Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

LipSyncLite

An Efficient Lip-movement-based Visual Speech Recognition

Dr. Gudavalli Madhavi
Associate Professor and HOD,
Department of Computer Science Engineering
University College of Engineering Narasaraopet, JNTUK
Narasaraopet, Andhra Pradesh, India

Devarapalli Vamsi, Reddy Veeraiah
Students
Department of Computer Science Engineering
University College of Engineering Narasaraopet, JNTUK
Narasaraopet, Andhra Pradesh, India

***Abstract* - Visual Speech Recognition (VSR), usually called as lip reading, focuses on understanding spoken language by analyzing visual information from speaker's lip movements, without depending on audio signals. This approach is particularly beneficial in environments where audio data is unavailable, corrupted by noise, or restricted due to privacy concerns. This research proposes a lightweight and modular deep learning framework for lip-movement-based speech recognition that emphasizes both accuracy and computational efficiency. The proposed system utilizes a Three-Dimensional Convolutional Neural Network (3D-CNN) to capture spatiotemporal features from sequences of grayscale mouth-region frames, effectively modeling both spatial lip structures and temporal motion patterns. To enhance contextual understanding, a Transformer- based sequence modeling component is employed, enabling the system to capture relationships across distant video frames. The model employs Connectionist Temporal Classification (CTC) loss, which allows alignment-free learning and eliminates the need for frame-level annotations. This significantly reduces data preparation complexity while supporting flexible prediction of variable-length text sequences. The system is evaluated using the GRID audiovisual dataset, and its performance is measured through commonly used evaluation metrics like Word Error Rate (WER) and Character Error Rate (CER), ensuring a comprehensive evaluation at both word and character levels. The proposed framework offers a balance between performance and computational efficiency, making it well-suited for real-time applications and deployment on resource-constrained devices. This work contributes toward the development of practical, scalable, and deployable visual speech recognition systems for use in areas like assistive communication, security, and human-computer interaction.**

Keywords* - *Visual Speech Recognition, Deep Learning, Transformer architecture, Connectionist Temporal Classification, 3D Convolutional Neural network

I. INTRODUCTION

Automatic Speech Recognition (ASR) systems have become a fundamental component of modern human–computer interaction, enabling machines to convert spoken language into text. These systems primarily rely on acoustic signals and perform well under controlled conditions. However, their effectiveness significantly degrades in challenging environments such as high background noise, poor audio quality, or situations where microphones cannot be used due to privacy or security constraints. These limitations highlight the need for alternative approaches to speech recognition that do not depend solely on audio input.

Although speech can be understood through audio alone, visual cues from lip movements significantly enhance perception, especially under challenging conditions. The McGurk effect illustrates the strong interaction between auditory and visual modalities in human speech perception [1], [2].

VSR addresses this challenge by interpreting speech through visual information, particularly the movements of the lips and surrounding facial regions. Since visual cues are unaffected by acoustic disturbances, VSR offers an effective approach for recognizing speech in noisy or silent environments.

Despite its advantages, lip reading is a complex task due to several inherent challenges. One of the primary challenge is the visual ambiguity of speech, where multiple phonemes can map to similar lip movements, known as visemes. The example for many-to-one mapping is shown in Figure 1.

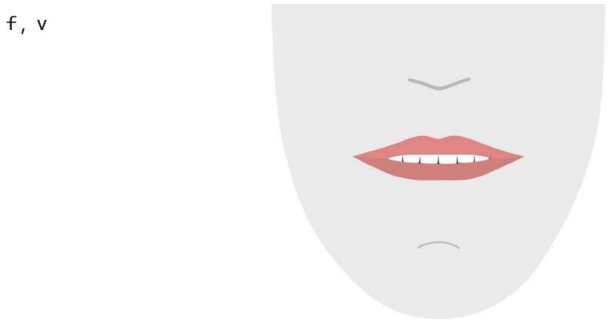

Figure 1: Example of viseme ambiguity in lip reading

Additionally, variations in lighting conditions, head pose, facial expressions, and speaker characteristics can further complicate accurate recognition. These factors make it difficult to extract reliable and discriminative features from visual data. And due to these humans can only achieve recognition rate around 35%.

Recent advancements in deep learning have significantly improved the performance of VSR systems. Convolutional Neural Networks (CNNs) are effective in extracting spatial features from images, while sequence modeling techniques such as Recurrent Neural Networks (RNNs) and Transformers

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

capture temporal dependencies in video sequences. In particular, Three-Dimensional Convolutional Neural Networks (3D-CNNs) enable joint learning of spatial and temporal features, making them much suitable for lip-reading tasks. Furthermore, Connectionist Temporal Classification (CTC) has enabled end-to-end training without requiring precise alignment between video frames and text labels.

