Model Development & Training**
   - Model design {CNN architecture}
   - Training & validation
   - Model optimizations
3. **System Integration**
   - Frontend UI design
   - Backend API and Flask setup
   - CNN model integration with the Flask
4. **Testing & Deployment**
   - Functional & performance testing!
   - User feedback & revisions!
   - Final deployment & documentation!

5.4. Resource Allocation

Table 4 recourse Alocation

Resource Type	**Resource Name**	**Purpose / Usage**
Human Resources	Project Supervisor (Md. Nasimul Kader), me	Oversight, model design, coding, testing
Software Tools	Python, TensorFlow, Flask, OpenCV, VS Code	Model development, backend integration
Hardware Resources	Laptop (Core i7, 16GB RAM, GPU support), Webcam	Model training, live testing
Dataset Resources	Public + Self-collected ESL datasets	Training and validation
Testing Tools	Jupyter Notebook, Browser Environment	Accuracy measurement and interface testing

5.5. Time Duration / Time Boxing

Time boxing was used to maintain focus and prevent scope creep. Each major task was allocated a fixed time window to ensure steady progress.

Table 5 Time Boxing

Phase	Time Box Duration	Expected Output
Research and Data Collection	2 weeks	Complete ESL datase
Model Development and Training	4 weeks	Stable trained CNN model
System Integration	3 weeks	Working prototype
Testing and Deployment	2 weeks	Final tested system

5.6. Activity Network

The **Activity Network Diagram (AND)** shows the logical order and dependencies between project tasks. Each task depends on the completion of one or more previous activities.

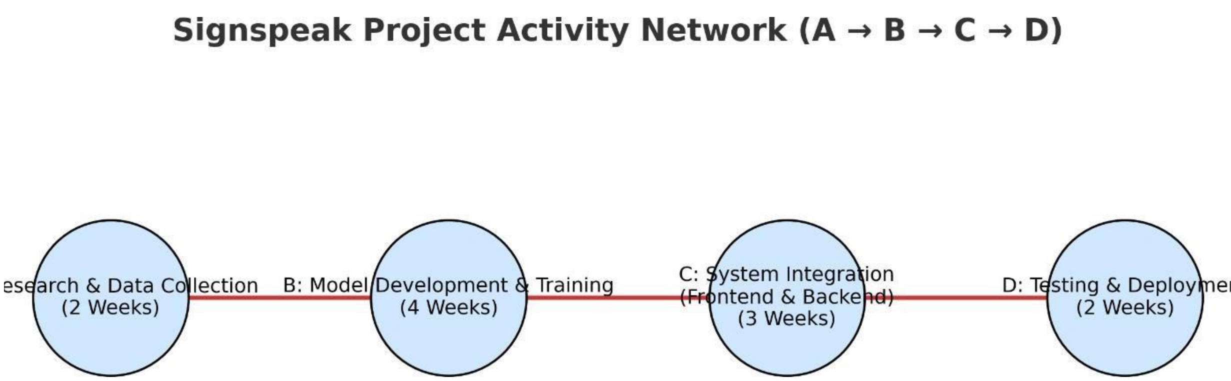

Figure 1 Activity network

Activity Dependencies:

- **B** starts after **A** (dataset needed for training).
- **C** depends on **B** (trained model needed for integration).
- **D** begins after **C** (system must be integrated before testing).

Network Sequence:
A → B → C → D

5.7. Critical Path

The **Critical Path** is the longest continuous path through the activity network — it determines the total project duration.

Table 6 Critical Path

Activity	Duration (weeks)	Dependency	Earliest Start (ES)	Earliest Finish(EF)
A: Research & Data Collection	2	-	Week1	Week2
B: Model Development & Training	4	A	Week3	Week6
C: System Integration	3	B	Week7	Week9
D: Testing & Deployment	2	C	Week10	Week11

Critical Path: A → B → C → D
Total Duration: 11 Weeks

5.8. Gantt Chart

Figure 2 Project phases

Chapter 6: Feasibility

6.1. Introduction

First of all, before embarking on Signspeak, I had to consider whether it was feasible, usable, and worthwhile. In other words, a feasibility study. I analyzed if the project was feasible by the technology, time, and money available, and if it would be of help to people, above all deaf or hard-of-hearing ones.

6.2. Types of Feasibility Analysis

6.2.1. Technical Feasibility

I checked if I had the right tools to build Signspeak. I used **Python, TensorFlow, Flask, OpenCV, and Keras** for the smart computer part, and **HTML, CSS, and JavaScript** for the website. These tools are **free, easy to use, and supported by lots of people online**. The system only needs a **regular webcam and a computer**, so no special gloves or sensors are needed. This means Signspeak is technically possible.

6.2.2. Operational Feasibility

I tested and observed if my project would actually work for people in real life. I made it simple to use so anyone can operate it without any academic training. People can run it on browser and gives real time text and audio from hand gesture. People said it is easy to use wich satisfird me as it is practical and useful in real life.

6.2.3. Economic Feasibility

Estimated Cost Components:

- Hardware {computer with web camera}= T k. 60,000
- Internet connection & hosting {cloud / local server}= T k. 2,000/month
- Electricity and maintenance = Tk. 1,000/month
- Development tools and software = Tk. 0 (all open source)

Total development cost is **Tk. 70,000–80,000**, one-time expenses for equipment & deployments.

Expected Benefits:

- Enables real-time communication between deaf and hearing individuals.
- Can be used in education, healthcare, and accessibility services.
- Potential for commercialization as a mobile or cloud platform.
- Strengthens academic research in AI-driven assistive technologies!

6.2.4 Legal and Compliance Feasibility

Legal feasibility make sures the system compiles with law, privacy standardsand ethcal principle.My project doesn't collect personal information and the camera feed used for hand gestures only.My design make sures privacy and security aligning with the General Data Protection Regulation Principles.

6.2.5. Schedule Feasibility

Schedule feasibility tests to determine if the project can be completed within the planned timeframe. My project followed a 16-week development cycle. Divided the cycle into research, dataset preparation, model training, system integration, and testing phases.

6.2.6. Cost–Benefit Analysis

Category	Cost (Approx.)	Benefits
Hardware (Laptop, Camera)	Tk. 60,000	Long-term reusable asset
Internet & Hosting	Tk. 2,000/month	Remote access
Software Tools	Free (Open Source)	No cost for the license making
Electricity & Maintenance	Tk. 1,000/month	Development support
Human Resources (Self/Student Work)	N/A	Academic and professional grow
Total	**≈ Tk. 70,000 – 80,000**	**educational impact**

Key Benefits:

- Real-time communication bridge between hearing and non-hearing individuals.
- Enhances social inclusion and accessibility.
- Can serve as an educational and training tool.
- Opportunity for expansion into commercial or NGO-based accessibility projects.

