Received 22 June 2024, accepted 24 July 2024, date of publication 1 August 2024, date of current version 20 August 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3436931

RESEARCH ARTICLE

LipSyncNet: A Novel Deep Learning Approach for Visual Speech Recognition in Audio-Challenged Situations

S. A. AMUTHA JEEVAKUMARI AND KOUSHIK DEY
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
Corresponding author: S. A. Amutha Jeevakumari (amutha.jeevakumari@vit.ac.in)

ABSTRACT In recent lip-reading technologies, deep learning methodologies have emerged as the key, transcending the limitations of traditional hybrid Deep Neural Network-Hidden Markov Model (DNN-HMM) frameworks based on Discrete Cosine Transform (DCT) features. LipSyncNet comprises a three-dimensional-Convolutional Neural Network (3D-CNN) that consists of a maximum depth of four layers and is responsible for extracting visual features by integrating EfficientNetB0, which results in excellent feature extraction capabilities. Following this, the network architecture incorporates a backend that utilizes a Bidirectional Long Short-Term Memory (Bi-LSTM)—a component of the recurrent neural network family—combined with Connectionist Temporal Classification (CTC) loss, enhancing its ability to perform classification tasks. The effectiveness of the proposed method is demonstrated through the evaluation of the Graphics Research International Database (GRID) corpus, a challenging word-level lip-reading dataset. Initially, facial features are extracted from the mouth area of an individual's face. Subsequently, these features are combined with available audio information to identify spoken words precisely. The lip-reading method aims to create a system that achieves accurate speech recognition by observing visual cues, thereby reducing the reliance on audio. The model utilizes information from various levels in a unified structure, enabling it to differentiate between words that sound alike and to improve its ability to handle changes in physical appearance.

INDEX TERMS Deep learning, bidirectional long short-term memory, long-short-term memory, visual cues, lip reading, 3D convolutional neural network, connectionist temporal classification.

I. INTRODUCTION

LIP reading involves the intricate process of interpreting spoken language solely through visual cues alone, such as lip movements, facial expressions, and the visibility of tongue and teeth movements. It is a cornerstone in enhancing communication, particularly when hearing information is absent or compromised. This visual component of speech, which is pivotal in human interaction, is essential in various applications, including aids for the hard of hearing, transcription of silent films, and support for speech recognition systems in noise-polluted environments. Visual Speech Recognition (VSR) involves interpreting spoken language using these visual cues. This method is particularly useful in noisy environments where audio cues are not reliable. Deep learning technologies have significantly advanced this field, enabling more sophisticated and accurate feature learning and extraction from visual speech data. However, the inherent challenge in this domain, underscored by phenomena such as the McGurk effect, lies in the ambiguity of visual speech, particularly with homophones, which are characters that produce similar lip movements despite producing distinct sounds. This emphasizes the importance of contextual and temporal analyses in word-level lip reading. This method uses visual information sequences to clarify speech and enhance the accuracy of machine-lip-reading systems. Owing to these

The associate editor coordinating the review of this manuscript and approving it for publication was Zeev Zalevsky.

factors and associated advancements in deep learning that facilitate effective learning and extraction of features, interest in lip reading has increased significantly in recent years [21], [22], [23], [24], [25].

A standard approach to lip reading involves analyzing the movements captured in a sequence of images and translating these visual data into words or sentences. One of the main challenges encountered in this process includes various imaging conditions such as inadequate lighting, pronounced shadows, motion blur, low image quality, and perspective distortion. Moreover, a critical inherent limitation affecting the accuracy is the presence of homophones. These words or phrases are distinct but involve identical or similar lip movements. For instance, in English, the phonemes "p" and "b" look the same visually, making words like "pat" and "bat" challenging to differentiate in lip reading without additional contextual information. This makes it difficult for humans to read the lips accurately, and previous studies have shown that humans can only achieve approximately 20% accuracy in lip reading [26].

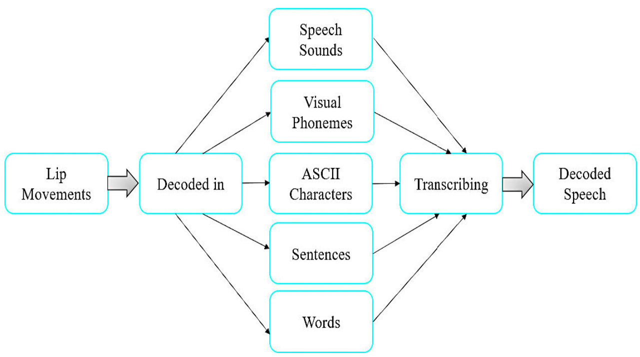

FIGURE 1. Overview of different classification schemas for lip movement interpretation.

The interpretation of lip movements varies, leading to the development of various classification frameworks within the field, such as categorizing phonemes or visemes, as referenced in [12] and [28]. Figure 1, a multitude of interpretations are assigned to lip movements, and various classification approaches are applied in the analysis of lip-reading. In the context of classification, back-end systems for lip reading are specifically designed to recognize sequential speech elements, such as words or sentences, which intrinsically possess a sequential structure. To process these sequences effectively, such systems typically employ sequence-processing neural networks, particularly the Recurrent Neural Network (RNNs).

To address specific problems, a model is being developed that utilizes fine- and coarse-grained spatiotemporal features to enhance its ability to distinguish between different inputs and increase its robustness. The proposed solution is a multi-grained spatiotemporal network explicitly designed for lipreading tasks. The network's front-end comprises a four-layer deep 3D-CNN that serves as the foundation for extracting visual features and is significantly enhanced by incorporating EfficientNetB0 to achieve unparalleled feature extraction capabilities. At its core, the model efficiently processes visual data to capture intricate details from visual input. Following this initial feature extraction phase, LipSyncNet incorporates an advanced back-end with a Bi-LSTM network. This element is designed to focus on processing the sequential characteristics of visual information, employing CTC loss for precise character classification. This dual-phase approach, which combines the depth and precision of 3D-CNNs with the temporal processing power of Bi-LSTM networks, enables LipSyncNet to capture and analyze visual cues with high fidelity and map these cues to specific characters, facilitating accurate speech recognition from visual information alone.

This paper's structure is outlined as follows: In Section II, a review of relevant work is presented, while Section III elucidates the methodology of the 3D-CNN-EfficientNetB0-Bi-LSTM network. Section IV showcases the experimental results and analysis, and lastly, Section V offers the paper's conclusion.

II. RELATED WORK

This study [1] reviews methods to improve speech recognition in noisy environments, highlighting the importance of visual cues. It traces the evolution from traditional techniques to advanced deep learning models incorporating lip reading. The research introduces a multi-head Key-Value (K-V) memory model and a joint cross-modal fusion model, tested on the LRS2 dataset, significantly reducing Word Error Rate (WER) compared to baseline models. The study demonstrates the proposed models' superior performance and competitive advantage in various noise conditions.

A cutting-edge lip-reading system has significantly advanced Automatic Speech Recognition (ASR) technology for noisy environments. The research analyzed cloud-based speech recognition services from major tech firms, using Google's Voice Command Dataset and improving keyword detection by integrating Microsoft's API with Google's Word2Vec. They introduced a unique lip-reading architecture combining three types of CNNs, achieving an average accuracy rate of 14.42% for Open Communication Speech Recognition (OCSR) APIs. This demonstrates the potential of combining audio and visual information to enhance ASR accuracy in noisy settings [2].

A groundbreaking method [3] for lip-reading markedly increases the precision in identifying words, phrases, and sentences from mute video clips by employing a dual-stream visual front-end network alongside a Dynamic Semantical-Spatial-Temporal Graph Convolutional Network (DST-GCN). This method excels in capturing appearance and motion and effectively modeling mouth dynamics. The Adaptive Semantic Segmentation Transform with Graph Convolutional Networks (ASST-GCN) module's learning of semantic and spatiotemporal relationships leads to superior performance on key datasets, achieving remarkable accuracy and word error rate improvements, showcasing its potential to transform lip-reading practices.

