IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Ai-Powered Lip Reading Assistant For Individuals With Hearing Impairment

Aditi Wangikar, Arnav Khavale, Sanika Sharma, Udayraj Gadekar, Udita Sinha
Department of Computer Science & Engineering
MIT ADT University
Pune 412201, India

Abstract: The ability to read spoken words visually through lip reading is an important skill for those with hearing impairments. Traditional lip-reading is very dependent on the skill of the individual and is prone to mistakes. An AI-powered lip-reading assistant that efficiently converts visual speech into text using deep learning algorithms is proposed in this paper. To process video inputs and generate corresponding transcriptions, the system combines Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs). The experimental findings demonstrated an increase in accuracy compared to existing methods. Real-time transcription services and assistive devices are just two of the many uses for the suggested system, which has the potential to transmogrify communication for the hard of hearing.

Index Terms **- Visual speech recognition, hearing impairment, CNN, LSTM, AI, deep learning, lipreading and speech recognition.**

I. INTRODUCTION

Visual Speech Recognition (VSR), another name for lip-reading, is the process of interpreting spoken language by visually observing lip movements without the use of auditory input. Due to its potential to close communication gaps, especially for those who are deaf or hard of hearing, this technique has gained a lot of traction in recent years. Beyond accessibility, lip-reading is essential for secure human-computer interaction, driver safety systems, surveillance, voice recovery in noisy environments, and silent speech interfaces.

In today's communication technology, Multimodal inputs are taking on greater significance. For performance improvement in intricate environments where audio may be unavailable or unreliable, visual cues like lip movements are being integrated with traditional audio-based systems. For example, the use of silent speech recognition is being investigated in noisy industrial environments or during secret operations where it is not feasible to communicate verbally. Visual speech recognition can also improve the performance of smart assistants in situations where there are overlapping conversations or poor audio quality.

Growing research in lip-reading is also fueling advancements in security and surveillance technologies. To support forensic investigations, VSR systems are being tested for their ability to extract information from video footage in situations where audio capture is scarce or nonexistent. Without the use of audio signals, lip-reading in driver assistance systems allows real-time driver monitoring to identify signs of distraction or drowsiness based on speech patterns.

Due to the inherent ambiguity of visemes, which are visual equivalents of phonemes and appear similarly on the lips (e.g., 'p', 'b', and 'm'), human lip-readers have historically shown limited accuracy. Earlier computer methods for lipreading used manually created features such as Hidden Markov Models and Active Appearance Models. These models, however, had difficulty with accuracy and real-time performance in dynamic environments.

Recent developments in deep learning, specifically in Transformer-based architectures, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have greatly enhanced machines' comprehension and interpretation of lip movements. Without the use of manually created features, these models can recognize complicated temporal and spatial patterns from video frames.

Numerous AI-powered lip-reading programs are currently in development. For instance, Google's DeepMind unveiled ‘Watch, Attend and Spell’, a deep learning-based model that showed encouraging outcomes on GRID and TCD-TIMIT as well. Other noteworthy systems include LipNet, which performs sentence-level lip-reading using spatiotemporal CNNs and RNNs. By fusing 3D convolutional networks with sequence modeling techniques, Microsoft's Visual Voice system and initiatives from universities such as the University of Oxford are also advancing the field of visual speech recognition.

The hurdles still exist despite such developments. Real-world performance is still constrained by variations in lighting, camera angles, speaker facial morphology, and blockages (such as masks or mustaches). More varied, openly accessible datasets that represent the linguistic and ethnic diversity of people around the world are also required.

This paper proposes a deep learning-powered AI lip-reading assistant to improve hearing-impaired users' communication accessibility and expand the use of VSR systems to security, silent interfaces, and assistive technology. With the use of CNNs and attention-based architectures, this system uses spatiotemporal modeling of lip movements to provide reliable, real-time speech interpretation independent of acoustic signals.

2. OBJECTIVE

The main goal of this work is to create an AI lip-reading assistant that uses deep learning to accurately interpret spoken words from real-time visual lip movements. While expanding its usefulness to applications in noisy environments, silent command interfaces, and surveillance systems, the proposed system seeks to improve communication accessibility for people with hearing impairments. Through the use of sophisticated deep learning architectures like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and attention mechanisms, the system focuses on recognizing visual speech with high accuracy across a variety of people, their languages, and various environments. The goal also includes developing a scalable and reliable model that can function well in real-world situations with varying lighting, occlusions, and camera angles, thus advancing the larger field of assistive technology and human-computer interaction.

