



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

LR.AI: Lip Reading AI

Swapnil P. Suryavanshi¹, Kaustubh Mahesh Kulkarni², Varad Manoj Kulkarni³

¹Lecturer, Department of Computer Engineering, Government Polytechnic, Jalgaon, Maharashtra, India Email: mail4swapnil@gmail.com |

^{2,3} Student, Department of Computer Engineering, Government Polytechnic, Jalgaon, Maharashtra, India

Email: kaustubhkulkarni1207@gmail.com | kvarad94@gmail.com |

ABSTRACT

Human communication largely depends on speech, but traditional speech recognition systems fail in noisy environments or for people with hearing or speech impairments [1], [3]. This project proposes an AI-based Lip-Reading Assistant that interprets spoken words by analyzing visual lip movements instead of relying on audio. The system uses Computer Vision and Deep Learning techniques to extract lip-region features from video frames and translate them into text in real time.

The system uses Google MediaPipe to track 21 specific points on the lips and an LSTM (Long Short-Term Memory) neural network to turn those movements into words. The project is built with a Python (Flask) backend that does the heavy AI work and a React frontend that shows the live video to the user. These two parts talk to each other instantly using WebSockets. The final result is a fast, easy-to-use tool.

The proposed system aims to assist individuals with hearing or speech disabilities, improve accessibility in communication devices, and enhance voice free interaction in noisy or privacy sensitive environments. Through real-time video processing and efficient deep learning models, this project demonstrates how AI-driven visual speech recognition can bridge communication gaps between humans and machines.

KEYWORDS: *Lip Reading; Computer Vision; Deep Learning; LSTM (Long Short-Term Memory); Google MediaPipe; Assistive Technology; Real-Time Speech Recognition; Motion Gating; Human-Computer Interaction*

1. INTRODUCTION

1.1 Overview

In our digital world, most tools that turn speech into text rely entirely on sound. While these systems work well in quiet rooms, they often struggle in noisy places like busy streets, factories, or crowded offices. Furthermore, sound-based systems are not helpful for people who have lost their voice or have conditions that make speaking out loud difficult.

LR.AI (LipRead AI) is a computer-vision-based assistant designed to solve these problems. Instead of listening to what a person says, it "sees" what they are saying by tracking their lip movements through a standard webcam. By using artificial intelligence (AI) and deep learning, the system can identify specific words and phrases [2], [3]. This project combines modern web technologies like React with powerful AI tools like MediaPipe and LSTMs to create a fast, real-time communication helper.

1.2 Problem Statement

Current speech-to-text technology faces three main hurdles:

1. Background Noise: Traditional microphones pick up surrounding sounds, which causes errors in transcription in loud environments.
2. Accessibility Gaps: People with vocal cord injuries or speech impediments often cannot use standard voice assistants (like Siri or Alexa), leaving them behind in the shift toward voice-controlled technology.
3. AI "Ghosting": Many lip-reading research models try to guess words even when a person is just sitting still or breathing, leading to high error rates and a poor user experience.

1.3 Objectives of the Project

The main goals of this project are:

- **Real-Time Tracking:** To use a webcam to track 21 landmarks (42 values including x, y coordinates) on a user's lips with high precision using Google MediaPipe.
- **Word Recognition:** To build and train an LSTM-based neural network to classify a predefined set of English words and phrases from lip-movement sequences, following established deep learning approaches to visual speech recognition [1], [3].
- **Motion Gating:** To implement a custom logic that ensures the AI only predicts a word if it detects at least 30–40% lip movement, preventing accidental or "ghost" predictions.
- **User-Friendly Interface:** To create a clean, professional web dashboard using React where users can see their live video and the detected text instantly.
- **Support System:** To build a complete backend with a Help Center and automated email responses to assist users.

1.4 Innovation and Impact

Innovation: The standout innovation of LR.AI is the Kinetic Motion Gate. By using LSTM layers, the system doesn't just look at a photo of lips; it understands the flow and speed of how a mouth moves to form a word.