In this research, a computationally efficient and modular visual speech recognition system is proposed. The system employs a 3D-CNN for spatiotemporal feature extraction, followed by a Transformer-based architecture for sequence modeling, and utilizes CTC loss for alignment-free decoding. The design focuses on achieving both recognition accuracy and computational efficiency.

The proposed approach aims to contribute to the development of practical and deployable lip-reading systems by addressing the limitations of existing complex architectures while maintaining reliable performance.

II. RELATED WORK

Early approaches to lip reading relied on traditional feature extraction techniques such as Principal Component Analysis (PCA), Discrete Cosine Transform (DCT), Active Appearance Models (AAM), and Hidden Markov Models (HMM) [3], [4]. While these methods were computationally efficient and interpretable, they struggled to capture complex spatiotemporal patterns in lip movements, resulting in limited performance in real-world scenarios.

Convolutional Neural Networks (CNNs) became widely adopted for extracting spatial features from images. Models combining CNNs with Recurrent Neural Networks like Long Short-Term Memory and Gated Recurrent Units, improved lip-reading performance by modeling temporal dependencies across video frames [5]–[7]. These hybrid architectures enabled end-to-end learning but often required large datasets and high computational resources. Wand et al. (2016) [8] demonstrated that LSTM-only networks can perform lipreading directly from raw mouth images without relying on CNNs.

LipNet [6] integrates spatiotemporal CNNs with Bidirectional GRUs and CTC loss [9]. This approach demonstrated strong performance on constrained datasets by directly mapping video sequences to text without requiring frame-level alignment. However, its generalization to unconstrained environments remains limited.

Graph-based methods, such as Adaptive Semantic-Spatio-Temporal Graph Convolutional Networks (ASST-GCN), have been proposed to model relationships between facial landmarks. These models capture fine-grained structural and temporal information of lip movements, achieving high accuracy. Nevertheless, they depend heavily on precise landmark detection and involve complex graph construction [10].

Recent research has also explored hybrid approaches combining 3D Convolutional Neural Networks (3D-CNNs) with efficient backbone networks, like MobileNet or EfficientNet, have shown promising results. Additionally, methods utilizing Connectionist Temporal Classification (CTC) enable alignment-free training, simplifying data annotation and improving scalability [11].

Several challenges still remain such as high computational complexity, dependence on large datasets, and limited robustness in real-world conditions. Many existing models are hard to deploy on resource-constrained devices.

Transformer-based architectures have further advanced the field by effectively capturing long-range temporal dependencies using self-attention mechanisms. Models such as video Transformers and multimodal frameworks have shown improved performance in both audio-visual and visual-only speech recognition systems. Recently, Thomas et al. (2025) proposed a phoneme-centric two-stage V-ASR framework that uses a Video Transformer with a CTC head to predict phonemes from visual input and then a fine-tuned large language model to reconstruct words and sentences, achieving promising results on LRS2 and LRS3 with far less labelled data [12].

The proposed work focuses on designing a lightweight, modular, and efficient VSR framework. By leveraging 3D-CNNs for spatiotemporal feature extraction, Transformer- based sequence modeling, and CTC for alignment-free decoding, the system aims to achieve a balance between accuracy, simplicity, and deployability.

III. METHODOLOGY

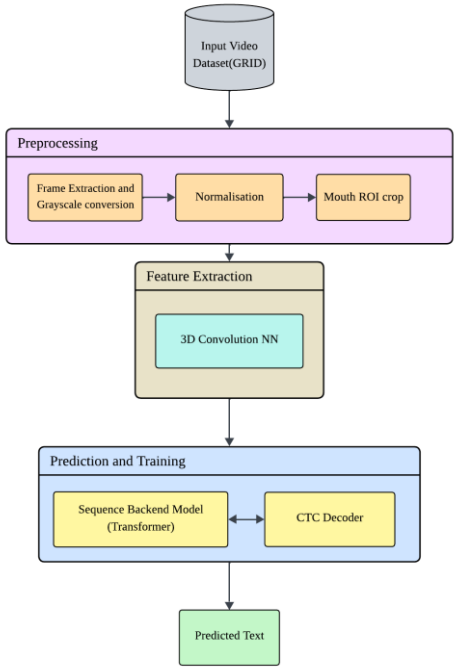

Figure 2 : Overview of the Lip Reading Pipeline.

This section dives deep into the methodological framework used in the development of LipSyncLite. This includes dataset selection, data preprocessing, region-of-interest extraction, feature extraction, and model architecture as depicted in the pipeline diagram(Figure 2).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

A. Dataset

The quality and structure of dataset is very crucial for a VSR system. In this work, the GRID (Graphics Research International Database) Audiovisual Sentence Corpus is utilized as the primary dataset for training and evaluation [13].

