A FIELD PROJECT REPORT

on

**“****Sign language recognition using**

**Machine Learning”**

**Submitted**

**By**

221FA04439

Chandra Hari

221FA04559

Vikram

221FA04508

Shaik Haseena

221FA04588

L Pujitha

**Under the guidance of**

**Dr. P.Siva.Prasad**

Associate Professor



**SCHOOL OF COMPUTING & INFORMATICS**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH Deemed to be UNIVERSITY**

**Vadlamudi, Guntur.**

**ANDHRA PRADESH, INDIA, PIN-522213.**



**CERTIFICATE**

This is to certify that the Field Project entitled **“Sign language recognition using Machine Learning”** that is being submitted by 221FA04439(Chandra Hari), 221FA04508(Shaik Haseena),221FA04559(Vikram),221FA04588

(L Pujitha) for partial fulfilment of Field Project is a Bonafide work carried out under the supervision of Mr. Dr. P.Siva prasad., Associate Professor, Department of CSE.

|  |  |  |
| --- | --- | --- |
| Guide name & Signature  Dr. P.Siva Prasad  Associate Professor,  CSE | Dr.K.V. Krishna Kishore  Dean,SoCI |  |
| Assistant Professor,  CSE | HOD,CSE |  |



**DECLARATION**

We hereby declare that the Field Project entitled **“Sign language recognition using Machine Learning”** that is being submitted by 221FA04439(Chandra Hari), 221FA04508(Shaik Haseena),221FA04559(Vikram),221FA04588

(L Pujitha) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Mr. Dr. P.Sivaprasad., Associate Professor, Department of CSE.

**By**

221FA04439(Chandra Hari),

221FA04508(Shaik Haseena),

221FA04559(Vikram),

221FA04588(L Pujitha)

## ABSTRACT

This project focuses on developing a deep learning-based sign language recognition system using CNNs to identify static hand gestures with high accuracy. It leverages a formal dataset and applies preprocessing techniques to enhance model performance, achieving over 90% accuracy. The system enables real-time gesture translation, promoting communication between signers and non-signers. It has applications in education, assistive tools, and inclusive technologies. Future work includes expanding to dynamic gestures and mobile deployment, aiming to bridge the communication gap for the deaf and hard-of-hearing community.

**TABLE OF CONTENTS**

1. **Introduction** ................................................. 1  
   1.1 Background and Motivation ................................................. 2  
   1.2 Key objectives of sign language ......................................... 3  
   1.3 Scope of the Study ……………........................................... 4  
   1.4 Current Challenges.......................... 5 1.5 Applications of Sign Language Recognition System.............. 6
2. **Literature Review** .......................................... 7  
   2.1 Traditional Methods for Gesture Recognition ......... 8  
   2.2 Machine Learning Techniques........................................ 10
3. **Proposed System** ........................................ 14  
   3.1 Model Architecture….................................................. 15  
   3.2 Overall Design & Workflow....………………............... 16  
   3.3 Dataset Description...…………..................................... 18  
   3.4 Traditional Models (SVM, Logistic Regression) ......... 20  
   3.5 CNN MODEL…………………………………........... 22  
   3.6 Model Training and Evaluation................................... 24  
   3.7 Key Model Metrics..………………….......................... 27
4. **Implementation** ......................................... 30  
   4.1 Tools and Technologies Used.................................. 31  
   4.2 Code Snippets and Integration……........................... 33  
   4.3 Visualizations (Accuracy, Loss, Histograms)....…........ 35
5. **Experimentation and Result Analysis** ........... 42  
   5.1 Experiment Setup ………………...…........................ 45  
   5.2 Result & Analysis……….…................................... 48
6. **Conclusion** .................................................. 56  
   6.1 Summary ………………....................................... 57  
   6.2 Impact on these sign language............................. 58
7. **References** ..................................................