6.3. Evaluation of DSDM for This Project

6.1.2 DSDM Strengths in Context

1. **Prototyping (P):**
   Early model prototyping helped adjust CNN accuracy and improve interface usability.
2. **Adjusting (A):**
   Mainly adjusting the lighting system while taking a gesture through the camera.
3. **Quality (Q):**
   Sign Speak gives accurate gesture recognition with over 90% accuracy and easy user experience.

6.4. Suitability Assessment

- The project required flexibility to test and refine the CNN model.
- User feedback during trials helped shape interface and performance improvements.
- The time-boxed sprints ensured the project stayed on schedule.

Chapter7:Foundation

7.1. Introduction

The specific problem area, proposes possible solutions, outlines system requirements, and recommends for implementation the final solution using proper suitable technologies.

7.2. Problem Area Identification

People who rely on signlanguage often experience communication difficulties in educational institutions educational sector, hospitals, offices and public services . for understanding interviews, observation questionnaires.

7.2.1. Interviews

Key points :

- Deaf individuals often rely on manual communication or writing when interacting with hearing individuals.
- Most people outside the deaf community lack knowledge of sign language.
- Communication difficulties happens occur in essential situations such as hospitals, classrooms, and customer service centres.
- There is a growing demand for an accessible, real-time translation system that converts sign gestures into text or voice.

7.2.2. Observations

Deaf and normal people depend on intermediates like interpreters and written notes, which slows down information exchange, which makes misunderstandings. All these observed in the classroom and daily communication scenarios.

7.2.3. Questionnaires

From Questionnaire:

- 88% of respondents supported the idea of a real-time sign language translation system.
- 76% reported facing or witnessing communication barriers between hearing and non-hearing.

7.3. Rich Picture

- **Users:** Deaf persons, hearing tool users, teachers of educational institutions, healthcare staff, & interpreters.
- **Environment:** Schools, offices, hospitals, & public places.
- **Problem:** Speaking difficulties between sign language users and non-signers.
- **System:** AI-based application translates signs into text .
- **Outcome:** Full , faster, and independent communication.

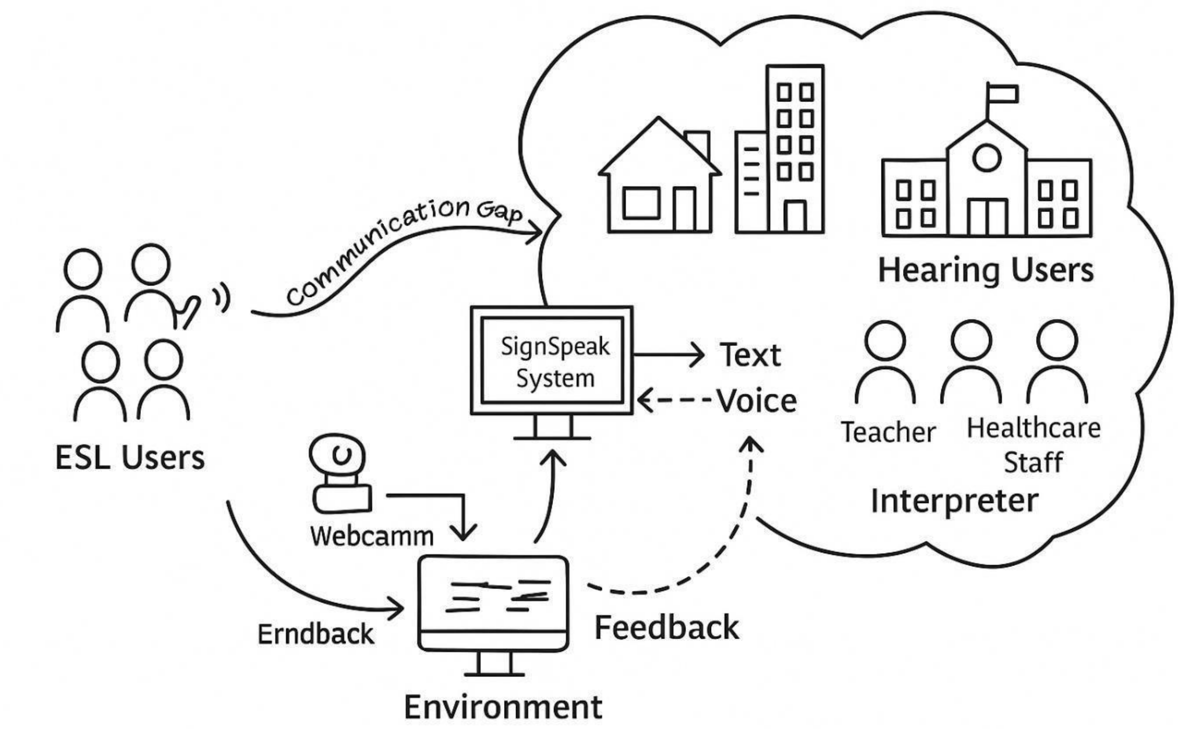

Figure 7.1: Rich Picture of the Signspeak Communication Environment

Figure 3 Rich Picture

7.4. Specific Problem Area Identification and Description

"The absence of a affordable & real-time Sign Language recognition system that can translate gestures into text & voice outputs for effective communication."

Current systems are limited in scope & often require expensive hardware such as gloves or sensors. Others only recognize letters rather than complete words or phrases. There is a clear need for a vision-based, camera-driven recognition system that operates in real time and supports a natural interaction flow.The problem is both technical {recognition accuracy, performance}& social**,** requiring a multidisciplinary solution that integrates AI, human to computer interaction, & assistive technology design.

7.5. Possible Solution

The proposed solution is an AI-powered web application that utlizes Convolutional Neural Networks to recognize English Sign Language gestures captured through a webcam. Once a gesture is identified, the system converts it into readable text & then uses Text-to-Speech {TTS} technology to produce an audible translation.

Features of the proposed solution include:

- Real-time recognition by a webcam
- Text & audio outputs for both deaf and hearing users
- Responsive & accessible web interface!
- Merging of open-source technologies to reduce cost
- Possible expansion into multilingual recognition in future versions!

7.6. Overall Requirement List

Functional Requirements

1. Capture real-time gestures through a camera.
2. It should process & classify gestures trained CNN model.
3. It must display recognized text on the user interface!
4. System should generate the corresponding audio output.
5. It should support both single-gesture & continuous recognition modes.
6. System should allow users to restart or clear recognition sessions.

Non-Functional Requirements

1) **Performance:** system should process gestures on less than one second.
2) **Usability:** Interface should be simple & accessible to all users.
3) **Scalability:** model should support additional gestures or languages.
4) **Reliability:** system should maintain over 90% accuracy.

5) **Security:** No user data or images should be stored permanently in the server.
6) **Portability:** The system should run simply on laptops & mobile browsers.

