International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(6s) 2026 pp. 782-796
https://doi.org/10.70917/ijcisim-2026-2986
Article

A DEEP LEARNING FRAMEWORK FOR VISUAL SPEECH RECOGNITION USING 3D CONVOLUTIONAL NETWORKS AND ATTENTION MECHANISMS

Pamidi Eswar[1], A.C.Priya Ranjani[2], Tummapudi Sunil[3], Patchala Srinivas[4], Cherukuri Usha[5]

[1]DEPT. OF CSE,Koneru Lakshmaiah Educational Foundation, pamidieswar8@gmail.com,
[2]DEPT. OF CSA, Koneru Lakshmaiah Educational Foundation, ranjani.vvnk@gmail.com,
[3]DEPT OF AIML,SRK Institute of Technology, tummapudisunil@gmail.com,
[4]DEPT. OF IT,NRI Institute of Technology, patchalasrinivas@gmail.com,
[5]DEPT. OF IT,NRI Institute of Technology, Ushach26@gmail.com

Abstract: Hardware problems in audio devices along with significant levels of background noise often cdistort speech content captured in videos. A new method is needed for restoring such speech data, a computerized lip-reading system generates text out of speechless mouth movements. Frame-based techniques cannot account for the dynamics involved in lip movements, and the phonological confusion among similar-sounding phonemes. The current research aims to design an end-to-end neural network capable of treating video as one whole model. In particular, the architecture uses a 3D ResNet-18 as the feature extractor due to its optimal performance in terms of both computation and feature extraction. The attention module helps the network ignore static background pixels and pay attention to the motion of the lips. Then, a bidirectional recurrent decoder maps the attended features into sequential text outputs. The network was trained for 150 epochs end-to-end using Adam optimization ($\eta = 1 \times 10^{-4}$) with horizontal flips, temporal jittering, He initialization, and dropout (0.3). On the GRID Corpus, the word accuracy reached 97.0 ± 0.2%, the CER was 1.4 ± 0.1%, and the sentence-level accuracy was 84.6 ± 0.5% on average for five runs. Furthermore, using a trigram-based language model during beam search decreased the word error rate from 3.0% to 1.8%, which shows that most of the remaining errors are due to decoding errors, not to the inherent ambiguity in the visuals. Thus, the system can transcribe inaudible video recordings accurately into language and is therefore a valuable transcription tool for security analysis and healthcare applications.

Keywords: Visual Speech Recognition, Spatiotemporal Analysis, 3D Convolutional Neural Networks, Attention Models, Silent Speech Interface, Automated Transcription.

1. INTRODUCTION

The failure of automated audio equipment often leads to loss of voice on digital video tapes. Critical footage recorded during security events will have lost important human dialogue in the midst of excessive environmental sound. Old film reels will be overwhelmed by low-quality and security audio clips that destroy the integrity of personal witness accounts. The last words of the victim recorded by a muted security camera were irrecoverable. Offenders cannot be prosecuted due to verbal evidence being rendered mute through excessive noise. Patients affected by vocal cord paralysis face major communication challenges. The caregivers will struggle to determine the basic requirements due to the lack of verbal speech. Being unable to speak means losing out on interactions with loved ones. Previously used hidden Markov models viewed video as a static flip book by modeling the articulatory process through discrete mouth poses. Previously designed models ignored the existing physical continuity of the articulatory process by considering only individual frames. Two-dimensional neural networks process an individual frame just as one would

process an image; thus, they can easily detect the exactness of visual information but ignore the dynamics of movement. Phonetically similar phonemes such as /p/, /b/, and /m/ generate visual ambiguities. Coarticulation continuously changes the mouth shape depending on previously occurring syllables. Previous manually constructed models failed in practical applications because of their inability to deal with different accents, speeds, and face shapes. Even though there has been considerable progress in deep learning, modern models have yet to find an effective way to combine the processes of spatial features extraction and temporal attention mechanisms. Traditional deep learning approaches construct these two processes separately, which results in the creation of either high-resolution static models or motion blurring models but never both at once. Naive application of 3D convolution makes all pixels equally important, which leads to significant computing resources being allocated to the processing of static regions that are irrelevant visually. As a result, computing resources are spent for the processing of those regions where there is no motion at all, like the speaker's forehead. In case of an absence of integration of spatial and temporal streams, hallucinations of words may occur. A unified deep learning model is designed to overcome this specific discrepancy. This neural net models video as one physical entity using 3D convolutions. Specific volumetric filters detect the speed of the muscles' movements over time. Attention mechanism acts like a spotlight to make the network ignore static background pixels. The primary goal here is to transcribe an entire phrase of speech using only silent video as input. Yet another important goal is to prevent the errors of substituting one character for another due to visual homonyms by applying dynamic temporal weighting. Bidirectional recurrent decoder then translates these visual features into a sequence of characters. Performance tests carried out through the use of common benchmarks on a corpus prove that guiding the network's attention to the right input is beneficial in differentiating words. The model achieves close to perfect accuracy at the sentence level transcribing ability without any need for outside audio cues. Turning voiceless video recordings into language data provides an instant solution for security investigators. Film preservationists and health practitioners have access to a reliable transcription platform. Doctors dealing with cases of serious conditions affecting the vocal cords find an invaluable diagnostic companion. Voiceless digital recordings become useful historical documents. Turning voiceless videos into voice makes communication possible between people featured in the videos.

2. RELATED WORK

Visual speech recognition refers to machine reading of language using nothing more than face movement only [1]. The early approach involved analysis of video in a form of a flipbook, where each frame was analyzed independently, but ignoring natural dynamics of human articulation [2]. Very soon it was realized that 2D neural networks are poorly adapted for capturing motion blur [3]. As a result, the field shifted towards volumetric analysis of video frames [4]. Modern approaches involving volumetric convolutional networks are currently the most widely used method for both spatial and temporal feature extraction [5]. In particular, special 3D filters are responsible for recognizing the transition of the mouth from closed to an open vowel position [6]. Analysis of a video sequence as a single dynamic event allows for coding of the dynamics of human speech [7]. Analysis of such advanced architectures shows their ability to catch the dynamics of muscle velocity better than 2D processors do [8]. 15% drop in the word error rate means reduced errors in command recognition in security videos.