The dataset consists of recordings from multiple speakers with controlled environmental conditions, ensuring consistency in lighting, background, and camera positioning. It includes approximately 34 speakers, each contributing around 1,000 utterances, resulting in nearly 33,000 utterances (excluding a speaker who had only voice data). Each video contains a short sentence with approximate duration of 3 seconds, recorded at 25 fps, yielding about 75 frames per sample.

The sentences follow a fixed grammatical structure:

```
"<command> <colour> <preposition>
    <letter><digit> <adverb>"
```

(e.g., “set white in o zero now”). This structured format ensures limited vocabulary complexity (51 unique words), making it apt for benchmarking VSR models. The vocabulary was defined with 27 unique categories, comprising the 26 lowercase English letters (a–z) along with an additional class representing a space. Additionally, the dataset provides corresponding text transcripts and alignment files, which are beneficial for training models using alignment-free methods such as CTC.

The dataset is preprocessed to select a subset of 5,000 samples for efficient experimentation while maintaining diversity across speakers.

B. Data Preprocessing

Preprocessing plays an important role in converting raw video data into a consistent format that can be effectively used by deep learning models The preprocessing pipeline ensures consistency in spatial and temporal dimensions while reducing noise and redundancy. Our preprocessing pipeline involved frame extraction, normalization, and sequence alignment.

1) Frame Extraction and Resizing: Videos were processed to extract a fixed sequence of 75 frames, ensuring a consistent temporal input length for the model. If a video has fewer than 75 frames, zero values frames are appended; if longer, it was truncated by uniform sampling.
2) Grayscale Conversion: To minimize the influence of lighting variations, computational complexity, and reduce data dimensionality, all video frames were converted from RGB to grayscale. This ensures the model focuses on the shape and motion of the lips rather than skin tone and lighting artifacts
3) Normalization: Normalization is performed by dividing each pixel intensity with 255.0 resulting values in range $[0, 1]$. This accelerates gradient descent convergence during training by maintaining consistent input distributions.
4) Alignment Processing: The accompanying `.align` files were parsed to generate training labels. Silence markers (‘sil’) and non-speech annotations were excluded. The remaining characters were mapped to integer indices using a `StringLookup` layer, creating a sequence of labels with a maximum length of 40 characters per utterence

These preprocessing steps convert raw videos into structured tensors of consistent size, enabling efficient batch processing (Batch size = 32) and model training.

C. ROI Crop (Region of Interest Extraction)

To concentrate on the most relevant visual features, the system isolates the Region of Interest (ROI) representing the mouth area from each frame of the video. This step eliminates irrelevant background details (e.g., hair, clothing, or background scenery) and improves model performance. The ROI extraction process involves:

- Face Detection and Landmark Localization: We used the `face_recognition` library (built upon dlib [14]) to detect facial landmarks. Specifically, we utilized the 68- landmark predictor to identify key points such as the eyes and nose tip.
- Mouth Region Localization: Rather than using a static bounding box, we implemented a dynamic cropping algorithm. The algorithm calculates the inter-ocular distance to estimate the facial scale. The mouth region is approximated below the nose tip with a crop size determined as a function of the eye distance (multiplied by a factor of 1.35). This approach ensures the crop adapts to different face sizes and camera distances.
- Fixed standard: The extracted mouth region is resized to a fixed resolution of 140 x 46 pixels. This specific aspect ratio preserves the width of the mouth during speech while minimizing the inclusion of the chin and nose bridge. The resulting input tensor for the model has a shape of $(75, 46, 140, 1)$.

By isolating the mouth region, the system reduces noise, lowers computational cost, and enhances the learning of discriminative lip movement features.

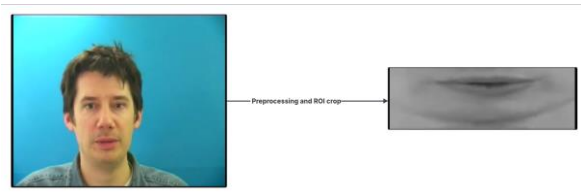

Figure 3 Preprocessing and ROI Extraction

D. Feature Extraction

In the proposed system, the feature extraction phase is pivotal for accurately interpreting lip movements from video frames. The model employs a 3D Convolutional

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Neural Network (3D-CNN) layers to extract spatial and temporal features from the input data.

