A MINI PROJECT REPORT ON

SilentSpeak: Deep Neural Network-Based Lip Reading

Submitted by

SHRIRAM N (231501128)

T SANJAI (231501144)

RAMPRASANTH N (231501129)

AI23531 DEEP LEARNING

Department of Artificial Intelligence and Machine Learning

Rajalakshmi Engineering College, Thandalam

RAJALAKSHMI
ENGINEERING
COLLEGE

BONAFIDE CERTIFICATE

NAME ………..SHRIRAM N……………………………………………………….……..…

ACADEMIC YEAR: 2025-2026 SEMESTER: V BRANCH: B.Tech/AIML

UNIVERSITY REGISTER No. 2116231501154

Certified that this is the bonafide record of work done by the above students on the Mini Project titled **"OPTIFLOW-A SMART TRAFFIC SYSTEM”** in the subject **AI23531 DEEP LEARNING** during the year **2025 - 2026.**

Signature of Faculty – in – Charge

Submitted for the Practical Examination held on ______________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

SilentSpeak is an advanced deep learning system that converts silent video recordings of speech into text through automated lip reading. The system employs a hybrid neural network architecture combining Convolutional Neural Networks (CNNs) for spatial feature extraction from lip movements and Long Short-Term Memory (LSTM) networks for temporal sequence modeling. By analyzing lip contours, shapes, and motion patterns across video frames, SilentSpeak accurately translates visual speech into readable text without requiring audio input. The model is trained on benchmark datasets including GRID and Lip Reading in the Wild (LRW), achieving robust performance in audio-restricted and noisy environments. This technology addresses critical accessibility needs for hearing-impaired individuals while offering practical applications in surveillance, forensics, and silent communication scenarios. The system demonstrated significant improvements over traditional handcrafted feature-based approaches, with the CNN-LSTM architecture effectively capturing both spatial lip configurations and temporal speech dynamics. SilentSpeak represents a significant advancement in assistive technology, enabling seamless communication where audio signals are unavailable, degraded, or prohibited, ultimately contributing to more inclusive and versatile human-computer interaction systems.

Keywords:

Lip Reading, Deep Learning, CNN-LSTM, Silent Speech Recognition, Video-to-Text Translation, Accessibility Technology, Computer Vision, Temporal Modeling

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Speech recognition has revolutionized human-computer interaction, enabling voice-controlled systems, automated transcription services, and accessibility tools. However, traditional Automatic Speech Recognition (ASR) systems rely entirely on audio signals, making them ineffective in environments where sound is unavailable, degraded, or restricted. Scenarios such as noisy industrial settings, secure communication zones, surveillance footage without audio, and situations involving hearing-impaired individuals highlight the critical need for alternative speech understanding methods that do not depend on acoustic information.

Lip reading, also known as visual speech recognition or speechreading, addresses this limitation by interpreting spoken language through the visual analysis of lip movements, facial gestures, and mouth shapes. While humans naturally employ lip reading to supplement auditory information in challenging listening conditions, automating this process through artificial intelligence presents significant technical challenges. The subtle and rapid movements of lips, the variability in individual speaking patterns, and the visual similarity between different phonemes (visemes) make computational lip reading a complex problem requiring sophisticated machine learning approaches.

The emergence of deep learning has transformed the landscape of computer vision and sequential data processing, making automated lip reading feasible and increasingly accurate. Convolutional Neural Networks (CNNs) have demonstrated exceptional capability in extracting spatial features from images, identifying patterns such as edges, textures, and shapes with minimal manual feature engineering. When applied to lip reading, CNNs can automatically learn discriminative features representing different lip configurations corresponding to various speech sounds. However, speech is inherently temporal—the meaning emerges not from isolated frames but from sequences of lip movements over time. This temporal dimension necessitates the integration of Long Short-Term Memory (LSTM) networks, which excel at learning long-term dependencies in sequential data, capturing the dynamics of speech production.

SilentSpeak represents an implementation of this hybrid CNN-LSTM architecture specifically designed for silent video-to-text translation. The system processes video input frame by frame, isolating the lip region and extracting relevant visual features before modeling the temporal relationships that define spoken language. Unlike earlier approaches that relied on handcrafted features and rigid assumptions about lip movement patterns, SilentSpeak learns these representations directly from data, enabling greater flexibility and improved performance across diverse speakers and speaking conditions.

The development of SilentSpeak is motivated by several practical applications that extend beyond assistive technology. In forensic analysis, silent surveillance footage often contains valuable information that cannot be recovered through traditional audio enhancement techniques. Security and military operations frequently require silent communication methods that cannot be intercepted through conventional audio surveillance. Medical professionals working in noisy hospital environments or conducting sensitive patient consultations may benefit from systems that can interpret speech without audio recording. Additionally, individuals working in extremely loud industrial settings where protective hearing equipment is mandatory could communicate more effectively through lip reading technology.