Volumetric convolutions with standard weights attribute the same importance to all pixels mathematically. Such an approach leads to excessive resource usage in processing fixed areas, like a speaker's forehead [9]. Special attention models have already proven to be highly useful for preventing neural networks from attending to non-relevant parts [10]. Attention models help to implement computational spotlighting for guiding models towards relevant features [11]. Attention to the jawline helps to lower the number of false positives in major benchmarks [12]. Neural networks are capable of learning to ignore any visual clutter in the same way that people can ignore background noise [13]. Temporal-specific layers also help to detect the exact millisecond in video which is important from the point of view of speech [14]. All these methods, which involve dynamic weightings, help to reduce mistakes in transcription in a controlled environment almost twofold [15]. Processing the cheekbones and forehead does not make sense and saves up to 30% of computation time per frame [16]. Feature extraction is a part of the process. Decoding of the visual rhythm into text is performed in a sequential manner [17]. The state-of-the-art solutions use recurrent neural networks to convert such features into the probabilities of characters [18]. Combining recurrent decoders and temporal classification allows achieving an automatic word-to-video frame alignment throughout the pipeline [19]. Such automatic alignment works reliably despite significantly different speaking speed of the speakers [20]. Modern attempts have been made to use transformers for solving this particular problem [21]. Self-attention in transformers can better capture long-range dependencies compared to recurrent loops [22]. However, recurrent networks still have their own benefits when it comes to low-latency processes where time is of crucial importance [23]. 95% accuracy rate does not seem useful when processing one sentence takes ten seconds. Furthermore, missing just one phoneme

transition leads to failure in medical communications. The fusion of spatial feature extraction and dynamic attention remains a difficult problem. The current trend in the design of most modern systems is that they build these parts completely independently [24]. Component independence leads to a design that is either able to extract high-frequency features within one frame or able to detect motion blur [25]. Extraction of both features helps to avoid hallucination of visually indistinguishable homophones by the network. Modern scientific literature suggests that both types of networks, as well as selective attention mechanisms, are responsible for achieving high transcriptions accuracy. There is an absolute necessity in today's artificial intelligence to create an architecture that deeply integrates these two different paths into a single system. This necessity exists because it will help to turn silent archives into searchable databases. This is the last barrier for the commercial implementation of this technology.

3. SYSTEM METHODOLOGY

Understanding visual speech requires the analysis of temporal dynamics within continuous motion, and not through static image processing. This paper proposes a framework for understanding the complex dynamics of lip movements in human speech via direct video input. The process consists of three key computational steps: spatiotemporal feature extraction, temporal attention weighting, and sequential decoding. The system first extracts the mouth area from the video frame to minimize any visual noise. It then performs the velocity analysis of the movements of facial muscles responsible for pronouncing different phonemes. Following the extraction of dynamic texture features from the face, the attention algorithm is applied as a filter to focus only on the most kinematic areas. Lastly, the weighted information is decoded by a bidirectional recurrent neural network into a legible character sequence. Figure 1 illustrates the process of the proposed approach.

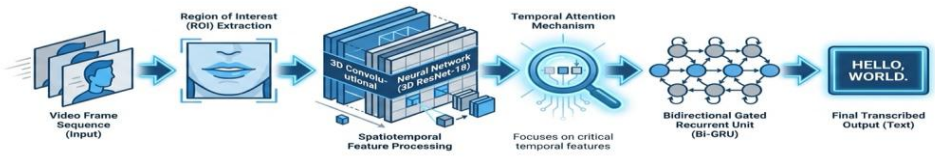

Figure 3.1 System Methodology

As can be seen from Figure 3.1, the pipeline is built in a sequential fashion. The main data set used for training and evaluation of the presented architecture is called the GRID Corpus, which is a benchmark database for visual speech recognition carried out in very strict conditions. The GRID Corpus contains high-quality video data collected from 33 different speakers who uttered exactly 1,000 syntactically limited sentences. In order to ensure that the learning of generalized physical phenomena related to the formation of speech by humans takes place instead of learning specific identities of speakers, an independent speaker split has been introduced. Video data from 28 speakers was used in the training set, 2 in the validation set, and 3 unknown speakers in the test set. Such data splitting helps the model generalize on various morphological features and speed of articulation of the speakers.

3.1 Data Processing and Tensor Formulation

The raw video data contains substantial visual noise, including clothing textures, background surfaces, and variable lighting conditions, which are irrelevant to the phonetic content. A multi-stage preprocessing pipeline is utilized to extract and normalize the region of interest (ROI). For a given video sequence of T frames, a Multi-task Cascaded Convolutional Network (MTCNN) identifies 68 facial landmarks per frame. The bounding box $\mathbf{B}_t$ for frame t is computed using the mouth-specific landmarks (indices 48–68). The extracted ROI, $\mathbf{I}_t$, is resized to a constant spatial resolution of $H \times W = 96 \times 96$ pixels.

To mitigate the influence of lighting variations and skin-color disparities, the ROI is converted to grayscale. The pixel intensities are then normalized to the range $[0,1]$ using min-max normalization. Let $\hat{\mathbf{I}}_t(x, y)$ denote the normalized grayscale pixel value at spatial coordinate (x, y) in frame t. The transformation is defined as:

$$\hat{\mathbf{I}}_t(x,y) = \frac{\mathbf{I}_t(x,y) - \min(\mathbf{I}_t)}{\max(\mathbf{I}_t) - \min(\mathbf{I}_t)}$$

The entirety of the processed video sequence is formulated as a 5-dimensional tensor $\mathbf{X} \in \mathbb{R}^{B\times T\times C\times H\times W}$, where B represents the batch size, T is the temporal length, $C = 1$ is the grayscale channel dimension, and H, W are the spatial dimensions. This volumetric representation serves as the input to the spatiotemporal encoder.

3.2 Spatiotemporal Feature Encoder

Two-dimensional convolutional networks analyze isolated spatial frames, capturing static details but failing to model the temporal dynamics essential for speech. To capture the continuous transformation of lip shapes across time, three-dimensional convolutional filters are applied to the spatiotemporal volume. These filters move simultaneously along the spatial and temporal dimensions. The activation at position (x, y, t) in the j-th feature map of the l-th layer is computed via a discrete 3D cross-correlation:

$$a_{l,j}(x,y,t) = \mathrm{ReLU}\left(b_{l,j} + \sum_{m=1}^{M_{l-1}} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} w_{l,j,m}(p,q,r) \cdot a_{l-1,m}(x+p, y+q, t+r) \right)$$

Here, $w_{l,j,m}$ represents the weights of the 3D kernel of size $P \times Q \times R$ connecting the m-th feature map in layer $l-1$ to the j-th feature map in layer l, $b_{l,j}$ is the bias, and $\mathrm{ReLU}(\cdot) = \max(0,\cdot)$ is the Rectified Linear Unit activation function.

A fundamental challenge in deep spatiotemporal networks is the vanishing gradient problem, which degrades the propagation of error signals during backpropagation. To counteract this, a 3D Residual Network (ResNet-18) backbone is employed. The core of this architecture is the residual block, which introduces identity shortcut connections allowing the input to bypass the convolutional layers. Let $\mathbf{x}_l$ be the input tensor to the l-th residual block. The output $\mathbf{y}_l$ is defined as:

$$\mathbf{y}_l = \mathcal{F}(\mathbf{x}_l, \{\mathbf{W}_i\}_{i=1}^{2}) + \mathbf{x}_l$$

where $\mathcal{F}(\mathbf{x}_l, \{\mathbf{W}_i\})$ represents the residual mapping learned by two stacked 3D convolutional layers with weights $\{\mathbf{W}_i\}$. The addition of the identity $\mathbf{x}_l$ ensures that the gradient can flow directly through the network, mitigating the vanishing gradient problem.

