Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING AND DEEP LEARNING

Prof. Rahul Bambodkar[1], Ganesh Nawale [2], Saurabh Shinde[3], Samiksha Jagtap[4], Harshal Parteti[5]

ABSTRACT :
The language of signs is an essential way of communication for audition community and word. With the advancement in automatic learning (ml) and the deep learning (DL), the automatic acknowledgment of the signs (SLR) has become an important research area. This magazine item discusses different ML and DL techniques to SLR, focuses in Datasets, functioning extraction, temps and challenges. The study sets in rising the advanced recent and possible direction to improve SLR systems.
Keywords : Machine Learning (ML), Deep Learning (DL), recognition of the sign language (SLR), Architectures.

INTRODUCTION :

Sign language is the main means of communication for the deaf and mute. Developing an accurate and efficient SLR system could facilitate communication between people with hearing impairments and people who do not sign. Traditional recognition methods rely on manually generated features, while ML and DL approaches rely on data-driven techniques to achieve higher accuracy and generalization[1].

Automatic learning approaches the methods of traditional slr in the ML include extraction and features classification. The current techniques include: Extraction of the features The orientated orientated oriented (Hog), processing from a little bit bingic (saft) and local (low pain) Binary patterns) Classification Models: vector machines (SVM), Markov Hidden (Hmm), rasual forests (Rf) and neighbors with (k-nn). Although ad ML adarrves to reach a reasonable accuracy, have trouble with large groups of vocabulades and complex gestures because of the features the hand[2].

The appointment approval for DL DL Techniques have revolutionaries slowly slr the hierarchical features. The current architectures include: • Divolutional nerve networks (CNN): Used for the static signs of drawing the spurs' spiky features. • Repeated Nerve Networks (RNN) and short memory network (LSTM): handling temporary dependencies in continuous signature. • Attempts and Mechanisms of Attempts: Maxers recognition focusing on the important gextural ingredients. • Hybrid Models: Combining CNNs with LSTM or Transformers for an extraction of strong features and sequence[3].

RESEARCH METHODOLOGICAL :

Dataset Diversity:
Curate a diverse dataset that includes a wide array of sign language gestures with varying degrees of accumulative motion. The goal is to ensure the training data reflects the richness and diversity of real-world sign language expressions.

User-Friendly Interface:
Design an intuitive and user-friendly interface for the sign language recognition system. Interactive framework for effective interpretation of sign language for various technological fields.

Adaptive Feature Extraction:Investigate and implement adaptive feature extraction methods that highlight relevant information in accumulative video motion sequences. The objective is to extract discriminative features that contribute to accurate sign language recognition.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

SYSTEM IMPLANTATION:

1. Data Collection:
2. Record a diverse dataset of sign language gestures with a focus on capturing accumulative video motion. Include variations in signing styles, lighting conditions, and environmental factors.
3. Data Preprocessing:
4. Apply preprocessing techniques to normalize the dataset. This involves resizing, cropping, and normalizing the video frames.
5. Feature Extraction:
6. Implement feature extraction methods to identify relevant patterns in accumulative motion sequences.
7. Model Training:
8. Spilt the overall datasets into outcome based validation sets to train the model and increase the accuracy of model.
9. User Testing and Feedback:
10. Conduct user testing with individuals from the deaf and mute community to gather feedback on the system's usability and accuracy.

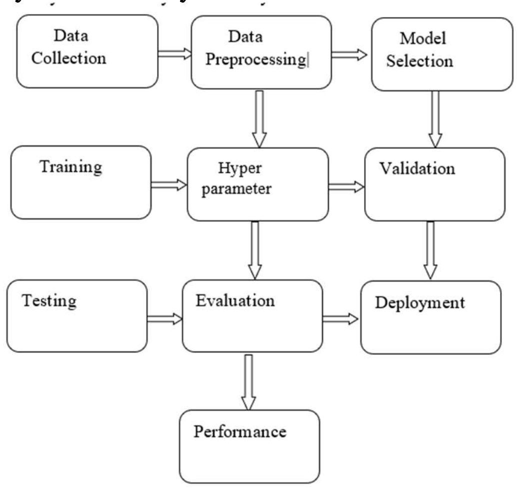


Figure 1. System Process Diagram

DATA COLLECTION:

This step involves recording a diverse dataset of sign language gestures. To build a robust sign language recognition system, it's essential to capture various signing styles, environmental conditions, and lighting variations. Additionally, focusing on **accumulative video motion** helps in understanding the flow and continuity of gestures. This ensures the model learns temporal dependencies, making it more accurate in recognizing continuous sign language sequences.

