International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Lip Reading From A Muted Video Using Ml

Syeda Fatima[1], R. Manasa[2], K. Sushma[3], T. Akshitha[4]
[1]Associate Professor; Bhoj Reddy Engineering College For Women Department Of Computer Science And Engineering (AI&Ml), Hyderabad, India.
[2,3,4]B.Tech Students; Bhoj Reddy Engineering College For Women Department Of Computer Science And Engineering (AI&Ml), Hyderabad, India.
reddyreddymanasa4@gmail.com

ABSTRACT
In many real-world situations, audio may not always be available or usable for understanding speech. Videos recorded in noisy environments, surveillance systems, or recordings without sound make it difficult to know what a person is speaking. Lip reading from muted videos is an important research area in computer vision and machine learning that aims to recognize speech by analyzing lip movements instead of depending on audio signals. The objective of this project is to develop a system that can detect and convert speech from muted videos into text using visual information from lip movements. The system accepts a silent video as input and processes it through several stages such as video frame extraction, preprocessing, lip region detection, motion analysis, and machine learning-based prediction. By analyzing the movement patterns of the lips across multiple frames of the video, the system predicts the spoken content and converts it into readable text.

The system is developed using Python along with libraries such as OpenCV, NumPy, TensorFlow, and Streamlit. OpenCV is used for video processing and frame extraction, NumPy is used for numerical computations, and the machine learning model analyzes the visual patterns of lip movements to predict the spoken text. Streamlit is used to create a simple web interface where users can upload muted videos and view the predicted text results. This project demonstrates that speech can be recognized from visual lip movements without relying on audio signals. Such a system can be useful in applications such as silent video analysis, surveillance systems, communication support for hearing-impaired individuals, and speech recognition in noisy environments. The project also provides a foundation for future improvements using advanced deep learning models to improve accuracy and support recognition of longer and more complex speech from muted videos.

INTRODUCTION

Lip reading from muted videos is an important application of computer vision and machine learning. In many situations, videos may not contain audio or the audio may not be clear due to noise or recording issues. In such cases, understanding what a person is speaking becomes difficult. Lip reading technology helps in solving this problem by analyzing the movement of a person's lips in a video and predicting the spoken content without relying on sound.

This project focuses on developing a system that can detect and convert speech from muted videos into text. The system processes the uploaded video by extracting frames, analyzing lip movements, and identifying visual patterns related to speech. Using machine learning techniques, the system predicts the text corresponding to the spoken words. The result is displayed to the user through a simple web interface, making the system easy to use for analyzing silent videos.

SCOPE OF THE PROJECT

This project focuses on developing a system that can recognize and convert speech from muted videos into text by analyzing lip movements. The system processes the uploaded video by extracting frames and analyzing the movement patterns of the lips to understand what the speaker is saying without using any audio input. It mainly concentrates on visual speech recognition using computer vision and machine learning techniques. The project includes video preprocessing, lip region analysis, motion detection, and prediction of spoken content using a trained model. A simple web interface is provided where users can upload muted videos and view the predicted text results. The system can be useful in situations where audio is unavailable, such as silent surveillance videos, communication support for hearing-impaired individuals, and environments with heavy noise. This project also provides a base for future improvements where advanced deep learning models can be used to increase accuracy and support longer sentences and real-time lip reading.

EXISTING SYSTEM

Most existing speech recognition systems depend on audio input to convert speech into text. These systems require microphones and clear sound to work properly. However, they cannot function when audio is missing, muted, or affected by background noise. In situations such as silent videos or surveillance recordings, these systems fail to recognize speech because they rely completely on sound signals. As a result, understanding spoken content becomes difficult without audio.

PROBLEMS IN EXISTING SYSTEM

- **Dependence on Audio:** Existing systems work only when clear audio is available.
- **Poor Performance in Noise:** Background noise reduces the accuracy of speech recognition.
- **Need for Microphones:** These systems require microphones to capture speech.
- **Not Suitable for Silent Videos:** They cannot detect speech from muted or silent videos.
- **Limited Accessibility:** Audio-based systems are less useful for hearing-impaired users.