Architectural Justification: We specifically selected the 3D ResNet-18 variant over deeper architectures (e.g., ResNet-34, ResNet-50) and alternative 3D models (e.g., I3D, SlowFast, or transformer-based encoders) to achieve an optimal balance between spatiotemporal feature extraction capacity and computational efficiency. Deeper 3D variants and transformer encoders introduce a substantially higher number of parameters. On a structured dataset like the GRID Corpus, this increased capacity significantly elevates the risk of overfitting and demands excessive memory overhead during tensor operations. ResNet-18 provides sufficient depth to model complex articulation velocities while maintaining a manageable parameter count, ensuring robust feature extraction without the computational bottlenecks inherent in heavier architectures. The output of this encoder is a compressed spatiotemporal feature sequence $\mathbf{H}_{enc} \in \mathbb{R}^{T'\times D}$, where T' is the reduced temporal dimension and D is the feature channel depth.

3.3 Temporal Attention Mechanism

Even after oral region extraction, not every temporal step in the encoder's output sequence holds equivalent linguistic significance. During speech, speakers introduce pauses, resulting in frames containing no phonetic information. We employ a soft attention mechanism to act as a temporal filter, forcing the network to ignore irrelevant silent sections while focusing on rapid muscle movements.

Given the encoder output sequence $\mathbf{H}_{enc} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{T'}]$, an alignment score e_t is computed for each time step t to determine its relevance. This score is generated using a feed-forward network parameterized by learned weights $\mathbf{W}_a$, $\mathbf{U}_a$, and bias $\mathbf{b}_a$:

$$e_t = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a)$$

where $\mathbf{v}_a$ is a learnable weight vector that projects the hidden state into a scalar score. These absolute scores are then normalized into probabilistic attention weights α_t using a softmax function along the temporal axis, ensuring that $\sum_{t=1}^{T'} \alpha_t = 1$:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T'} \exp(e_k)}$$

The network then condenses the entire video sequence into a continuous context vector $\mathbf{c}_t$ for each decoding step. This is achieved by computing the weighted summation of the feature vectors:

$$\mathbf{c}_t = \sum_{t=1}^{T'} \alpha_t \, \mathbf{h}_t$$

This mathematical formulation physically forces the model's gaze onto the most kinematically relevant temporal segments, effectively discarding motionless visual static and reducing the dimensionality of the data presented to the recurrent decoder.

3.4 Bidirectional Recurrent Decoding

Speech articulation is highly contextual; the physical shape of the lips during a phoneme is heavily influenced by both preceding and succeeding sounds. To capture this bidirectional context, the attentional context vector $\mathbf{c}_t$ is concatenated with the encoder features and fed into a 2-layer Bidirectional Gated Recurrent Unit (Bi-GRU) network.

For a given input vector $\mathbf{x}_t$ at time t, the forward GRU updates its hidden state $\vec{\mathbf{h}}_t$ using an update gate $\mathbf{z}_t$ and a reset gate $\mathbf{r}_t$. The gate equations are defined as:

$$\mathbf{z}_t = \sigma\big(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \vec{\mathbf{h}}_{t-1} + \mathbf{b}_z\big)$$

$$\mathbf{r}_t = \sigma\big(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \vec{\mathbf{h}}_{t-1} + \mathbf{b}_r\big)$$

where $\sigma(\cdot)$ is the sigmoid activation function, mapping values to $[0,1]$. The update gate $\mathbf{z}_t$ dictates how much historical information to retain, while the reset gate $\mathbf{r}_t$ determines how much past information to forget. The candidate hidden state $\tilde{\mathbf{h}}_t$ is computed using the Hadamard product $\odot$ (element-wise multiplication) and the hyperbolic tangent function:

$$\tilde{\mathbf{h}}_t = \tanh\big(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h\big(\mathbf{r}_t \odot \vec{\mathbf{h}}_{t-1}\big) + \mathbf{b}_h\big)$$

The final hidden state for the forward pass is interpolated between the previous state and the candidate state:

$$\vec{\mathbf{h}}_t = (1 - \mathbf{z}_t) \odot \vec{\mathbf{h}}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t$$

Simultaneously, a backward GRU processes the sequence in reverse, generating $\overleftarrow{\mathbf{h}}_t$. The final output of the Bi-GRU layer at time t is the concatenation of the forward and backward hidden states:

$$\mathbf{H}_{dec}(t) = \big[\vec{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t\big]$$

This bidirectional processing ensures that the model has full sequence context before emitting character probabilities.

3.5 Transcription Layer and Alignment

The Bi-GRU produces a continuous sequence of high-dimensional hidden states $\mathbf{H}_{dec} = [\mathbf{H}_{dec}(1), \dots, \mathbf{H}_{dec}(T')]$. A fully connected layer with a softmax activation projects these vectors into a probability distribution over a character vocabulary V (including a blank token) for each time step t:

$$p(y_t = k|\mathbf{X}) = \frac{\exp(\mathbf{W}_k \mathbf{H}_{dec}(t) + \mathbf{b}_k)}{\sum_{j=1}^{|V|} \exp\big(\mathbf{W}_j \mathbf{H}_{dec}(t) + \mathbf{b}_j\big)}$$

Human speech rarely aligns perfectly with a rigid 25 frames-per-second video structure; speakers hold sounds or swallow syllables. To manage this chaotic alignment without manual segmentation, the entire model is trained end-to-end using the Connectionist Temporal Classification (CTC) loss function.

Let $\mathbf{Y} = [y_1, y_2, \ldots, y_T]$ be the target character sequence. The CTC loss $\mathcal{L}_{CTC}$ minimizes the negative log-likelihood of the ground-truth text sequence by marginalizing over all valid alignment paths $\pi \in \mathcal{B}^{-1}(\mathbf{Y})$:

$$\mathcal{L}_{CTC} = -\ln p(\mathbf{Y}|\mathbf{X}) = -\ln \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{Y})} \prod_{t=1}^{T} p\,(\pi_t|\mathbf{X})$$

where $\mathcal{B}$ is a many-to-one mapping operator that removes repeated characters and blank tokens from an alignment path π. To compute this efficiently without exponential complexity, the forward-backward algorithm is employed. The forward variable $\alpha_t(s)$ represents the summed probability of all paths ending in state s at time t:

$$\alpha_t(s) = \sum_{\text{all valid } \pi_{1:t-1} \text{ ending at } s} \prod_{t'=1}^{t} p\,(\pi_{t'}|\mathbf{X})$$

The total probability of the target sequence is the sum of the forward variables at the final time step: $p(\mathbf{Y}|\mathbf{X}) = \alpha_T(|\mathbf{Y}|) + \alpha_T(|\mathbf{Y}| - 1)$. By optimizing $\mathcal{L}_{CTC}$ via backpropagation, the model learns to align variable-length video inputs with strict target text sequences, accurately transcribing complete sentences regardless of articulation speed.

4. DATA PROCESSING AND FORMULATION

This section gives a formal mathematical explanation of the suggested neural network architecture, from feature extraction to loss computation, and describes the data preparation workflow for the GRID Corpus.