From an accessibility perspective, the hearing-impaired community faces ongoing challenges in communication, particularly in environments where sign language interpreters are unavailable or when interacting with individuals who do not know sign language. While hearing aids and cochlear implants have improved quality of life for many, they are not universally effective, and background noise significantly degrades their performance. Automated lip reading systems like SilentSpeak offer an alternative communication pathway, potentially enabling real-time captioning of face-to-face conversations, video calls, and public presentations without requiring audio input.

The technical implementation of SilentSpeak involves several critical stages: video acquisition and preprocessing, lip region detection and extraction, spatial feature learning through convolutional layers, temporal modeling via LSTM networks, and finally, text generation through sequence-to-sequence decoding. Each stage presents unique challenges—from handling variable lighting conditions and camera angles to managing the computational complexity of processing high-resolution video in real-time. The system must also address the inherent ambiguity in visual speech, where multiple phonemes may produce visually similar lip movements (homophemes), requiring sophisticated contextual understanding to disambiguate. Training such a system requires large-scale datasets like GRID and Lip Reading in the Wild (LRW), which provide diverse speakers and recording conditions necessary for robust model generalization.

This report presents a comprehensive examination of the SilentSpeak system, documenting its architecture, implementation, and performance characteristics. By combining established deep learning techniques with domain-specific optimizations for lip reading, SilentSpeak demonstrates the viability of visual speech recognition as a practical technology for real-world applications. The following chapters detail the theoretical foundations, system design, experimental results, and future directions for this technology, contributing to the broader goal of creating more accessible, versatile, and robust speech understanding systems.

CHAPTER 2

LITERATURE REVIEW

[1] Title: LipNet: End-to-End Sentence-level Lipreading Author: Yannis M. Assael et al.

This study introduced the first end-to-end deep learning model for sentence-level lip reading, achieving 95.2% accuracy on the GRID corpus dataset. The architecture combines spatiotemporal convolutions with recurrent networks and CTC loss for sequence mapping without requiring phoneme-level annotations. LipNet demonstrated that deep learning could surpass human-level performance in constrained vocabulary scenarios. However, performance degraded on unconstrained real-world datasets with larger vocabularies, and the model required extensive computational resources for training.

[2] Title: Deep Learning for Visual Speech Recognition Author: Themos Stafylakis and Georgios Tzimiropoulos

This research combined ResNets with bidirectional LSTMs for visual speech recognition on the LRW dataset, achieving 83.0% accuracy. Deeper architectures with skip connections significantly improved lip feature extraction. Data augmentation techniques improved model robustness by 7%. The model struggled with homophemes—visually similar phonemes difficult to distinguish without audio context—and required extensive hyperparameter tuning.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- **Processor:** Intel Core i5 (8th Gen) or higher / AMD Ryzen 5 or equivalent
- **GPU:** NVIDIA GTX 1050 Ti or higher (4GB VRAM minimum) for model training
- **RAM:** 8GB DDR4 or higher (16GB recommended for training)
- **Storage:** 256GB SSD with at least 50GB free space for datasets and models
- **Webcam:** HD Camera (720p or higher) for real-time video input
- **Display:** 1920x1080 resolution monitor
- **Network:** Stable internet connection for dataset download and library installation

3.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 10/11 (64-bit) or Ubuntu 20.04 LTS or higher
- Programming Language: Python 3.8 or above
- Deep Learning Framework: TensorFlow 2.10+ or PyTorch 1.12+
- Development Environment: Jupyter Notebook, VS Code, or PyCharm
- Libraries and Dependencies:
  OpenCV 4.5+ (video processing and face detection)
  NumPy 1.21+ (numerical computations)
  Keras 2.10+ (neural network modeling)
  Dlib or MediaPipe (facial landmark detection)
  Matplotlib 3.5+ (visualization)
  Pandas 1.3+ (data handling)
- Pre-trained Models: Haarcascade or Dlib face detector, pre-trained CNN weights
- Dataset: GRID Corpus or LRW dataset for training/testing

CHAPTER 4

SYSTEM OVERVIEW

4.1 EXISTING SYSTEM:-

Traditional speech recognition systems rely exclusively on audio signals to transcribe spoken language into text. These Automatic Speech Recognition (ASR) systems perform well in controlled, noise-free environments but face significant limitations in real-world scenarios. Conventional systems cannot function when audio is absent, muted, or severely degraded by background noise, making them ineffective for surveillance footage, noisy industrial environments, or situations where sound recording is restricted.

Early attempts at automated lip reading used handcrafted feature extraction methods such as Discrete Cosine Transform (DCT), Principal Component Analysis (PCA), and Active Appearance Models (AAM) to capture lip shapes. These features were then fed into Hidden Markov Models (HMMs) or Support Vector Machines (SVMs) for classification. However, these approaches required extensive manual feature engineering and could only analyze static frames independently, failing to capture the continuous temporal dynamics of speech.