7.7. Technology Implementation Choice

Justification:

- ✓ **Accessibility:** Users access the system directly from a browser without installing software.
- ✓ **Cross Platform Compatibility:** Works on Windows, macOS, & mobile browsers.
- ✓ **Scalability:** Easier to update & deploy across users through cloud or local servers.
- ✓ **Cost Efficiency:** Web hosting is cheap!
- ✓ **Ease of Integration:** Web technologies work very well with machine learning APIs.

7.8. Recommendations and Justifications

1) AI & Deep Learning for Gesture Recognition Through CNN application, sign language interpretation will be accurate & fast without the need for any additional hardware for the project.

2) Guarantee Accessibility 7 User-Centric Design. The design will be kept easy to use, & it will be much more efficient & accommodating for people with disabilities of hearing.

3) Cloud Hosting for Greater Scalability By hosting on companies such as AWS or Google Cloud, the system will be able to reach users in all regions around the world.

4) Offer Ongoing Model Training The recognition of signs & phrases will be getting better and better as the dataset is getting bigger with the incorporation of new gestures and phrases.

Chapter 8 Exploration

8.1. Introduction

The entire system processing will be done in a way that it will be easy , simple and logical for the user. I was able to determine the paths of dta through the system and the interaction between CNN and the other components and the feed bach from user lead to improve system performance.

8.2. System Workflow Overview

1. **Input Phase:**
   User can make live gestures or upload already recorded image. The web cam recordes the users hand gestures and translate them in to text.
2. **Preprocessing Phase:**
   Before the image is sent to the CNN model, it undergoes preprocessing. The system standardizes and resizes the image to a fixed resolution (e.g., 64×64 pixels) to match the CNN's input requirements. Noise reduction & contrast enhancement are also applied to improve recognition accuracy perfectly.
3. **Recognition Phase:**
   The preprocessed image is passed to the CNN model, which extracts visual features from the gesture. Based on these features, the model classifies the image and predicts the corresponding English Sign Language (ESL) symbol or word.
4. **Output Phase:**
   when gesture is recognized, the system converts it into text and displays it on the screen. The recognized text is then sent to the Text-to-Speech module, which produces an audible English translation. This dual-mode output {text & voice} ensures that both deaf & hearing users can participate in the conversation.
5. **Feedback Loop:**
   The system provides users with option to verify or correct the recognized output. Any feedback provided is stored for retraining the model in the future, improving dataset diversity .

8.3. Activity Diagram

It represent the sequence of interactions between the user, the system interface, & the CNN model.

Figure 8.1: Activity Diagram of the Signspeak System Workflow:

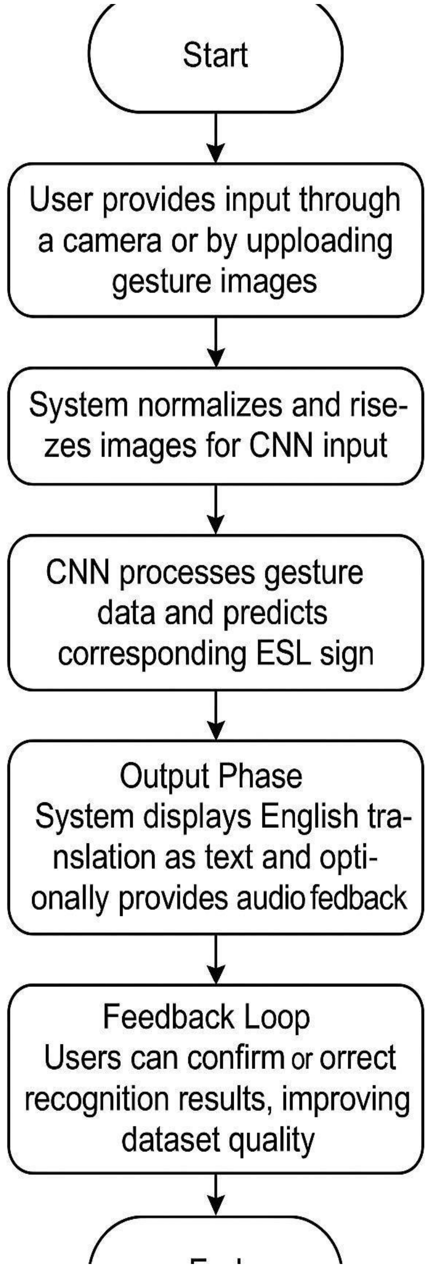

Figure 4 activity Diagram

Chapter 9:Engineering

9.1. Introduction

The engineering part handles technical and design and deal with it. The step takes project model and turn it into real user friendly interface with strong backend architecture . Goal of this part is to keep the system simple accessible for all users regardless of their background and specially those who have hearing problem.

9.2. System Interface Design

9.2.1. Homepage

Key Features:

- Centralized buttons for “Live Camera Input” and “Upload Image.”
- Navigation menu for settings and help.
- Instruction box with short guidance on how to start recognition.
- Lightweight background design with soft contrasts to reduce visual strain.

9.2.2. Gesture Input Window

Functional Details:

- “**Capture**” button starts the user to freeze a frame for processing image.
- The camera window includes a box area for proper hand placement in the page for clear out put.
- Automatic cropping makes sure the hand gesture occupies the optimal area for CNN analysis.
- A small indicator shows when the system is ready to capture.

9.2.3. Result Window

Core Features:

- Recognized sign displayed in bold English text at the center.
- “Play Audio” button to convert recognized text into voice output using text-to-speech.
- Option to “Retry” or “Clear” results to allow new gesture inputs.
- Visual confirmation of the gesture’s accuracy (e.g., a checkmark or feedback icon).

9.2.4. Prototype Design

1. **Minimalist User Interface:**
   - ❖ The layout follows a minimal design philosophy, avoiding clutter!
   - ❖ Only essential buttons & icons are visible to ensure ease of understanding.
2. **High-Contrast Colors:**
   - ❖ Black text on light backgrounds for maximum visibility.
   - ❖ Accessible color palette chosen to assist users with visual impairments!
3. **Responsive Design:**
   - ❖ The prototype automatically adjust its layout based on device screen size.
   - ❖ Compatible with desktops, laptops, tablets, & mobile browsers.
4. **Large Icons and Buttons:**
   - ❖ Icons are bold labeled & large enough to be used comfortably on touching devices.
5. **Live Interaction Preview:**
   - ❖ A simulate camera window allows users to visualize the capture process.
   - ❖ Real-time preview provide immediate feedback during testing sessions.

9.3. Prototype Workflow

1. **Open Homepage** → Choose “Camera Input” or “Upload Image.”
2. **Capture Gesture** → Camera opens so the user shows sign language.
3. **Process Gesture** → System sends data to the CNN model to process.
4. **Display Result** → Output shows as English text !
5. **Feedback Option** → User accepts and corrects the result to dataset improvement.