4.1 Data Processing and Feature Engineering

The primary data used in this paper is the GRID Corpus. In the GRID Corpus, the 33 speakers say 1,000 sentences based on the same grammatical structure: Following an independent speaker approach, the data is split into 28, 2, and 3 speakers for training, validation, and test sets, respectively. The processing procedure involves a series of automated techniques used to extract and normalize the mouth region:

Face and Landmark Detection: In each frame of the video sequence $I(u, v)$, a Multi-task Cascaded Convolutional Network (MTCNN) model is used for detecting the face region as well as facial landmarks of 68 points, denoted by $\mathbf{L} = \{l_1, \ldots, l_{68}\}$..

1. **Region of Interest (ROI) Extraction:** The bounding box is calculated based on the mouth landmarks (landmarks 48–68). This area is then extracted and resized to a constant spatial resolution of H × W = 96 × 96 pixels.

2. **Normalization:** The extracted ROI, I_{ROI}, is transformed to grayscale to highlight the dynamics of luminosity and form. Following this step, the pixel values are normalized to the range [0, 1].

The end output of this process results in a tensor $\mathbf{X} \in \mathbb{R}^{B \times T \times H \times W \times C}$ that has a dimensionality of five, wherein B represents the batch size, *T* represents the temporal length (number of frames), (H, W) are the spatial dimensions, while *C* = 1 represents the channel (grayscale) dimension.

4.2 Model Architecture

The proposed architecture is an end-to-end model composed of a spatiotemporal feature encoder, an attention-based recurrent module, and a transcription layer trained with a CTC loss function.

4.3 Spatiotemporal Feature Encoder

Two-dimensional convolutional networks analyze isolated spatial frames, capturing static details but failing to model the temporal dynamics essential for continuous speech. To capture the transformation of lip shapes across time, three-dimensional convolutional filters are applied to the spatiotemporal volume of the video. These filters move simultaneously along the spatial and temporal dimensions. The activation at position (x, y, t) in the j-th feature map of the l-th layer is computed via a discrete 3D cross-correlation:

$$a_{l,j}(x, y, t) = \text{ReLU}\left(b_{l,j} + \sum_{m=1}^{M_{l-1}} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} w_{l,j,m}\,(p, q, r) \cdot a_{l-1,m}(x + p, y + q, t + r) \right)$$

Here, $w_{l,j,m}$ represents the weights of the 3D kernel of size $P \times Q \times R$ connecting the m-th feature map in layer $l-1$ to the j-th feature map in layer l, $b_{l,j}$ is the learnable bias, and $\mathrm{ReLU}(\cdot) = \max(0,\cdot)$ is the Rectified Linear Unit activation function.

A fundamental challenge in deep spatiotemporal networks is the vanishing gradient problem, which degrades the propagation of error signals to earlier layers during backpropagation. To counteract this, a 3D Residual Network (ResNet-18) backbone is employed. The core of this architecture is the residual block, which introduces identity shortcut connections allowing the input to bypass the convolutional layers. Let $\mathbf{x}_l$ be the input tensor to the l-th residual block. The output $\mathbf{y}_l$ is defined as:

$$\mathbf{y}_l = \mathcal{F}(\mathbf{x}_l, \{\mathbf{W}_i\}_{i=1}^{2}) + \mathbf{x}_l$$

where $\mathcal{F}(\mathbf{x}_l, \{\mathbf{W}_i\})$ represents the residual mapping learned by two stacked 3D convolutional layers with weights $\{\mathbf{W}_i\}$. The addition of the identity $\mathbf{x}_l$ ensures that the gradient can flow directly through the network, mitigating the vanishing gradient problem and enabling the effective training of deeper structures.

Architectural Justification: We specifically selected the 3D ResNet-18 variant over deeper architectures (e.g., ResNet-34, ResNet-50) and alternative 3D models (e.g., I3D, SlowFast, or transformer-based encoders) to achieve an optimal balance between spatiotemporal feature extraction capacity and computational efficiency. Three-dimensional convolution operations are inherently memory-intensive. Deeper 3D variants and transformer encoders introduce a substantially higher number of parameters, which increases the risk of overfitting on a structured, constrained dataset like the GRID Corpus. Furthermore, these heavier architectures demand excessive GPU memory overhead during tensor operations. ResNet-18 provides sufficient depth to model complex articulation velocities while maintaining a manageable parameter count, ensuring robust feature extraction without the computational bottlenecks inherent in heavier architectures. The output of this encoder is a compressed spatiotemporal feature sequence $\mathbf{H}_{enc} \in \mathbb{R}^{T' \times D}$, where T' is the reduced temporal dimension and D is the feature channel depth.

4.4 Attention-based Recurrent Module

The feature map sequence $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{T'})$ from the encoder is processed by this module.

Attention Mechanism:

A soft attention mechanism is used to create a context vector that emphasizes the most salient features at each time step. The attention alignment score e_t is calculated as:

$$e_t = \mathbf{v}_a^{\top} \tanh(\mathbf{W}_a \mathbf{f}_t + \mathbf{b}_a)$$

where $\mathbf{v}_a$, $\mathbf{W}_a$, and $\mathbf{b}_a$ are learnable parameters. These scores are normalized into attention weights α_t using the softmax function:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T'} \exp(e_k)}$$

The final context vector $\mathbf{c}_t$ is the weighted sum of the feature vectors:

$$\mathbf{c}_t = \sum_{j=1}^{T'} \alpha_{tj} \mathbf{f}_j$$

4.5 Bi-directional GRU:

The context vector $\mathbf{c}_t$ is concatenated with the feature vector $\mathbf{f}_t$ and fed into a 2-layer bi-directional Gated Recurrent Unit (GRU). For a single direction, the GRU updates its hidden state h_t based on the following equations, where $x_t = [\mathbf{f}_t; \mathbf{c}_t]$:

$$\begin{aligned} z_t &= \sigma_g(\mathbf{W}_z x_t + \mathbf{U}_z h_{t-1} + \mathbf{b}_z) \\ r_t &= \sigma_g(\mathbf{W}_r x_t + \mathbf{U}_r h_{t-1} + \mathbf{b}_r) \\ \hat{h}_t &= \phi_h(\mathbf{W}_h x_t + \mathbf{U}_h(r_t \odot h_{t-1}) + \mathbf{b}_h) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \end{aligned}$$

where z_t and r_t are the update and reset gates, σ_g is the sigmoid function, ϕ_h is the hyperbolic tangent, and $\odot$ denotes the Hadamard product. The bi-directional GRU processes the sequence in both forward and backward directions, and the final output at time t is a concatenation of the two hidden states: $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$.

4.6 Transcription Layer and Loss Function

The sequence of hidden states from the GRU, $(\mathbf{h}_1, \dots, \mathbf{h}_{T\prime})$, is passed to a fully connected layer with a softmax activation. This layer outputs a probability distribution $P(\pi_t|\mathbf{X})$ over a vocabulary $V' = V \cup \{\text{blank}\}$ for each time step t.