3. TOOLS AND LANGUAGE

1. Languages Used:

- Python: Model inference, preprocessing, and backend logic.
- HTML: Structure of front-end web pages.
- JavaScript: Webcam recording and backend communication.

2. Tools, Libraries, and Frameworks Used:

- OpenCV: Recording and analyzing video frames.
- NumPy: Managing numerical operations and arrays.
- ONNX Runtime: Running the deep learning model.
- Flask: Backend APIs & routing.
- ONNX: Pretrained lip reading model's format.
- Long Short-Term Memory or LSTM: A deep learning model architecture.
- TensorFlow/PyTorch: Used during model training.
- Convolutional Neural Network or CNN: Extract features from each video frame showing lip movements.

4. PROCESS AND ARCHITECTURE

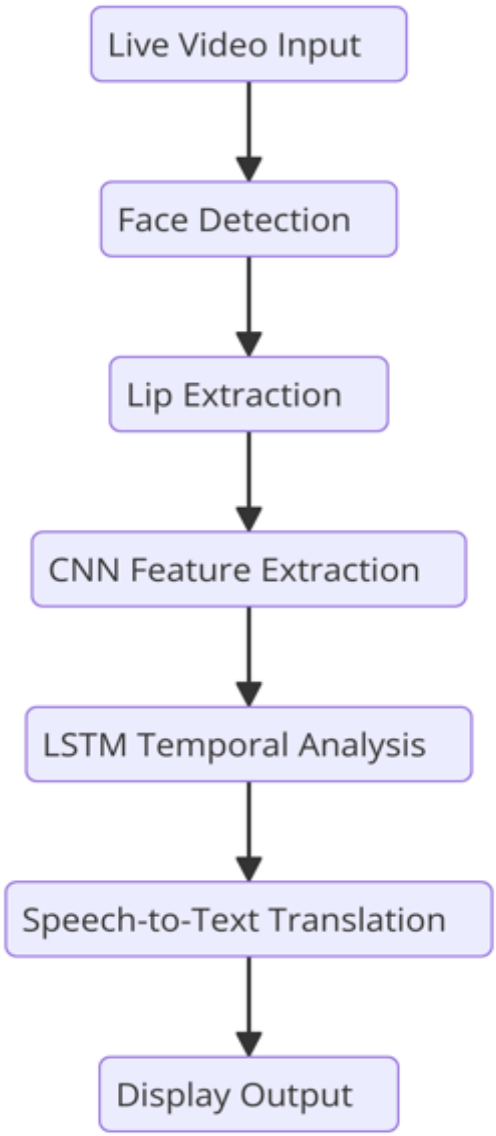

Fig. 4.1 Flow chart of the UI

The suggested AI-powered lip-reading assistant processes visual speech and produces precise text outputs by utilizing a mix of deep learning architectures, programming languages, and frameworks. A thorough, step-by-step workflow outlining the integration and operation of every system component can be found below:

1. User Interface (Frontend Interaction Layer)

- Languages Used: JavaScript, HTML.
- Functionality:
  - A basic web interface with HTML for structure and JavaScript for dynamic behavior is used to interact with the user.
  - JavaScript records live video from the webcam and is frequently used in conjunction with the browser's MediaDevices API.
  - Asynchronous JavaScript calls (such as `fetch()` or `WebSocket`) are used to send these recorded video frames to the backend server.

2. Layer of Backend Processing

- Languages & Frameworks: Flask, Python.
- Functionality:
  - Python is used to run models, handle incoming video streams, and carry out preprocessing.
  - The backend API is provided by the lightweight Python web framework Flask. It interacts with the AI model, manages HTTP requests, and routes data.
  - Video frames are sent to the backend, which uses pre-established procedures to process them before feeding them into the AI model.

3. Feature extraction and video preprocessing

- Tools Used: NumPy, OpenCV.
- Functionality:
  - OpenCV(Open Source Computer Vision Library detects the required features/objects, makes them compatible with the library by converting them into grayscale and resize them in order to fit the input frame.
  - Before feeding the data to the model, NumPy helps with numerical operations like frame stacking, pixel normalization, and array transformations.