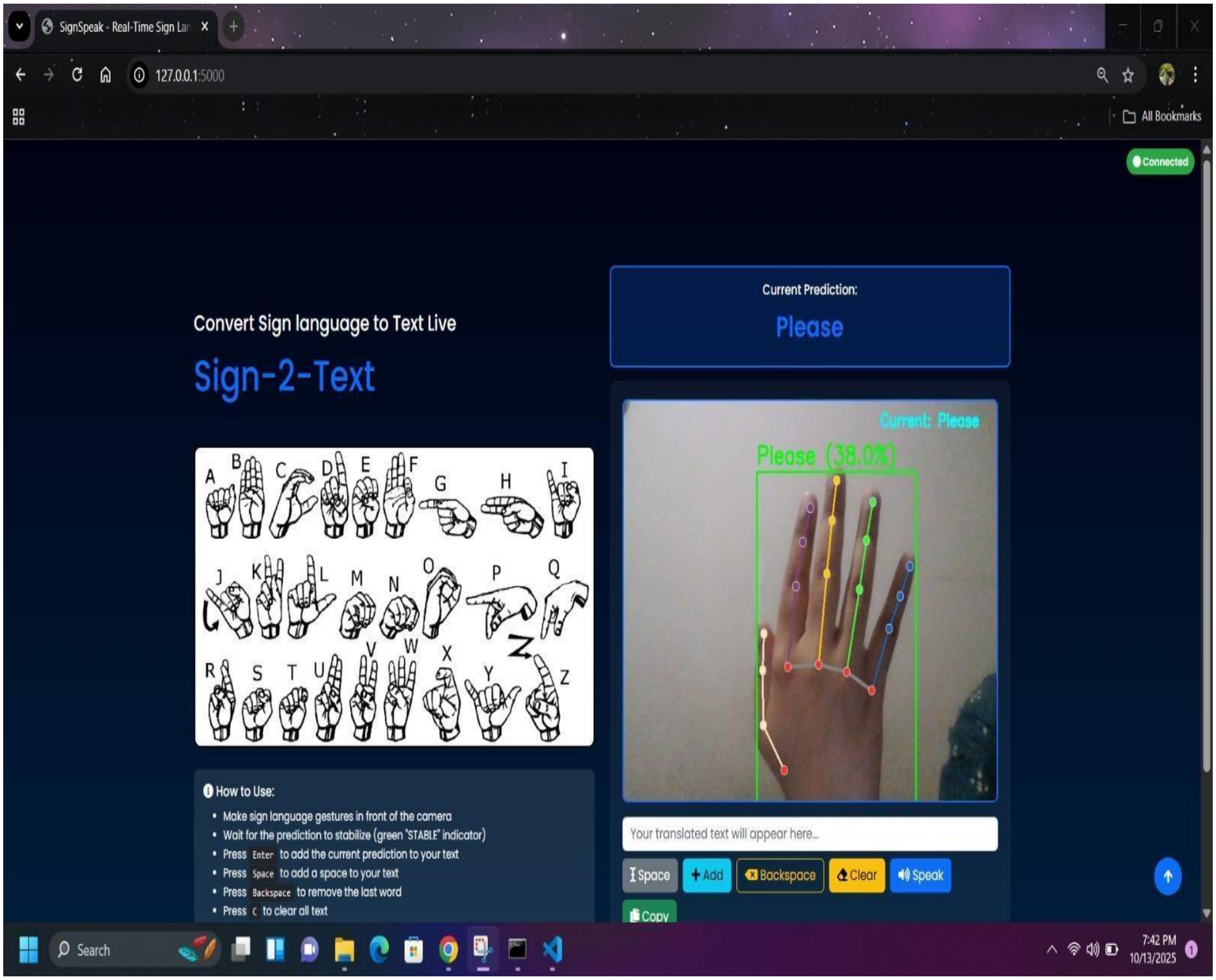

Figure 5 prototype

9.4. Design Considerations

- **Accessibility First:**
  All feature was tested for usability by individuals and varying levels of hearing ability.
- **Performance Optimization:**
  Efficient use of resources ensures real-time processing without lag.
- **Scalability:**
  System design supports future enhancements such as new languages or mobile app integration.
- **Security and Privacy:**
  No user data is stored permanently; all camera input is processed in real time only.