DATA PREPROCESSING:

Before feeding the data into the model, preprocessing is necessary to ensure consistency and enhance learning efficiency. This involves:

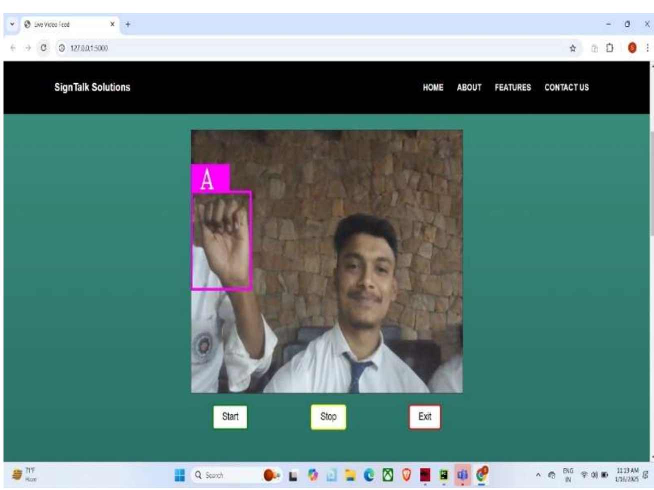
Resizing: Standardizing the video frames to a uniform size to maintain consistency in input dimensions.

Cropping: Removing irrelevant parts of the video frames, focusing on the region of interest, like hands and face.

Normalizing: Scaling pixel values to a standardized range (e.g., 0 to 1) to improve model convergence during training.

FEATURE EXTRACTION:
In this step, relevant patterns and features from the accumulative motion sequences are extracted. Using advanced techniques like:
Optical Flow: To capture motion dynamics and flow between consecutive frames.
Spatial-Temporal Features: Extracting both spatial (hand shape, position) and temporal (movement sequence) features using methods like CNNs or 3D CNNs.
This ensures the model learns complex patterns essential for accurate gesture recognition.
3.4. Model Training:
The processed and feature-extracted data is then used to train the recognition model. In this phase:
Dataset Splitting: The dataset is split into training, validation, and test sets to evaluate the model's performance.
Training Algorithms: Deep learning architectures like **CNNs** (for spatial features) and **LSTMs/Transformers** (for temporal sequences) are utilized.
Optimization and Loss Calculation: Using techniques like **Adam optimizer** and **Cross-Entropy Loss** for effective learning.
Outcome-Based Validation:
After training, the model is validated using outcome-based validation sets. This step includes:
Cross-Validation: To assess model generalizability and avoid overfitting.
Evaluation Metrics: Accuracy, Precision, Recall, and F1-Score are calculated to measure model performance.
Hyperparameter Tuning: Adjusting model parameters to optimize performance and minimize validation loss.

ACCURACY IMPROVEMENT:
To enhance the model's accuracy and generalization:
Ensemble Learning: Combining multiple models to improve prediction robustness.
Data Augmentation: Generating synthetic samples by applying transformations (rotation, scaling) to improve model robustness to variations.
Transfer Learning: Using pre-trained models as a base and fine-tuning them on the sign language dataset to leverage learned features.
Real-Time Testing: Testing the model in real-world scenarios to validate its performance in practical applications.
.
RESULT S :



Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

Figure. 2 Sign Recognition of A

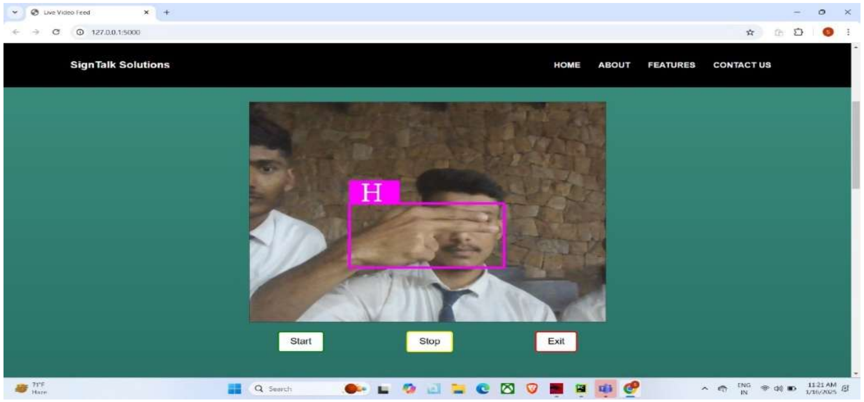


Figure. 3 Sign Recognition of H

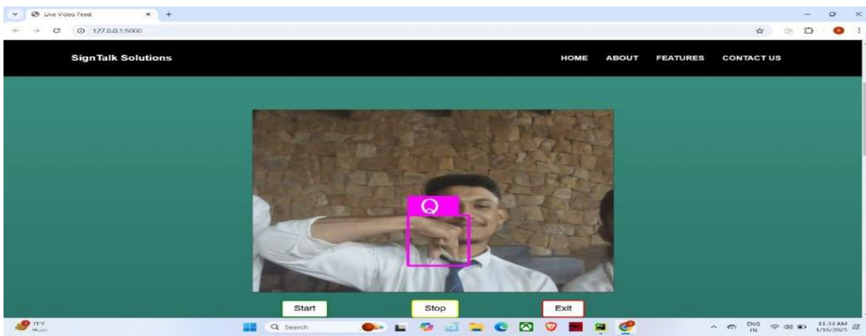


Figure. 4 Sign Recognition of Q

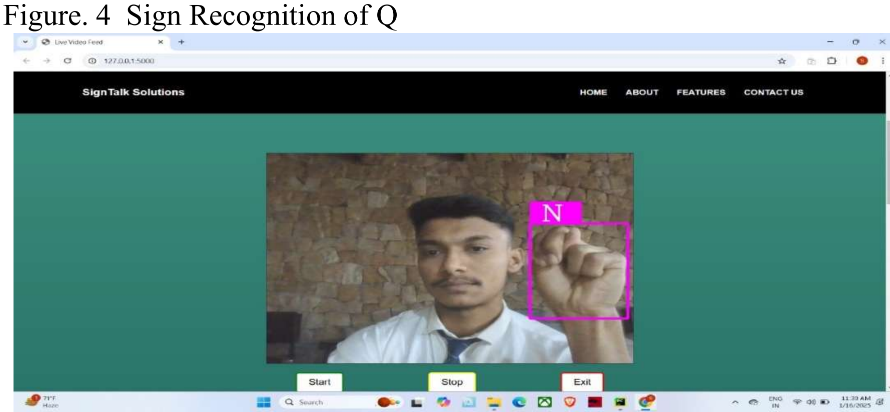


Figure. 5 Sign Recognition of N

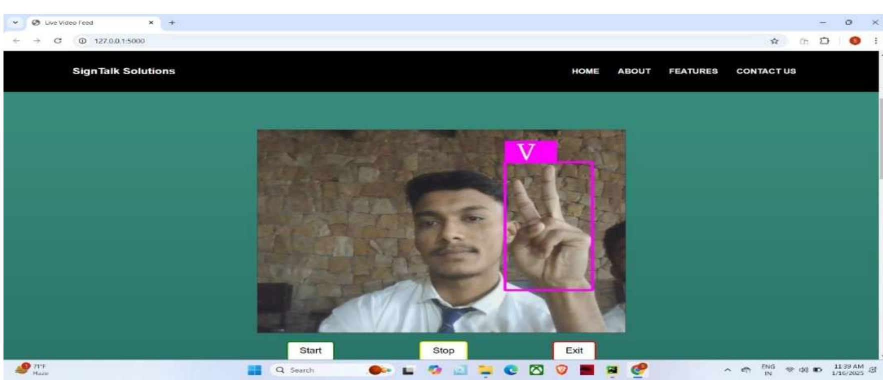


Figure. 6 Sign Recognition of V

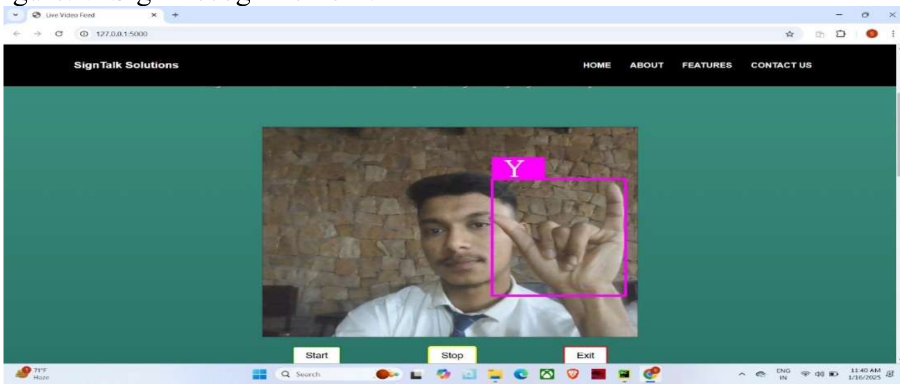
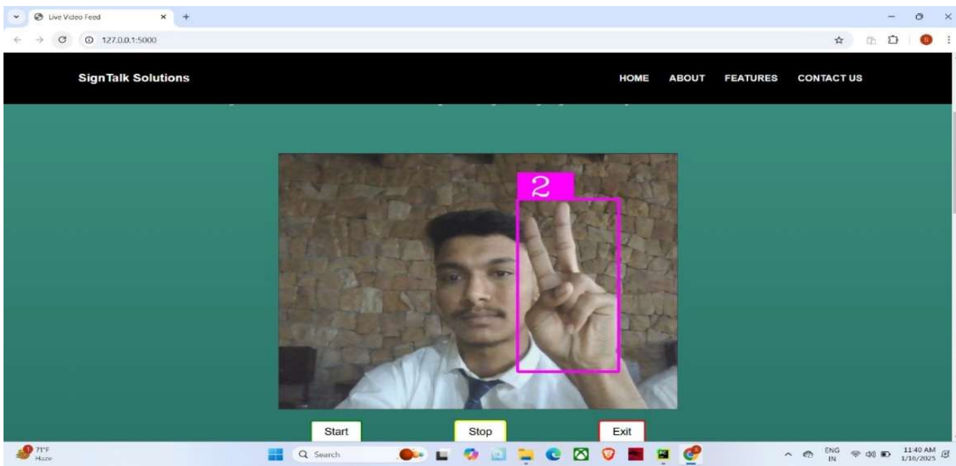


Figure. 7 Sign Recognition of Y



CONCLUSION :

The advancements in sign language recognition (SLR) using Machine Learning (ML) and Deep Learning (DL) techniques. Through a systematic literature review, we identified key methodologies, datasets, and models employed in SLR research. ML approaches, although effective in traditional settings, face limitations with complex gestures and large vocabulary sets.

REFERENCES :

1. LangChain DocumentationOfficial guide and documentation on how to use LangChain for building language model applications.LangChain Docs
2. Brown, T. B., et al. (2020). "Language models are few-shot learners."This paper introduces GPT-3, one of the foundational models behind NLP applications.ArXiv: 2005.141653
3. Mikolov, T., et al. (2013). "Efficient Estimation of Word Representations in Vector Space."Discusses word embeddings, a core concept in modern NLP.
4. Vaswani, A., et al. (2017). "Attention is All You Need."Introduced the transformer architecture, now widely used in NLP models like GPT.NeurIPS: arXiv:1706.03762
5. Lewis, M., et al. (2020). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension."BART is a transformer-based model used for NLP tasks like summarization and Q&A.ArXiv