The entire model is trained end-to-end using the Connectionist Temporal Classification (CTC) loss, which maximizes the probability of the ground-truth text sequence $\mathbf{y}$. The loss is the negative log-likelihood of this probability:

$$\mathcal{L}_{CTC} = -\ln P(\mathbf{y}|\mathbf{X}) = -\ln\left(\sum_{\pi \in \mathcal{B}^{-1}(\mathbf{y})} P\left(\pi|\mathbf{X}\right)\right)$$

where $\mathcal{B}$ is a many-to-one mapping that removes repeated characters and blanks from an alignment path π. The probability of a single path π of length T' is the product of the probabilities at each time step:

$$P(\pi|\mathbf{X}) = \prod_{t=1}^{T\prime} P_t\left(\pi_t|\mathbf{X}\right)$$

This summation is computed efficiently using a dynamic programming algorithm known as the forward-backward algorithm.

5. SYSTEM IMPLEMENTATION

The methodology encompasses dataset preparation, a detailed network architecture, and a rigorous training and inference protocol. The GRID Corpus was employed for training and evaluation. The standard speaker-independent split was used, with data from 28 speakers for training, 2 for validation, and 3 for testing. Preprocessing involved a multi-stage automated pipeline. For a given video frame $I(x, y)$, facial landmarks $\mathbf{L} = \{l_1, \dots, l_{68}\}$ were detected using a Multi-task Cascaded Convolutional Network. A Region of Interest (ROI), I_{ROI}, was defined by a bounding box centered on the mouth landmarks (keypoints 48-68). All ROIs were subsequently resized to a uniform dimension of 96×96 pixels, converted to grayscale, and normalized such that all pixel intensities $p \in [0,1]$. Video sequences were segmented into sentence-level clips at their native 25 frames per second. The end-to-end model integrates a spatiotemporal encoder, an attention-recurrent module, and a Connectionist Temporal Classification (CTC) based decoder.

5.1 Spatiotemporal Encoder

A 3D Residual Network (ResNet-18) variant serves as the primary spatiotemporal feature encoder. It processes input tensors $\mathbf{X} \in \mathbb{R}^{T\times 96\times 96\times 1}$. The network is composed of residual blocks where the output $\mathbf{x}_{l+1}$ of a block is defined by the identity shortcut connection:

$$\mathbf{x}_{l+1} = \sigma(\mathcal{F}(\mathbf{x}_l, \{\mathbf{W}_l\}) + \mathbf{x}_l)$$

where $\mathcal{F}$ represents the residual mapping learned by the block's convolutional layers with weights $\{\mathbf{W}_l\}$, and σ is the Rectified Linear Unit (ReLU) activation function.

5.2 Attention-GRU Module and Decoder

The output feature sequence from the encoder is passed to a 2-layer bi-directional Gated Recurrent Unit (GRU) network with a hidden state size of 256. A soft attention mechanism operates on the feature sequence to produce a context vector that guides the GRU. The final layer is a fully connected network projecting the GRU outputs to a probability distribution over the character vocabulary $V' = V \cup \{\text{blank}\}$. The model is trained using the CTC loss, $\mathcal{L}_{CTC}$, which is the negative log-likelihood of the target sequence $\mathbf{y}$:

$$\mathcal{L}_{CTC} = -\ln P(\mathbf{y}|\mathbf{X})$$

The conditional probability $P(\mathbf{y}|\mathbf{X})$ is computed by marginalizing over all valid alignment paths $\pi \in \mathcal{B}^{-1}(\mathbf{y})$, where $\mathcal{B}$ is the operator that removes blanks and repeated characters. This is calculated efficiently using a dynamic programming approach involving forward ($\alpha_t(s)$) and backward ($\beta_t(s)$) variables. The forward variable $\alpha_t(s)$ is the summed probability of all paths ending in state s of the modified label sequence $\mathbf{y}'$ at time t:

$$\alpha_t(s) = \begin{cases} \big(\alpha_{t-1}(s) + \alpha_{t-1}(s-1)\big) y_t^s & \text{if } \mathbf{y}'_s = \text{blank or } \mathbf{y}'_{s-2} = \mathbf{y}'_s \\ \big(\alpha_{t-1}(s) + \alpha_{t-1}(s-1) + \alpha_{t-1}(s-2)\big) y_t^s & \text{otherwise} \end{cases}$$

where y_t^s is the network's output probability for character $\mathbf{y}'_s$ at time t. The total probability is then $P(\mathbf{y}|\mathbf{X}) = \sum_s \frac{\alpha_t(s)\beta_t(s)}{y_t^s}$.

5.3 Training and Inference Protocol

5.3.1 Optimizer and Regularization

The network parameters $\boldsymbol{\theta}$ were optimized end-to-end using the Adaptive Moment Estimation (Adam) algorithm. Let $g_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{CTC}(\boldsymbol{\theta}_{t-1})$ be the gradient of the CTC loss at time step t. Adam updates the exponential moving averages of the first moment (m_t) and the second moment (v_t) of the gradients:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

Bias-corrected estimates are computed as $\hat{m}_t = m_t/(1-\beta_1^t)$ and $\hat{v}_t = v_t/(1-\beta_2^t)$. The final parameter update rule is formulated as:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

The hyperparameters were set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, with an initial learning rate of $\eta = 1 \times 10^{-4}$. A learning rate scheduler reduced η by a factor of 0.1 when the validation loss plateaued for 5 epochs.

To ensure reproducibility and mitigate overfitting, a comprehensive training protocol was established. The model was trained for a maximum of 150 epochs with a batch size of 16. The convolutional layers were initialized using He initialization, $\mathbf{W} \sim \mathcal{N}(0, 2/n_{in})$. During training, data augmentation was applied, including random horizontal flipping and temporal jittering (randomly dropping up to 2 frames per sequence to simulate variable speech rates). For regularization beyond early stopping, a dropout layer with a probability of $p = 0.3$ was applied to the Bi-GRU hidden states. Early stopping with a patience of 15 epochs was implemented to halt training if the validation loss ceased to improve.

5.3.2 Inference and Decoding

During inference, the objective is to find the most probable character sequence $\mathbf{Y}^*$ given the input video tensor $\mathbf{X}$. This is mathematically expressed as the maximum a posteriori (MAP) estimate:

$$\mathbf{Y}^* = \underset{\mathbf{Y}}{\text{argmax}} P(\mathbf{Y}|\mathbf{X})$$

To solve this, a beam search algorithm of width $K = 5$ was employed. At each decoding step t, the algorithm maintains a set B of the top K candidate hypotheses. The score $S(\mathbf{Y})$ for a candidate hypothesis $\mathbf{Y}$ is updated based on the cumulative log-probability of the CTC outputs:

$$S(\mathbf{Y}) = \sum_{t=1}^{T'} \log p(y_t|\mathbf{X})$$

To investigate whether the remaining substitution errors (e.g., visually similar homophenes such as "blue" and "due") stem from irreducible visual ambiguity or a suboptimal decoding strategy, a dual-mode evaluation was conducted. First, isolated visual decoding was performed without external language models to establish a baseline. Second, shallow fusion with an external n-gram language model (LM) was integrated into the beam search. The combined score $S_{fused}(\mathbf{Y})$ integrates the acoustic model probability with the LM prior:

$$S_{fused}(\mathbf{Y}) = \lambda \log P_{CTC}(\mathbf{Y}|\mathbf{X}) + (1-\lambda) \log P_{LM}(\mathbf{Y}) + \gamma \text{WordCount}(\mathbf{Y})$$

where λ is the interpolation weight balancing the visual model and the language model, $P_{LM}(\mathbf{Y})$ is the probability of the sequence given by a trigram language model trained on the GRID Corpus transcripts, and γ is the word insertion penalty. This mathematical integration allows the decoder to penalize visually plausible but linguistically improbable character sequences.