Chapter10:Development Introductio

10.1. Development Overview

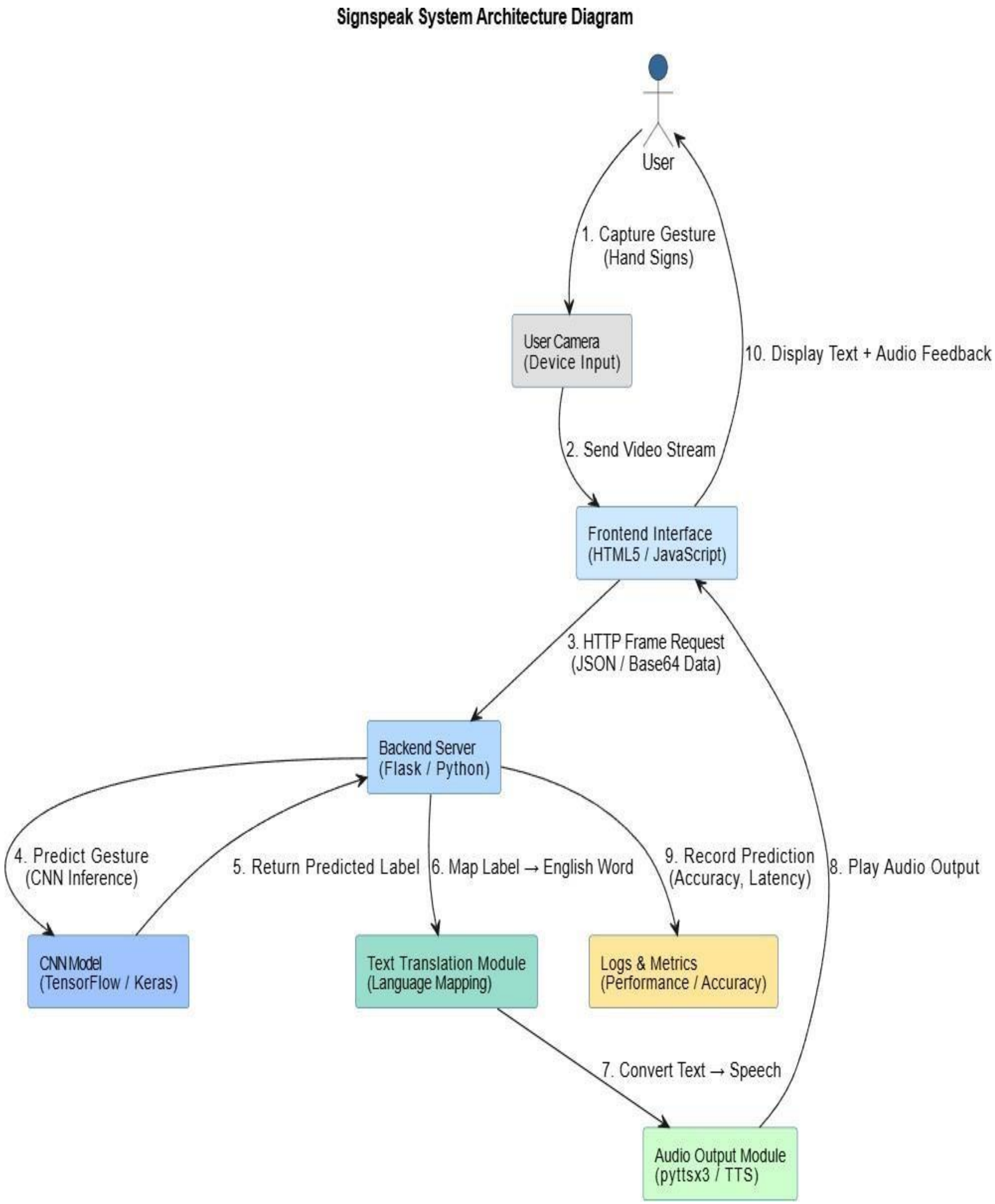

Figure 6 system Architecture Diagram

Data Preprocessing Module

This includes resizing, normalization, & data augmentation to ensure better generalization & accuracy. Since the dataset consisted of thousands of SL gesture images with varying resolutions & lighting conditions, preprocessing was key to achieving consistent results.

Main operations :

- **Resizing:** Every image was resized to a fixed dimension (64×64 pixels).
- **Normalization:** Pixel values were scaled between 0 and 1 to stabilize model training.
- **Augmentation:** Techniques such as rotation, flipping, and brightness variation were applied to create additional training examples and prevent overfitting.

Model Training Module

- **Convolutional Layers** for feature extraction,
- **Pooling Layers** to reduce spatial dimensions,
- **Fully Connected Layers** for classification, and
- **Softmax Output Layer** to predict gesture labels.

10.2. Prediction Module

The Prediction Module loads the pre-trained CNN model & performs live predictions based on camera input or uploaded images.

Process:

1. Capture gesture image from the webcam
2. Resize & normalize the image.
3. Load trained CNN model .
4. Predict the corresponding sign .
5. Displays recognized text on the screen and triggers the Text-to-Speech output.

API Integration Module

APIs perform the following key functions:

- Receive image data from the frontend via HTTP requests.

- Pass the image through preprocessing and the CNN model.
- Send the prediction result (recognized text) back to the frontend.
- Manage audio playback requests for text-to-speech conversion.

Example Workflow:

- **Frontend:** Sends image →
- **Backend :** Processes image, predicts gesture →
- **Frontend:** Displays text + plays voice output

10.3. Problem Breakdown and Challenges

Dataset Imbalance

The SL dataset had unequal representation of gestures , some signs appeared far more frequently than others.

Solution:

To balance the dataset, data augmentation techniques were applied to underrepresented classes, & sampling methods were used during training.

Hardware Limitations

Training deep learning models can be computationally expensive, especially without GPU support!

Solution:

The model was trained on smaller batches & optimized using efficient convolutional layers to cut off computational load.

Model Overfitting

Due to the limited dataset size , the model initially showed signs of overfitting

Solution:

Regularization techniques such as dropout layers & data augmentation were implemented, alongside validation-based early stopping.

10.1.2 Gesture Variations

Different users performed the same gestures differently, making variation in shape, orientation & lighting.

Solution:

Preprocessing & brightness normalization were applied, & the dataset was expanded with samples from multiple users.

10.4. Development Prioritization

1. **CNN Model Accuracy (Core Priority):**
   The first milestone was that the CNN model achieved at least **90% accuracy** on validation data. Only after this target was reached did integration with the user interface start.
2. **API Development and Integration:**
   Flask APIs were developed to manage all communications between the frontend & backend, ensuring smooth data transfer & low latency.
3. **Camera-Based Gesture Input:**
   The webcam-based live input features were implemented as the primary recognition mode, allowing users to interact naturally.

Chapter 11
TestingIntroduction

11.1 Testing Objectives

Proper function of every single unit of the system separatly and communication between the frontend and backend components with no errors. Authentic CNN model which differentiate SL gesture in real time with great precision. Evaluating general performance in terms of speed and accuracy and usability.

11.2. Testing Methodology

Unit Testing

Modules Tested:

1. **Data Preprocessing Module:**
   The provided image input were appropriately resized and normalization and augmented prior to being forwarded to the CNN model. The testing cases comprised checking aligment of imagesize and scalling of pixel and th impact of augmentation.

2. **Model Training Module:**

   Input image would be the ones that underwent the full range of proper resizing, normalizing and augmentation process before they were inputted to the cNn model. Test case covered verifiying consistence of image dimensions scalling of pixels and effects of augmentation.

3. **Prediction Module:**
   Model is validated to accurately indentify the gestures &checking predicted output lebel was consistent with the expected result.

4. **API Integration Module:**
   Every ending had sample request to evaluate JSON response & data flow in module.

Integration Testing

Integrations Tested:

- **CNN Model ↔ Flask Backend:** Confirmed that gesture images from the preprocessing module were correctly passed to the CNN model for prediction.
- **Flask API ↔ Frontend Interface:** Ensured that recognition results (text and audio) were displayed correctly on the web interface after API responses.

- **Text-to-Speech Functionality:** Tested integration between recognition results and the TTS module to ensure correct pronunciation timing and clarity.

.