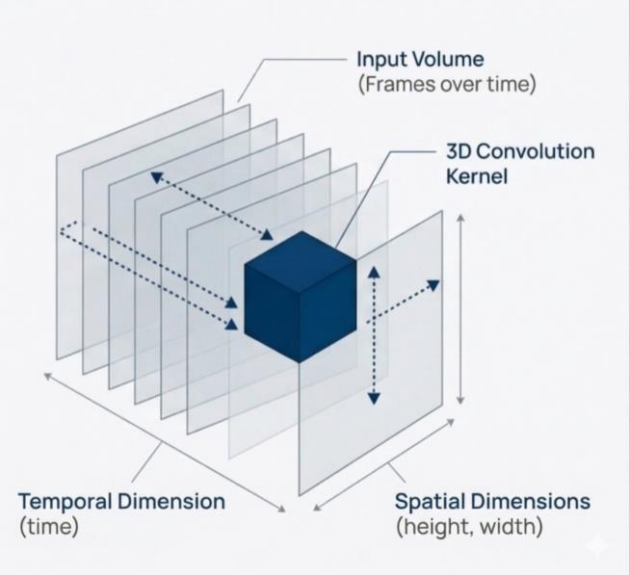

Figure 4 : 3D-CNN block diagrams

1) Sequence of video frames(75 frames, each of size 46 x 140 pixels in grayscale) is given as input. The initial layers employ 3D convolutions to process this volumetric data, which captures the motion dynamics across the frame sequence. The 3D-CNN frontend consists of three convolutional blocks with filter counts of 32, 64, and 96, respectively. Within each block, convolutional layers are applied first, followed by batch normalization and max-pooling to refine and downsample the features. These layers extract spatiotemporal features by sliding 3D kernels over the input volume, thereby capturing the temporal changes in lip shape and spatial patterns simultaneously. Specifically, the max-pooling layers reduce the spatial dimensions progressively (from 46 x 140 down to 5 x 17) while preserving the temporal length (75 frames), forcing the network to learn compact, abstract visual features.
2) The output from the 3D-CNN layers is reshaped into a series of feature vectors for further temporal processing. This involves using a `TimeDistributed` wrapper to apply operations independently to each time step. A `GlobalAveragePooling2D` layer is applied to the spatial dimensions (5 x 17) of each frame, collapsing them into a single 96-dimensional vector per frame. This step effectively removes the spatial height and width, leaving only the temporal sequence. Following this, a `TimeDistributed Dense` layer projects these 96-dimensional vectors into a 256-dimensional embedding space. This transformation prepares the features for the subsequent temporal backend, aligning the vector dimension with the model's internal feature configuration.

E. Model Architecture

The model is built for lip reading from video frame sequences, featuring a hybrid architecture that combines 3D Convolutional Neural Networks (3D-CNN) to capture spatiotemporal features with a Transformer Encoder for modeling temporal dependencies. It takes as input a sequence of 75 grayscale frames, each sized 46 x 140 pixels, and processes them through multiple network layers.

1) Spatiotemporal Feature Extraction: The initial layers of the model use 3D convolutions to process volumetric input data. This is particularly useful for tasks involving temporal sequences where capturing relationships across height, width, and time is crucial. The frontend consists of three Conv3D blocks with 32, 64, and 96 filters, respectively. Within each block, batch normalization and ReLU activation are applied, then MaxPool3D is used to reduce spatial dimensions while maintaining the temporal length.
2) Feature Aggregation and Projection: Unlike traditional approaches that flatten features or use heavy backbones, this model employs `TimeDistributed_GlobalAveragePooling2D`. This operation averages the spatial dimensions of each frame independently, collapsing the spatial grid (5 x 17) into a single 96-dimensional vector per time step. A `TimeDistributed Dense` layer then projects these vectors into a 256-dimensional feature space, preparing the sequence for the Transformer backend.
3) Transformer Encoder for Sequence Modeling: The core backend utilizes a Transformer Encoder architecture rather than recurrent layers (LSTMs). This choice allows the model to capture long-range temporal dependencies using self-attention mechanisms. Positional embeddings are added to the input sequence to retain temporal order information. The encoder consists of two blocks, each comprising Multi-Head Attention and Feed Forward Networks (FFN), enabling the model to weigh the importance of specific lip movements across the entire sequence context.
4) Classification and CTC Loss: The Transformer-processed features are projected onto the vocabulary space using a final Dense layer. To handle word transcripts that do not align directly with video frames, we employ a Connectionist Temporal Classification (CTC) loss. This method addresses alignment challenges and prevents repeated predictions by summing up all possible correspondences between input frames and target text [9]

1) 3D CONVOLUTIONAL NETWORKS:

CNNs stand at the forefront of performing convolutional operations on visual data, which is crucial for advancing computer vision tasks. In particular, object recognition tasks utilize 2D Convolutional Layers (conv2d) to process the image's channel *Z*. Extending this framework, a conv3d incorporates an additional dimension into the convolution process, accommodating temporal dynamics or depth information. The 3D convolution is defined by:

$$Conv3d(u,v)_{s,t,u} = \sum_{z=1}^{Z}\sum_{a=1}^{h_v}\sum_{b=1}^{w_v}\sum_{c=1}^{d_v} v_{z,a,b,c} \cdot u_{z,s+a,t+b,u+c}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

In this equation, the introduction of `u` as a new dimension allows the kernel to slide across time, capturing motion features essential for lip-reading.