A new lip-reading method [4] was introduced to synthesize 3D convolution with vision transformer technology to enhance the machine's lip-reading capabilities. This method integrates the extraction of spatiotemporal features with the advantages of both convolutions and transformers and proceeds with sequence analysis through a Bidirectional Gated Recurrent Unit (Bi-GRU). It reached unprecedented accuracy levels of 88.5% on the Lip Reading in the Wild (LRW) dataset and 57.5% on the naturally-distributed large-scale benchmark for LRW-1000 dataset, showcasing its capacity to substantially improve lip-reading precision and offer significant enhancements across a range of applications.

The Deep Multimodal Contrastive Learning for Retrieval (DMCLR) dataset, introduced by Haq et al., includes 1,000 video clips from 100 everyday dialogues by 10 individuals. Their lip-reading model achieved 94.2% accuracy on the DMCLR dataset, using a spatial and temporal convolution layer combined with an SE-ResNet-18 framework and a backend with Bi-GRU layers, 1D convolutional layers, and fully connected layers. This approach outperformed previous models on the DMCLR dataset and showed strong results on the LRW and LRW-1000 datasets, highlighting its effectiveness in predicting everyday Mandarin conversations [5].

An innovative Cantonese Lip Reading Dataset (CLRW) with 800 words and 400,000 instances has advanced lip-reading technologies. The Two-Branch Global-Local (TBGL) model integrates a global branch for spatial information and a local branch for lip motions using bidirectional knowledge distillation loss. TBGL achieved 88.4% accuracy on the LRW dataset and 49.1% on the CAS-VSR-W1K dataset, surpassing state-of-the-art performance on CAS-VSR-W1K and matching it on LRW. This work improves lip-reading accuracy and efficiency through a substantial dataset and novel architecture [6].

This method [7] combines lip segmentation with word lip reading using a hybrid active contour model for precise mouth region identification and a CNN-Bi-GRU combination for feature extraction and classification. Tested against the LRW dataset, it achieves a remarkable 90.38% recognition accuracy, setting a new benchmark in word recognition from mouth sequences and demonstrating the potential for broader applications.

Automated lip-reading techniques that employ deep learning methodologies underwent comprehensive assessment, revealing the importance of CNNs and RNNs for feature extraction and classification with attention-based transformers and Temporal Convolutional Networks (TCNs) as alternatives for classification. Exploration of various datasets highlights the need for extensive data for model training and testing. Comparative analysis shows transformers' superiority in sentence-level lip-reading and TCNs' effectiveness in managing long-term dependencies, indicating the evolving landscape of lip-reading technologies [8].

A novel approach [9] utilizing Deep Convolutional Neural Networks (DCNNs) has effectively identified adversarial attacks on audio-visual speech recognition systems, focusing on LRW and GRID datasets. This method outperformed existing systems in detecting adversarial attacks with high precision, recall, accuracy, and F1-score metrics. This enhances the security and reliability of audiovisual speech recognition technologies by improving adversarial attack detection.

In 2021, Fenghour and team [10] delved into a deep-learning framework aimed at tackling the task of transforming visemes into words within automatic lipreading systems. Identifying a critical limitation, their research introduces a framework utilizing an attention-enhanced GRU, which boosts the accuracy of word recognition to 79.6%, achieving a notable enhancement of 15.0% over earlier methodologies. This strategy significantly betters the efficiency of converting visemes to words while diminishing both training and operational durations, indicating a promising avenue for advancements in automatic lipreading technologies.

Hybrid Lip-Reading Network(HLR-Net) [11] introduced a novel hybrid lip-reading model designed to enhance communication accessibility for hearing impairments by translating video-captured lip movements into subtitles. This model combines preprocessing, an encoder with inception, gradient, and Bi-GRU layers, and a decoder with attention mechanisms and CTC, thereby achieving superior performance on the GRID corpus dataset, registering a Character Error Rate (CER) of 4.9%, a WER of 9.7%, and a Bleu score of 92% for speakers not previously encountered., and even better results for overlapped speakers, HLR-Net significantly advances lip movement transcription and subtitle generation, improving accessibility for those with hearing impairments.

Previous systems have focused on classifying isolated speech segments and transitioning to entire sentences, with accuracy challenges. Fenghour et al. [12] proposed a neural network-based lip-reading system that classifies speech into visemes, showing significant improvements in word accuracy on the British Broadcasting Corporation (BBC)-LRS2 dataset. This system, which is robust to illumination changes, represents a novel approach for lip-reading sentences, demonstrating promising results and advancing the research area.

Leveraging a fuzzy convolutional neural network [13] achieves breakthrough accuracy in lip image segmentation, especially in complex scenarios. This approach combines fuzzy logic with CNNs to handle uncertainties and learn features effectively, achieving a 98.4% accuracy on a diverse dataset. This fusion represents a significant advancement in lip reading and visual speaker authentication, enhancing the under challenging conditions.

Deep learning models, such as 2D-CNNs, 3D-CNNs, and hybrid 3D-2D CNN systems, are essential for enhancing lipreading accuracy. Studies using datasets like LRW and OuluVS2 show that deep learning frameworks, particularly CNN and LSTM combinations, outperform traditional methods in metrics like Word Recognition Rate (WRR) and sentence accuracy rate. Additionally, a model combining residual networks with LSTM for audiovisual speech recognition, trained on the LRS3-TED dataset, significantly reduces WER compared to baseline models. These advancements demonstrate the effectiveness of deep learning in improving the accuracy and efficiency of lipreading and audiovisual speech recognition by leveraging visual lip movement signals [14].

In advancing lip-reading technology, an innovative model [15] employing TCN replaces Bi-GRU layers, enhancing the training efficiency. The LRW and a LRW-1000 datasets show accuracy increases of 1.2% and 3.2%, respectively, and variable-length augmentation is used to improve generalization. This model represents a significant step forward by offering a robust and simplified solution for real-world applications in which audio cues are compromised.

In the study [16], an advanced model for audiovisual speech recognition was developed to decode spoken words and phrases by observing facial movements. The research compared two approaches: one using CTC loss and another using a sequence-to-sequence (seq2seq) strategy with the transformer's self-attention mechanism. Tested on the new BBC-LRS2 dataset, these models outperformed previous efforts in standard lip-reading evaluations, significantly reducing word error rates. This highlights the advantage of integrating auditory and visual cues for speech recognition, especially in noisy environments, with practical applications in transcription and automated speech recognition systems.

Mudaliar et al. introduced a deep learning methodology using the ResNet framework with 3D convolutional layers for encoding and GRU for decoding, trained on the LRW dataset. The model achieved a 90% accuracy rate on the BBC dataset and 88% on their own dataset, outperforming existing models. The study suggests further improvements by adding more variety to the dataset and exploring the impact of facial hair on model performance.

The paper [18] introduces the Densely Connected Temporal Convolutional Network (DC-TCN) for lip-reading isolated words. To improve existing TCNs, the authors added dense connections and a Squeeze-and-Excitation block for better classification. The DC-TCN achieved state-of-the-art performance with 88.36% accuracy on the LRW dataset and 43.65% on the LRW-1000 dataset, surpassing all baseline methods.

This study [19] proposed a deep learning framework for lip-reading to enhance speech clarity amidst background noise. The framework consists of a deep learning-based lip-reading regression model to calculate clean audio features and an enhanced visually derived Wiener filter for speech enhancement. Tested on the AV ChiME3 corpus, which includes real-world dynamic noise, this method outperformed standard audio-centric techniques like spectral subtraction and LMMSE. The authors also discuss ongoing efforts to improve the lip-reading framework's precision and suggest future research into context-sensitive, autonomous audio-visual speech amplification.