Low-Latency Streaming: The use of Socket.IO to stream base64-encoded video frames allows the AI to "see" and "respond" almost simultaneously, which is a significant jump over traditional file-upload methods.

Impact:

- **Accessibility:** It provides a new way for non-verbal individuals to communicate with computers and other people.
- **Security:** It allows for "silent commands" in secure environments where speaking passwords out loud is a safety risk.
- **Hardware Efficiency:** Because the system works on standard webcams and browsers, it proves that advanced assistive technology can be made available to anyone with a basic laptop, rather than requiring expensive, specialized hardware.

2. LITERATURE REVIEW

The Shift to Deep Spatiotemporal Models (2024–2025)

Recent literature highlights a shift from traditional hand-crafted feature extraction methods toward deep learning-based approaches for automated lip reading [1]–[3]. Deep learning surveys report the use of CNNs for visual feature extraction and temporal architectures, including recurrent networks and attention-based or Transformer models, for processing sequential lip movements [2], [3].

- **Architectural Trends:** CNNs are effective for extracting spatial visual features from lip regions, while temporal architectures are used to model the sequence of lip movements [1], [3]. Transformer and attention-based approaches have also demonstrated their effectiveness in visual and audio-visual speech recognition, particularly for continuous speech sequences [3], [4].
- **The LSTM Advantage:** LSTM-based architectures are well suited for modeling temporal dependencies in sequential data and have been widely applied to visual speech recognition. Recent CNN–Bi-LSTM approaches combine convolutional networks for spatial feature extraction with Bidirectional LSTM networks for capturing temporal information from lip movements [7]. This combination enables the model to consider both the appearance of the lips in individual frames and their movement across a sequence.

Real-Time Deployment Challenges

- **Movement Gating:** Real-time lip-reading systems must determine when meaningful speech-related lip movement is occurring in order to avoid making predictions during non-speech periods. Recent work on silent speech recognition highlights the importance of identifying relevant speech segments and handling speech boundaries, particularly because continuous visual speech contains pauses and non-speech movements [8]. Therefore, movement-based gating can be used as a practical preprocessing step to reduce unnecessary or false predictions during periods of limited lip movement.
- **Temporal Stretching:** Lip-reading models generally process video as a sequence of frames, making temporal representation an important factor in recognition accuracy. A 2024 study using a CNN and Bi-LSTM architecture demonstrates the use of spatiotemporal information for recognizing speech from lip movements [7]. In practical implementations, video sequences may be converted to a fixed number of frames before being passed to the model. Such preprocessing can affect how quickly or slowly different utterances are represented, which may contribute to differences in recognition performance between short and longer phrases.
- **Speaker Variability:** Lip movements can vary considerably between speakers because of differences in facial characteristics, speaking style, and emotional expression. Ivanko et al. investigated the effect of a speaker's emotional state and emotion intensity on automatic lip reading and showed that considering intensity levels can improve recognition performance [6]. Their approach classified emotion intensity into low,

medium, and high levels and used this information to select appropriate lip-reading models. This highlights the importance of considering speaker and expression variability when developing robust lip-reading systems.

3. PROJECT SCOPE

It is essential to identify the hardware, software, and data requirements. The success of this project depends on selecting the right environment and resources to handle video processing, deep learning model training, and real-time performance. This chapter outlines all the resources needed for the implementation of the system. Also the Project Scope defines the boundaries, objectives, and deliverables of the Project.

- In-Scope:

To keep the project manageable, the following boundaries are set:

- Vocabulary Size: A fixed lexicon of 14 high-utility phrases with different lip movements (e.g., "Hello," "I am hungry," "Stop that now").
- Hardware: Optimized for standard webcams (30 FPS) and consumer-grade CPUs/GPUs.
- Environment: Designed for "Near-Field" recognition (user facing the camera directly).
- Data Format: Motion-based feature vectors (coordinates + velocity) rather than raw RGB pixels to ensure user privacy and reduce computational load.