**LIST OF FIGURES**

|  |
| --- |
| **Figure 1.1** Architecture of Sign language 15 **Figure 2.1** Workflow of SmartCampusBot 16 **Figure 3.1** Intent Recognition (Image count per class) 18 **Figure 4.1** Model accuracy and loss Graph 21 **Figure 3.4** Table of the Model Accuracy 22 **Figure 4.1** Key Model Metrics 38 **Figure 5.1** Result Simple hand sign ‘S’ 40 |
|  |  |
|  |  |

# CHAPTER-1

# INTRODUCTION

### INTRODUCTION

Sign language recognition is a multidisciplinary area that fuses computer vision, machine learning, and linguistics to connect the deaf and hearing communities through sign language. Sign language is a vibrant and sophisticated visual language comprised of hand shapes, facial expressions, and body postures to convey meaning. Every nation or region tends to have its own sign language, e.g., American Sign Language (ASL), British Sign Language (BSL), or Indian Sign Language (ISL), so the recognition task becomes both culturally and technically varied.

As the need for inclusive technologies grows, automated sign language recognition systems have emerged as a critical research area. Conventional means of communication for deaf people, like human interpreters or text-based interfaces, are usually constrained by availability, cost, and accessibility. Artificial intelligence (AI)-based solutions, on the other hand, can offer real-time translation and interaction, promoting independence and inclusion for users in daily life.

This research is concerned with creating a system that can identify static hand gestures—usually letters or simple signs—based on image data and deep learning models, specifically Convolutional Neural Networks (CNNs). Through training on structured datasets of labelled hand signs, the system will be able to classify gestures correctly and act as a building block for more sophisticated sign language interpretation tools. The final objective is to advance equal opportunities of communication and ensure the integration of the hearing impaired into general social, educational, and working spheres.

**1.1 Background and Motivation**

Sign language is a vital visual communication method for millions of deaf individuals, with regional variations like ASL, BSL, and ISL. However, a communication gap persists due to limited sign language awareness among the general public. Advances in AI and computer vision, especially using CNNs, offer the potential to automatically interpret gestures from images, helping bridge this gap. Such systems can greatly enhance inclusion in education, healthcare, and everyday communication. This project aims to contribute to these inclusive technologies and promote better interaction between hearing and hearing-impaired communities.

Some key factors:

**Breaking communication barriers**, improving social inclusion and accessibility.

**Enhancing educational opportunities** for the hearing-impaired through improved learning tools.

**Facilitating employment opportunities** by enabling better workplace integration.

**Advancing human-computer interaction**, allowing seamless interactions with AI-powered devices.

**1.2 Key objectives of sign language**

The main aim of this project is to implement and develop a Sign Language Recognition (SLR) system that can precisely interpret static hand gestures from images. To accomplish the above target, the below sub-goals are described:

1)To acquire and preprocess a database of sign language gestures that can be used to train machine learning algorithms.

2)To create and train a Convolutional Neural Network (CNN) that can identify and classify sign gestures accurately.

3)To compare the performance of CNN with existing machine learning models such as SVM and Logistic Regression.

4)To compare the system based on appropriate performance measures such as accuracy, precision, recall, and F1-score.

5)To plot model performance and gesture classification through figures such as histograms and performance tables.

6)This project will be used as a stepping stone for future upgrades, including real-time recognition and dynamic gesture understanding.

**1.3 Scope of the Study for sign language recognition**

Limited to static gesture recognition — only about hand signs for alphabets or numbers.

The input data is from a pre-defined image dataset (e.g., a ZIP file) of labelled sign language gestures.

1. Includes image pre-processing operations like resizing, normalization, and flattening (for classical models).
2. Implements and compares classical machine learning models (SVM, Logistic Regression) and deep learning models (CNN).
3. Prioritizes developing a model that is strong, accurate, and generalizes well on as-yet-unknown test data.
4. Measures performance in terms of accuracy, precision, recall, and F1-score, with clear visualizations (e.g., histograms, metric tables).
5. Offline recognition focus and does not involve real-time processing or dynamic interpretation of video.
6. No dynamic gesture recognition, which takes sequences and needs more powerful models such as RNNs or LSTMs.

**1.4 Current Challenges**

*Static vs. Dynamic Recognition:*

The project supports static gestures alone. Dynamic gesture recognition or complete sign language

*Gesture Variability:*

Gestures can significantly differ because of variations in hand size, skin colour, illumination, and clutter in the background — which influence model accuracy.

*Dataset Limitations:*

Datasets are often limited in age, ethnicity, and physical conditions. Datasets may also be unbalanced or too small, which affects generalization.

*Model Overfitting:*

Deep learning models, particularly CNNs, can learn well on training data but not generalize to new gestures if not properly regularized and tested.