PROPOSED SYSTEM

The proposed system is designed to recognize speech from muted videos by analyzing the lip movements of the speaker, thereby enabling speech understanding in the absence of audio signals. This approach addresses the limitations of traditional speech recognition systems, which rely heavily on acoustic input and fail in silent or noisy environments. The system provides an intelligent and accessible solution for interpreting visual speech, making it particularly useful in applications such as surveillance, video analysis, assistive technologies, and communication in sound-restricted environments.

The process begins with a user-friendly web interface that allows users to upload muted video files. Once the video is submitted, it undergoes a series of preprocessing steps, including frame extraction and enhancement to improve visual clarity. Each frame is then analyzed using computer vision techniques to detect and isolate the region of interest, specifically the speaker's lips. Accurate lip detection is crucial, as it ensures that only relevant motion features are considered for further processing.

After isolating the lip region, the system employs advanced machine learning models to analyze temporal and spatial patterns of lip movements. These models are trained on large datasets of visual speech to learn correlations between lip motion sequences and corresponding textual representations. By capturing subtle variations in mouth shape, movement, and timing, the system can effectively interpret the spoken content. Techniques such as sequence modeling and pattern recognition play a vital role in improving prediction accuracy.

The extracted features are then processed to generate the predicted speech text, which is displayed to the user through the interface in a clear and readable format. The system may also include mechanisms for storing results and maintaining a history of processed videos for future reference. Additionally, optimization techniques can be applied to ensure that the system performs efficiently, delivering results within a reasonable time frame without compromising accuracy.

This visual speech recognition approach offers significant advantages, particularly in scenarios where audio data is unavailable, corrupted, or unreliable. It can be beneficial for assisting individuals with hearing impairments, enhancing video accessibility, and supporting forensic or security analysis. Furthermore, the integration of computer vision and machine learning enables continuous improvement of the system through training and adaptation to diverse speaking styles and conditions.

In conclusion, the proposed system presents an innovative and practical solution for speech recognition from muted videos. By leveraging lip movement analysis and intelligent algorithms, it transforms visual information into meaningful text, thereby expanding the capabilities of modern communication and data interpretation systems.

REQUIREMENT ANALYSIS

The proposed system for speech recognition from muted videos is designed with well-defined functional and non-functional requirements to ensure efficient and reliable performance. The system consists of several functional modules that work collaboratively to achieve the desired outcome. The **User Module** enables users to upload muted video files and view the corresponding predicted speech text generated by the system. Once a video is uploaded, the **Video Processing Module** extracts individual frames and performs necessary preprocessing to prepare the data for further analysis. The **Lip Detection Module** then identifies and isolates the lip region from each frame, focusing on mouth movements that are essential for speech prediction. These extracted features are analyzed by the **Machine Learning Module**, which interprets lip movement patterns and predicts the corresponding spoken text using trained models. The results are then presented to the user through the **Result Display Module**, which ensures clear and accessible visualization of the predicted text via a web interface. Additionally, the **Database Module** is responsible for storing uploaded video details and prediction results, thereby maintaining a history for future reference and analysis.

In terms of non-functional requirements, the system is designed to deliver high performance by processing uploaded videos and generating predicted text within a minimal time frame. Accuracy is a critical factor, and the system aims to provide reliable and precise speech predictions based on lip movements. Usability is also emphasized, ensuring that the interface is simple, intuitive, and user-friendly, allowing users to easily upload videos and view results. The system is built to be reliable, capable of handling multiple video uploads simultaneously while maintaining consistent performance without failures. Furthermore, portability is considered, enabling the system to operate efficiently on standard computing