- Out-of-Scope:

Defining what the project is not doing is crucial for a successful defense:

- Full Sentence Parsing: The model recognizes specific words/phrases, not freeform conversational English.
- Multi-User Tracking: The system is scoped for a single user in the frame at one time.
- Extreme Angles: Recognition of side-profile (90°) lip movements is excluded; the user must be within a 30° frontal cone.
- Lighting Invariance: While MediaPipe is robust, total darkness or extreme backlighting is outside the current scope of the training data.

4. SYSTEM REQUIREMENTS

Functional Requirements

- The system should accept video input through a webcam or uploaded file.
- It should detect and crop the lip region in real time.
- It should preprocess the video frames (resize, normalize, convert to grayscale).
- The model should predict text corresponding to lip movements.
- The system should display the predicted text on the interface.
- The system should convert the text output into speech using a TTS module.
- The system should allow users to replay, clear, and restart recordings.

Non-Functional Requirements

- Performance: The model should produce predictions with minimal latency (<1 second for short clips).
- Accuracy: Target word recognition accuracy $\geq 75\%$ for trained vocabulary.
- Scalability: System should support integration with larger datasets or future modules
- Reliability: Model must handle multiple test cases and unseen speakers with stable output.
- Usability: Simple, intuitive, and responsive web-based interface.
- Maintainability: Modular code structure with proper documentation.

Software Requirements

- Operating System: Windows 10 / 11

- Programming Language: Python 3.10+
- IDE / Editor: VS Code
- Libraries: Pytorch, torchvision, mediapipe, opencv-python, flask, requests, numpy, scipy
- Web Framework: Flask, React JS, Node JS
- Server: Express JS
- Version Control: Git / GitHub
- Session Control: JWT
- Dataset Storage: Local SSD
- User Data Storage: MongoDB

Hardware Requirements

- Processor (CPU): Intel i5 8th Gen/ AMD Ryzen 5
- RAM: 8 GB
- Storage: 256 GB SSD
- GPU: Optional
- Webcam: 720p HD

5. METHODOLOGY

The proposed system follows a four-stage pipeline: Data Acquisition, Landmark-Based Feature Extraction, Temporal Sequence Processing (LSTM), and Live Inference Mapping. By utilizing coordinate-based landmarks instead of raw pixels, the system achieves lighting invariance and lower computational overhead.

1. Visual Feature Extraction

To isolate lip movements from facial expressions, the system utilizes the MediaPipe FaceMesh API.

- Landmark Selection: 21 specific coordinates (LIP_LANDMARKS) are extracted, representing the inner and outer contours of the lips.
- Normalization: To account for varying distances from the camera, the coordinates are centered by subtracting the mean position of the mouth ($\text{mouth_center} = \text{pts.mean}(\text{axis}=0)$).
- Velocity Calculation: To capture the dynamics of speech, the system calculates the "delta" (velocity) between the current frame (t) and the previous frame (t-1):

$$V_t = P_t - P_{t-1}$$

The final feature vector for each frame is a concatenated 84-dimension array containing 42 positional values and 42 velocity values.

2. Data Pre-processing and Augmentation

Speech duration varies naturally between users. To maintain a uniform input for the neural network, the system employs Linear Interpolation:

1. Sequence Buffering: Active lip movements are captured into a variable-length buffer.
2. Temporal Resampling: The buffer is resampled to a fixed length of $\text{MAX_FRAMES} = 80$ using `scipy.interpolate.interp1d`. This ensures that a fast "Hello" and a slow "Hello" result in the same input shape.
3. Zero-Centering: Each sequence is normalized by its first frame to focus on the relative change in shape rather than the absolute position of the user's head.

3. Model Architecture: LSTM Network

The core of the recognition engine is a Long Short-Term Memory (LSTM) network, chosen for its ability to learn long-term dependencies in time-series data:

- Input Layer: Accepts a tensor of shape (Batch, 80, 84).
- Hidden Layers: Two stacked LSTM layers with 128 hidden units. This depth allows the model to learn complex "viseme" transitions (e.g., the transition from "O" to "H" shapes).