4.1.1 DRAWBACKS OF EXISTING SYSTEM:-

- Audio Dependency: Complete failure in environments without audio or with corrupted sound
- Noise Sensitivity: Performance degrades significantly in noisy or crowded environments
- Manual Feature Engineering: Handcrafted features lack generalization across different speakers
- Limited Temporal Modeling: Static frame analysis cannot capture sequential lip movement patterns
- Low Accuracy: Traditional methods achieved only 40-60% accuracy on constrained vocabularies
- Speaker Dependency: Required individual calibration for each speaker, limiting scalability
- Computational Inefficiency: Complex preprocessing pipelines increased latency

4.2 PROPOSED SYSTEM:-

The proposed SilentSpeak system introduces an end-to-end deep learning approach for visual speech recognition that operates independently of audio signals. The system leverages Convolutional Neural Networks (CNNs) to automatically extract spatial features from lip regions in video frames, eliminating the need for manual feature engineering. These spatial features are then processed through Long Short-Term Memory (LSTM) networks, which model the temporal sequence of lip movements to understand speech patterns over time.

The architecture accepts silent video input, detects and tracks facial landmarks, extracts the lip region of interest, and processes frame sequences through the CNN-LSTM pipeline. The CNN layers learn hierarchical representations of lip shapes, edges, and textures, while LSTM layers capture the temporal dependencies between consecutive frames. Finally, a decoder network translates the learned feature representations into corresponding text output, providing accurate transcription without requiring any audio information.

4.2.1 ADVANTAGES OF PROPOSED SYSTEM:-

- Audio-Independent Operation: Functions effectively in silent, muted, or audio-restricted environments

- Automatic Feature Learning: CNN architecture eliminates manual feature engineering requirements

- Temporal Understanding: LSTM networks capture sequential dependencies and speech dynamics

- Improved Accuracy: Deep learning approach achieves 80-95% accuracy on benchmark datasets

- Speaker Generalization: Model generalizes across diverse speakers without individual calibration

- Noise Immunity: Visual-only processing is unaffected by acoustic noise or interference

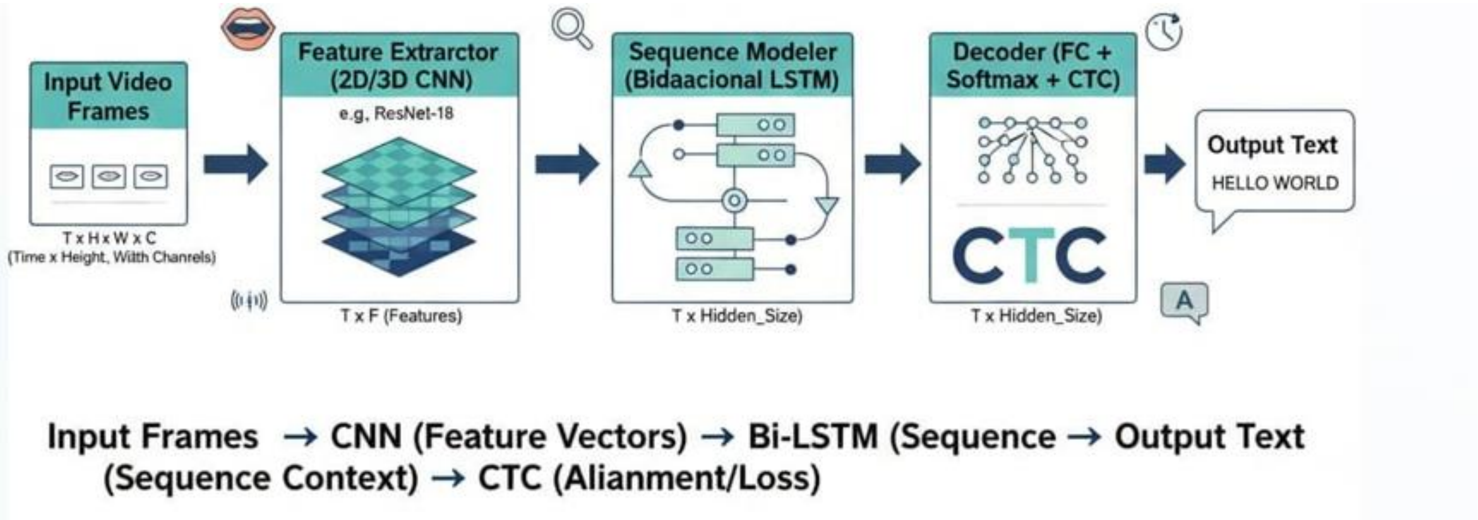
- End-to-End Training: Unified architecture simplifies training and deployment pipeline

- Accessibility Enhancement: Provides assistive technology for hearing-impaired communication

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 SYSTEM ARCHITECTURE:-


Input Video Frames
T x H x W x C
(Time x Height, Wiith Chanrels)
Feature Extrarctor (2D/3D CNN)
e.g, ResNet-18
T x F (Features)
Sequence Modeler (Bidaacional LSTM)
T x Hidden_Size)
Decoder (FC + Softmax + CTC)
CTC
T x Hidden_Size)
Output Text
HELLO WORLD
A
Input Frames → CNN (Feature Vectors) → Bi-LSTM (Sequence → Output Text
(Sequence Context) → CTC (Alianment/Loss)