devices without the need for high-end hardware configurations.
The computational resource requirements include both hardware and software specifications necessary for system implementation. On the hardware side, a minimum of an Intel Core i5 processor or equivalent is required, along with at least 8 GB of RAM to support model training and processing tasks, and approximately 50 GB of free storage space for data and application files. On the software side, the system requires an operating system such as Windows 10 or above, along with Python 3.10 or later as the primary programming language. Essential libraries and frameworks include TensorFlow, NumPy, OpenCV, and Streamlit for data processing, model development, and application deployment. Development activities can be carried out using tools like Visual Studio Code for coding and debugging, while Streamlit is used for deploying and running the web application locally. The system utilizes SQLite as the database for efficient data storage and management.

ARCHITECTURE
SOFTWARE ARCHITECTURE

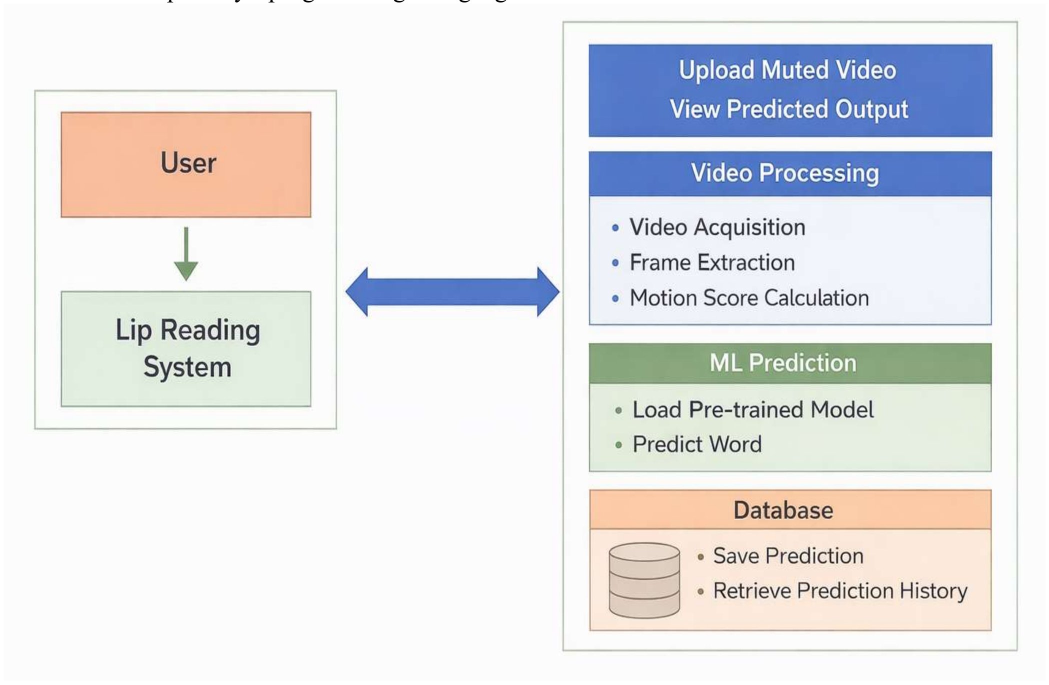


Figure-1
3.1.2 TECHNICAL ARCHITECTURE

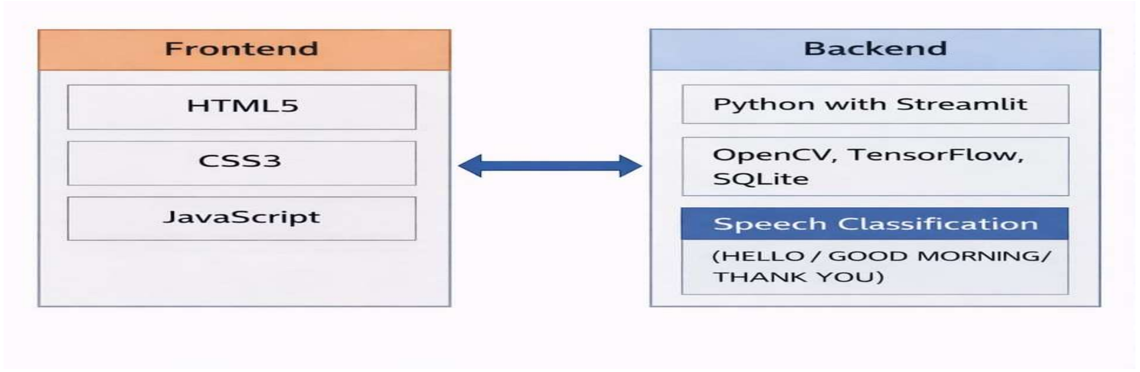


Figure-2
In the fig 3.1.2, computers, software or networks, the overall design of a computing system and the logical and physical interrelationships between its components. A system architecture is the conceptual model that defines the structure, behaviour. and more views of a system. An architecture description isa formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. A system architecture can consist of system components and the sub-systems developed that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLS).
UML DIAGRAMS

ISSN 2347–3657

Volume 14, Issue 1, 2026

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Metamodel and a notation. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the system. The main purpose of a use case diagram is to show what system functions are performed for which actor roles of the actors in the system can be depicted.

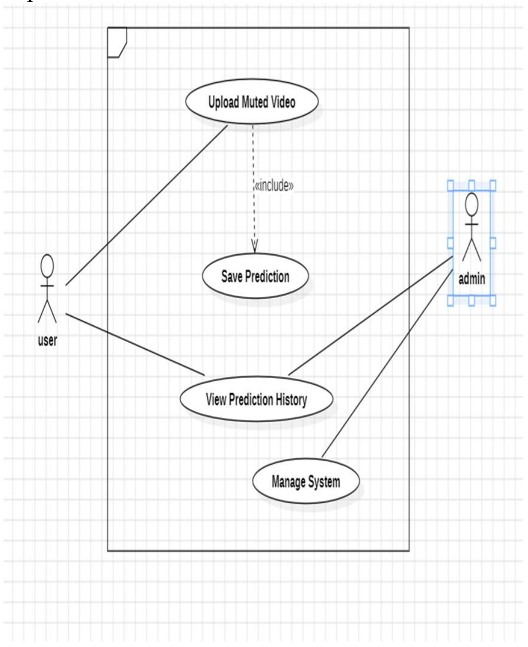


Figure-3

Implementation