Howell et al. [20] presented a method for automatic lip-reading using visual units and confusion modeling with Weighted Finite-State Transducers (WFSTs) to address missing visual speech information. Tested on ISO-211 and RM-3000 datasets, the approach showed a significant improvement in recognition accuracy over conventional systems, with a word accuracy rate exceeding 76%. The study also evaluated the limitations of visemes due to lexical ambiguity and decreased accuracy, highlighting the potential of WFSTs combined with confusion modeling to enhance lip-reading accuracy.

III. METHODOLOGY

The methodology for our lip-reading detection system unfolds into four critical stages, ensuring a comprehensive approach from data collection to deployment. The process began with acquiring a video dataset that laid the foundation for our study.

Afterwards, the dataset was refined to guarantee that it was in the most suitable format for analysis. Subsequently, the development of the Model Architecture takes place, which is crucial in determining the efficacy of the lip-reading detection system. The system was evaluated with great attention to detail as we compared the output letter by letter to the actual output. This meticulous approach ensured the accuracy and reliability of our model. The final step in our methodology is the deployment of the model, which marks the culmination of our development process and the beginning of its application to real-world scenarios. In addition, our system design and testing were rigorously applied to video frames to ensure robustness and efficiency.

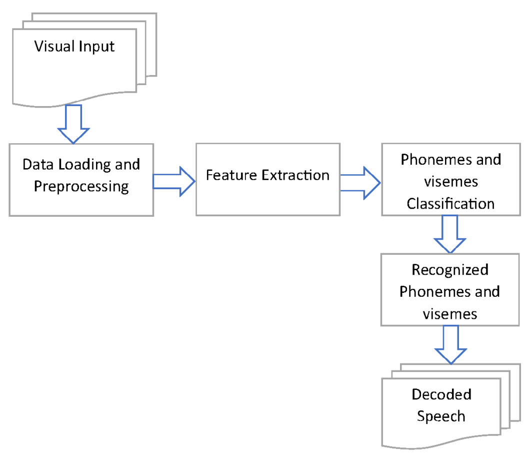

FIGURE 2. Framework of lip-reading system.

Figure 2 outlines the deep-learning process of lip reading, and the lip Region of Interest (ROI) was initially identified and extracted from the video. This was followed by the extraction of deeper features from the experimental data, which is a crucial step in understanding the nuances of lip movements. Subsequently, the temporal and spatial features were extracted from the front end of the system. These extracted features are then fed into the back-end and concatenated at each time step for classification. This framework in Figure 2 highlights the sequential steps in our methodology and underscores the importance of each phase in achieving an accurate lip-reading detection.

A. DATASET

The GRID corpus, introduced by Cooke et al. in 2006, stands out in lipreading datasets with rich audio and video-recording collections. This dataset is notable for its depth, featuring 34 speakers, each delivering 1,000 sentences, resulting in 28h of video containing 34,000 sentences. The GRID Corpus surpasses many other datasets focused on single words and is limited in size, offering comprehensive coverage with significantly advanced lipreading performance benchmarks. Little demonstration of the GRID Corpus dataset in Figure 3 has shown below.

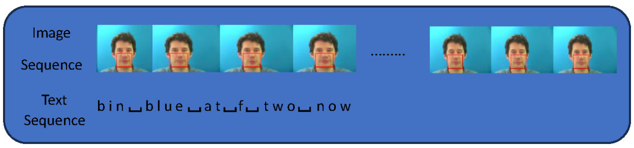

FIGURE 3. **Examples for lip-reading. The example is randomly sampled from the GRID dataset.**

To evaluate LipSyncNet, the GRID corpus is crucial, as it offers sentence-level complexity and a wealth of unmatched data. The sentences follow a structured grammar across six categories: command, color, preposition, letter, digit, and adverb, enabling the generation of 64,000 potential sentences. This structure allows for a diverse range of sentence combinations, such as "set blue by A four please," "place red at C zero again," and "set blue with H seven again," thanks to a consistent six-word sentence format and a lexicon of 51 unique words. The dataset, with 33,000 samples available after excluding one speaker who provided only voice data, is a critical resource for in-depth lipreading research and provides a controlled setting for a comprehensive analysis. Delving deeper into the GRID audio-visual dataset, each sentence was carefully structured to follow the sequence of "Command + Color + Preposition + Letter + Digit + Adverb," for instance, "set black into X four, please." The dataset features 51 unique words, categorized into four commands, four colors, four prepositions, 25 letters, 10 numbers, and four adverbs. Sentences are constructed randomly from these categories, with each spoken phrase lasting three seconds. With an extensive vocabulary of 33,000 utterances and 33 speakers, the GRID corpus is a comprehensive and invaluable asset for propelling the field of lipreading studies and technologies.

B. DATA PREPROCESSING

In audio-visual speech recognition, preprocessing video data to focus on essential features such as lip movements is paramount, as it enables the model to disregard irrelevant data such as the background. This preprocessing step as shown in Figure 4, which is crucial for enhancing system performance, involves loading video data using the dlib library, extracting frames, converting their color from RGB to grayscale with TensorFlow, and cropping to the lip region. Following this, frames are normalized and scaled by calculating the mean and standard deviation, respectively, ensuring that the model remains unbiased towards any specific video or frame, thus improving accuracy and generalization.

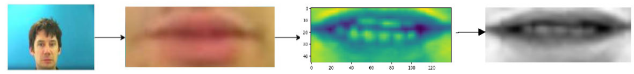

FIGURE 4. **Preprocessing of videos.**

In preparation for our deep learning model using Keras, we created a lexicon comprising 40 characters: including space, the alphabet (a-z), punctuation marks ('?', '!', and an apostrophe), and numbers (1-9), where each symbol is assigned an index ranging from 1 to 40 for mapping purposes. This meticulous vocabulary setup is crucial for accurately mapping the numerical representations of characters during the model-training process.

Building upon this foundation, we meticulously prepared our dataset by implementing functions to map characters to numbers and vice versa, carefully loading the dataset with a focus on alignment. Entries marked 'sil' for silence were excluded, and the remaining characters were converted into numerical representations. This meticulous preparation extends to establishing a data pipeline that is crucial for managing data flow during training and allows TensorFlow to randomly select samples, thereby enhancing the learning diversity of the model.

FIGURE 5. **Mouth region crop.**

Moreover, we emphasize the segmentation of the mouth region in the images, as shown in Figure 5, which is a critical step in decoding visual speech models. This static segmentation ensured the model focused solely on the oral region. Utilizing Imageio and the 'mimsave' function, we created animated Graphics Interchange Format(GIFs) of mouth movements to provide a dynamic dataset from which our model can learn. These animations enable our model to accurately interpret and replicate speech-associated mouth movements, highlighting its potential for speech recognition and animation applications. This integrated approach, from pre-processing to dataset preparation and model training, underscores our commitment to advancing the capabilities of audio-visual speech recognition technologies.

C. MOUTH SEGMENTATION

The mouth segmentation phase is crucial in this approach to visual speech recognition, especially given the variability in mouth shapes and the inclusion of teeth or tongue. To address these challenges, we have implemented several robust techniques and conducted extensive validations, as detailed below.

1. Segmentation Technique: This methodology employs precise mouth region segmentation, utilizing the dlib library for accurate facial landmark detection. This step ensures consistent isolation of the mouth region across different individuals and scenarios. The extracted region is resized to 140 × 46 pixels to match the input dimensions expected by our model's CNN layers.

2. Handling Variability: To handle variability in mouth shapes and the presence of teeth and tongue, our preprocessing pipeline involves converting video frames to grayscale, normalizing, and scaling them. We achieve this by calculating the mean and standard deviation of the frames, ensuring the model remains unbiased towards specific frames and focuses on relevant features.

3. Data Augmentation: We employ data augmentation techniques to simulate various real-world conditions, including different lighting scenarios, angles, and mouth configurations. This allows the model to learn to generalize better, enhancing its robustness against variations in mouth appearance.