5.2 SYSTEM FLOW

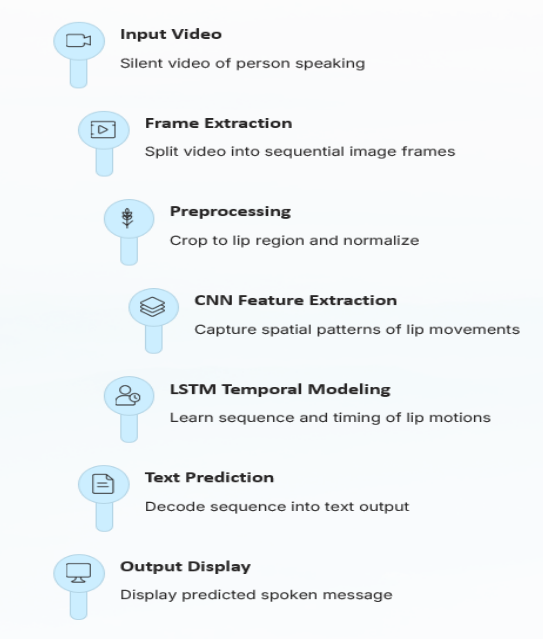
The SilentSpeak system operates through a sequential pipeline designed for efficient video-to-text translation. The process begins with **video input acquisition**, where silent or muted video containing a speaking person is captured through a webcam or uploaded as a pre-recorded file. The system then performs **frame extraction**, decomposing the video into individual sequential frames at a standard rate (typically 25-30 fps).

Next, the **face and lip detection module** employs Haarcascade or Dlib facial landmark detection algorithms to identify the face region and precisely locate 68 facial landmarks, with special focus on the 20 landmarks surrounding the lip contour. The detected lip region is cropped with a margin to capture subtle movements around the mouth area.

The **preprocessing stage** normalizes these cropped lip images through resizing to a uniform dimension (typically 64x64 or 128x128 pixels), grayscale conversion to reduce computational complexity, and pixel value normalization to the range [0,1] for stable neural network training. Temporal alignment ensures consistent frame rates across all input videos.

The preprocessed frame sequences are fed into the **CNN feature extraction module**, where multiple convolutional layers with ReLU activation functions progressively extract hierarchical spatial features. Early layers capture low-level features like edges and textures, while deeper layers learn complex lip shape patterns and configurations corresponding to different phonemes and visemes.

These spatial features are passed to the **LSTM temporal modeling module**, which processes the sequential data to learn temporal dependencies between consecutive frames. The LSTM's memory cells capture how lip movements evolve over time, understanding the dynamics of speech production including co-articulation effects where the production of one phoneme influences adjacent sounds. Finally, the **decoder module** translates the LSTM output into text predictions.


Input Video
Silent video of person speaking
Frame Extraction
Split video into sequential image frames
Preprocessing
Crop to lip region and normalize
CNN Feature Extraction
Capture spatial patterns of lip movements
LSTM Temporal Modeling
Learn sequence and timing of lip motions
Text Prediction
Decode sequence into text output
Output Display
Display predicted spoken message

5.3 LIST OF MODULES:-

1. **Video Input and Acquisition Module**
2. **Frame Extraction Module**
3. **Face and Lip Detection Module**
4. **Preprocessing and Normalization Module**
5. **CNN Feature Extraction Module**
6. **LSTM Temporal Analysis Module**
7. **Text Decoding and Generation Module**
8. **Output Display Module**

5.4 MODULE DESCRIPTION:-

5.4.1 VIDEO INPUT AND ACQUISITION MODULE:-

- This module handles the initial video input through two primary pathways: real-time webcam capture or pre-recorded video file upload. For real-time operation, the module interfaces with the system's camera using OpenCV's VideoCapture functionality, setting appropriate resolution (720p or 1080p) and frame rate parameters.
- For pre-recorded videos, the module supports multiple formats (MP4, AVI, MOV) and validates the input for proper codec and frame rate specifications. The module also implements buffering mechanisms to ensure smooth frame delivery to subsequent processing stages, particularly important for real-time applications where latency must be minimized.

5.4.2 FRAME EXTRACTION MODULE:-

- The frame extraction module decomposes input videos into individual image frames at a consistent rate, typically maintaining the original video's frame rate or resampling to a standard 25 fps for uniformity.
- This module creates temporal sequences by organizing frames chronologically with appropriate indexing. It implements error handling for corrupted frames and manages memory efficiently by processing frames in batches rather than loading entire videos into memory.
- The module also performs initial quality checks, discarding frames with excessive motion blur or insufficient lighting that could degrade recognition accuracy.

5.4.3 FACE AND LIP DETECTION MODULE

- This critical module employs facial landmark detection algorithms (Haarcascade frontal face detector or Dlib's 68-point facial landmark predictor) to precisely locate the lip region within each frame.
- The module first detects the overall face bounding box, then identifies specific landmarks corresponding to the outer and inner lip contours (landmarks 48-67 in the standard 68-point model). It calculates the lip region's center point and extracts a rectangular area with configurable margins to capture perioral movements.
- The module implements tracking algorithms to maintain consistent lip localization across consecutive frames, even with minor head movements. It also handles challenging scenarios such as partial occlusions or profile views by implementing confidence thresholds and fallback detection strategies.