The proposed system is implemented using a combination of modern programming technologies, machine learning frameworks, and web development tools to ensure efficient processing and user interaction. The core development is carried out using **Python**, a high-level and versatile programming language widely used for machine learning, data processing, and application development. For handling video and image-related operations, the system utilizes **OpenCV**, an open-source computer vision library that enables frame extraction and image preprocessing required for analyzing lip movements. **NumPy** is employed for numerical computations, particularly for converting image frames into arrays and performing mathematical operations essential for preparing input data for the machine learning model. The predictive component of the system is developed using **TensorFlow**, a powerful machine learning framework that facilitates the design and execution of models capable of analyzing lip movement patterns and generating corresponding speech text. To provide a user-friendly interface, **Streamlit** is used, allowing users to upload muted videos and view the predicted output through an interactive web application. Additionally, **SQLite** serves as a lightweight database for storing video details and prediction results, ensuring efficient data management and retrieval.

The system follows a structured data preprocessing pipeline to ensure accurate and efficient analysis. Initially, users upload a muted video through the web interface. The uploaded video is then processed using OpenCV to extract individual frames, which serve as the primary input for further analysis. These frames undergo preprocessing steps such as resizing and format conversion to make them suitable for machine learning operations. During this stage, unnecessary background information is minimized to improve the focus on relevant features. The system then performs lip region detection, identifying and isolating the area of interest in each frame to capture meaningful mouth movement data. Following this, feature extraction is carried out by converting the processed frames into numerical arrays using NumPy. These arrays are fed into the machine learning model developed using TensorFlow, which analyzes temporal and spatial patterns in lip movements to predict the corresponding spoken text. The output generated by the model is then displayed to the user through the web interface.