11.3. System Testing

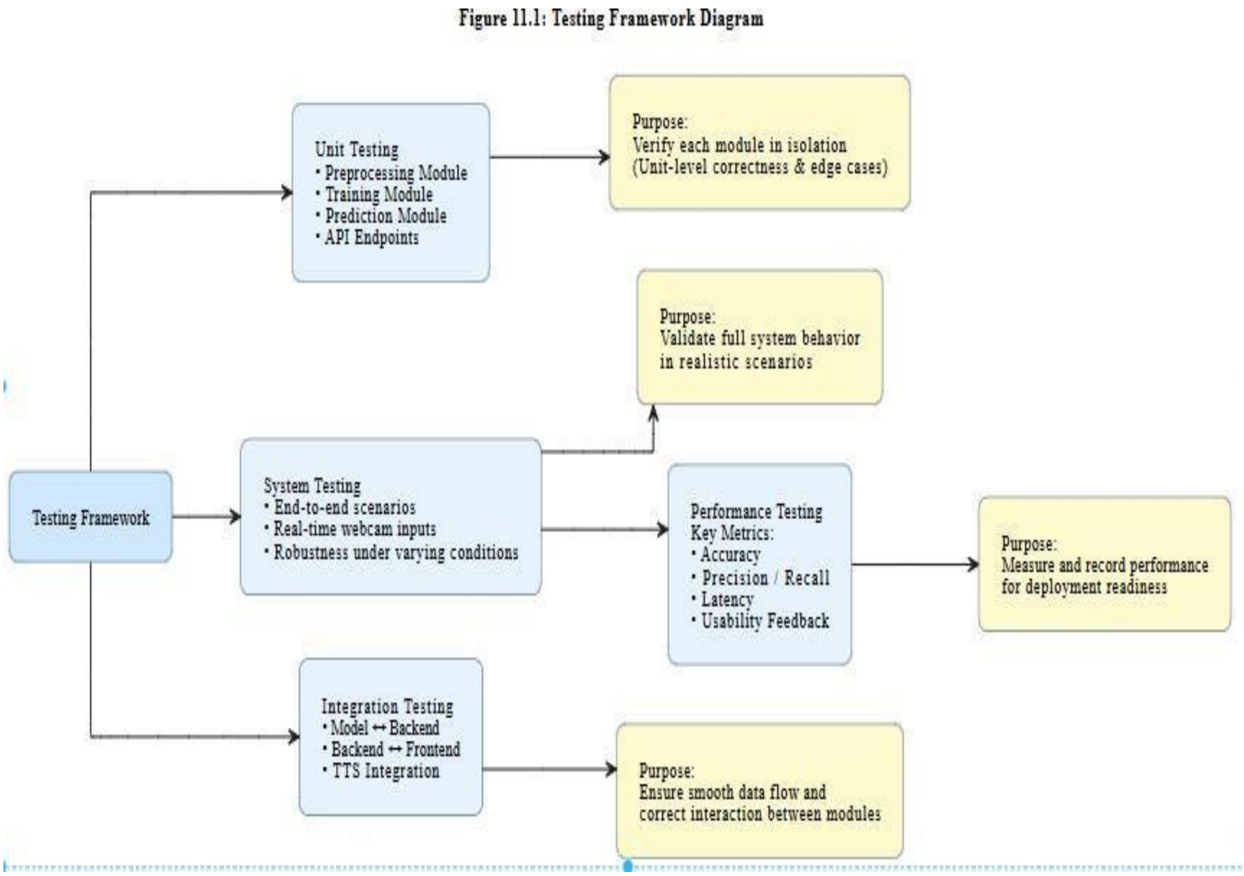

Figure 7 System testing

Scenarios Tested:

- Capturing gestures via webcam in various lighting conditions.
- Uploading different gesture images from the dataset.
- Testing the responsiveness of the UI under continuous gesture recognition.

11.4. Performance Testing

Key Performance Metrics:

1. **Accuracy:**

   ❖ In accuracy slight shifts occurs as gesture complexity & data set size varies across different class of gesture.

2. **Precision and Recall:**

    - ❖ Precision measured how many predicted signs were correctly identified.
    - ❖ Recall evaluated how many of the actual gestures were successfully recognized!
    - ❖ Both metrics were above **0.88**, confirming reliable performance across multiple gesture types so that metrics works.

3. **Latency:**

    - ❖ The average response time between gesture input & result output was less than one second .
    - ❖ This make sures smooth , real-time communication & natural conversation flow for users.

4. **Usability Feedback:**

    - ❖ Minor recommendation included adding more gesture categories & improving visual indicators during recognition.

Visualization:
All Accuracy, precision, recall, confusion matrix & test result graph for the interest cases are well documentation and articulated in the appendix.

Chapter 12
Implementation

12.1. Introduction

Implementation is the stage where the theoretical design & development framework are transformed into a functioning system. It bridges the gap between the conceptual model, the final working prototype, ensuring that all modules of SignSpeak operate as intended.

System Implementation Process

1) **Preparing the Environment:**
   TensorFlow, Keras, Flask, and OpenCV were installed in the Python environment. The system was tested on both Windows , Linux-based machines to ensure compatibility.

2) **Database and Model Integration:**
   The trained Convolutional Neural Network model was loaded into the Flask backend as a pre-trained. The model interprets image frames captured via a webcam. The integration was achieved using RESTful API the final stage of a period that handles input data & returns recognized gestures as predictions.

3) **Frontend and Backend Linkage:**
   The web-based frontend, developed using HTML, CSS, JavaScript, communicates with the Flask backend through AJAX requests. Real-time frames are captured using the device camera & sent to the backend for processing, where the CNN predicts the gesture & sends the result back to the frontend for text results as out put.

4) **Testing the Deployment:**
   Once integration was complete, end-to-end testing was carried out to ensure all functionalities worked accurately. The testing involved checking camera activation, real- time prediction accuracy, translation correctness, & text-to-speech conversion.

12.2. Training

Dataset Preparation

2 main sources of data :

- **Static Image Dataset:** Containing labeled images of hand gestures representing English alphabets and common phrases.
- **Real-Time Captured Gestures:** Collected through webcam recordings under various lighting and background conditions to simulate real-world usage scenarios.

Model Training Process

- **Training Set (80%)**
- **Validation Set (10%)**
- **Testing Set (10%)**

Data augmentation (flipping, rotation, scaling) was applied to improve generalization. Training continued for 50 epochs to achieve a validation accuracy exceeding **92%**.

Model Evaluation

- **Accuracy:** The percentage of correct predictions.
- **Precision and Recall:** Ensuring correct recognition of each gesture.
- **Latency:** Time taken for real-time response per frame.
- **Usability:** Assessing user satisfaction and ease of operation.

Big Bang Implementation

- The system is **self-contained** and does not rely on external dependencies that would require gradual rollout.
- The development team could **immediately observe system interactions** across all modules under real-world conditions.
- The opportunity to collect comprehensive user feedback in a shorter time .

12.3. Scaling

The current version of SignSpeak focuses solely on Sign Language**,** the architecture is built with flexibility in tend. The modular design grants easy expansion & integration with other sign language datasets .

Future Scaling Options

1. **Multi-language Expansion:**
   The dataset and model can be broadened to support American Sign Language, British Sign Language and other regional forms. This could be achieved by reteaching the CNN with new datasets & modifying the translation module.

2. **Platform Expansion:**
   - Mobile apps using TensorFlow Lite or ONNX Runtime.
   - Cloud-based API that allows third-party developers to integrate Signspeak functionality into their platforms.

3. **Cloud Infrastructure:**
   Large-scale adoption, cloud combination {AWS, Google Cloud, or Azure} will authorize distributed training, continuous deployment, & easier data updates.

Benefits of Scaling

- Upgrade global accessibility.
- Multilingual communication.
- Real time collaboration across devices.
- Increased social impact for the hearing-impaired people's community worldwide.

Load Balancing

SignSpeak system is designed for academic & research purposes, handling a limited number of synchronized users. Thus, large-scale load balancing mechanisms are not yet required. Still, the design includes potential for expansion if user demands increase in future variations.

Present Configuration

The present setup runs on a local Flask server hosted on a single machine at a time. Model inference is handled by the CPU or GPU of a parallel system. Since synchronous requests are minimal, resource usage remains stable & within acceptable limits.