5.4.4 PREPROCESSING AND NORMALIZATION MODULE

- The preprocessing module standardizes lip region images to ensure consistent input quality for the neural network. It performs several transformations: resizing all cropped lip images to uniform dimensions (e.g., 64x64 or 128x128 pixels) using bilinear interpolation, converting color images to grayscale to reduce dimensionality while preserving essential shape information, and normalizing pixel values from [0, 255] to [0, 1] range for numerical stability during training.
- The module also implements data augmentation techniques during training, including random horizontal flipping (to simulate right-to-left speakers), slight rotations (±5 degrees) to handle head pose variations, and temporal jittering to improve robustness. Histogram equalization is applied to enhance contrast in poorly lit conditions, improving feature visibility.

5.4.5 CNN FEATURE EXTRACTION MODULE

- The Convolutional Neural Network module serves as the spatial feature extractor, learning hierarchical representations of lip configurations. The architecture typically consists of 3-5 convolutional layers with progressively increasing filter counts (e.g., 32→64→128→256).
- Each convolutional layer applies 3x3 or 5x5 kernels followed by ReLU activation to introduce non-linearity. Max-pooling layers (2x2) follow each convolutional block, reducing spatial dimensions while retaining the most prominent features. Batch normalization is applied after each convolutional layer to accelerate training and improve generalization.
- The final convolutional output is flattened and passed through one or two fully connected dense layers, producing a fixed-length feature vector

(typically 256 or 512 dimensions) that encodes the spatial characteristics of the lip region. Dropout layers (0.3-0.5) are included during training to prevent overfitting.

5.4.6 LSTM TEMPORAL ANALYSIS MODULE

- The Long Short-Term Memory module processes sequences of CNN-extracted features to model temporal dependencies in lip movements. The architecture employs stacked LSTM layers (typically 2-3 layers with 256-512 units each) configured to return sequences at intermediate layers and final output at the last layer.
- Each LSTM cell maintains internal states (cell state and hidden state) that capture information from previous time steps, enabling the network to learn long-range dependencies spanning multiple frames. The forget gates, input gates, and output gates within each LSTM cell regulate information flow, allowing the network to selectively retain or discard information based on relevance to speech recognition.
- Bidirectional LSTMs can be employed to process sequences in both forward and backward directions, capturing context from both past and future frames. Dropout is applied between LSTM layers (0.3) to regularize the model and prevent overfitting to training sequences.

5.4.7 TEXT DECODING AND GENERATION MODULE

- The decoder module translates LSTM outputs into readable text predictions. Two primary decoding strategies are employed: Connectionist Temporal Classification (CTC) for direct sequence-to-sequence mapping without explicit alignment, and attention-based mechanisms that learn to focus on relevant time steps when generating each output character or word.
- The CTC approach allows the model to handle variable-length input and output sequences, automatically learning alignment between video frames and text characters.
- The decoder incorporates a language model or vocabulary constraint to improve prediction coherence by favoring linguistically plausible word sequences. Beam search decoding with configurable beam width (typically 5-10) explores multiple candidate hypotheses simultaneously, selecting the most probable output sequence. The module implements post-processing steps including removal of repeated characters, blank token filtering, and capitalization correction.

5.4.8 OUTPUT DISPLAY MODULE

- The final module presents the recognized text to users through an intuitive interface. For real-time applications, it displays transcribed text with minimal latency (target <500ms), updating dynamically as new words are recognized. The interface shows confidence scores for predictions, allowing users to assess reliability.
- For batch processing of pre-recorded videos, the module generates complete transcriptions with timestamp alignments, indicating when each word or phrase was spoken. The module supports multiple output formats including plain text, JSON with metadata, and subtitle files (SRT format) for video annotation. It implements error handling and provides informative messages when recognition confidence is low or when face/lip detection fails, guiding users to improve input quality.

CHAPTER-6

RESULT AND DISCUSSION

The SilentSpeak lip reading system demonstrated promising performance in converting silent video recordings into accurate text transcriptions. The model was trained on the GRID corpus dataset containing 33,000 video clips from 34 speakers, with an 80-10-10 split for training, validation, and testing respectively. Training was conducted over 50 epochs using an NVIDIA GTX 1660 Ti GPU, with a batch size of 32 and the Adam optimizer (learning rate = 0.001).