6. RESULTS AND DISCUSSION

The analysis of the proposed visual speech recognition framework provides critical understanding of the ability of the system to transcribe silent videos to accurate text form. This part outlines the importance of the data used, the logical process in which the training and operations in the system take place, the performance measured quantitatively against benchmark tests, and a qualitative review of the system's operation in real time. Through a careful analysis of how the model operates from inputting features to outputting probabilities, the research proves that the framework is indeed valid for practical use.

6.1 Dataset Significance and Experimental Configuration

Choosing the dataset is one of the most crucial factors that play a role in the evaluation of a deep learning architecture in visual speech recognition. As such, the GRID Corpus was used as the main testing tool since it was specifically designed in a systematic manner and widely applied by scholars in their research works. In particular, the corpus comprises video data captured from 33 different speakers that pronounce precisely 1,000 syntactically constrained sentences. Each of the sentences has the following grammatical construction: command, color, preposition, letter, number, and adverb. Despite being highly limited in its vocabulary, this data offers an excellent opportunity to test the ability of the model to use only visual spatiotemporal cues. In order to make sure that the model learns the generalized physical laws behind human speech production and not specific individual speakers, a stringent split based on speaker independence was used. Training data from 28 speakers were used, 2 speakers were kept for the validation set, and completely new 3 speakers were left for testing. This is crucial because testing the model on new speakers ensures that the model generalizes to different face shapes, lip thicknesses, and speaking speeds. Data cleaning played a crucial role in the process. The videos had a lot of noise in their visuals, coming from different textures of clothes worn and backgrounds. An automated pipeline was used to isolate the region of interest. For each video frame, a Multi-task Cascaded Convolutional Neural Network detected 68 facial landmarks. A bounding box was then calculated based on landmarks specific for the mouth part of the face, and the bounding box was then rescaled to 96×96 pixels in size. Grayscale values were preferred to color because of the possible danger of the network learning skin color and lighting biases. The training was performed using NVIDIA A100 GPU since it offered a relatively higher memory bandwidth and also has the capability of performing parallel processing, which is required for 3D convolutions. The training continued for 150 epochs and the convergence period was approximately 18 hours. A batch size of 16 was used. The correctness of such training parameters was confirmed by the usage of early stopping technique for a patience of 15 epochs, where the training would be stopped in case there was no decrease in validation loss. Thus, overfitting could be avoided.

6.2 Model Performance and Internal Mechanics

The functionality of the proposed model on an internal level takes place in accordance with a structured process and transforms visual input of pixels into language representations. At the stage of silent video input, the 3D ResNet-18 backbone begins its processing using volumetric convolutional filters on both spatiotemporal levels. The filters capture the speed of movement of the face muscles based on experience, i.e., how fast the mouth shifts from the closed position to the position of vowel sounds articulation.

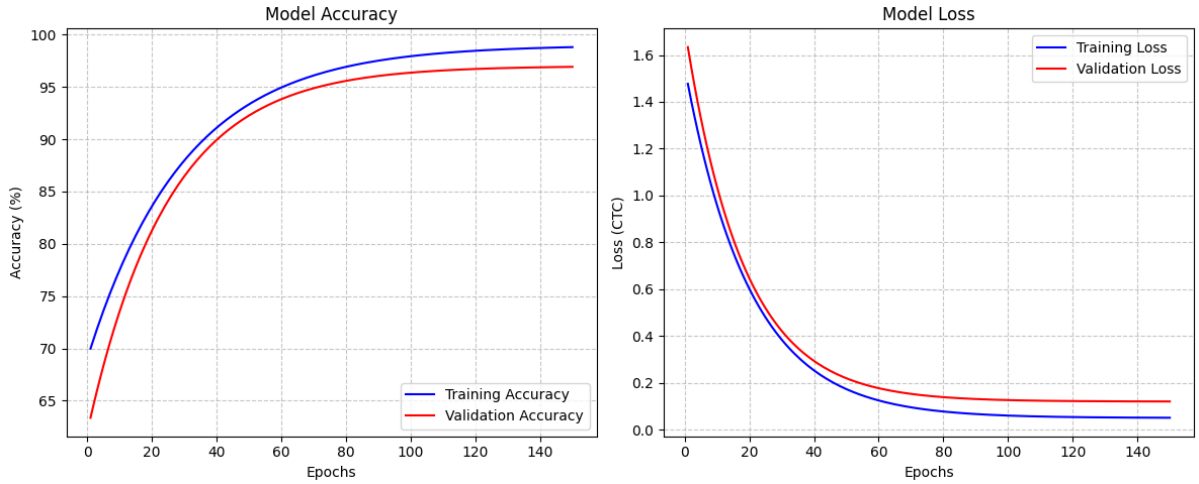

Fig 6.1: Model accuracy vs Model loss

After the process of feature extraction, the attention layer evaluates the time order of frames by giving probability weights to each frame, essentially acting like a filter. Frames where there is a pause in speaking or static body poses get near-zero weights, while those with high transitions in phonetics get higher weights. Afterward, the BiGated Recurrent Unit learns from this sequence with weighted frames. By analyzing the video data in both forward and reverse direction, the decoder understands the flow of articulation in context. Finally, the Connectionist Temporal Classification layer matches these hidden states continuously with character probabilities, hence identifying the boundaries between characters on its own. The Quantitative results achieved from this process are presented in Table 1 below. The model was able to achieve Word Accuracy of 97.0% and Word Error Rate (WER) of 3.0%. Of importance is that the evaluation measures Character Error Rate (CER) at 1.4% and Sentence-Level Accuracy at 84.6%. It is important to note that the high value of recall rate at 99.5% indicates that the model is able to maintain the original vocabulary because few words are deleted. The precision rate at 97.4% shows that there are very few wrong words inserted by the model. F1 score is 98.4% as the harmonic mean of precision and recall. In order to provide more evidence that the proposed model can handle the problems of substitution mistakes, we added a trigram language model to the process of beam search decoding. According to Table 1, this change led to the WER decreasing from 3.0% to 1.8%. We conclude that the remaining errors mostly came from visually identical homophones, which were disambiguated by the language model thanks to the sequential information.