2) TRANSFORMER ENCODER:

The Transformer architecture excels in sequence modeling by using self-attention mechanisms to capture long-range dependencies without the sequential constraints of RNNs. At the core of the Transformer Encoder is the Multi-Head Attention mechanism, which enables the model to simultaneously focus on information from multiple representation subspaces across different positions. The scaled dot-product attention is defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are the query, key, and value matrices derived from the input sequence [15]. Each Transformer Encoder block in this model comprises Multi-Head Attention, followed by a Position-wise Feed-Forward Network (FFN), residual connections, and layer normalization. The FFN performs two linear transformations with a ReLU activation in between:

$$FFN(x) = \max(0, xW_1 + b_1)\,W_2 + b_2$$

This design allows the model to dynamically assign importance to specific lip shapes across the full 75-frame sequence, rather than processing the frames strictly in order.

3) CONNECTIONIST TEMPORAL CLASSIFICATION:

Connectionist Temporal Classification (CTC) is commonly employed in lipreading because it models temporal sequences without requiring explicit frame-level alignment. At each timestep, a softmax layer converts the network outputs into a probability distribution over possible labels, including a special blank token to account for unaligned frames

Let the input sequence be x = (x1, x2, . . . , xT) and the target label sequence be y = (y1, y2, . . . , yU), with U ≤ T .

CTC computes the probability of y given x by summing over all possible alignment paths π that collapse to y:

$$P(y|x) = \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi|x)$$

Here, B is the collapse function that removes repeated labels and blank tokens. Assuming conditional independence across timesteps, the probability of a particular alignment path π is:

$$P(\pi|x) = \prod_{t=1}^{T} P(\pi_t|x)$$

The CTC loss is defined as the negative log-likelihood of the target sequence:

$$\mathcal{L}_{CTC} = -\log P(y|x)$$

This approach allows the model to learn monotonic alignments and handle variable-length input-output mappings, making it particularly effective for lipreading tasks where the exact timing of spoken words is unknown [9].

TABLE I: MODEL STRUCTURE

Layer (type)	Output Shape	Param #
input_layer	(None, 75, 46, 140, 1)	0
conv3d	(None, 75, 46, 140, 32)	2,432
batch_normalization	(None, 75, 46, 140, 32)	128
activation	(None, 75, 46, 140, 32)	0
max_pooling3d	(None, 75, 23, 70, 32)	0
conv3d_1	(None, 75, 23, 70, 64)	55,360
batch_normalization_1	(None, 75, 23, 70, 64)	256
activation_1	(None, 75, 23, 70, 64)	0
max_pooling3d_1	(None, 75, 11, 35, 64)	0
conv3d_2	(None, 75, 11, 35, 96)	165,984
batch_normalization_2	(None, 75, 11, 35, 96)	384
activation_2	(None, 75, 11, 35, 96)	0
max_pooling3d_2	(None, 75, 5, 17, 96)	0
time_distributed	(None, 75, 96)	0
time_distributed_1	(None, 75, 256)	24,832
time_distributed_2	(None, 75, 256)	0
transformer_encoder_1	(None, 75, 256)	527,104
transformer_encoder_2	(None, 75, 256)	527,104
output_dense	(None, 75, 29)	7,453

Total params: 1,311,037 (5.00 MB)
Trainable params: 1,310,653 (5.00 MB)
Non-trainable params: 384 (1.50 KB)

F. Training Time Analysis

The model is trained on an L4 GPU in Google Colab, and the training time for each epoch was monitored.

TABLE II: EPOCH TIME ANALYSIS

Greedy Decoding		Beam Decoding	
Epoch	Time per Epoch	Epoch	Time per Epoch
1	237s	1	116s
2	116s	2	116s
3	116s	3	116s
4	116s	4	116s
5	115s	5	116s
.	.	.	.
90	115s	10	115s

The average time for an epoch is 117 seconds, and the total time for 100 epochs is 11,650 seconds.

$$\frac{11{,}650}{3600} = 3.23 \text{ hours}$$

IV. EXPERIMENT RESULTS AND ANALYSIS

A. Implementation Details

The ROI (Lip region) is extracted using dlib's face_recognition library. They are resized to a consistent 140 × 46 pixels and saved in Google Drive and loaded to system RAM to accelerate the training process. The first 10 frames extracted for a sample video are shown in the figure.