4. Dynamic GIF Creation: Utilizing Imageio and the 'mimsave' function, we create dynamic GIFs of mouth movements. This approach allows the model to capture the temporal dynamics of speech, learning from animated sequences that include a range of mouth shapes and movements, such as the visibility of teeth and tongue.

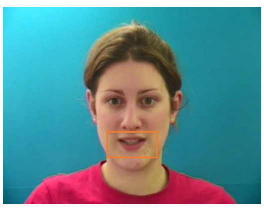

FIGURE 6. Prominent teeth visibility.

In one example, a subject with prominently visible teeth during speech was used to assess the model's performance. The model accurately segmented the mouth region and correctly interpreted the lip movements without being affected by the teeth' visibility in Figure 6. The dynamic GIFs enabled the model to learn and generalize from these variations effectively, demonstrating its robustness against the inclusion of teeth in the visual data.

This model was extensively validated using the GRID Corpus, featuring a diverse range of speakers and mouth shapes.

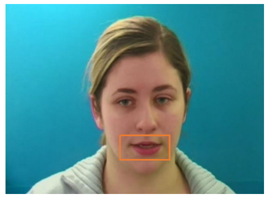

FIGURE 7. Significant tongue and teeth movements.

The results demonstrated high accuracy in recognizing lip movements, showcasing the model's robustness in various scenarios. The inclusion of teeth or tongue did not significantly impact the model's performance in Figure 7, validating the effectiveness of this approach.

D. FEATURE EXTRACTION

In the proposed LipSyncNet model, the feature extraction phase is crucial for accurately interpreting lip movements from video frames. The model utilizes a combination of 3D-CNNs and EfficientNetB0 to capture both spatial and temporal features effectively.

1. The input to the model is a sequence of video frames (75 frames, each of size 46 × 140 pixels in grayscale). The initial layers of the model use 3D convolutions to process the volumetric data, which captures the temporal dynamics across the sequence of frames. The 3D-CNN consists of multiple convolutional layers with batch normalization and max-pooling layers. These layers extract spatiotemporal features by considering the temporal changes and spatial patterns within the video frames. Specifically, the model has four conv3d, each followed by batch normalization and max-pooling to reduce the spatial dimensions and retain the essential features.

2. The output from the 3D-CNN layers is then prepared for EfficientNetB0 by converting it into a suitable format. This involves, using a TimeDistributed layer to apply the same operation to each frame in the sequence independently, adjusting the single-channel (grayscale) output from the 3D-CNN to a three-channel format expected by EfficientNetB0. This is achieved by repeating the grayscale channel, and resizing the frames to 224 × 224 pixels, which is the input size required by EfficientNetB0. EfficientNetB0, pre-trained on the ImageNet dataset, is then applied to each frame to extract high-level features. EfficientNetB0 is known for its efficient feature extraction capabilities due to its compound scaling method that balances network depth, width, and resolution.

3. The features extracted by the 3D-CNN and EfficientNetB0 are concatenated to form a unified feature representation. This combination leverages the temporal features from the 3D-CNN and the high-level spatial features from EfficientNetB0, providing a comprehensive representation of the input video sequence.

4. The concatenated features are then fed into Bi-LSTM layers. Bi-LSTM is effective in learning the temporal dependencies in both forward and backward directions, which enhances the model's ability to capture the sequential nature

of lip movements. The Bi-LSTM layers process these features over time, producing an output that captures the temporal dynamics of the sequence.

E. MODEL ARCHITECTURE

This model was designed for lip reading from sequences of video frames, employing a sophisticated architecture that integrates conv3d, EfficientNetB0 for feature extraction, and Bi-LSTM layers for temporal sequence processing. The input to the model was a sequence of 75 frames, each with a 46 × 140 pixel grayscale image. The model processes these inputs through a series of layers, shown in the architecture Figure 8 and block diagram Figure 9.

1. 3D-CNN for Volumetric Data Processing: The initial layers of the model use 3D convolutions to process volumetric or sequential input data. This is particularly useful for tasks that involve temporal sequences or 3D spatial data, where capturing the relationships across three dimensions is crucial.

2. Preparation for EfficientNetB0: Before passing the output of the 3D-CNN layers to EfficientNetB0, the data format must be adjusted. EfficientNetB0 expects 2D images (height × width × channels).

3. The 3D-CNN output is flattened or pooled to reduce it to 2D if necessary. TimeDistributed layers are used to apply the same 2D operation (such as resizing and repeating) across each time step independently.

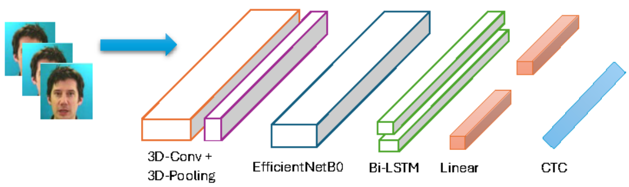

FIGURE 8. Architecture of the proposed 3D-EfficientNetB0-Bi-LSTM-CTC network.

4. The images are resized to the input size expected by EfficientNetB0 (224 × 224 pixels) and the channel dimension is adjusted(repeating the single channel to match the three-channel input expected by EfficientNetB0). EfficientNetB0 for Feature Extraction: EfficientNetB0, pre-trained on a large dataset, such as ImageNet, extracts rich features from 2D images. The top layers of EfficientNetB0 can be fine-tuned to the task while keeping the earlier layers frozen to leverage the pretrained weights.

5. Combining Features for Final Prediction: The features extracted by the 3D-CNN and EfficientNetB0 components can be combined using concatenation or another method to make the final predictions. This approach allows the model to leverage the deep spatiotemporal features extracted by the 3D-CNN and high-level features extracted by Efficient-NetB0.

The dense layer used for classification played a crucial role in the prediction process of the proposed model. We have incorporated a unique loss function CTC, which is particularly effective for processing word transcripts that do not directly correspond to frames. This approach also helps to minimize and eliminate repetitive predictions.

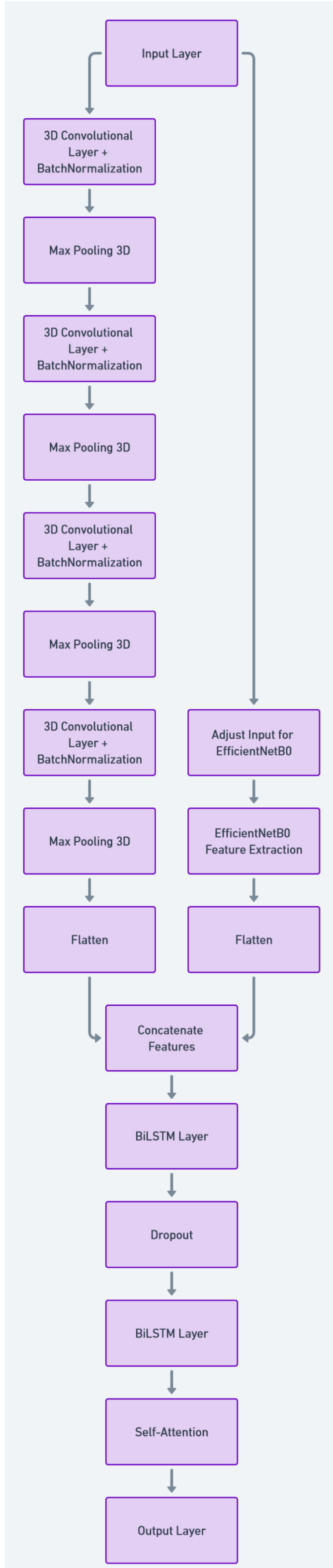

FIGURE 9. Block Diagram of the proposed model.

TABLE 1. Model structure.