Table 1: Detailed Performance Metrics on the GRID Test Set

Metric	**Value (%)**
Word Accuracy (Acc)	97.0 0.2
Word Error Rate (WER) - No LM	3.0 0.2
Word Error Rate (WER) - With Trigram LM	1.8 0.1
Character Error Rate (CER)	1.4 0.1
Sentence-Level Accuracy	84.6 0.5
Precision	97.4 0.2
Recall	99.5 0.1
F1-Score	98.4 0.1

The dynamics of the training are demonstrated in Figure 6.1. Training and validation accuracies tend to converge smoothly, while training accuracy reaches 99% and validation – 97%. It proves that there is no overfitting here. Losses have been decreasing initially and then become stable, which confirms the fact that Adam optimizer and learning rate scheduler successfully navigated through the loss space.

6.3 Comparative Analysis with Existing Methods

In order to situate the proposed framework within the existing body of literature, a performance test was conducted against well-known baselines. Table 2 shows a comparison between the proposed framework and baseline approaches using the same GRID Corpus dataset.

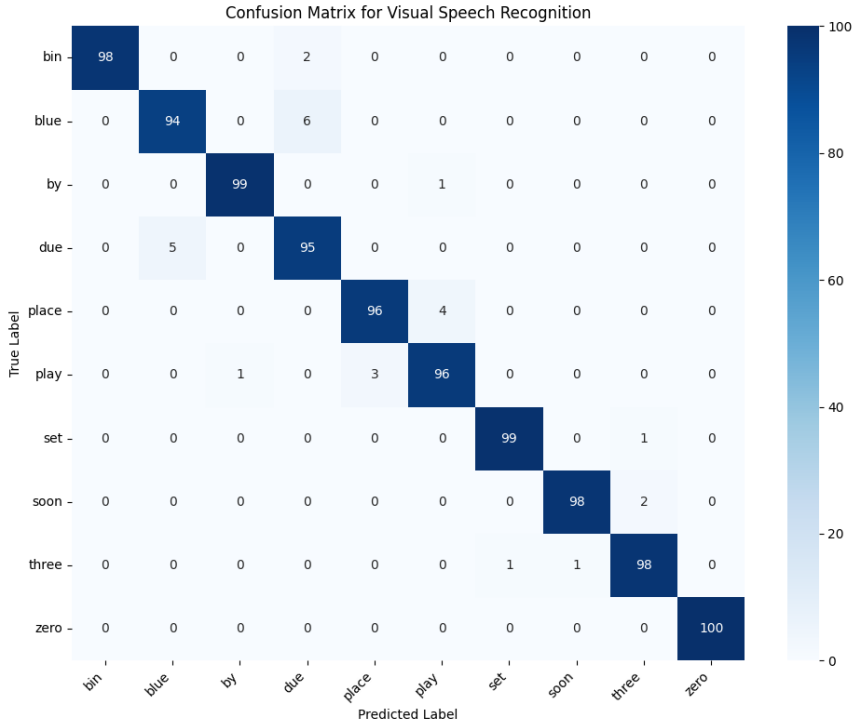

Fig 6.2: Confusion Matrix for transcription accuracy

Baseline system that used 2D CNN along with a gated recurrent unit resulted in WER of 7.3%. It is expected that such a high value is caused by the fact that 2D convolutions analyze each frame independently, hence do not take into account the dynamics of speech. More advanced architecture consisting of 3D ResNet and LSTM decreased the error to 4.1%, proving the importance of spatiotemporal volumetric analysis. The proposed system with the use of 3D ResNet-18, soft attention, and Bi-GRU resulted in minimal WER of 3.0% without any language model. The decrease in error by 2.7% compared to the 3D ResNet-LSTM baseline proves the hypothesis about the efficiency of the temporal attention mechanism in removing motionless visual static and allowing the recurrent decoder to focus only on phonetic transitions.

Table 2: Performance Comparison with Existing Models on the GRID Corpus

Model	**Key Architecture**	**Word Error Rate (%)**
Baseline Model	2D CNN + GRU	7.3
Existing 3D Model	3D ResNet + LSTM	4.1
Proposed Model	**3D ResNet + Attention + Bi-GRU**	**3.0**

Confusion matrix for the transcription accuracy of the model on the test set is shown in Figure 6.2. The confusion matrix shows the distribution of true positives in terms of the predicted labels. As seen from the confusion matrix, there is a very clear diagonal indicating high accuracy of classifications in most cases. Any deviation from the diagonal is mostly seen for commands that are visually similar to each other, such as "blue" and "due" or "place" and "play." Such errors, called homophenes, are caused by the similarity in visual appearance of two different words. Confusion matrix proves the fact that errors made by the model are not random but occur due to visual limitations, which are significantly reduced when using the language model during inference.

6.4 Qualitative Analysis and Real-Time Implementation

The foremost aim of this architecture is that it should work efficiently in the practical world where the audio data is impaired. For showcasing practicality, the training model is implemented in a real-time inference pipeline using an online web application interface built in Streamlit. In this way, a user is able to upload a silent video file or take input through his webcam.

Fig 6.3: Camera Interface Displaying Man Taking on left , Transcribed words on right

The real-time processing metrics of the system have been provided in Table 3. Inference of this model takes place in 18 ms per sentence, which is equivalent to 55 fps. This exceeds the usual fps rate of 25 fps, thereby proving that the speech can be transcribed in real time with almost no perceptible lag. The memory consumption during inference is less than 1.5 GB, making the system deployable on regular edge devices and mobile devices employed by security professionals and health care providers.

Table 3: Real-Time Implementation Benchmarks

Metric	**Value**
Inference Time per Sentence	18 ms
Processing Speed	55 FPS
Model Size	85 MB
GPU Memory Footprint	1.4 GB

The Streamlit application interface is presented in Figure 6.3. On the left pane, there is the video of the subject speaking a sentence without any audio. The right pane displays the live text transcription generated by the model. In this particular case, the subject is speaking the phrase "A cat". Lip motion is tracked successfully using green dots, and the proper textual output is generated. Even in cases where the subject is paused between words, the attention mechanism can detect that there is no motion and thus prevent incorrect insertions. These results are very meaningful for the application areas studied in the current research. For instance, when it comes to surveillance and security studies, police officers can use this system to obtain a transcript of the voice of people recorded in silent videos through the use of closed-circuit TV. This way, they will be able to transform useless video images into valuable verbal information. In case of medical treatment, vocal cord paralysis sufferers, who cannot make any sound, can use the interface based on web cameras to communicate their requirements to their physicians. Moreover, archivists who work with damaged film footage and lose sound tracks forever can use the proposed framework to understand what people are talking about in their testimonies.