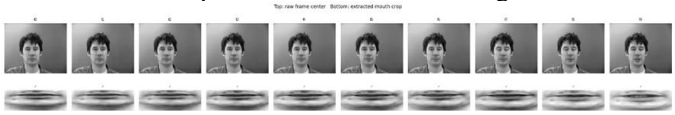

Figure 5 : First 10 frames extracted for a sample video.

The model uses 3D CNNs for feature extraction, followed by a Transformer encoder for sequence modeling, and a final dense layer to get the probabilities of each character in the vocabulary. The model was trained using the Adam optimizer with an initial learning rate of 0.0001. If the validation loss does not decrease

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

for 5 epochs, the learning rate is halved. TensorFlow's native CTC greedy search is used for 90 epochs, and beam search with a beam width of 10 is used to decode the model's predictions into actual text.

Custom callbacks are used to produce example predictions, print them at the end of each epoch, save the training metrics, and monitor the
ps in understanding the overfitting and underfitting of the model. The model weights are saved whenever the epoch's validation loss decreases, for resumption if the cloud runtime disconnects.

B. Baselines

Xu et al. introduced a reference approach by training a Cascaded Attention-CTC model on the GRID dataset. Margam et al. proposed a baseline method that combines 3D and standard 2D convolutional neural networks on the GRID dataset. Despite its effectiveness, this approach has certain limitations.

A more recent baseline is LipSyncNet, introduced by Jeevakumari and Dey[11], which employs a 3D convolutional neural network enhanced through EfficientNetB0 to improve feature representation. The features are then processed by a BLSTM network combined with a CTC mechanism.

A summary of the experimental results from these baseline techniques is provided in Table.

TABLE III: EVALUATION OUTCOMES OF BASELINES FOR A VIDEO (TRUE: "BIN RED AT N NINE AGAIN")

Sl.	Predicted	Method Used	Dataset
a.	bin red at n nime again	LipNet + Wave2Lip	GRID
b.	bin red at n nine again	LipSyncNet	GRID
c.	bim red at nine again	Adapted LipNet	GRID

C. Training Setup And Analysis

The model training and testing used a uniformly selected subset of 5,000 videos from the GRID corpus. It is divided into train, test, and validation sets. The test set containing 500 videos is immediately saved to disk to avoid data leakage.

TABLE IV: DATASET SPLIT

Split	Number of Samples	Percentage
Train	3600	72.0%
Val	900	18.0%
Test	500	10.0% (on disk)

If the validation loss does not decrease for 5 epochs, the learning rate is halved. The epochs where learning rate changed are shown in Table VI. The training loss curve for Greedy Decoding training is shown in Figure 6 along with validation loss curve, and beam decoding in Figure 7.

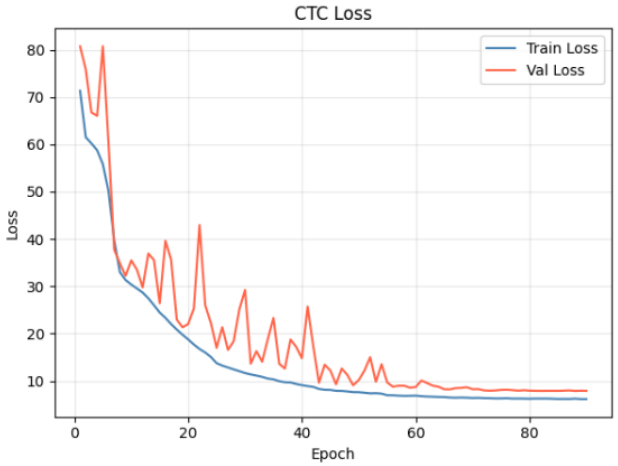

Figure 6: Training and Validation Loss Curves for Greedy phase.

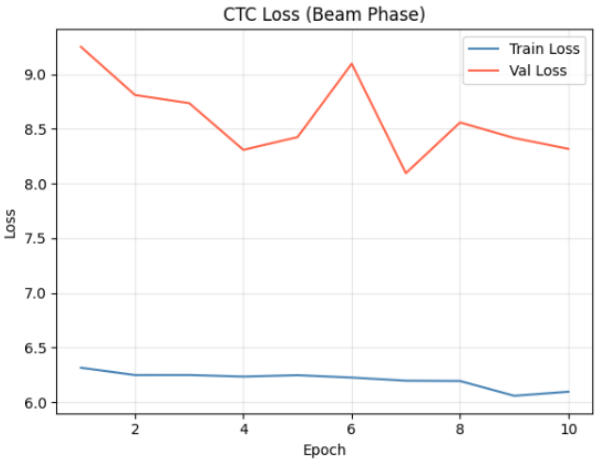

Figure 7 : Loss Curves under Beam Search.