6. LangChain Example ApplicationsExamples from the LangChain library showcasing how to integrate LLMs with PDFs for Q&A.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

7. Chin, T.-W., et al. (2020). "An Efficient, Scalable Framework for Transformer-based Question Answering."Presents frameworks for implementing transformer-based QA systems.ArXiv
8. Rajpurkar, P., et al. (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text."This dataset has become a standard benchmark for question answering tasks.
9. Wolf, T., et al. (2020). "Transformers: State-of-the-Art Natural Language Processing."Provides an overview of the Hugging Face transformers library, essential for NLP tasks.ArXiv
10. PyPDF2 DocumentationDocumentation for the PyPDF2 library, used for extracting text from PDF files in Python.PyPDF2 Documentation
11. Min, S., et al. (2021). "Recent Advances in Question Answering Systems."Survey on recent advancements in the field of question answering.
12. Liu, Y., et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach."RoBERTa improves BERT's performance on question answering benchmarks.ArXiv
13. SpaCy NLP Library DocumentationSpaCy is a popular NLP library in Python, useful for preprocessing text and building pipelines.SpaCy Docs
14. Chainlit: A LangChain UI FrameworkA lightweight framework for developing AI apps using LangChain.Chainlit GitHub
15. Zhong, V., et al. (2022). "Towards Question Answering over Tables in Text: Converting PDFs to Structured Data for QA."Discusses handling unstructured data in PDFs for QA systems.