For video handling within the application, Streamlit provides built-in functionalities that simplify user interaction. The st.file_uploader() function allows users to upload video files directly through the interface, while the uploaded content is temporarily stored (e.g., as *temp.mp4*) for processing. The st.video() function is used to display the uploaded video, enabling users to verify the input before viewing the predicted results.

Overall, the integration of these technologies ensures a seamless workflow, from video upload and preprocessing to prediction and result display, making the system efficient, scalable, and user-friendly.

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

```python
import streamlit as st

st.title("Lip Reading From Muted
Video")

video_file = st.file_uploader("Upload
Muted Video", type=["mp4"])

if video_file is not None:
    with open("temp.mp4", "wb") as f:
        f.write(video_file.read())
    st.video("temp.mp4")
```

WEB APPLICATION FRONTEND
Developed using Streamlit, a Python web framework for building interactive ML apps.
Provides a simple user interface for uploading muted video files.
Displays uploaded video, predicted text output, and prediction history.
Handles user interaction and visualization of results.

BACKEND ROUTES
Backend implemented using Python scripts.
Handles video processing, motion analysis, and prediction logic.
Main route processes the uploaded video and sends it to the prediction pipeline.
Backend also communicates with the SQLite database to store and retrieve prediction results.

```python
import cv2

def extract_frames(video_path):
    cap = cv2.VideoCapture(video_path)
    frames = []

    while True:
        ret, frame = cap.read()
        if not ret:
            break
        frames.append(frame)

    cap.release()
    return frames
```

```python
import numpy as np

def compute_motion_score(frames):
    motion = 0
    for i in range(1, len(frames)):
        diff = np.abs(frames[i] -
frames[i-1])
        motion += np.mean(diff)

    return motion / len(frames)
```

```python
import sqlite3

conn =
sqlite3.connect("predictions.db")
cursor = conn.cursor()

cursor.execute("INSERT INTO
predictions(video_name, result) VALUES
(?, ?)",
               (video_name,
predicted_word))

conn.commit()
conn.close()
```

SYSTEM DEPLOYMENT
The application is deployed locally using the Streamlit server, enabling a lightweight and efficient execution environment without the need for complex infrastructure. Users can access the system through a standard web browser interface, ensuring ease of use and accessibility. The system operates on a standard computer equipped with a Python environment and the necessary libraries installed, making it practical and cost-effective for implementation. Furthermore, the application does not require any external server or cloud-based infrastructure, which simplifies deployment, enhances data privacy, and reduces dependency on internet connectivity.

ISSN 2347–3657

Volume 14, Issue 1, 2026

4.7 SUMMARY OF COMPONENT INTEGRATION

Component	**Function**
Streamlit	Provides the web interface for uploading muted videos and displaying the prediction results.
Python Backend	Controls the overall workflow including video processing, prediction logic and database interaction.
OpenCV	Extracts frames from the uploaded video and performs preprocessing for motion analysis.
NumPy	Performs numerical operations on image frames preprocessing for motion analysis.
SQLite Database	Stores prediction history including video name, predicted word, and timestamp.

Table-1

5. TESTING

The system was tested at multiple levels to ensure that it performs as intended and satisfies all specified functional requirements. A systematic and structured testing strategy was adopted to evaluate the performance, accuracy, and reliability of each component. Key modules such as video upload functionality, frame extraction process, motion analysis logic, machine learning prediction accuracy, and overall web interface performance were thoroughly examined. Each of these components plays a critical role in the end-to-end workflow, and their proper functioning is essential for achieving accurate speech recognition from muted videos.