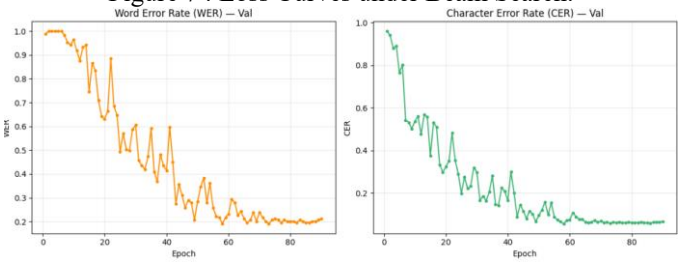

Figure 8 : WER and CER curves while training

TABLE V: EPOCHS WHERE LEARNING RATE CHANGED

Epoch	Learning Rate	Loss	Val Loss
23	1.00E-04	15.0478	22.2194
24	5.00E-05	13.7570	16.9930
42	2.50E-05	8.2938	9.6019
54	1.25E-05	6.9735	9.6768
64	6.25E-06	6.5940	8.2333
71	3.12E-06	6.3596	7.9837
78	1.56E-06	6.2523	8.0203
87	1.00E-06	6.2499	7.8697

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

TABLE VI: COMPARISON WITH LIP-SYNC-NET

Model	Streams	Parameters	Time	CER	WER
LipSyncNet [11]	2	307,595,340	46.8h	—	0.082
LipSyncLite	1	1,311,037	3.2h	0.09	0.257

D. Results and Discussion

The features output by the Transformer encoder are mapped to the vocabulary space through a fully connected layer with a softmax activation applied at each timestep. Because the correspondence between input video frames and output text is not known, we use Connectionist Temporal Classification (CTC) loss, which allows sequence-to-sequence learning without requiring explicit exact alignment.

The CTC loss is formulated as the negative log-likelihood of the target sequence:

$$\mathcal{L}_{CTC} = -\log P(y|x)$$

The computation of $P(\mathbf{y}|\mathbf{x})$ is efficiently performed using a forward–backward dynamic programming algorithm. This formulation allows the model to handle variable-length sequences and learn monotonic alignments, which is ideal for lipreading.

The Word Error Rate is :

$$WER = \frac{S_w + D_w + I_w}{N_w}$$

The Character Error Rate is:

$$CER = \frac{S_c + D_c + I_c}{N_c}$$

where S is Substitutions, D is Deletions, I is Insertions, and N is Total Reference Count.

Test Results:

- *Greedy Decode*:
  - Test WER 0.2579 ± 0.1596
  - Test CER 0.0942 ± 0.0912
- *Beam Search (width=10)*:
  - Test WER 0.2589 ± 0.1592
  - Test CER 0.0943 ± 0.0910

The evaluation results reveal a distinct performance gap between Character Error Rate (CER) and Word Error Rate (WER). While the model achieves a low CER, the WER is negatively impacted by errors in the fourth word of the sentence structure (the isolated letter). Per Sample WER and CER distributions are depicted in the figures 9 and 10.

1. The Challenge of Visually Similar Letters: A significant portion of the errors involves confusion between letters that look identical on the lips. For example, the model frequently misidentifies the letter ‘u’ as ‘q’. The phonetic sounds are visually very similar; /k/ involves no visible lip movement, causing frames to look almost identical.

2. Lack of Context in Isolated Letters: The model performs significantly better when recognizing letters embedded within full words compared to isolated letters. When speaking an isolated letter, the speaker starts from a neutral position and ends in a neutral position, removing surrounding visual context that the Transformer model relies on.

3. Impact on Word Error Rate: The dataset structure amplifies these errors. Since the sentences follow a fixed pattern, getting the isolated letter wrong counts as a full word error, even if the rest of the sentence is correct. This explains why the CER remains low while the WER is comparatively higher.