4. CNN combined with LSTM Model Architecture for the Model's Inference Layer

- Format of the Model: ONNX
- Framework for Execution: ONNX Runtime
- Usability:
  - Usually trained with PyTorch or TensorFlow, the deep learning model is exported to the ONNX (Open Neural Network Exchange) format for deployment across platforms.
  - The model combines Long Short-Term Memory (LSTM) units to capture temporal patterns across multiple video frames with Convolutional Neural Networks (CNNs) for spatial feature extraction (e.g., lip shape, movement).
  - The pretrained model is efficiently run by the ONNX Runtime, allowing for quick and portable inference in a variety of settings.

5. Interpretation and Display of Output

- The predicted text (spoken word or sentence) is sent back to the frontend by the backend after inference is finished.
- Based only on visual speech cues, the user receives immediate feedback as the text is shown on their screen in real time.

6. Additional upload mechanism

- Users can also upload their video to encode the lip movements and receive the transcription from the model.

5. ALGORITHM

1. Face Detection by using MTCNN - Multi-Task Cascaded Convolutional Network

- Detects faces in video frames using a hierarchical CNN-based structure.
- Extracts prominent facial features, e.g., the lips, for subsequent processing.

2. Lip Area Extraction by using OpenCV Facial Feature Detection

- Identifies and separates the lip region for evaluation.
- Aligns the extracted lip area to the average to keep the model input consistent.
- Feature Extraction (ResNet-50 - CNN-Based)
- Collects spatial data from strings of lip movement.
- Apply convolutional filters to detect lip movement patterns between frames.

3. Temporal Analysis by applying Bi-directional LSTM
   - Captures temporal dependencies in lip movement patterns.
   - CNN feature extraction for capturing word-level differences.
   - Speech-to-Text Conversion of the person in frame.
   - Maps strings of lip movements to words and phonemes.
   - Employing probabilistic sequence alignment when generating text transcriptions.

4. Error Correction & Contextual Refinement by using Transformer-Based NLP Model - BERT/GPT
   - Learns from past predictions of words.
   - Improves the accuracy of transcribed words by correcting spellings and determining the right spoken words.

5. User Interface by using the GUI - Tkinter/PyQt
   - Offers real-time transcription of spoken language. Offers accessibility features like text highlighting and speech playback.