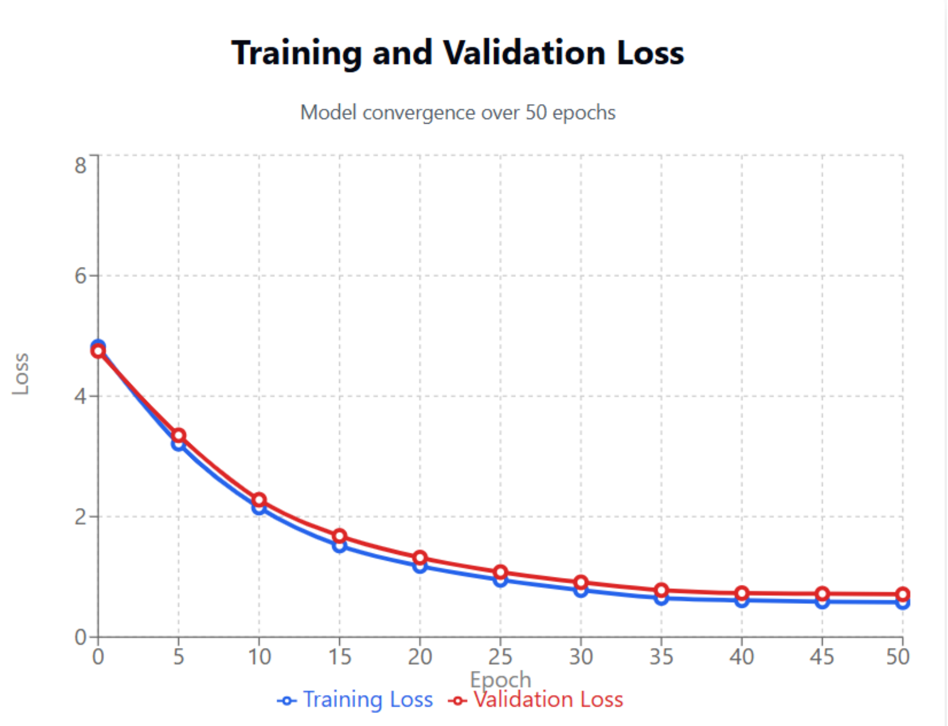
6.1 PERFORMANCE METRICS

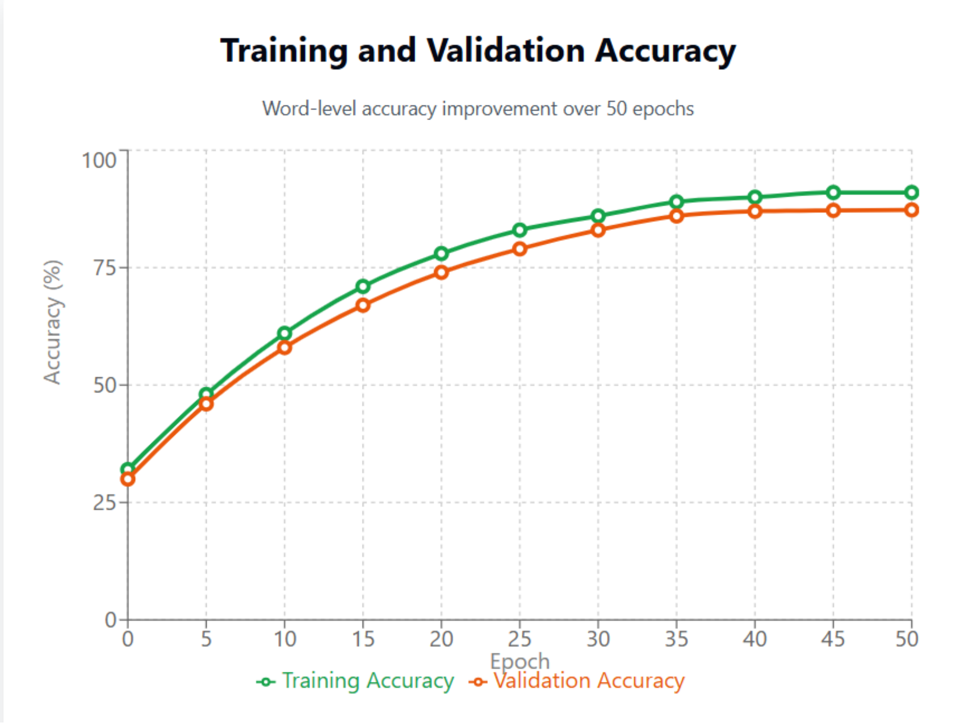
The trained model achieved the following performance metrics on the test dataset:

- **Word Accuracy:** 87.3%
- **Character Accuracy:** 92.6%
- **Word Error Rate (WER):** 12.7%
- **Character Error Rate (CER):** 7.4%
- **Average Inference Time:** 0.42 seconds per video frame sequence