Layer (type)	Output Shape	Param #
Input Layer	(None, 75, 46, 140, 1)	0
3D-CNN_1 + Activation	(None, 75, 46, 140, 64)	1,792
BatchNormalization_1	(None, 75, 46, 140, 64)	256
MaxPool3D_1	(None, 75, 23, 70, 64)	0
3D-CNN_2 + Activation	(None, 75, 23, 70, 128)	221,312
BatchNormalization_2	(None, 75, 23, 70, 128)	512
MaxPool3D_2	(None, 75, 11, 35, 128)	0
3D-CNN_3 + Activation	(None, 75, 11, 35, 256)	884,992
BatchNormalization_3	(None, 75, 11, 35, 256)	1,024
MaxPool3D_3	(None, 75, 5, 17, 256)	0
3D-CNN_4 + Activation	(None, 75, 5, 17, 512)	3,539,456
BatchNormalization_4	(None, 75, 5, 17, 512)	2,048
MaxPool3D_4	(None, 75, 2, 8, 512)	0
TimeDistributed(Flatten) for 3D-CNN	(None, 75, 8192)	0
EfficientNetB0 Input Adjustment	(None, 75, 46, 140, 3)	0
TimeDistributed for EfficientNet (Resize)	(None, 75, 224, 224, 3)	0
EfficientNetB0 Feature Extraction	(None, 75, 7, 7, 1280)	4,049,361
TimeDistributed(Flatten) for EfficientNetB0	(None, 75, 62720)	0
Concatenate	(None, 75, 70912)	0
Bi-LSTM_1 + Dropout	(None, 75, 1024)	5,971,968
Bi-LSTM_2 + Dropout	(None, 75, 1024)	10,485,760
TimeDistributed(Dense) + Activation	(None, 75, vocabulary_size+1)	Dependent on vocabulary_size

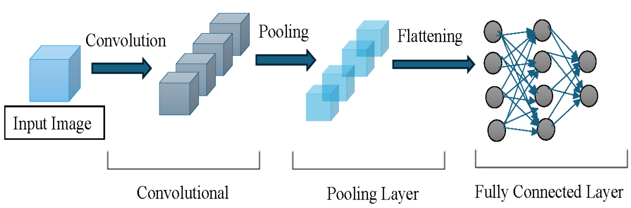

FIGURE 10. Diagram illustrating the workflow of a 3D-CNN.

In Table 1 there are 307,595,340 total parameters, of which 304,824,800 are trainable, and 2,780,531 are nontrainable, likely frozen during training to retain previously learned features or owing to the Batch Normalization layers.

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, which utilize images to discern and categorize objects, benefit from the essential functionality of 2D Convolutional Layers (conv2d) in processing the image's channel Z.

For an input u and a set of weights v, defined within the space $\mathbb{R}^{du\times hu\times wu}$, the 2D convolution operation is mathematically expressed as Eq. (1):

$$\mathrm{conv2d}(u, v)_{s,t} = \sum_{z=1}^{z}\sum_{a=1}^{hv}\sum_{b=1}^{wv} v_{z,a,b}\cdot u_{z,s+a,t+b} \tag{1}$$

Here, s and t denote the spatial dimensions of the output.

Extending this framework, a conv3d incorporates an additional dimension into the convolution process, as depicted in Figure 10, which outlines the operational framework of a 3D-CNN. The 3D convolution is defined by the following Eq. (2):

$$\mathrm{Conv3d}(u, v)_{s,t,u} = \sum_{z=1}^{z}\sum_{a=1}^{hv}\sum_{b=1}^{wv}\sum_{c=1}^{dv} v_{z,a,b,c} \cdot u_{z,s+a,t+b,u+c} \tag{2}$$

In this equation, the introduction of u as a new dimension accommodates the inclusion of either temporal dynamics or depth information in the data and weights, offering a more detailed analysis capability for data exhibiting temporal changes or volumetric characteristics.

2) EfficientNetB0

EfficientNetB0 introduces a novel scaling methodology in neural networks, focusing on a balanced enhancement across network width, depth, and resolution dimensions. This model innovates with a compound scaling coefficient, denoted by θ, to modulate the network's dimensions in a unified manner. The objective is to optimize model performance and efficiency in image classification tasks without disproportionately increasing computational demands.

The essence of EfficientNetB0's scaling strategy is encapsulated in the following relations, which describe how each dimension scales for θ:

$$a = \delta^{\theta} \tag{3}$$
$$b = \epsilon^{\theta} \tag{4}$$
$$c = \zeta^{\theta} \tag{5}$$

where δ, ϵ, and ζ are constants that guide the scaling of depth, width, and resolution respectively. These parameters were fine-tuned through an exploratory search within the initial EfficientNetB0 framework.

Central to EfficientNetB0 is the Mobile Inverted Bottle-neck Convolution (MBConv) block, denoted as MBConv(y), where y is the input to the block. The architecture unfolds through a succession of MBConv blocks of varying dimensions, leading to global average pooling and a dense classification layer. The architectural flow can be delineated as:

$$\mathrm{EfficientNetB0}(y) = \mathrm{Dense}(\mathrm{GlobalAvgPooling}(\mathrm{MBConv}_m(\ldots(\mathrm{MBConv}_1(y))\ldots))) \tag{6}$$

as shown in Eq. (6) here, $\mathrm{MBConv}_1, \ldots, \mathrm{MBConv}_m$ represent the sequence of MBConv blocks, culminating in a Dense layer for classification. This approach underlines the model's commitment to scaling efficiency, ensuring that improvements in capacity are matched by gains in computational efficiency.

3) BIDIRECTIONAL LONG SHORT-TERM MEMORY NETWORK

Bi-LSTMs are an advancement of the standard LSTM model, designed to improve the processing of sequential data by utilizing both previous (backward) and upcoming (forward) contexts. This dual-directional approach allows Bi-LSTMs to grasp dependencies from both directions, enhancing prediction accuracy for sequences.

Illustrated in Figure 11 is the structure of the Bi-LSTM network.

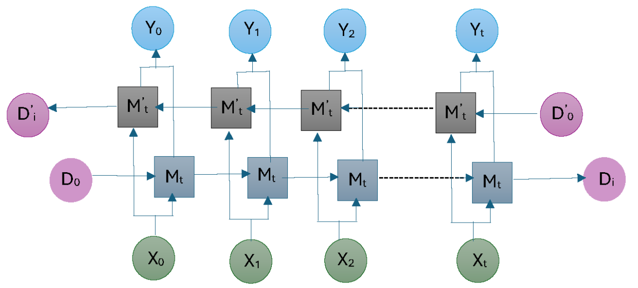

FIGURE 11. **Architecture of the Bidirectional-LSTM layer.**

Bi-LSTM networks perform computations based on a series of mathematical operations that guide the information flow through the network. These operations are reformulated as:

$$g_t = \sigma(U_{gx}X_t + U_{gh}M_{t-1} + U_{gc}D_{t-1} + d_g) \quad (7)$$
$$q_t = \sigma(U_{qx}X_t + U_{qh}M_{t-1} + U_{qc}D_{t-1} + d_q) \quad (8)$$
$$D_t = q_t \odot D_{t-1} + g_t \odot \tanh(U_{dx}X_t + U_{dh}M_{t-1} + d_d) \quad (9)$$
$$p_t = \sigma(U_{px}X_t + U_{ph}M_{t-1} + U_{pc}D_t + d_p) \quad (10)$$
$$M_t = p_t \odot \tanh(D_t) \quad (11)$$

as shown in Eq. (7) through (11). Here, σ represents the sigmoid function, and g_t, q_t, D_t, p_t, and M_t denote the input gate, forget gate, cell state, output gate activations, and hidden state at time t, respectively. The U matrices and d vectors stand for the updated weights and biases, fine-tuned through the training phase.