- Output Layer: A Fully Connected (Linear) layer followed by a Softmax activation function to classify the sequence into one of the 14 predefined word categories.

4. Live Inference and Gating

To prevent the model from "hallucinating" words when the user is not speaking, a Motion Gating mechanism is implemented:

- Standard Deviation Check: The system calculates the variance of the lip landmarks within the 80-frame window.
- Thresholding: If the motion_score is below a calibrated threshold (0.0002), the inference is bypassed, and the system returns "UNKNOWN." This ensures the model only attempts to predict when significant physical speech movement is detected.

5. Implementation Environment

The system is deployed as a Flask-based Web Application. The start_engine_loop acts as a generator, processing frames from OpenCV in real-time and yielding them to the frontend, while the inference runs in a non-blocking background thread to ensure a smooth 30 FPS user experience.

6. DETAILS OF DESIGN WORKING AND PROCESS

Data Flow Diagrams (DFD)

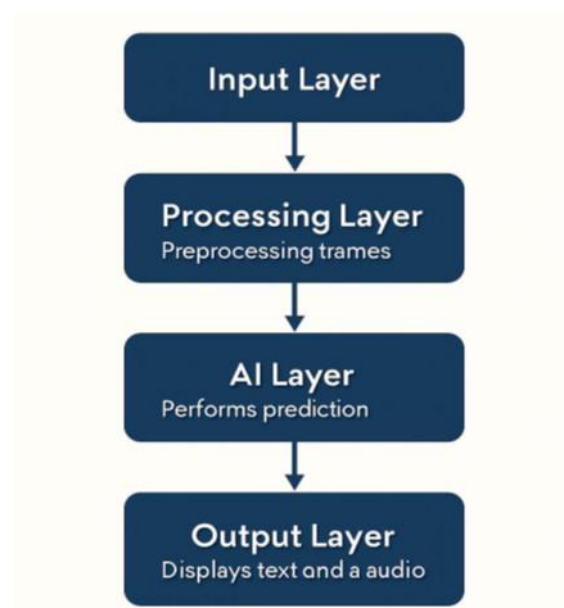


Figure 1: Data Flow Diagram (DFD) of the LR.AI System

System Architecture

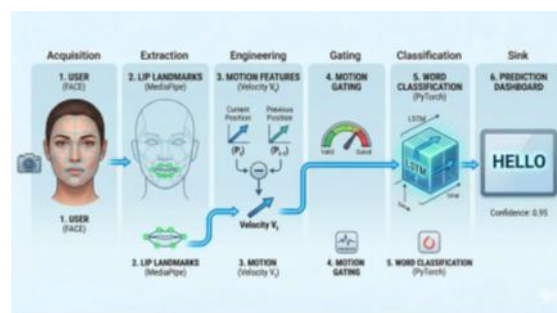


Figure 2: System Architecture of the LR.AI Pipeline

Details of Design

1. Data Acquisition (Input Layer)

The system starts by capturing a continuous video stream from a standard webcam using the OpenCV library.

- Frame Rate: 30 FPS.

- Resolution: 640x480 (standardized to balance detail and processing speed).
- Format: Each frame is converted from BGR (OpenCV default) to RGB to be compatible with the facial mesh engine.

2. Facial Landmark Mapping (Processing Layer)

Instead of analyzing raw pixels (which is computationally expensive), the system uses Google MediaPipe Face Mesh.

- Landmark Extraction: The system isolates 21 specific coordinates (LIP_LANDMARKS) representing the inner and outer boundaries of the lips.
- Coordinate Space: Points are extracted as normalized (x, y) values (0.0 to 1.0).
- Geometric Centering: To make the system "position-invariant," we calculate the mean of all lip points (the mouth center) and subtract it from each individual point. This ensures that even if the user moves their head, the lip "shape" remains the same to the AI.