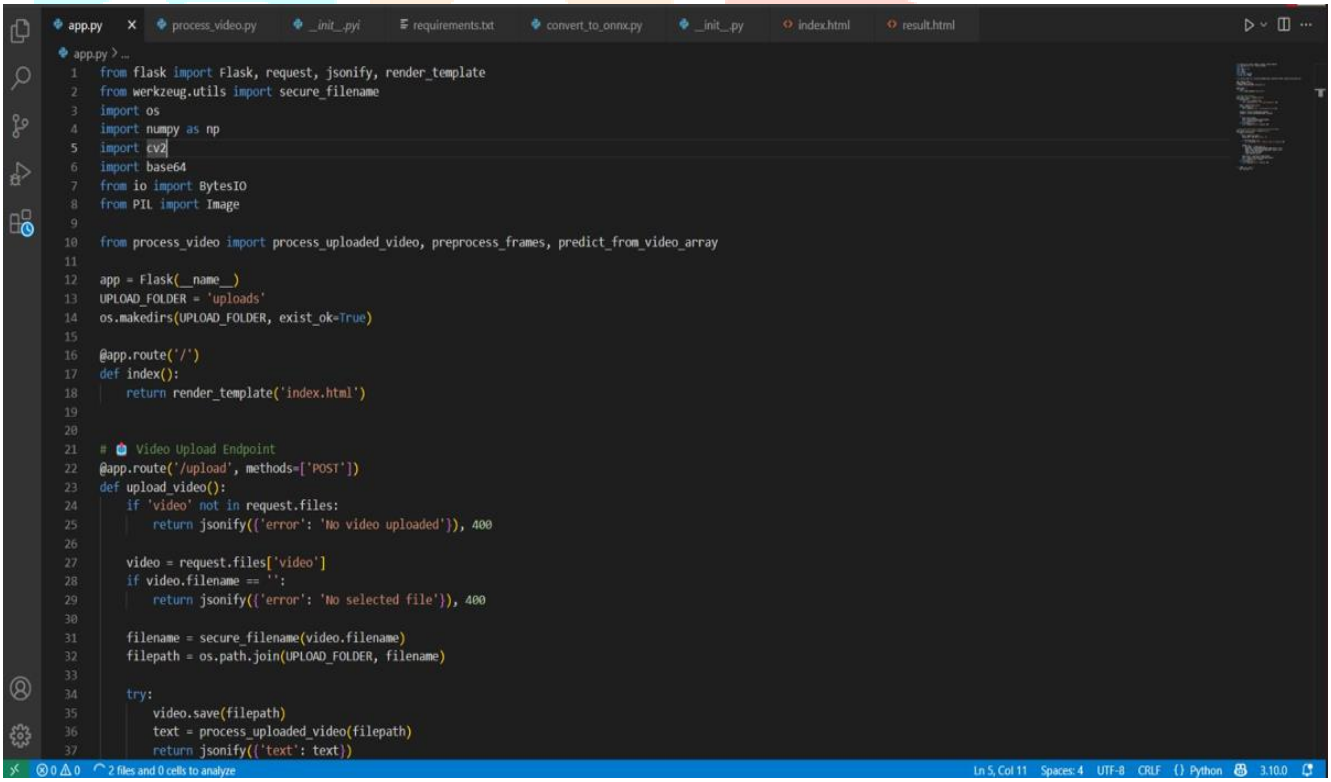

Fig. 5.1 Code for backend API

app.py 1 process_video.py _init_.pyi requirements.txt convert_to_onnx.py _init_.py index.html result.html