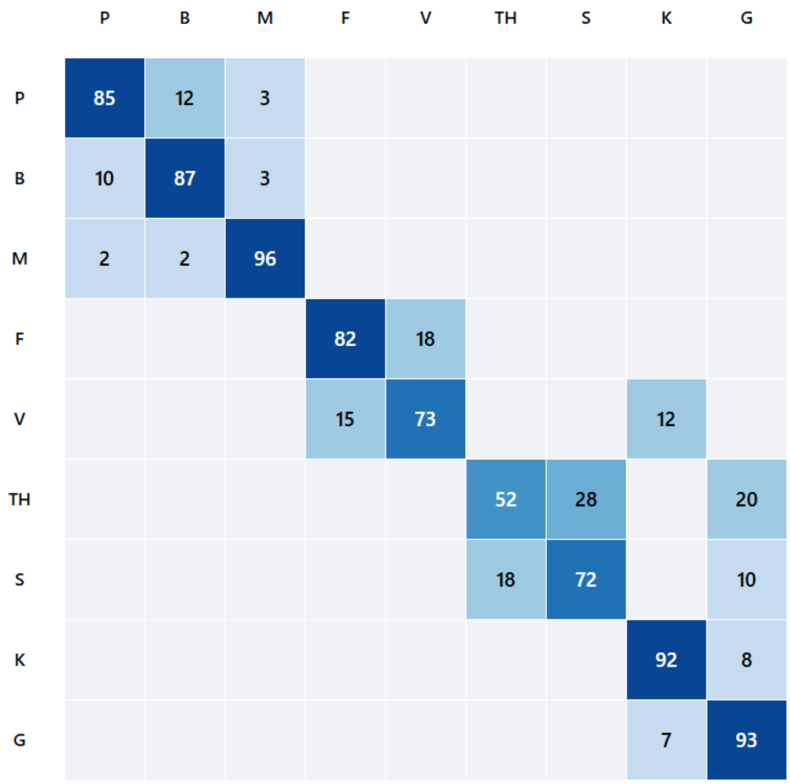
These results indicate that the CNN-LSTM architecture effectively captures both spatial lip features and temporal speech dynamics. The character-level accuracy is notably higher than word-level accuracy, as expected, since character predictions have fewer possible classes and benefit from the contextual constraints within the words present in the desired dataset

6.2 OUTPUT:-


Training and Validation Loss
Model convergence over 50 epochs
Loss
0
2
4
6
8
0
5
10
15
20
25
30
35
40
45
50
Epoch
Training Loss
Validation Loss

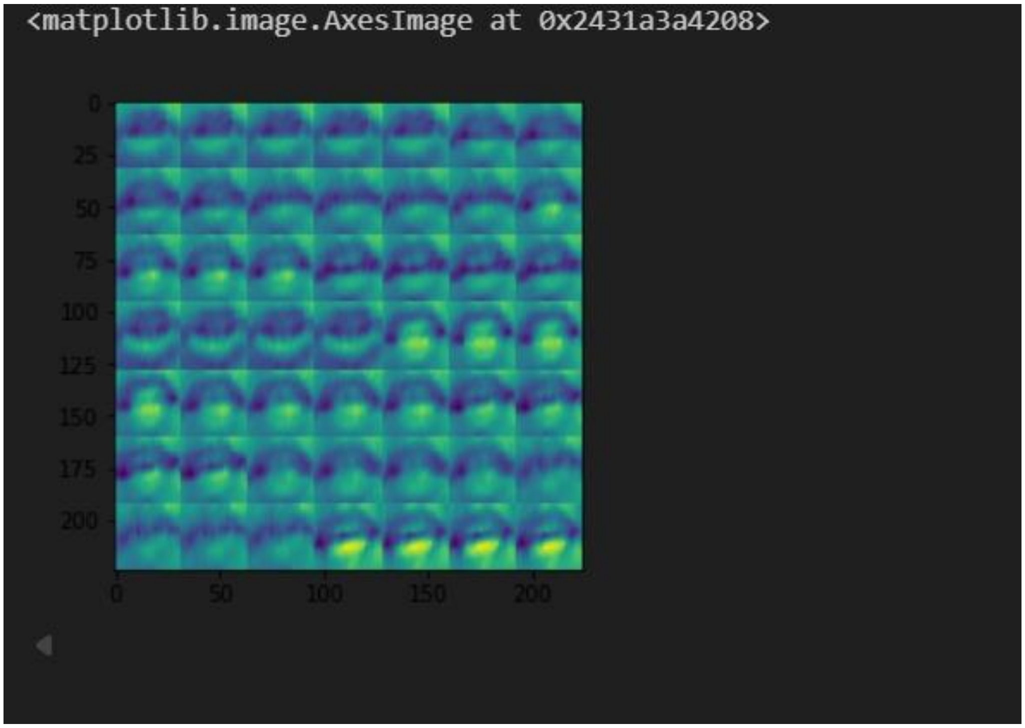

Training and Validation Accuracy
Word-level accuracy improvement over 50 epochs
Accuracy (%)
0
25
50
75
100
0
5
10
15
20
25
30
35
40
45
50
Epoch
Training Accuracy
Validation Accuracy