Following processing in the Bi-LSTM layers, the sequence is subjected to a linear transformation and then passed through a softmax activation to derive the final output. The probability distribution for a sequence of length T is elucidated as:

$$r_t = \text{Linear}(M_t) \quad (12)$$
$$y_t = \text{softmax}(r_t) \quad (13)$$

as shown in Eq. (12) and (13). In this context, y_t represents the model's output at time t, indicating a probability vector across the designated class set. For tasks such as character recognition, y_t would display probabilities for all potential characters, including a unique 'blank' symbol for instances where no character is recognized.

4) CONNECTIONIST TEMPORAL CLASSIFICATION

The CTC function, initially prominent in speech recognition, has been effectively adapted for lip-reading owing to its proficiency in processing temporal sequences. The CTC methodology is geared towards mapping sequences of variable lengths to their corresponding labels without requiring predetermined alignment.

CTC leverages the network's output, transforming it into a probability distribution over possible label sequences. This process involves the integration of a softmax layer with additional units designated for a special blank label, expanding the original label set $\mathcal{L}'$.

Given an input sequence z of length U, a network characterized by a Bi-LSTM is employed. This network, with inputs j, outputs p, and weight matrix v, facilitates the mapping of an input sequence to an output sequence as described in Eq. (14):

$$\mathcal{M}_v : (\mathbb{R}^q)^U \rightarrow (\mathbb{R}^r)^U \quad (14)$$

where $z = \mathcal{M}_v(x)$ outputs.

The probability of observing a specific label sequence is computed using Eq. (15):

$$q(\sigma|z) = \prod_{u=1}^{U} \mu_u^{\sigma_u}, \forall u \epsilon C^U. \quad (15)$$

The cumulative probability for a label sequence m is obtained by aggregating the probabilities across all paths, as shown in Eq. (16):

$$q(m|z) = \sum_{\sigma \in \gamma^{-1}(m)} q(\sigma|z) \quad (16)$$

Identifying the most probable labeling, $\hat{m}$ (z), involves maximizing the computed probabilities as defined in Eq. (17):

$$\hat{m}(z) = \arg\max_{m \in \mathcal{L}'^U} q(m|z). \quad (17)$$

The loss function for the CTC objective is formulated as shown in Eq. (18):

$$\eta(z) = -\ln \hat{m}(z). \quad (18)$$

The goal in training a CTC network revolves around minimizing the negative log-likelihood for the desired label sequence m, as specified in Eq. (19):

$$\eta_{ctc} = -\ln(q(m|z)). \quad (19)$$

This adaptation underscores the versatility of the CTC function, extending its application from speech recognition to the domain of lip reading and showcasing its effectiveness in temporal sequence analysis.

TABLE 2. Computation time analysis.

Epoch	Time per Epoch (seconds)	Cumulative Time (seconds)
1	1685.2	1685.2
2	1685.2	3370.4
3	1685.2	5055.6
4	1685.2	6740.8
5	1685.2	8426.0
...	...	...
100	1685.2	168520

F. TRAINING TIME ANALYSIS

The training process was monitored to evaluate the computational performance. The total time required to train the model for 100 epochs was calculated based on the average time per epoch, as shown in Table 2.

- Average Time per Epoch: Approximately 1685.2 seconds
- Total Computation Time for 100 Epochs shown in Eq (20):

$$100 \times 1685.2 \text{ seconds} = 168520 \text{ seconds} \tag{20}$$

Converting the total computation time into hours as shown in Eq. (21):

$$\frac{168520 \text{ seconds}}{3600 \text{ seconds/hour}} \simeq 46.81 \text{ hours} \tag{21}$$

Thus, the total computation time required for training the model over 100 epochs was approximately 46.81 hours.

We have used the following notations and definitions given in the notation Table 3.

IV. EXPERIMENT RESULTS AND ANALYSIS

A. IMPLEMENTATION DETAILS

In the code provided, ROI was extracted from the video frames using the facial landmarks of the library to identify the mouth region. This ROI was then resized to a consistent size of 140 × 46 pixels, which is the input size expected by the model's CNN layers. To accelerate the training process, the code preprocesses the dataset by detecting the ROI areas and saving them on the disk. Only the preprocessed ROI areas were loaded during training, which reduced the computational overhead of performing ROI detection at each training iteration.

The model uses a combination of conv3d and EfficientNetB0 for feature extraction, followed by Bi-LSTM layers for sequence modeling and a final density layer to output the probabilities of each character in the vocabulary. The model was trained using an Adaptive Moment Estimation (Adam) optimizer with an initial learning rate of 0.0001. Training involves feeding the preprocessed ROI data into the model and adjusting the model's weights based on the CTC loss function, which is designed to handle sequence prediction tasks such as lip reading. The code uses TensorFlow's native CTC beam search decoder to decode the model predictions into actual text. This decoder operates with a beam width of 100. Therefore, it considered the top 100 most likely sequences at each time step during the decoding process.

It also includes a custom callback to produce example predictions at the end of each epoch, which can help monitor the performance of the model during training. Additionally, the model weights are saved at regular intervals using ModelCheckpoint callback, allowing for the resumption of training if needed.

B. ABLATION STUDIES

In this part of the content, we conduct ablation studies to assess the effectiveness of individual components. As a result, we have developed two models, namely 3D-CNN-Bi-LSTM-CTC and 3D-CNN-EfficientNetB0-Bi-LSTM-CTC, and we have performed performance comparisons between them.

Figure 12 displays a visual representation illustrating the variant models available for comparison. The term "variant" refers to different versions or configurations of the models being compared. Specifically, the paper compares two models: 3D-CNN-Bi-LSTM-CTC and 3D-CNN-EfficientNetB0-Bi-LSTM-CTC. We conducted ablation studies using the GRID Corpus dataset, in which each architecture was trained for 100 epochs. To illustrate these differences, we plotted curve graphs to showcase the training and evaluation losses for each epoch across iterations. From the ablation studies mentioned previously in Figures 13 and 14, it was determined that the 3D-EfficientNetB0-Bi-LSTM-CTC model, which employs the more sophisticated and efficient EfficientNetB0 for high-level feature extraction, can deliver top-tier performance in the GRID Corpus datasets. When evaluating the architectural designs of both 3D-CNN-Bi-LSTM-CTC and 3D-CNN-EfficientNetB0-Bi-LSTM-CTC, it is evident that the latter, with EfficientNetB0 as the feature extractor, provides more prosperous feature extraction, quicker convergence, and significantly improved performance. Both models utilize Bi-LSTM for feature learning.

Figures 13 and 14 illustrate the loss graphs for two different models during training, specifically designed for lip-reading tasks. Figure 13 depicts the loss graph for the 3D-CNN-Bi-LSTM-CTC model, while Figure 14 shows the loss graph for the 3D-CNN-EfficientNetB0-Bi-LSTM-CTC model. Both graphs plot the training and validation loss over 100 epochs, with the x-axis representing the number of epochs and the y-axis showing the loss value.

The training loss curves in both figures demonstrate a consistent decrease, indicating effective learning from the training data. This behavior is typical in deep learning, reflecting the models' ability to minimize error as they optimize their parameters. This is particularly important in lip-reading applications, where accurately capturing and distinguishing subtle variations in lip movements is crucial for recognizing speech.

The validation loss curves follow a similar decreasing trend but with slight fluctuations. These fluctuations can be attributed to the varying complexity of the validation data compared to the training data. A steady decrease in validation loss suggests good generalization to unseen lip-reading data.

TABLE 3. Notation table.