During testing, the video upload module was evaluated for its ability to handle different video formats, sizes, and durations without errors or delays. The frame extraction process was tested to ensure that frames are consistently and accurately extracted from the uploaded videos using computer vision techniques. Special attention was given to the quality and consistency of extracted frames, as they directly influence the effectiveness of subsequent processing stages. The motion analysis logic, particularly the lip detection and tracking mechanisms, was carefully validated to confirm that the system correctly identifies and focuses on the lip region across varying lighting conditions, angles, and facial orientations.

The machine learning model was tested extensively to assess its prediction accuracy and robustness. This involved evaluating the model with different input samples to ensure that it can generalize well and produce reliable text outputs based on lip movement patterns. Performance metrics such as prediction correctness, response time, and consistency were considered to determine the effectiveness of the model. Additionally, the system’s ability to handle edge cases, such as unclear lip movements or low-quality video inputs, was also analyzed.

The web interface was tested for usability, responsiveness, and proper integration with backend processes. This included verifying that users can easily upload videos, view the processed results, and interact with the application without technical difficulties. The interface was also evaluated for its ability to display results clearly and in real time, enhancing the overall user experience.

Overall, this comprehensive testing approach ensures that each module functions correctly in isolation while also working seamlessly when integrated. It helps identify potential issues early, improves system reliability, and ensures that the final application delivers accurate, efficient, and user-friendly performance.

The primary objectives of testing were to verify that the system correctly accepts muted video inputs and processes them efficiently. It was also essential to ensure that frames are accurately extracted from the uploaded videos and that the lip motion analysis module functions as expected. Additionally, the testing aimed to validate that the machine learning model generates accurate predictions based on lip movement patterns. The proper functioning of the web interface was also examined to confirm that results are displayed clearly to the user. Furthermore, the system was tested to ensure that

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

prediction results are successfully stored in the database for future reference.
Testing was conducted in multiple stages, starting with unit testing, where individual components of the system were tested separately to identify and resolve any issues at the module level. For example, a test case was executed using a muted video containing clear lip movements, where the expected output was a specific word such as "HELLO." This helped in validating the correctness of the prediction module.
Following unit testing, integration testing was performed to ensure that different components of the system work seamlessly together. This included verifying the interaction between the Streamlit-based user interface and the Python backend, ensuring proper coordination between the video processing module and the prediction logic, and validating the integration of the machine learning model with the database for storing results. Through this multi-level testing approach, the system's reliability, accuracy, and overall performance were effectively ensured.

FUNCTIONAL TESTING

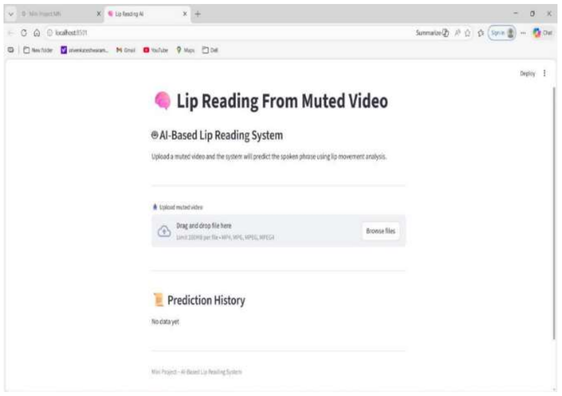
Validated all functionalities as per functional requirements:

Table-1

Test Name	Input	Expected output	Actual Output	Result
Video Upload	Muted Video	Video displayed	Displayed successfully	Pass
Frame Extraction	Uploaded video	Frames extracted	Frames extracted	Pass
Motion Analysis	Video frames	Motion score generated	score generated	Pass
Prediction	Motion score value	Correct word predicted	Correct output	Pass
Database Storage	Prediction result	Stored in database	Stored successfully	Pass
UI Display	Predicted word	Displayed on Screen	Displayed correctly	Pass

Table-3

6. SCREENSHOTS


Lip Reading From Muted Video
AI-Based Lip Reading System
Upload a muted video and the system will predict the spoken phrase using lip movement analysis.
Upload muted video
Drag and drop file here
Limit 200MB per file • MP4, MPG, MPEG, MPEG4
Prediction History
No data yet
Mini Project – AI-Based Lip Reading System