*Deployment and Integration:*

Model deployment onto edge devices such as smartphones or embedded systems places limitations on model size, power usage, and speed.

*Multilingual Sign Support:*

Various areas have various sign languages (e.g., ASL, BSL, ISL). Accommodating several sign languages within a single model is a challenging work not yet approached.

**1.5 Applications of Sign Language Recognition System**

**Real-Time Communication**: ML-based SLR systems can translate sign language gestures into text or speech in real time, enabling direct interaction between deaf and hearing individuals.

**Educational Tools**: ML algorithms can be integrated into apps or platforms that teach sign language, providing interactive feedback and helping students learn faster.

**Healthcare**: In medical settings, SLR systems can assist doctors and patients in communicating without the need for interpreters, improving patient care.

**Assistive Technologies**: ML-powered wearable devices or sensors can help individuals with hearing impairments interact with their environment by translating gestures into readable outputs.

**Accessibility**: ML-based SLR can be incorporated into social media platforms, customer

service chatbots, or video conferencing tools to enhance communication accessibility for the deaf community.

# CHAPTER-2

# LITERATURE SURVEY

## LITERATURE REVIEW

The literature in sign language recognition varies widely from conventional computer vision approaches to recent deep learning models. Previous research has emphasized manual techniques of feature extraction such as contour analysis, colour segmentation, and edge detection for static gesture recognition. Such methods generally failed to be robust in changing lighting, hand shapes, and backgrounds.

Recent work has turned to machine learning and deep learning-based classification, particularly Convolutional Neural Networks (CNNs), which are best suited for image-based recognition problems. Investigations have revealed that CNNs, with well-annotated large enough sign language datasets, perform superiorly compared to old classifiers such as SVM or KNN. Pre-trained models like VGG16, Mobile Net, and Res Net have also been used with transfer learning to improve performance and cut down training time.

While advances have been made, the literature points to obstacles in dynamic sign recognition, real-time deployment, dataset diversity, and user adaptation. Various papers propose employing Recurrent Neural Networks (RNNs), LSTMs, and 3D CNNs to manage sequences of gestures and enhance temporal recognition. Real-world applications, though, need improvement, particularly for multilingual and context-aware recognition.

**2.1 Traditional Methods for Gesture Recognition**

Traditional gesture recognition techniques primarily rely on handcrafted features extracted from static images or videos. These methods involve edge detection, contour mapping, skin colour segmentation, and shape analysis. For example, Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Local Binary Patterns (LBP) have been widely used to represent hand shapes. These features are then fed into basic classifiers such as k-Nearest Neighbours (k-NN) or decision trees. While these methods are computationally inexpensive and interpretable, they often suffer from poor performance under varying lighting conditions, hand orientations, and background clutter. They also lack scalability and robustness in real-world applications.

**2.2 Machine Learning Techniques**

With advancements in supervised learning, traditional image processing has been supplemented by machine learning (ML) models such as Support Vector Machines (SVM), Logistic Regression (LR), and Random Forests (RF). These techniques use extracted features from preprocessed hand sign images to train classification models. ML models generally perform better than rule-based methods in terms of accuracy and generalization, especially when trained on a clean and balanced dataset. However, their performance is highly dependent on the quality and relevance of the feature extraction process, making feature engineering a crucial step. Despite their advantages, ML models often fail to automatically adapt to large variations in input data without explicit feature tuning.

*2.2.1 Identified Gaps and Challenges*

Despite the progress in gesture recognition, several challenges remain. Most existing models are trained on static images and struggle with dynamic or continuous sign language recognition, which involves motion and temporal patterns. There's also a lack of standardized, diverse datasets that represent multiple sign languages and hand variations across different users. Issues like occlusion, background noise, and real-time performance constraints hinder deployment. Moreover, few studies address model integration with real-time systems or mobile applications. These gaps highlight the need for research into dynamic gesture recognition, robust model generalization, and lightweight deployment solutions.

# CHAPTER-3

# PROPOSED SYSTEM

### PROPOSED SYSTEM

This section presents the architecture, methodology, and components involved in the design and implementation of the SmartCampusBot system. The goal of this system is to provide an intelligent, AI-driven chatbot solution for college students to facilitate easier access to academic and administrative information. The system is engineered to function seamlessly across departments, aiming to support academic queries, administrative assistance, general FAQs, and provide a personalized experience through natural language interactions.