Notation	Section	Definition
u	3DCNN	Input to the convolutional layer in the convolution operation
v	3DCNN	Weights in the convolutional layer in the convolution operation
$\mathbb{R}^{d_u \times h_u \times w_u}$	3DCNN	The space within which the input u and weights v are defined
$\text{conv2d}(u, v)_{s,t}$	3DCNN	2D convolution operation (Equation 1)
s	3DCNN	Spatial dimension of the output (width)
t	3DCNN	Spatial dimension of the output (height)
z	3DCNN	Channel dimension, representing depth in the input data
h_v	3DCNN	Height of the convolutional filter
w_v	3DCNN	Width of the convolutional filter
$v_{z,a,b}$	3DCNN	Weight at position (z, a, b) in the convolutional filter
$u_{z,s+a,t+b}$	3DCNN	Input data at position $(z, s + a, t + b)$
$\text{conv3d}(u, v)_{s,t,u}$	3DCNN	3D convolution operation (Equation 2)
c	3DCNN	Additional dimension introduced in the 3D convolution (depth/temporal dimension)
d_v	3DCNN	Depth of the convolutional filter
$v_{z,a,b,c}$	3DCNN	Weight at position (z, a, b, c) in the 3D convolutional filter
$u_{z,s+a,t+b,u+c}$	3DCNN	Input data at position $(z, s + a, t + b, u + c)$
θ	EfficientNetB0	Compound scaling coefficient
a	EfficientNetB0	Depth dimension scaling factor
b	EfficientNetB0	Width dimension scaling factor
c	EfficientNetB0	Resolution dimension scaling factor
δ	EfficientNetB0	Constant guiding the scaling of depth
ϵ	EfficientNetB0	Constant guiding the scaling of width
ζ	EfficientNetB0	Constant guiding the scaling of resolution
$\text{MBConv}_1, \ldots, \text{MBConv}_m$	EfficientNetB0	Sequence of MBConv blocks
GlobalAvgPooling	EfficientNetB0	Global average pooling layer
Dense	EfficientNetB0	Dense layer for classification
g_t	Bi-LSTM	Input gate activation at time t
q_t	Bi-LSTM	Forget gate activation at time t
D_t	Bi-LSTM	Cell state at time t
p_t	Bi-LSTM	Output gate activation at time t
M_t	Bi-LSTM	Hidden state at time t
σ	Bi-LSTM	Sigmoid function
U_{gx}	Bi-LSTM	Weight matrix for input X_t in the input gate
U_{gh}	Bi-LSTM	Weight matrix for hidden state M_{t-1} in the input gate
U_{gc}	Bi-LSTM	Weight matrix for cell state D_{t-1} in the input gate
d_g	Bi-LSTM	Bias vector for the input gate
U_{qx}	Bi-LSTM	Weight matrix for input X_t in the forget gate
U_{qh}	Bi-LSTM	Weight matrix for hidden state M_{t-1} in the forget gate
U_{qc}	Bi-LSTM	Weight matrix for cell state D_{t-1} in the forget gate
d_q	Bi-LSTM	Bias vector for the forget gate
U_{dx}	Bi-LSTM	Weight matrix for input X_t in the cell state
U_{dh}	Bi-LSTM	Weight matrix for hidden state M_{t-1} in the cell state
d_d	Bi-LSTM	Bias vector for the cell state
U_{px}	Bi-LSTM	Weight matrix for input X_t in the output gate
U_{ph}	Bi-LSTM	Weight matrix for hidden state M_{t-1} in the output gate
U_{pc}	Bi-LSTM	Weight matrix for cell state D_t in the output gate
d_p	Bi-LSTM	Bias vector for the output gate
X_t	Bi-LSTM	Input at time t
M_{t-1}	Bi-LSTM	Hidden state at time $t - 1$
D_{t-1}	Bi-LSTM	Cell state at time $t - 1$
$\odot$	Bi-LSTM	Element-wise multiplication
$\tanh$	Bi-LSTM	Hyperbolic tangent function
r_t	Bi-LSTM	Linear transformation of the hidden state M_t
$\text{Linear}(M_t)$	Bi-LSTM	Linear transformation applied to hidden state M_t
y_t	Bi-LSTM	Model's output at time t, indicating a probability vector across the designated class set
$\text{softmax}(r_t)$	Bi-LSTM	Softmax activation function applied to r_t
$\mathcal{L}'$	CTC	Expanded label set including a special blank label
z	CTCCTC	Input sequence
U	CTC	Length of the input sequence
j	CTC	Inputs to the Bi-LSTM network
p	CTC	Outputs from the Bi-LSTM network
v	CTC	Weight matrix
$\mathcal{M}_v$	CTC	Mapping function characterized by a Bi-LSTM (Equation 14)
$(\mathbb{R}^q)^U$	CTC	Space of the input sequence
$(\mathbb{R}^r)^U$	CTC	Space of the output sequence
σ	CTC	Specific label sequence
$q(\sigma \| z)$	CTC	Probability of observing a specific label sequence (Equation 15)
$\mu_u^{\sigma_u}$	CTC	Network output probability at time u for label σ_u

TABLE 3. ***(Continued.)*** **Notation table.**

$\mathcal{C}^U$	CTC	Set of all possible paths for length U
m	CTC	Label sequence
$\gamma^{-1}(m)$	CTC	Set of all paths corresponding to label sequence m
$q(m\|z)$	CTC	Cumulative probability for a label sequence (Equation 16)
$\hat{m}(z)$	CTC	Most probable labeling (Equation 17)
$\eta(z)$	CTC	Loss function for the CTC objective (Equation 18)
η_{ctc}	CTC	Negative log-likelihood for the desired label sequence m (Equation 19)

In both models, the curves stabilize, indicating a balance between model complexity and generalization capability.

Comparing the two models, the 3D-CNN-EfficientNetB0-Bi-LSTM-CTC model (Figure 14) shows fewer fluctuations and lower overall loss values compared to the 3D-CNN-Bi-LSTM-CTC model (Figure 13). This suggests that incorporating EfficientNetB0 enhances feature extraction, leading to quicker convergence and improved performance. These enhanced capabilities are particularly beneficial in lip-reading tasks, where precise feature extraction from lip movements is critical.

The analysis of these loss curves provides valuable insights into the models' learning processes and helps in diagnosing issues such as overfitting or underfitting. Monitoring these curves allows for informed decisions about adjusting hyper-parameters, stopping training at the right time, and improving overall model performance in lip-reading applications.

C. BASELINES

Xu et al. [29] established a baseline by training a model using Cascaded Attention-CTC on the GRID dataset [27], which contains samples with a fixed number of frames. It is commonly understood that the frame count in typical lip-reading videos varies, making the training and recognition of videos with undefined frame numbers critical for future exploration.

Margam et al. [30] applied a foundational strategy using a combination of 3D and basic 2D-CNN to the GRID dataset [13], securing a 91.4% accuracy level with data previously unexamined from the same dataset. However, it has limitations highlighted by Xu et al. [29]. The experimental outcomes of these baseline methods on the respective datasets are summarized in Table 4.

Upon evaluating our model's performance, we observed that the overall effectiveness was somewhat below expectations, despite the input videos being processed accurately for the output sentences shown in Table 5. This result can be attributed to the model's training on a portion of the GRID Corpus dataset [27], which comprises approximately 1,000 videos.