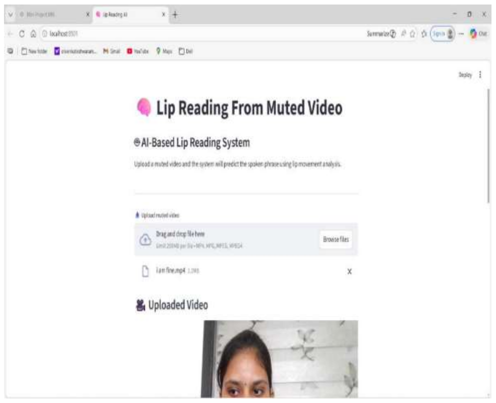
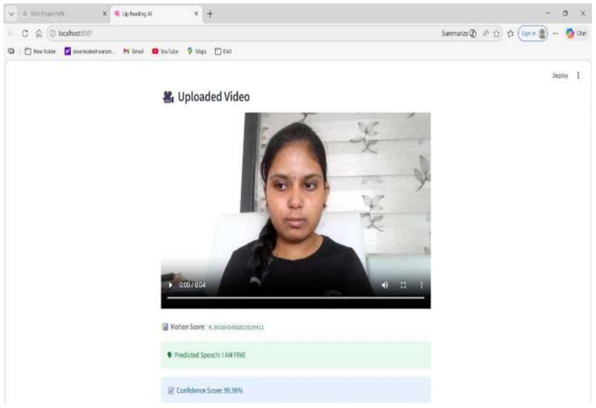

Lip Reading From Muted Video
AI-Based Lip Reading System
Upload a muted video and the system will predict the spoken phrase using lip movement analysis.
Upload muted video
Drag and drop file here
Limit 200MB per file • MP4, MPG, MPEG, MPEG4
I am fine.mp4
Uploaded Video

Figure-6.1


Uploaded Video
Motion Score: [illegible]
Predicted Speech: I AM FINE
Confidence Score: 99.98%

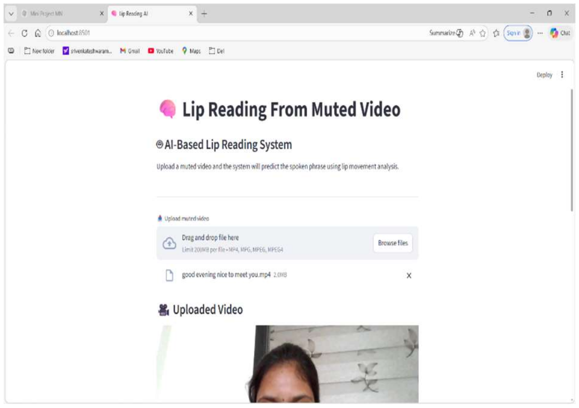
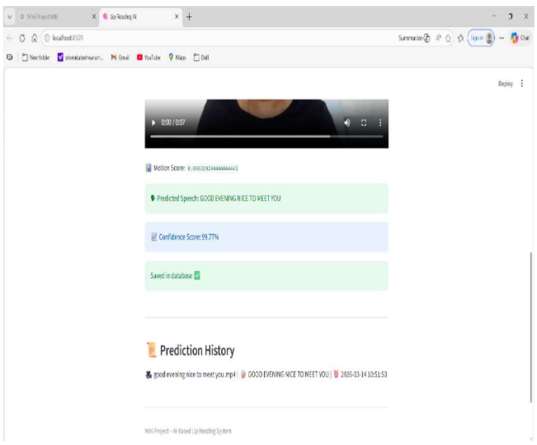