process_video.py > ...

```python
# process_video.py
import onnxruntime as ort
import numpy as np
import cv2

# Load the ONNX model
session = ort.InferenceSession("model/lip_reading_model.onnx")
input_name = session.get_inputs()[0].name

print("Model input shape:", session.get_inputs()[0].shape)
print("Model output shape:", session.get_outputs()[0].shape)

# Character set used by the model (28 classes: 26 letters + space + blank for CTC)
chars = list("abcdefghijklmnopqrstuvwxyz ")  # no CTC blank handling since not trained with CTC

def preprocess_frames(frames):
    processed = []
    for frame in frames:
        # Resize correctly: (width, height) = (100, 50) → shape becomes (50, 100, 3)
        frame = cv2.resize(frame, (100, 50))  # (W, H)
        processed.append(frame)

    video_array = np.array(processed)        # (75, 50, 100, 3)
    video_array = np.expand_dims(video_array, axis=0)  # (1, 75, 50, 100, 3)
    return video_array.astype(np.float32)

def predict_from_video_array(video_array):
    outputs = session.run(None, {input_name: video_array})  # shape: (1, 75, 28)

    predicted_class = np.argmax(outputs[0], axis=-1)  # (1, 75)
    predicted_text = ''.join([chars[i] for i in predicted_class[0]])  # Convert each frame's output to char
    return predicted_text.strip()

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

Ln 69, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.10.0

Fig. 5.2 Code for video processing

6. RESULTS

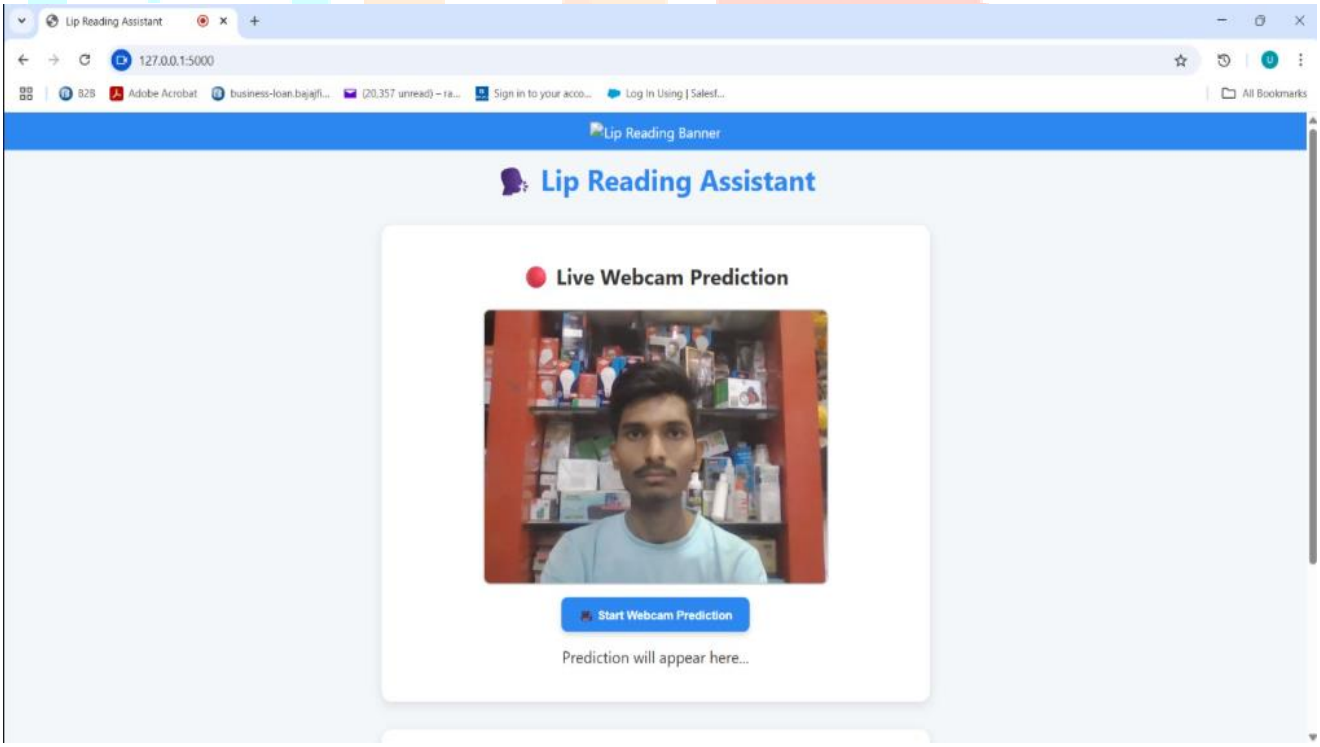

Fig. 6.1 Interface for Lip-Reading Assistant

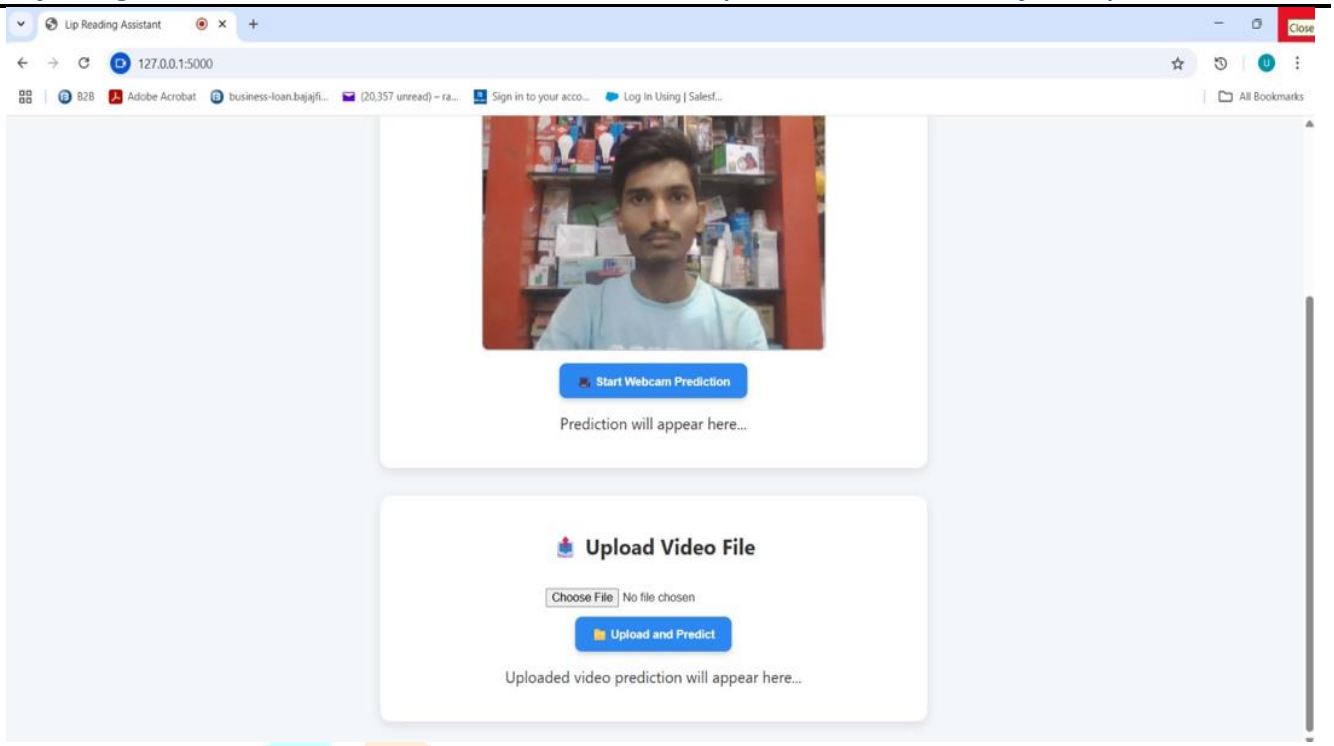

Fig. 6.2 Prediction using Webcam / Uploading Video File

There will be a textual representation of the spoken words deduced from a series of video frames displaying lip movements as the result of the suggested lip-reading system. The transcription will appear at the place of "Prediction will appear here…" just below 'Start Webcam Prediction'. The system decodes the learned spatiotemporal features to produce precise word or sentence-level predictions after running the video input through the deep learning model (CNN + LSTM). These predictions, generated frame-by-frame are instantly rendered on the interface in sync with the user's spoken input.

7. ACKNOWLEDGEMENT

We want to sincerely thank everyone who helped us with this project, **"AI-POWERED LIP READING ASSISTANT FOR INDIVIDUALS WITH HEARING IMPAIRMENT"**. We want to start by extending our heartfelt thanks to our mentor, **Prof. Aditi Wangikar**, for her unwavering support, insightful advice and constant guidance throughout this project. Our comprehension and performance of the work were greatly influenced by her knowledge and guidance.

We are also appreciative of each team member's commitment and spirit of cooperation. The combined efforts and contributions of the following people enabled this project:

- **Arnav Khavale**
- **Sanika Sharma**
- **Udayraj Gadekar**
- **Udita Sinha**

Every member contributed special talents to the team and this project is evidence of our collaboration, tenacity and common goal. We also want to express our gratitude to our institution for giving us the tools and assistance we needed to finish this project successfully.