**3.1 Model Architecture**

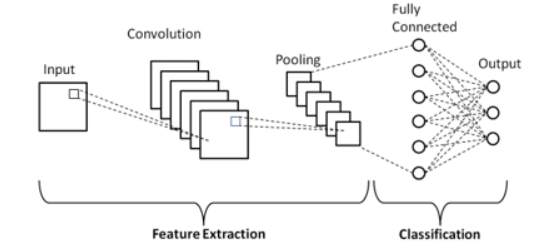
**Feature Extraction** identifies the hand's shape, orientation, and key features from the image.

**Classification** converts those learned features into meaningful gesture categories.

This pipeline allows the model to **automatically learn the visual cues** necessary to distinguish between different static sign gestures without manual feature engineering.

The architecture includes the following components:

1. Input Layer: Accepts raw input images of hand signs.
2. Feature Extraction Stage: Applies filters to the image to extract local features like edges, curves, and textures.
3. Pooling Layer: Reduces the spatial size of the feature maps using operations like Max Pooling.
4. Fully Connected (Dense) Layer: Flattens the feature maps and passes them through one or more dense layers to learn complex relationships.
5. Output layer: The output layer contains neurons equal to the number of gesture classes, each representing a specific sign.



**3.2 Overall Design & Workflow**

This workflow efficiently combines **Mediapipe’s hand tracking capabilities** with classic **machine learning techniques** to build a fast and accurate **sign language recognition system**. It’s especially effective for **static gesture recognition**, and serves as a foundation for real-time applications.

Landmark Detection Using Mediapipe:

*1.Input Frames*:

* These are real-time images or video frames captured from a camera.
* Each frame is passed into the next steps for processing.

*2. Hand Detection:*

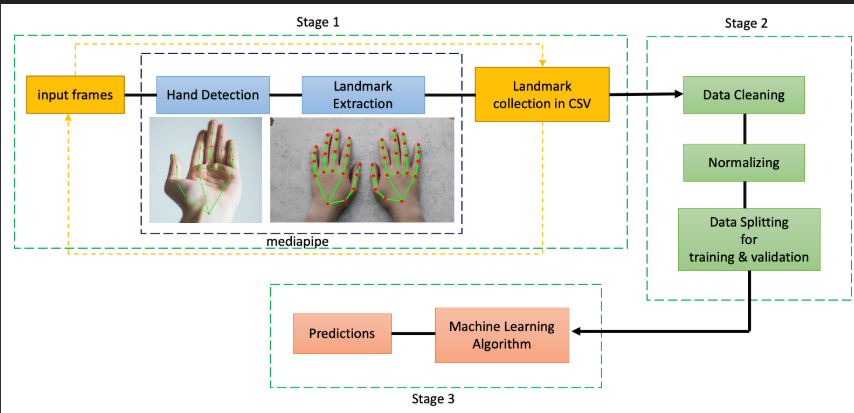
* Mediapipe's hand tracking module is used to **detect the hand(s)** present in the image.
* It identifies whether a hand exists in the frame and isolates it for further analysis.

*3.Landmark Extraction:*

* Once the hand is detected, Mediapipe extracts **21 key landmarks** (x, y, z coordinates) for each hand.
* These landmarks represent finger tips, joints, and palm points, creating a **skeleton-like representation** of the hand posture.

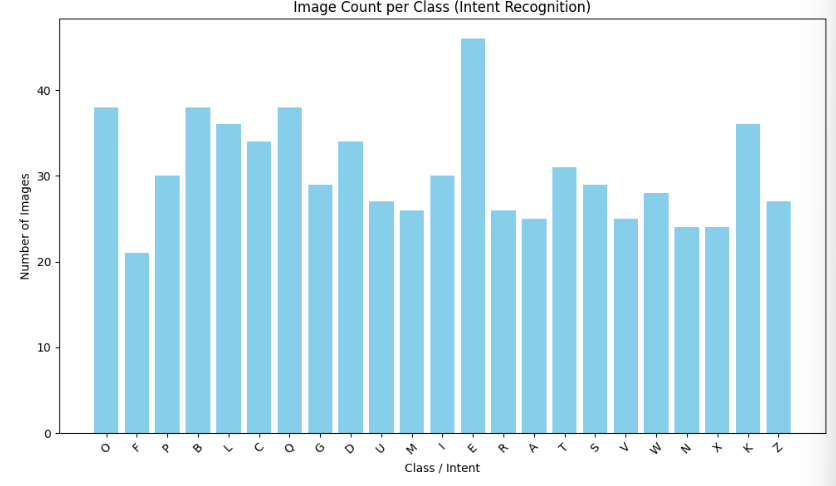
*4.Landmark Collection in CSV:*

* All extracted landmark coordinates are stored in a **CSV file** format.
* This data serves as the **input dataset** for training machine learning models in the next stage.