Future Load Balancing Plan

When the system transitions into a public or enterprise-level application, load balancing will be implemented to make certain reliability under heavy traffic. Possible future improvements include:

- **Cloud Hosting : T**he backend on AWS EC2 or Google Cloud Run to handle multiple concurrent users!

- **Auto-Scaling :** Adjusting server resources dynamically based on traffic volumes!

Chapter13:Critical Appraisal and Evaluation

13.1. Introduction

Assesment & evaluation , indeed are the fundamental constituents of project since they give an unbiassed mirror of its outcome success and limitations. The system accomplished its planned objectives identifies difficulties encountered the upsurge 7 finally points out areas which could be made better.

13.2. Objectives That Could Be Met

1. SL recognition model CNN:
Design & training of the CNN model for SL gestures recognition were successful, where the model could reliably recognize these gestures from both static images and real-time camera inputs images.

2. Model integration with a web-based frontend:
Integration of a trained CNN into a web application was done smoothly by employing Flask as the backend framework. And front-end interface rendered it possible for users to communicate with the model live without needing to install any specific software!

3. Real-time SL gesture recognition with text and audio feedback:
Recognition of gestures by the system was accompanied immediately by visual & auditory feedback. The signs that were detected were first turned into text & audio through the text-to-speech capability.

13.3. Success Rate Against Each Objective

Table 7Success

Objective	**Evaluation Metric**	**Result**
CNN Model Accuracy	Recognition accuracy	**>90% accuracy** on trained ESL dataset
Real-Time Usability	Frame processing latency	**Low latency**, average 0.3 seconds per frame
Accessibility	Ease of use, interface design	**User-friendly**, no technical expertise required
Integration	System coherence and stability	**Stable performance**, no major runtime errors

How Much Better It Could Have Been Done

- **Dataset Improvement:**
  Bigger and diversified SL dataset will greatly enhance the accuracy of recognition.

- **Hardware Acceleration:**
  GPU or TPU technology could lead to reduction in the durations of both training and interface prod
- **Data Augmentation & Real-Time Calibration:**
  Advanced augmentation techniques and real-time calibration for lighting and hand position could further minimize recognition errors in uncontrolled environments.

Why It Could Not Be Done

1. **Limited Access to High-Quality ESL Datasets:**
   Publicly available SL datasets are frequently small & at odds with, limiting the model's exposure to complex contrast in gesture styles & backgrounds.
2. **Hardware Limitations:**
   Training process was conducted on standard personal computing hardware system without advanced GPU acceleration. This restricted the depth & size of the CNN model & limited the number of experiments that could be run for the system.
3. **Time Constraints:**
   The project was conducted within a fixed academic schedule, restricted large experimentation & large-scale optimization efforts!

13.4. Objectives That Have Been Missed

1. **Full Support for Dynamic Gestures:**
   Firstly focuses on static gestures corresponding to the alphabet & short words. Continuous zestful gestures, such as phrase-level signing, not implemented due to dataset limitations.
2. **Multilingual Translation Beyond English:**
   The translation module for now supports only English output! Integration of multilingual translation features remains unimplemented in this version of software!

Why These Objectives Were Missed

1. **Complexity of Dynamic Gesture Recognition:**
   Continuous gesture recognition requires sequential datasets! This necessitates advanced architectures, such as Recurrent Neural Networks or Long Short-Term Memory models, which were beyond the project's hardware capabilities, being used in the system.
2. **Multilingual Integration Challenges:**
   Elaborating translation beyond English requires integrating Natural Language Processing & multilingual text-to-speech modules. These require large and pre-trained language models & additional training data, which are unavailable within the project scopes.

What Could Have Been Done to Complete Those Objectives

1. **Collaborating with Deaf Organizations so that the operation can happen:**
   Recruiting with sign language institutions could provide access to high-quality, continual SL datasets. Alike collaborations would enhance dataset diversity & authenticity!

2. **Integration of NLP Models:**
   Grping pre-trained NLP frameworks {e.g., BERT, GPT-based translators, or Google's Translation APIs} could help incorporate multilingual text translation features!

3. **Hardware Upgrades:**
   Employing dedicated GPUs or cloud-based training environments like AWS Sage Maker !

How Better Are the Features of the Solution

- **Real-Time Recognition:**
  System can interpret gestures as they come about, offering an interactive experience without delays.

- **Audio Feedback Integration:**
  The text-to-speech functionality enhances universality, allowing hearing individuals to understand sign language output audibly.
- **Lightweight Architecture:**
  CNN-based system requires minimal computational power to compute , making it suitable for academic, personal, & portable use cases to make it light weighted.
- **Scalability:**
  Designed with modularity, the system can be upgraded easily to include more gestures, languages, & features!

Which Features Could Not Be Touched

1. **Gesture-to-Speech in Multiple Languages**
2. **Continuous Recognition of Sentence-Level ESL**

Why These Features Could Not Be Touched

1. **Resource Constraints:**
   Project relied on limited local hardware to use, which restricted the implementation of multi-language TTS & continuous gesture recognition modules.
2. **Time Limitations:**
   Project's academic timeline did not allow for extensive model retraining & integration of multiple NLP systems due to time insufficiency.
3. **Dataset Availability:**
   Continuous putting datasets are rare & require specialized recording setups, which were unavailable during development stages.

What Could Be Done to Touch Those Features

1. **Expand Dataset Collection:**
   Collaborate to develop large-scale SL datasets including continuous sentence signing videos to store in dataset.
2. **Hybrid Model Development:**
   Combine CNN for geometric physical recognition with RNN or Transformer-based models for temporal sequence understanding to develop a hybrid model.
3. **Cloud-Based NLP Integration:**
   Use APIs such as Google Cloud Translate, AWS Comprehend, or OpenAI's language models to add multi-language translation & dependent speech generation capabilities.

Chapter 14
Lessons Learned

14.1. Introduction

Sign Speak creation journey portraying the event, findings, individual development that took place over the entire project frame. The knowledge obtained in the phases before the same time after implementations.

Pre-Project – Review – Closing

Pre-Project Stage

14.2. Review Phase

At first, tests with limited datasets showed the model's overfitting & lack of generalization as the main problems. So, I turned to data augmentation and hyperparameter tuning that helped significantly to up the performance. System front and back end also underwent several iterations as part of evolution. In the beginning, the interface was unresponsive and hard to understand. By testing and getting feedback, I developed the user experience, coming up with a simple and user-friendly design.

Closing Stage

At the closing stage system managed to combine computer vision, machine learing and web tech together to meet it's goal. The trip from dreaming the idea to putting it into action was proof to close relatiosip between and practicalapplication. This system become something that cloud give power to the people with hearing disability. This insight is the foundation of the technology revolution of most significant when it serves a cause which is bigger than itself.