Last but not least, we are grateful to the researchers and developers whose earlier work and open-source tools had a significant impact on our learning and development process.

8. CONCLUSION

By offering a technologically advanced solution for people who are unable to understand spoken language through sound, the project "AI-Powered Lip Reading Assistant for Individuals with Hearing Impairment" tackles a significant obstacle in inclusive communication. This system effectively converts silent lip movements into readable text using cutting-edge deep learning techniques, allowing people with hearing loss to comprehend spoken content without the use of conventional hearing aids or sign language interpreters.

The assistant incorporates important technologies like LSTMs (Long Short-Term Memory networks) to record the temporal patterns across frame sequences and CNNs (Convolutional Neural Networks) to extract spatial features from lip regions in video frames. This combination guarantees that precise speech is produced using both the lips' shape and their movement over time. After being trained with frameworks such as TensorFlow or PyTorch, the model is transformed into the ONNX format for lightweight deployment, and it is run using ONNX Runtime to provide quick and effective performance.

The system is well-suited for real-time use, with Python managing the backend logic, Flask acting as the API framework, and tools like OpenCV and NumPy handling image processing and numerical computations. The system is easy to use and accessible thanks to the frontend interface, which was created with HTML and JavaScript and enables webcam interaction.

With this project, we showed that visual speech recognition can be an effective tool for developing assistive technologies that support accessibility and independence. The system has the potential to be expanded to multiple languages and can function in a variety of environments.

In conclusion, the foundation for future applications in accessibility, silent communication in noisy environments, human-computer interaction, and surveillance is laid by this AI-powered assistant. Larger datasets and additional development could make the system a very useful practical tool that closes communication gaps and makes a significant contribution to social inclusion.

9. REFERENCES

References within the Main Content of the Research Paper:

[1] G. Geetha, A. Kaur, and P. Subramanian, "Deep learning techniques for visual speech recognition: A comprehensive survey," Journal of Ambient Intelligence and Humanized Computing, vol. 15, no. 2, pp. 843–858, Feb. 2024.

[2] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 1, pp. 36–50, Jan. 2022.

[3] "ONNX: Open Neural Network Exchange." [Online]. Available: https://onnx.ai

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] A. Wadhwa and V. Prabhakaran, "Real-time lip reading using spatiotemporal features and deep learning," Procedia Comput. Sci., vol. 173, pp. 162–170, 2020.