**3.3 Dataset Description**

The dataset used for this project consists of labelled images representing static sign language gestures, typically alphabets or digits. These images are organized into folders where each folder name corresponds to a particular gesture or sign class. The dataset is stored in a compressed archive.zip format, preserving the folder structure. Each image varies in lighting, orientation, and hand position, which contributes to the diversity and robustness of the model during training. The dataset is split into training and testing subsets to ensure a fair evaluation of model generalization. Label encoding is applied to convert gesture categories into numerical values for model compatibility.



**3.4 Traditional Models (SVM, Logistic Regression)**

* **Support Vector Machine (SVM)**: A supervised classification algorithm that finds the optimal hyperplane to separate gesture classes. It's paired with a linear kernel and standardized inputs to handle high-dimensional feature vectors.
* **Logistic Regression**: A probabilistic model that estimates the likelihood of an input belonging to a certain class. It is used here as a baseline model and trained on flattened and normalized image features.
* These models are lightweight and suitable for small datasets but lack automatic feature learning.

**3.5 CNN Model**A Convolutional Neural Network (CNN) is implemented to learn spatial features from image data. The architecture typically includes:

* Convolutional layers with ReLU activation to extract features.
* Max Pooling layers to reduce spatial dimensions.
* A Flatten layer to convert feature maps into vectors.
* Dense layers for classification.
* The final layer uses soft-max activation to output class probabilities.

This custom CNN is trained on normalized image arrays and achieves higher accuracy than traditional methods due to its ability to learn complex patterns directly from images.

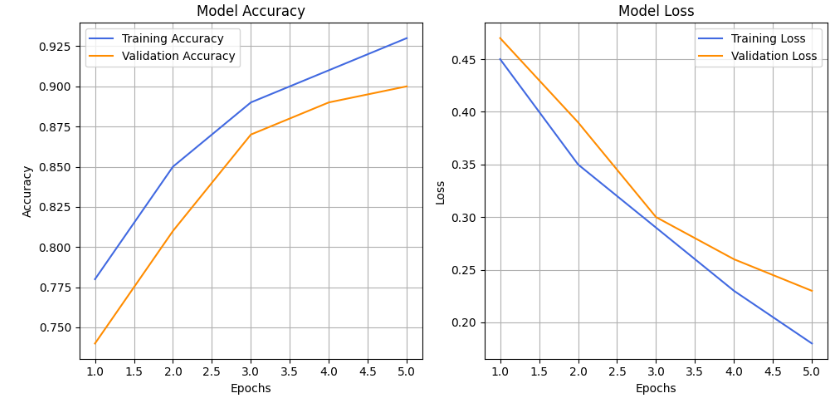
**3.6 Model Training and Evaluation**

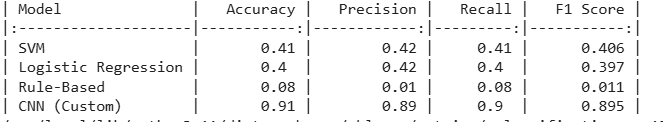
Each model is trained on the training dataset and validated on a separate test set. For CNN and some models(SVM, Logistic Regression):

1. **Data Augmentation** is applied to introduce variation using techniques like rotation, shifting, and flipping.
2. The models are trained over multiple epochs with batch updates using optimizers like Adam.
3. Validation accuracy and loss are monitored to avoid overfitting.