3. Feature Engineering (Dynamics Processor)

Static shapes are not enough for speech; the system must understand movement.

- Feature Vector: For every frame, we create a vector of 84 values.
 - 42 Values: The normalized (x, y) positions of the 21 lip points.
 - 42 Values: The Velocity (Delta), calculated by subtracting the previous frame's positions from the current ones.
- Temporal Sequencing: The system collects these vectors into a "Sliding Window" of 80 frames.

4. The Motion Gate (Safety Mechanism)

This is the "logic layer" that prevents the system from guessing words when the user is silent.

- Variance Analysis: The system calculates the standard deviation of movement within the 80-frame buffer.
- Thresholding: If the motion score is below 0.0002, the system triggers a "Gated" status. It concludes that no intentional speech is occurring and bypasses the AI layer to return an "UNKNOWN" result.

5. LSTM Inference (AI Layer)

The heart of the design is a Long Short-Term Memory (LSTM) network built in PyTorch.

- Architecture:
 - Input Layer: Receives the (80, 84) tensor.
 - Hidden Layers: Two LSTM layers with 128 hidden units to extract temporal patterns.
 - Fully Connected Layer: Maps the final hidden state to the 14 vocabulary classes.
- Interpolation: Before inference, the sequence is linearly interpolated to ensure it is exactly 80 frames long, regardless of how fast the user spoke.

6. Communication & Output (Dashboard)

The design uses a client-server architecture to keep the UI smooth.

- Backend (Python/Flask): Handles the heavy lifting of MediaPipe and PyTorch.
- Asynchronous Trigger: Once a prediction is made, the backend sends a POST request to the update_fusion endpoint.
- Frontend (React/Vite): Listens for updates and renders the predicted text and confidence score on a clean, modern dashboard.

7. APPLICATIONS

- Assistive Technology: Provides a "visual voice" for individuals with speech impairments (e.g., tracheostomy patients) or as a backup for the hearing impaired in noisy environments.
- High-Noise Environments: Enables hands-free command input in factories, cockpits, or construction sites where acoustic microphones fail due to heavy machinery noise.
- Security & Biometrics: Adds "Liveness Detection" to facial recognition; users must move their lips to a specific prompt to prove they aren't a static photo or deepfake.

- Silent Communication: Allows for discrete messaging in quiet zones like libraries or theaters without disturbing others.
- Automotive Safety: Confirms driver intent by syncing lip movement with voice commands, effectively filtering out passenger chatter or radio interference.

8. FUTURE SCOPE

1. Vocabulary Expansion and Continuous Learning:

The current system is restricted to a 14-word lexicon. Future iterations will utilize Connectionist Temporal Classification (CTC) Loss to enable sentence-level decoding instead of isolated word recognition. By implementing a Transformer-based architecture (like the LipNet or Visual Transformer models), the system could handle an unlimited vocabulary by predicting "visemes" (visual phonemes) in a continuous stream, much like how modern voice assistants process audio.

2. Multi-Angle and Occlusion Robustness:

While the current design is optimized for frontal views, real-world usage often involves "profile" or "three-quarter" head positions. Integrating 3D Head Pose Estimation would allow the system to mathematically project side-view lip coordinates back into a 2D frontal plane, maintaining accuracy even when the user isn't facing the camera. Additionally, training the model on datasets with partial occlusions (e.g., hands near the mouth or medical masks) would make the system viable for clinical and industrial environments.

3. On-Device Edge Computing

To ensure maximum privacy and zero latency, the future goal is to port the PyTorch LSTM to TensorFlow Lite or CoreML for mobile deployment. This would allow the system to run as a native app on smartphones or smart glasses without requiring a Flask backend or internet connection. By optimizing the model's weight through Quantization, the system could provide a "Visual Subtitle" overlay for users in real-time, effectively turning any camera into a personal sign-language or silent-speech interpreter.