Lip Reading From Muted Video
AI-Based Lip Reading System
Upload a muted video and the system will predict the spoken phrase using lip movement analysis.
Upload muted video
Drag and drop file here
Limit 200MB per file • MP4, MPG, MPEG, MPEG4
good evening nice to meet you.mp4 2.0MB
Uploaded Video

Figure-6.2


Lip Reading AI

localhost:8501

Motion Score: 0.6663292444444443

Predicted Speech: GOOD EVENING NICE TO MEET YOU

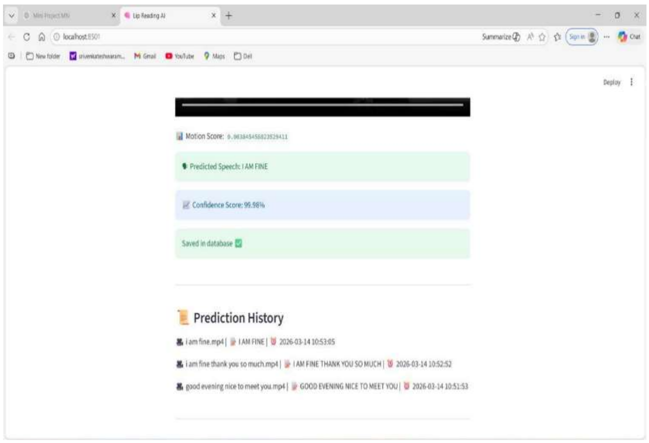
Confidence Score: 99.77%

Saved in database ✅

Prediction History

good evening nice to meet you.mp4 | GOOD EVENING NICE TO MEET YOU | 2026-03-14 10:51:53

Mini Project - AI-Based Lip Reading System


Lip Reading AI

localhost:8501

Motion Score: 0.983845456823529411

Predicted Speech: I AM FINE

Confidence Score: 99.98%

Saved in database ✅

Prediction History

i am fine.mp4 | I AM FINE | 2026-03-14 10:53:05

i am fine thank you so much.mp4 | I AM FINE THANK YOU SO MUCH | 2026-03-14 10:52:52

good evening nice to meet you.mp4 | GOOD EVENING NICE TO MEET YOU | 2026-03-14 10:51:53

Figure-6.3

1. CONCLUSION AND FUTURE SCOPE

This mini project successfully demonstrates that speech can be recognized from visual lip movements without using audio. The system analyzes facial motion and predicts basic spoken words using machine learning techniques. Though the current system supports limited vocabulary, it proves the concept of visual speech recognition. With advanced deep learning models and larger datasets, this system can be expanded to recognize complete sentences, helping hearing-impaired people and enabling silent video analysis. The combination of video processing, machine learning, and web interface technologies enables the system to effectively recognize speech from muted videos.

FUTURE SCOPE:

The proposed system can be further enhanced by incorporating advanced deep learning techniques to improve the accuracy and efficiency of lip reading. In the future, larger and more diverse datasets can be used to train the model so that it can recognize different speaking styles, facial expressions, and variations in lip movements. This will help the system provide more accurate predictions for a wider range of users and improve its overall reliability.

The project can also be extended to recognize complete sentences and continuous speech instead of predicting limited text from muted videos. Another possible improvement is implementing real-time lip reading from live video streams, which can be useful in applications such as video conferencing, security systems, and assistive technologies. In addition, integrating the system with more advanced computer vision models and cloud-based platforms can help handle larger data and improve processing speed, making the system more scalable and practical for real-world applications.

REFERENCES

[1] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[2] M. Abadi, P. Barham, J. Chen et al., "TensorFlow: A System for Large-Scale Machine Learning," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

[3] A. Streamlit Inc., "Streamlit: The Fastest Way to Build Data Apps in Python," *Streamlit Documentation*, 2023.

[4] A. Ng, "Computer Vision Basics," *Coursera Online Course*, Stanford University, 2022.

[5] R. Lutz, "Python Programming Language," *W3Schools Online Web Tutorials*, 2024.

[6] Y. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-End Sentence-Level Lipreading," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.