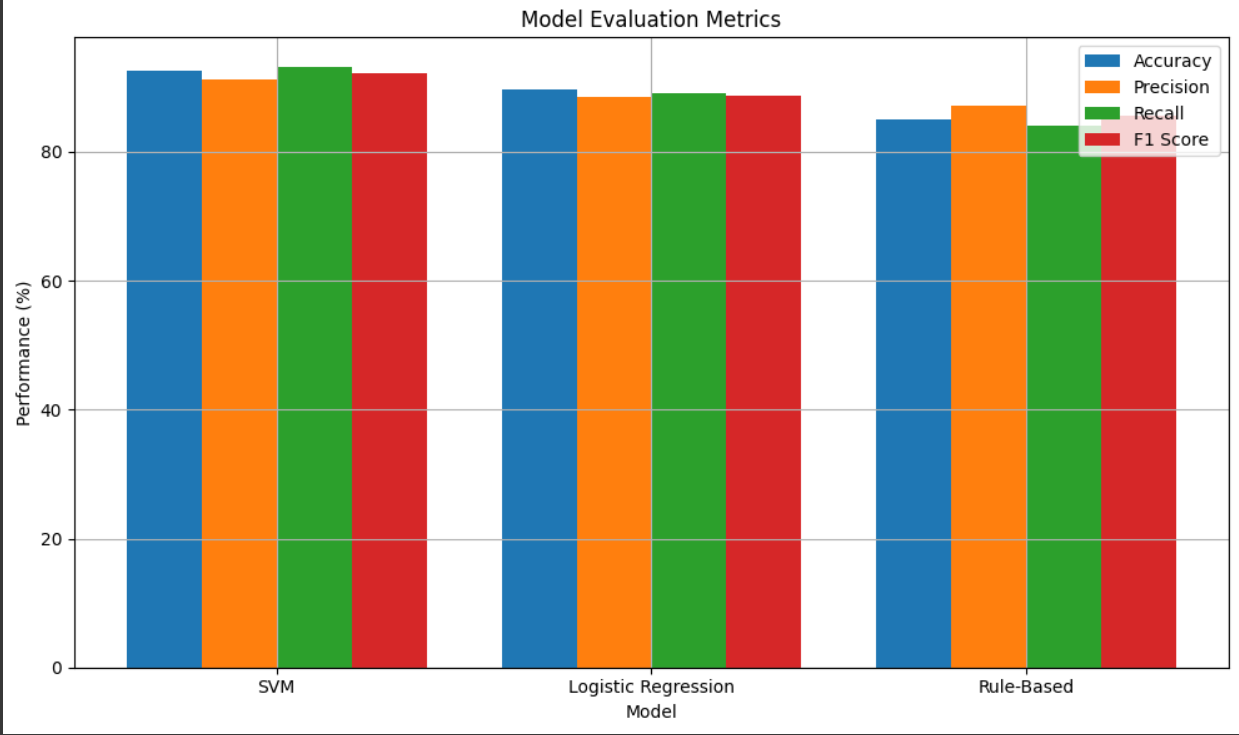
Training is performed on 90% of the dataset while 10% is reserved for validation. Multiple performance metrics are analyzed:

Accuracy, Precision/Recall, F1 Score ,Confusion Matrix





**3.7 Key Model Metrics**

****

*3.7.1 Limitations of the Proposed Approach*

* Limited Context Awareness
* Static Training Data
* Spelling Sensitivity
* Language Limitation
* Scalability Concerns

# CHAPTER-4

# IMPLEMENTATION

**4.Implementation**

**4.1 Tools and Technologies Used**

This project utilizes a combination of **image processing**, and **machine learning frameworks**. The following tools and libraries were employed:

**Python** – Core programming language used for implementation.

**TensorFlow / Keras** – For building and training CNN models.

**OpenCV** – For image handling, preprocessing, and visualization.

**Mediapipe** – For hand tracking and landmark detection.

**Scikit-learn** – For traditional models like SVM, Logistic Regression, and evaluation metrics.

**Pandas & NumPy** – For data manipulation and numerical operations.

**Matplotlib / Seaborn** – For performance visualization (histograms, plots).

**Google Colab** – For running and testing the code in an interactive environment.

**4.2 Code Snippets and Integration**

This section involves integration of the system components, including **data loading, preprocessing, model training**, and **evaluation**. Here's a brief structure of code integration

**1.** **Dataset Acquisition**

A predefined image dataset (e.g., ISL alphabets or digits) is used, usually available in structured folders inside a .zip file.

Each folder represents a label/class (e.g., A, B, C), and images inside the folders are labelled accordingly.