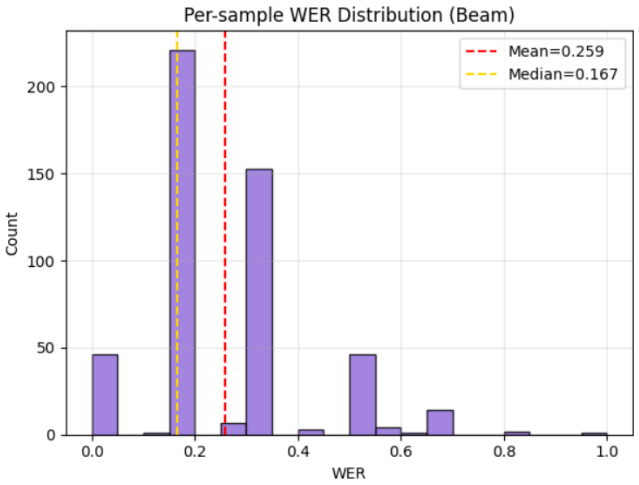

Figure 9 : Per sample WER distribution

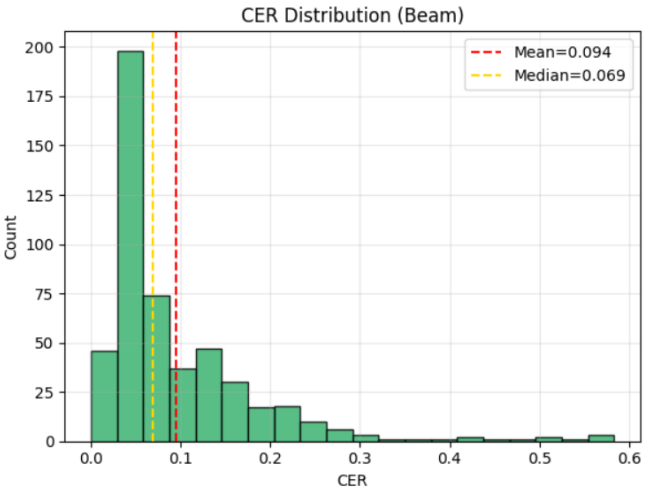

Figure 10 : Per sample CER distribution

V. CONCLUSION

This study presented a computationally efficient lip-reading architecture that integrates 3D-CNNs with a Transformer-based backend. A key achievement of this work is the drastic reduction in model complexity compared to existing large-scale architectures. Our proposed model utilizes only 1.3 million parameters and completes training in approximately 3 hours. This stands in stark contrast to the reference LipSyncNet, which requires 307 million parameters and 46.8 hours of training time. Despite its lightweight design, the system achieves a competitive Character Error Rate (CER) of 0.09, demonstrating that high-level visual speech recognition does not strictly require massive computational resources.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Future work will focus on bridging the gap between character-level accuracy and Word Error Rate (WER). Potential improvements include adopting a dual-stream feature extraction approach, potentially integrating pre-trained backbones like EfficientNetB0 to enhance spatial feature richness. Additionally, optimizing the architecture specifically for Tensor Processing Units (TPUs) could further leverage computational efficiency. Exploring phoneme-level mapping also presents a promising direction, as phonemes correlate more naturally with visemes than characters do. Finally, extending the training process to include "in-the-wild" datasets and diverse languages will be essential steps toward making the system robust for real-world deployment.

VI. REFERENCES

[1] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," Nature, vol. 264, pp. 746–748, 1976.
[2] S. Hilder, R. Harvey, and B.-J. Theobald, "Comparison of human and machine-based lip-reading," in AVSP 2009, Norwich, UK, 2009, pp. 86–89.
[3] X. Hong, H. Yao, Y. Wan, and R. Chen, "A PCA Based Visual DCT Feature Extraction Method for Lip-Reading," in 2006 Int. Conf. on Intel. Information Hiding and Multimedia, CA, USA, 2006, pp. 321–326.
[4] Q. Zeng, J. Du, and Z. Wang, "HMM-based Lip Reading with Stingy Residual 3D Convolution," in 2021 APSIPA ASC, Tokyo, Japan, 2021, pp. 1438–1443.
[5] S. Petridis, Z. Li, and M. Pantic, "End-to-end visual speech recognition with LSTMS," in 2017 IEEE ICASSP, New Orleans, LA, USA, 2017, pp. 2592–2596.
[6] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-end sentence-level lipreading," 2016, arXiv:1611.01599.
[7] N. K. Mudaliar, K. Hegde, A. Ramesh, and V. Patil, "Visual Speech Recognition: A Deep Learning Approach," in 2020 5th ICCES, Coimbatore, India, 2020, pp. 1218–1221.
[8] M. Wand, J. Koutn´ık, and J. Schmidhuber, "Lipreading with Long Short Term Memory," arXiv preprint arXiv:1601.08188, 2016.
[9] A. Graves, S. Ferna´ndez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in Proc. of 23rd ICML, 2006.
[10] C. Sheng, X. Zhu, H. Xu, M. Pietika¨inen, and L. Liu, "Adaptive Semantic-Spatio-Temporal Graph Convolutional Network for Lip Reading," IEEE Trans. Multimedia, vol. 24, pp. 3545–3557, 2022.
[11] S. A. A. Jeevakumari and K. Dey, "LipSyncNet: A Novel Deep Learning Approach for Visual Speech Recognition in Audio-Challenged Situations," IEEE Access, vol. 12, pp. 110891–110904, 2024.
[12] M. Thomas, E. Fish, and R. Bowden, "VALLR: Visual ASR Language Model for Lip Reading," arXiv preprint arXiv:2503.21408, 2025.
[13] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," The Journal of the Acoustical Society of America, 2006.
[14] D. E. King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, 10, 1755–1758, 2009.
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, 2017.