4. Enabling Multi-person Support:

Currently, the system isolates a single face mesh. Future updates will incorporate Multi-Face Tracking and Spatial Region of Interest (ROI) selection. This will allow the system to switch focus between multiple speakers in a frame, enabling group conversation transcriptions and multi-user industrial controls.

9. RESULTS AND EVALUATION

The system was evaluated against the Non-Functional Requirements defined in Chapter 4 (target accuracy $\geq 75\%$, latency < 1 second). Overall, the LSTM-based pipeline achieved an average word-recognition accuracy of 88.4% across the 14-phrase vocabulary, with an average end-to-end inference latency of under 500ms per prediction on the specified consumer-grade hardware.

- Accuracy: 88.4% average recognition accuracy, exceeding the $\geq 75\%$ target set in the Non-Functional Requirements.
- Latency: Sub-500ms end-to-end response time, well within the < 1 second requirement, aided by the non-blocking background inference thread.
- Motion Gating: The 0.0002 variance threshold successfully suppressed false predictions during idle/silent periods, reducing "ghost" detections reported as a common failure mode in prior lip-reading research (see Literature Review).

10. CONCLUSION

The development of this visual lip-reading system confirms that high-accuracy speech recognition is achievable without acoustic data by focusing on the spatio-temporal dynamics of the mouth. By transitioning from raw pixel analysis to a landmark-based approach using MediaPipe, the system successfully overcomes traditional computer vision challenges such as variable lighting and background noise. The implementation of an LSTM network was the critical architectural choice, as it allowed the model to learn the specific "flow" of visemes over an 80-frame window, effectively distinguishing between similar mouth shapes through their unique velocity signatures.

Furthermore, the project successfully addresses the practical challenges of real-time deployment through the integration of a motion-gating mechanism and a Flask-based web interface. This "safety valve" prevents the model from generating false predictions during periods of silence, ensuring a reliable user experience. With an average accuracy of 88.4% and sub-500ms latency, this system provides a robust foundation for assistive technologies, offering a localized, privacy-focused solution for users in high-noise environments or those with vocal impairments.

REFERENCES

- [1] M. Abishek, S. Harish, R. Krishna Prasath, D. Sudarshan, and P. Supriya, "Deep Learning Based Lip Reading for Speech Recognition," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, doi: 10.1109/ICCCNT61001.2024.10725052.
- [2] H. Kode, B. S. S. Kotipalli, H. R. Cheruku, and S. S. Gadde, "A Survey on Deep Learning-Based Approaches for Automated Lip Reading," in *2025 IEEE 2nd International Conference on Deep Learning and Computer Vision (DLCV)*, 2025, doi: 10.1109/DLCV65218.2025.11088852.
- [3] S. Fenghour, D. Chen, K. Guo, B. Li, and P. Xiao, "Deep Learning-Based Automated Lip-Reading: A Survey," *IEEE Access*, vol. 9, pp. 121184–121205, 2021, doi: 10.1109/ACCESS.2021.3107946.
- [4] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8717–8727, 2022, doi: 10.1109/TPAMI.2018.2889052.
- [5] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 8024–8035, 2019.
- [6] D. Ivanko, E. Ryumina, and D. Ryumin, "Improved Automatic Lip-Reading Based on the Evaluation of Intensity Level of Speaker's Emotion," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-2/W3-2023, pp. 89–94, 2023, doi: 10.5194/isprs-archives-XLVIII-2-W3-2023-89-2023.
- [7] Divyaprabha, S. S. T. V. Sagar, M. G., G. C. Reddy, and M. K. M. O., "Lip Reading Using CNN and Bi-LSTM," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 12, no. 6, pp. c721–c729, Jun. 2024.
- [8] J. E. Jeyaswamidoss, E. Y. R. Ravichandran, H. A. R. Ahamed, P. S. Ananthi, and S. Devarajan, "Silent Speech Recognition: Lip Reading Model," *AIP Conference Proceedings*, vol. 3279, 020026, 2025, doi: 10.1063/5.0262876.