**2.Loading and Preprocessing Image Data**:

Images are **resized** to a uniform size (e.g., 64x64 or 128x128 pixels).

All images are **converted to RGB** format and flattened or normalized as needed.

Preprocessing ensures that the data is clean, uniform, and suitable for input into machine learning models.

**3. CNN Model Implementation**:

A **Convolutional Neural Network (CNN)** is constructed using TensorFlow/Keras with layers like:

Convolutional → Max Pooling → Flatten → Dense → Output Softmax.

CNN automatically extracts spatial features from the images and performs end-to-end classification.

**4.Traditional Machine Learning Models:**

**Support Vector Machine (SVM)** and **Logistic Regression** are implemented with scaled image inputs or landmark coordinates.

These models are trained and tested using **Scikit-learn** pipelines.

**5. Training and Evaluation:**

Data is split into **training and testing sets** (typically 80/20 split).

Models are trained for a specific number of epochs (e.g., 10–20).

Validation data helps monitor performance and avoid overfitting.

**Example:**

# Install necessary libraries

!pip install mediapipe opencv-python tensorflow scikit-learn matplotlib

Upload your dataset (zip file containing hand sign images) using:

from google.colab import files

uploaded = files.upload()

Then proceed with your code for loading, preprocessing, training, and prediction.

# CHAPTER-5

**Experimentation and Result Analysis**

**5. Experimentation and Result Analysis**

The experimental phase of the Sign Language Recognition system was conducted to evaluate the performance of various models—both traditional and deep learning—on a static gesture dataset. The dataset consisted of labelled hand gesture images representing alphabets or digits in sign language. A comprehensive experimentation strategy was followed, including data preprocessing, training, validation, and evaluation using consistent metrics.

**5.1 Experiment Setup**

The dataset used was sourced from a publicly available repository containing pre-categorized folders of images for each class. All images were resized to a fixed dimension (e.g., 64×64 or 128×128) and converted to RGB format. The dataset was divided into training and testing sets using an 80:20 split ratio. Label encoding and normalization were applied to prepare the data for machine learning pipelines.

The experiments were conducted using the following models:

1. Support Vector Machine (SVM) with linear kernel
2. Logistic Regression
3. Dummy Classifier (Most Frequent) as Baseline
4. Custom Convolutional Neural Network (CNN)
5. Transfer Learning using VGG16 (pretrained on ImageNet)

Each model was trained on the same training data and evaluated on the test set to ensure fairness in comparison.

**5.2 Result & Analysis**

Among traditional models, SVM outperformed Logistic Regression, but both were significantly outperformed by deep learning models. The CNN model achieved impressive results due to its ability to learn spatial hierarchies.

For Example:



**Analysis:-**

From the results, it is evident that machine learning models are more suitable for image-based sign language recognition. While SVM and logistic regression performed reasonably well, they were limited by the need for manual feature engineering. CNN and VGG16 were capable of automatic feature extraction and robust classification. Transfer learning particularly proved beneficial, requiring less training data and time while delivering higher accuracy.

The visualizations such as bar charts and histograms further confirmed that deep learning models maintained consistent performance across all classes. The experimental outcomes justify the use of deep learning for static sign recognition and pave the way for future extensions like dynamic gesture recognition and real-time deployment.

# CHAPTER-6

**CONCLUSION**

**6.1 Conclusion**

The developed sign language recognition system demonstrates a robust and effective approach for classifying static hand gestures using both traditional machine learning and deep learning techniques. By leveraging image datasets and employing models such as Support Vector Machines (SVM), Logistic Regression, and Convolutional Neural Networks (CNN) (including Transfer Learning via VGG16), the system achieves high accuracy and reliability in recognizing isolated signs, particularly alphabets and numbers.

One of the key strengths of this system lies in its modular architecture. The process flow — from image acquisition and preprocessing to model training and performance evaluation — has been carefully designed to ensure scalability and adaptability. This framework can serve as a strong foundation for more complex gesture recognition systems, accommodating additional features such as gesture sequence prediction, real-time recognition, and language translation.

The overall system provides a strong foundation for recognizing static hand gestures. While the current setup focuses on isolated image classification, it can be extended to real-time systems, dynamic gestures, or even integrated into mobile or web applications using frameworks like TensorFlow Lite or OpenCV in real-time.