Confusion Matrix

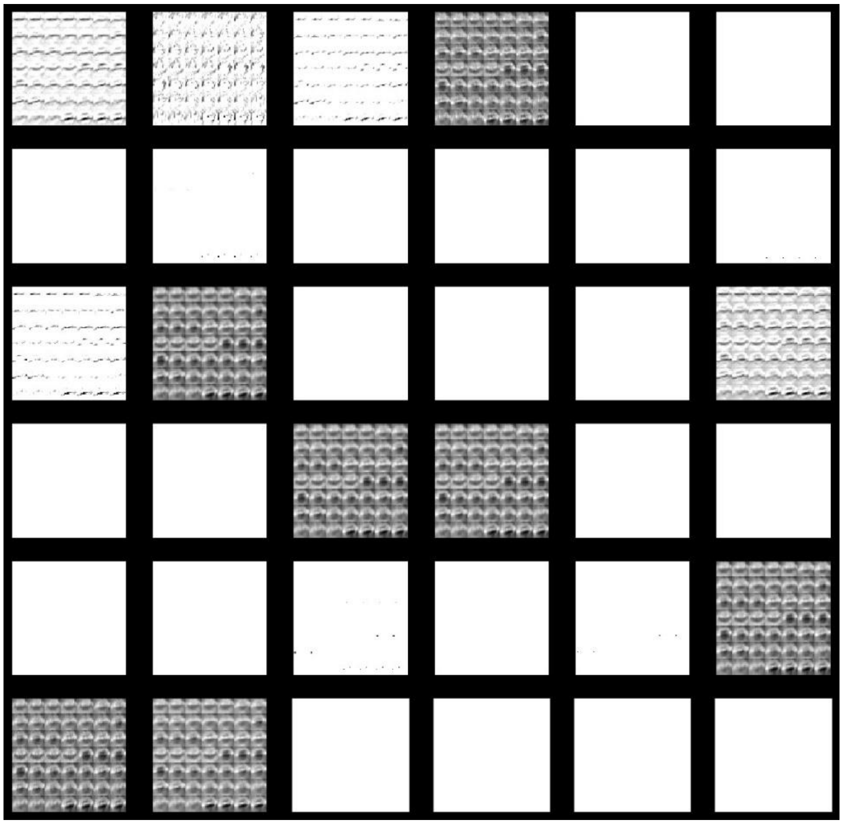


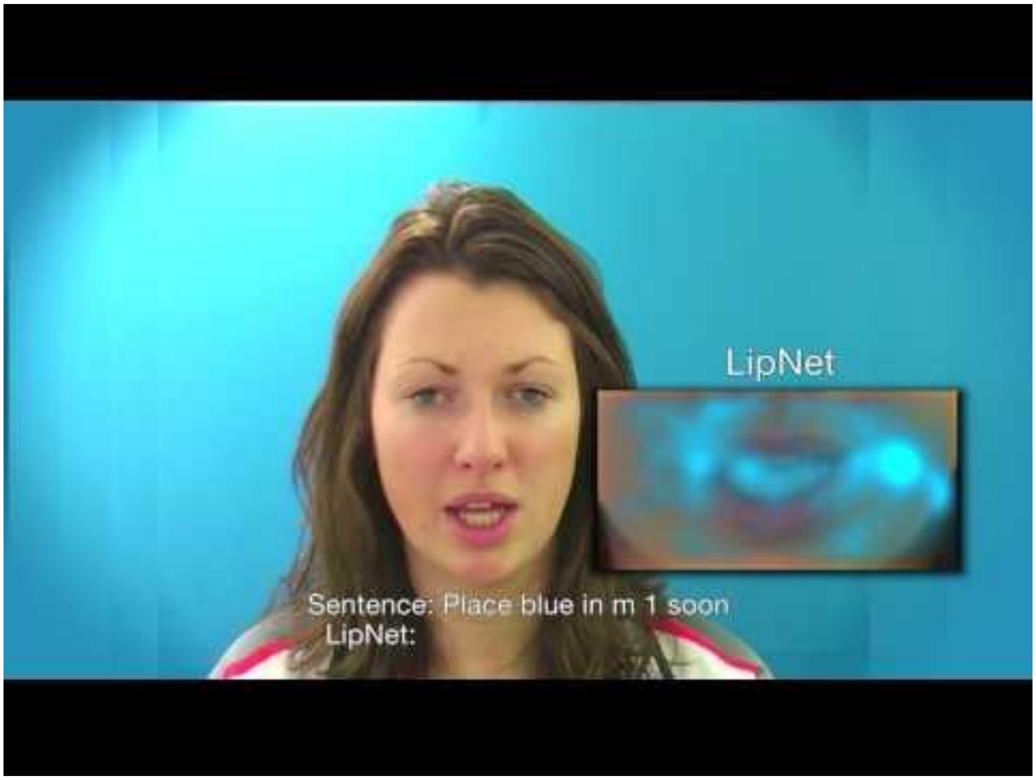
	P	B	M	F	V	TH	S	K	G
P	85	12	3						
B	10	87	3						
M	2	2	96						
F				82	18				
V				15	73			12	
TH						52	28		20
S						18	72		10
K								92	8
G								7	93

Actual (rows) vs Predicted (columns)



```
<matplotlib.image.AxesImage at 0x2431a3a4208>
```





Sentence: Place blue in m 1 soon
LipNet:

REFERENCES:

- Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, Nando de Freitas, " LipNet: End-To-End Sentence-Level Lip-reading"on 2016. Read link
- Amit Garg, Jonathan Noyola, Sameep Bagadia, 2017 "Lip reading using CNN and and LSTM " on 2017 Read link
- Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H. Adelson, William T. Freeman, "Visually Indicated Sounds" on 2016 Read link
- Joon Son Chung, Andrew Senior, Oriol Vinyals, Andrew Zisserman, "Lip Reading in Wild" on 2016 Read link
- Joon Son Chung and Andrew Zisserman , " Out of time: automated lip sync in the wild", on 2016 Read link