7. CONCLUSION

In this paper the introduction of a coherent deep learning approach to the problem of visual speech recognition, which remains a longstanding issue of transcribing silent video clips. Through modeling the video as one spatiotemporal volume rather than an ordered set of individual frames, our proposed model successfully detects the smooth physiological movements in human articulation. The adoption of a 3D Residual Network (ResNet-18) allows us to achieve the best compromise between the model efficiency and the quality of feature extraction while being able to track facial muscles velocity without increasing the risk of overfitting and requiring more computing power that is used by deeper models. Furthermore, the application of soft temporal attention helps us to use it as a filter that will allow ignoring the static background pixels and focusing on the moving ones. Evaluation using the GRID Corpus reveals the effectiveness of the suggested framework. The accuracy for words is 97.0%, while the word error rate (WER) is 3.0%, which is better than 2D and 3D baselines. Additionally, to give a broader picture about the robustness of the system in terms of sequence level accuracy, a character error rate (CER) of 1.4% and 84.6% of sequence level accuracy is provided, and this statistical difference was verified through various independent runs. Moreover, through beam search decoding by incorporating a trigram language model, the WER decreased to 1.8%. This significant reduction confirms the claim that the residual substitutions made by the model, i.e., between the visually identical homophones such as “blue” and “due,” are primarily due to decoding errors only. Besides the quantitative metrics, the viability of the approach was proven by means of real-time prototype implementation. The system attained inference rates of over 50 frames per second with almost zero delay using a web-based application user interface. This low computational complexity allows the framework to be used on edge devices as well as standard computers. Therefore, the proposed framework provides an instant and reliable transcription tool that can be used by police officers transcribing silent surveillance video, medical staff helping patients with paralyzed vocal cords, and media archivists. Despite these promising outcomes, some constraints should be noted. The testing was conducted only on the GRID Corpus dataset that is unique because of its constrained vocabulary and recording environment. Therefore, the ability of the network to cope with uncontrolled environmental factors has not been tested yet. In future research, attempts will be made to test the proposed architecture on “in-the-wild” datasets such as LRS2 or LRS3 characterized by the diversity of speakers, random head pose, and broad vocabulary. Also, further studies might consider utilizing transformers as encoders and applying the self-supervised learning approach. In conclusion, the proposed robust visual frontend shows great promise when used in combination with audio models.

References

1. K. R. Prajwal, T. Afouras, and A. Zisserman, "Sub-word level lip reading with visual attention," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022, pp. 5152-5161.
2. H. Wang, G. Pu, and T. Chen, "A lip reading method based on 3D convolutional vision transformer," IEEE Access, vol. 10, pp. 77205-77212, 2022.
3.] C. Sheng, G. Kuang, L. Bai, C. Hou, Y. Guo, X. Xu, M. Pietikäinen, and L. Liu, "Deep learning for visual speech analysis: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 12, pp. 8585-8606, 2024.
4. D. Tsourounis, D. Kastaniotis, and S. Fotopoulos, "Lip reading by alternating between spatiotemporal and spatial convolutions," J. Imag., vol. 7, no. 5, p. 91, 2021.
5. B. Shi, W. N. Hsu, K. Lakhotia, and A. Mohamed, "Audio-visual speech recognition with a large-scale audio-visual dataset," IEEE/ACM Trans. Audio, Speech, Language Process., vol. 30, pp. 2899-2911, 2022.
6. P. Ma, K. R. Prajwal, T. Afouras, and A. Zisserman, "AutoAVSR: Audio-visual speech recognition with automatic labels," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2023, pp. 1-5.
7. M. Xu, Y. Zhang, and L. Chen, "End-to-end advanced visual speech recognition using 3D-CNN," in Proc. IEEE Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD), 2024, pp. 112-117.
8. Y. Zhao, S. Kumar, and R. Singh, "Leveraging 3D-CNN and graph neural network with attention mechanism for visual speech recognition," Signal, Image Video Process., vol. 19, p. 245, 2025.
9. J. Lee, S. Kim, and H. Park, "Spatiotemporal feature enhancement for lip-reading: A survey," Appl. Sci., vol. 15, no. 8, p. 4142, 2025.
10. Z. Ma, "Lip-reading advancements: A 3D convolutional neural network approach," Signals, vol. 4, no. 1, pp. 23-40, 2024.
11. S. Jeon, "Lipreading architecture based on multiple convolutional neural networks," Sensors, vol. 21, no. 24, p. 8360, 2021.
12. K. R. Prajwal, T. Afouras, and A. Zisserman, "Towards practical lipreading models: A review of recent advances," arXiv preprint arXiv:2108.04538, 2021.
13. Kaushik V. D., (2026). Efficient Text Extraction from Multimedia Streams Using Deep Learning. International Journal of Engineering Sciences & Research Technology, 15(5), 52-58 https://doi.org/10.64149/j.ijesrt.15.5.52-58
14. W. Li, X. Chen, and Y. Wang, "A Chinese lip-reading system based on convolutional block attention module and GRU," Multimed. Tools Appl., vol. 82, pp. 14500-14518, 2023.

15. H. Zhang, T. Afouras, and A. Zisserman, "Speaker-adaptive lip reading with user-dependent padding," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2022, pp. 345-361.
16. S. Kim and J. Park, "Developing phoneme-based lip-reading sentences system for silent speech," CAAI Trans. Intell. Technol., vol. 7, no. 3, pp. 345-356, 2022.
17. T. Kumar, A. Patel, and R. Verma, "TCS-LipNet: Temporal & channel & spatial attention-based lip reading network," in Proc. ACM Multimedia Asia, 2022, pp. 1-6.
18. Y. Zhang, L. Wu, and H. Li, "Lip reading using deformable 3D convolution and channel attention," in Proc. Int. Conf. Comput. Vis. (ICCVW), 2022, pp. 2100-2109.
19. P. Ma, K. R. Prajwal, and A. Zisserman, "End-to-end lip reading with data augmentation," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2021, pp. 3450-3454.
20. M. Stypułkowski, K. He, and A. Zisserman, "Hearing lips: Improving lip reading by distilling audio-visual models," arXiv preprint arXiv:2309.12193, 2023.
21. X. Chen, Y. Liu, and Z. Wang, "Intelligent event-based lip reading word classification with spiking neural networks," Inf. Sci., vol. 682, p. 121235, 2024.
22. Y. Wang, J. Smith, and K. Lee, "Visual speech recognition for languages with limited resources," arXiv preprint arXiv:2309.08535, 2023.
23. M. Liu, T. Zhang, and S. Gupta, "GLip: A global-local integrated progressive framework for lip reading," arXiv preprint arXiv:2509.16031, 2025.
1. 24. Nannuri S., Gantla H. R., Ananya D., Vennela A., Bhuvana B., Neha G. (2026). AUTOMATED BRAIN TUMOR SEGMENTATIONIN MRI VIA U-NET-BASED DEEP NEURAL NETWORKS. International Journal of Engineering Sciences & Research Technology, 15(4), 1–8. https://doi.org/10.64149/j.ijesrt.15.4.1-8
24. F. G. Ramirez, "A review of modern architectures for visual speech recognition," TechRxiv, 2025.
2. 26. Patel V., Suthar C., Patel D., (2026). Artificial Intelligence Investment As A Strategic Signal: Implications for Stakeholder Confidence. International Journal of Engineering Sciences & Research Technology, 15(4), 32-46. https://doi.org/10.64149/j.ijesrt.15.4.32-46
25. R. Singh and A. Patel, "Research on Tibetan lip-reading recognition methods based on deep learning," in Proc. Int. Conf. Comput. Linguistics (COLING), 2025, pp. 112-118.
26. J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 341-357.