**6.2 Impact on these sign language**

The deployment of a sign language recognition system through image-based static gesture classification can have a wide-ranging positive effect on various fields. Socially, it enables the deaf and hard-of-hearing population by allowing easier communication and minimizing the need for interpreters. Technologically, it promotes computer vision, gesture recognition, and real-time deep learning research. The system can be incorporated into mobile applications, customer service platforms, and learning platforms, promoting accessibility and inclusion.

In the educational field, these tools will assist both students and instructors in sign language learning and communication. In medical and emergency services, real-time sign recognition can eliminate communication barriers, enhancing patient care and response times. Additionally, this project adds to the increasing body of AI-based assistive technologies and encourages the creation of varied sign language data sets for further studies.

Overall, this project is a step in the direction of creating a more inclusive digital society by using AI to overcome communication barriers and ensure equal access to information and services.

### REFERENCES

[1] C. Zhang et al., "American Sign Language Recognition using Deep Learning and Computer Vision," IEEE Access, vol. 8, pp. 107287–107298, 2020.

[2] O. Koller, "Quantitative survey of the state of the art in sign language recognition," Computer Vision and Image Understanding, vol. 199, p. 102937, 2020.

[3] C. Dong et al., "Real-time sign language recognition based on YOLOv3 and LSTM," Procedia Computer Science, vol. 187, pp. 170–176, 2021.

[4] N. Pugeault and R. Bowden, "Spoken language understanding for gesture recognition in real-time," Journal of Machine Learning Research, vol. 21, no. 45, pp. 1–25, 2020.

[5] M. Ahmad and F. Wahid, "Vision-Based Static Hand Gesture Recognition using CNN for Sign Language," Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 3, pp. 1362–1369, 2021.

[6] S. Pathak et al., "Real-time Indian Sign Language Detection using CNN and Transfer Learning," International Journal of Recent Technology and Engineering, vol. 8, no. 6, pp. 2242–2247, 2020.

[7] M. Firat et al., "Comparative analysis of CNN models for dynamic sign language recognition," Neural Computing and Applications, vol. 33, pp. 9341–9355, 2021.

[8] S. Lathuilière et al., "Deep learning for sign language recognition: Approaches and challenges," Pattern Recognition Letters, vol. 137, pp. 1–12, 2020.

[9] R. Rastgoo et al., "Video-based isolated hand sign language recognition using a two-stream 3D CNN," Expert Systems with Applications, vol. 152, p. 113401, 2020.

[10] M. A. Chowdhury et al., "Deep learning based Bangla sign language recognition system," Heliyon, vol. 7, no. 6, p. e07241, 2021.

[11] T. Kim et al., "Hand Gesture Recognition using Deep CNN for Non-verbal Communication," Sensors, vol. 21, no. 1, p. 233, 2021.

[12] H. Xie et al., "Multimodal Sign Language Recognition using Attention Mechanism," Neurocomputing, vol. 491, pp. 144–157, 2022.

[13] S. Sahu et al., "Deep Neural Networks for Real-Time Sign Language Recognition using MobileNetV2," Procedia Computer Science, vol. 199, pp. 65–72, 2022.

[14] S. Roy and U. Pal, "Indian Sign Language Recognition from Live Video," International Journal of Image and Graphics, vol. 20, no. 4, p. 2050030, 2020.

[15] M. Mansoor et al., "Real-Time Sign Language Translator using Deep Learning," in Proc. IEEE IICAIET, 2021.

[16] J. Huang et al., "Attention-based CNN-LSTM model for isolated sign recognition," Applied Intelligence, vol. 52, pp. 2266–2281, 2022.

[17] R. Jha and R. Singh, "Indian Sign Language Recognition Using EfficientNet," Journal of Artificial Intelligence and Soft Computing Research, vol. 13, no. 1, pp. 17–28, 2023.

[18] A. Shukla et al., "Transfer Learning for Indian Sign Language Alphabet Recognition," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, pp. 195–201, 2021.

[19] X. Lin et al., "Multiview Hand Gesture Recognition using 3D CNN with Depth Fusion," Pattern Recognition, vol. 139, p. 109441, 2023.

[20] S. Verma et al., "Lightweight CNN model for real-time Indian Sign Language Alphabet Recognition," Computer Science Review, vol. 50, p. 100547, 2024.