14.3. What Have I Learned

Technical Learning

1. The Importance of Dataset Diversity in Machine Learning Performance.
2. The Need for Iterative Prototyping to Refine Accuracy and Usability.
3. How Small-Scale Academic Research Can Still Make a Meaningful Impact on Accessibility

Personal Learning

This project allowed me Working on Sign Speak fostered patience, resilience, & adaptability. Managing unexpected technical challenges & learning new frameworks under time constraints strengthened problem-solving. Solving & research abilities. It also underscored the value of collaboration and the importance of clear and simple documentation for maintaining consistency throughout complex stages of the project.o grow in many aspects especially in terms of of patience nad adaptability.New gramework limited time frame made me more capable of solving problem. My problem solving And research skill were also improved.

What Problems I Faced

1. Lack of an Extensive ESL Dataset.
2. Computational Limitations While Training CNNs.
3. Designing an Interface That Balances Simplicity with Functionality.
4. Integration Complexity Between Components.

14.4. What Solutions Occurred

1. Applied Data Augmentation Techniques to Artificially Expand Dataset Size.
2. Optimized CNN Architecture to Train Efficiently on Limited Hardware.
3. Adopted a Minimalist UI Approach to Prioritize Usability.
4. Implemented Incremental Testing and Logging Mechanisms.

Chapter15:Conclusion

15.1. Summary of the Project

Sign Speak system is real-time communication tool which can detect Sign language Gestures & translate them into text version. Using CNN model system captures sign gestures.

The development procidure :

- **Research & Data Collection** : Review of existing literature & building of ESOL feature of the dataset for data collection for the process.
- **Model Development & Training** : Design, training and testing maked the CNN model developments.
- **System Integration** : FLASK API integration so that the system can connect with web-based links.
- **Testing & Deployment** : As I wanted to provide a functional integrity, swiftness and the ever-well-receive accessibility to user testing is very important part .

Goal of the Project

- ✓ Designing a CNN-based model to recognize SL gestures smoothly.
- ✓ Integrating the trained model into a responsive web-based interface so it can run by a browser.
- ✓ Allowing instant text.

Success of the Project

Functionality, performance and user experience done its achievement exceptional successes . CNN model during testing reached accuracy more than 90%. It's also proven to be very effective under real time situation! The integration of the system frontend and backend clearly. Here camera captured the gesture and backend recognige gesture immidietly using the pre installed models learning and frontend shows the resul by translating the gestures . this system make the communication easy between signer and non signers. User are very happy to use it and satisfied with usability, speed and simplicity.

15.2. Value of the Project

Academic Value

My project made the marriage between machine learning nad human computer interaction. The possibility of fine tuning the cnn frameworks for diagnosis and solv the every day problems of human life!And creating a starting point for future studies guestire recognition and assistive Ai system.

Social Value

My project make sures equality among people and the right to communicate without any barrier . It made the deaf or speech impired people to communicate with full strength even without any aid of any interpreting person or separate aid. It has a huge impact on the education , healthcare and daily communication of signer and non signer people.

Technological Value

The light weighted design ensures that it will be accessible on all platfroms and relasticly opens the possible of future development in web or mobile app. It become great innovation of kindness by modern Ai tecniques.

15.3. My Experience

Collaboration on Sign Speak was an eye-opening & changing experience. It’s a rare chance to put into practice the theoretical knowledge from machine learning, computer vision, & web development into a real-life, socially beneficial system. In the journey, I faced many limitations such as those of the dataset, slow training of the model, and syncing of frontend and backend. But the challenges made me stronger in my problem-solving & analytical skills. I gained skills in being flexible unexpected issues, getting the best result with limited resources and being aware of the ethical implications in technologies. Whole process also developed my knowledge in practical design and in the way technology should be made for the different types of users. Every obstacle turned into a lesson learning part of my life and every enhancement was considered a contribution to accessibility research. This endeavor has not only improved my technical capabilities but also brought me closer to the human-centered revolution. It make me realised that even small-scale academic research could have an impactful contribution to the society when it is driven by empathy, purpose, and creativity!

Reference:

1) Starner, T., Weaver, J., & Pentland, A. (1998). *Real-time American Sign Language recognition using desk and wearable computer based video*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(12), 1371–1375.
2) Kumar, P., & Jain, S. (2017). *A survey on hand gesture recognition using computer vision*. International Journal of Engineering and Technology, 9(1), 41–47.
3) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. Advances in Neural Information Processing Systems, 25, 1097–1105.
4) Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. International Conference on Learning Representations (ICLR).
5) Zhang, Z., Liu, Z., & Chen, L. (2020). *A real-time continuous gesture recognition system using deep learning*. IEEE Access, 8, 186086–186097.
6) Rahman, M. M., & Hassan, M. R. (2019). *Hand gesture recognition using convolutional neural networks*. International Journal of Computer Applications, 178(7), 1–6.
7) Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2015). *Sign language recognition using convolutional neural networks*. European Conference on Computer Vision (ECCV), Workshop Proceedings.
8) Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). *Neural sign language translation*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7784–7793.
9) Howard, A. G., et al. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.04861.
10) Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
11) Bradski, G. (2000). *The OpenCV library*. Dr. Dobb's Journal of Software Tools.
12) Ramalingam, B., & Jacob, S. (2021). *Sign language recognition using deep learning and computer vision*. Journal of Ambient Intelligence and Humanized Computing, 12(11), 10519–10530.

213-16-582

ORIGINALITY REPORT

10%	8%	4%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	Submitted to University of Greenwich Student Paper	1%
3	ijarsct.co.in Internet Source	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
5	www.coursehero.com Internet Source	<1%
6	assets-eu.researchsquare.com Internet Source	<1%
7	Yang Wang, Wei Jiang, Deyou Li. "Sign Language Translator System Using Computer Vision and LSTM Neural Networks", International Journal of Information Technologies and Systems Approach, 2025 Publication	<1%
8	www.ijrte.org Internet Source	<1%
9	Suman Lata Tripathi, Om Prakash Kumar, Allwin Devaraj Stalin, Tanweer Ali. "Innovations in Computer Vision, Communication Systems, and Computational Intelligence - Proceedings of the First International Conference on Computer Vision, Communication System and Computational Intelligence (CVCNCE 2025), 08–09 May 2025, Tirunelveli, India", CRC Press, 2025 Publication	<1%

10	"Smart Data Intelligence", Springer Science and Business Media LLC, 2026 Publication	<1 %
11	arxiv.org Internet Source	<1 %
12	sist.sathyabama.ac.in Internet Source	<1 %
13	merimaxpub.org Internet Source	<1 %
14	Submitted to Dublin Business School Student Paper	<1 %
15	S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufacturing", CRC Press, 2025 Publication	<1 %
16	uet.edu.al Internet Source	<1 %
17	www.americaspg.com Internet Source	<1 %
18	Submitted to University of Northampton Student Paper	<1 %
19	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
20	www.mdpi.com Internet Source	<1 %
21	wpsites.ucalgary.ca Internet Source	<1 %
22	ijrpr.com Internet Source	<1 %
23	ora.ox.ac.uk Internet Source	<1 %
24	sep.com Internet Source	<1 %