D. TRAINING SETUP AND ANALYSIS

This study utilized a selected portion of the GRID Corpus dataset [13], comprising 450 video samples for training and 550 for testing from 1000 video samples. This subset was chosen because of its representative diversity, which ensured the robustness of our findings without compromising computational efficiency. In our method, it used the Adam optimizer [32] to optimize the model for as many as

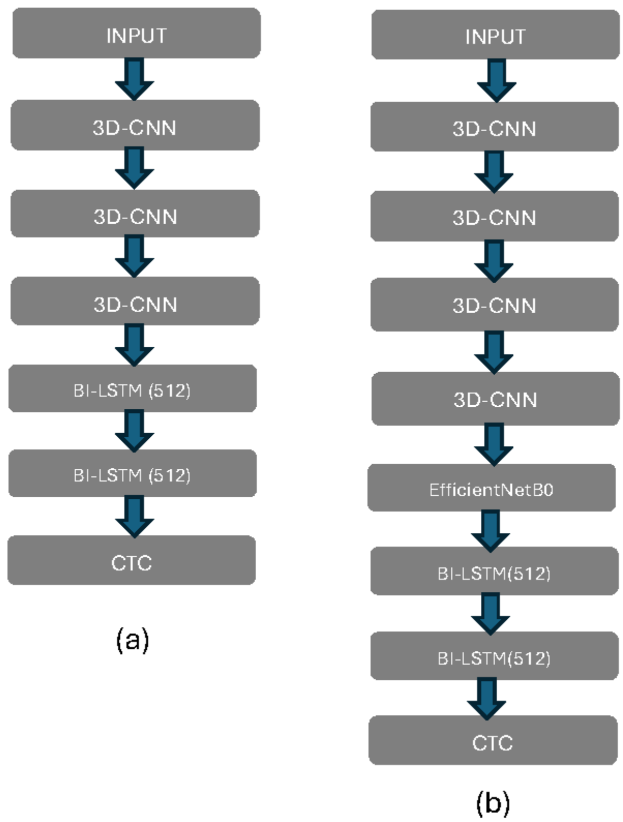

FIGURE 12. **Diagram of the variant models for comparison (a) 3D-CNN-Bi-LSTM-CTC Model, (b) 3D-CNN-EfficientNetB0-Bi-LSTM-CTC model.**

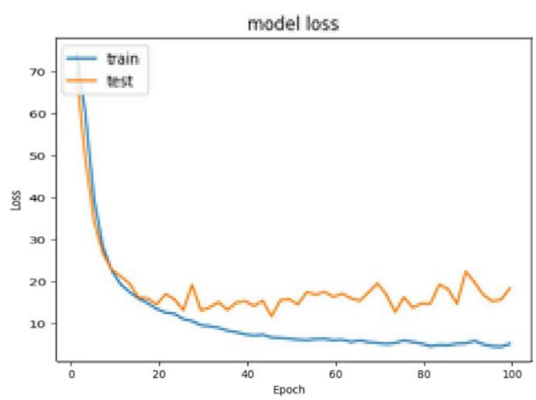

FIGURE 13. **Loss graph of 3D-CNN-Bi-LSTM-CTC.**

FIGURE 14. Loss graph of 3D-CNN-EfficientNetB0-Bi-LSTM-CTC.

TABLE 4. Preliminary findings from the initial experiment.

Author(s)	Dataset	Accuracy
Xu et al.[29]	GRID	89.6%
Gergen et al.[31]	GRID	86.4%
Margam et al.[30]	GRID	91.4%

TABLE 5. Experimental comparison using the GRID dataset.

Author(s)	Dataset	Accuracy
Xu et al.[29]	GRID	89.6%
Gergen et al.[31]	GRID	86.4%
Margam et al.[30]	GRID	91.4%
Our architecture	GRID	96.7%

100 epochs with a fixed learning rate of 0.0001. This method was intended to have a middle ground between steady improvement and effective training. Through an iterative process of minimizing a specifically chosen loss function, the model optimizes the parameters to enhance the performance of the designated task. The decision to extend the training up to 100 epochs was made to give the model ample time to assimilate the training data and adjust its predictions accordingly. It's worth mentioning that there's no one-size-fits-all for the perfect number of epochs. It depends on many things, like how big and complicated your dataset is and the problem being addressed.

Fundamentally, our strategy leveraged the Adam optimizer for the iterative refinement of the model parameters over many epochs, thus improving task-specific performance. The performance of LipSyncNet was also verified using a sample video from the GRID CORPUS Dataset, with the results detailed in Table 6. Throughout 100 epochs, our methodology yielded a WER 8.2% and impressive average accuracy of 96.7%, surpassing the current leading method by 3.3% percentage points.

E. RESULTS AND DISCUSSION

The model was deployed using Streamlit, an open-source framework in Python, to construct a web application showing the efficacy of the trained model. This application processes videos from the dataset into GIFs that isolate the mouth region, as shown in Figure 15. These preprocessed GIFs

TABLE 6. Evaluation outcomes for a randomly selected video from the GRID CORPUS Dataset.

Sl. No.	Text Predicted/Actual Text	Method Used
a.	bin red at n nime again	LipNet integrated with Lip Sync Technology (Wave2Lip [12])
b.	bin red at n nine again	LipSyncNet
c.	bim red at nine again	LipNet [22] adapted for GRID CORPUS
d.	bin red at n nine again	Verified Text

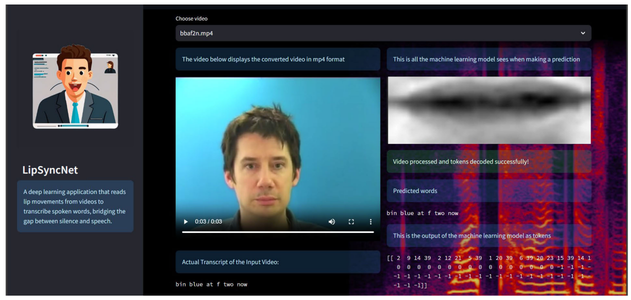

FIGURE 15. User interface design of the LipSyncNet model.

serve as inputs for the deep learning model, translating visual information into textual output for the user. Consequently, in instances of inaccurate output, the initial step involves verifying the correct segmentation of lip movements.

V. CONCLUSION

This research extensively analyzes the development and assessment of the 3D-EfficientNetB0-Bi-LSTM-CTC model for lipreading by employing advanced deep-learning methodologies. The proposed architecture demonstrated a remarkable accuracy rate 96.7% on the GRID Corpus dataset. Our initial phase involved leveraging the LipNet [22] model, during which we identified multiple practical application challenges. Furthermore, by integrating the Dlib facial landmark detector [33], we enhanced the ability of the model to remain agnostic to the speaker's position within the video.

Future work can be divided into three main areas. Initially, Constructing a model that leverages audio and visual cues for speech recognition presents an opportunity for comprehensive sentence-level prediction. Such a model would be particularly beneficial for improving video subtitle accuracy in noisy environments. Second, exploring training on even larger datasets can further elevate the performance metrics. Lastly, exploring the replacement of Bi-LSTM with transformer-based architectures presents a promising path towards achieving state-of-the-art outcomes.

REFERENCES

[1] D. Li, Y. Gao, C. Zhu, Q. Wang, and R. Wang, "Improving speech recognition performance in noisy environments by enhancing lip reading accuracy," *Sensors*, vol. 23, no. 4, p. 2053, Feb. 2023, doi: 10.3390/s23042053.

[2] S. Jeon and M. S. Kim, ''End-to-end sentence-level multi-view lipreading architecture with spatial attention module integrated multiple CNNs and cascaded local self-attention-CTC,'' *Sensors*, vol. 22, no. 9, p. 3597, May 2022, doi: 10.3390/s22093597.

[3] C. Sheng, X. Zhu, H. Xu, M. Pietikäinen, and L. Liu, ''Adaptive semantic-spatio-temporal graph convolutional network for lip reading,'' *IEEE Trans. Multimedia*, vol. 24, pp. 3545–3557, 2022, doi: 10.1109/TMM.2021.3102433.

[4] 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, doi: 10.1109/ACCESS.2022.3193231.

[5] M. A. Haq, S.-J. Ruan, W.-J. Cai, and L. P. Li, ''Using lip reading recognition to predict daily Mandarin conversation,'' *IEEE Access*, vol. 10, pp. 53481–53489, 2022, doi: 10.1109/ACCESS.2022.3175867.

[6] Y. Xiao, L. Teng, A. Zhu, X. Liu, and P. Tian, ''Lip reading in cantonese,'' *IEEE Access*, vol. 10, pp. 95020–95029, 2022, doi: 10.1109/ACCESS.2022.3204677.

[7] M. Miled, M. A. B. Messaoud, and A. Bouzid, ''Lip reading of words with lip segmentation and deep learning,'' *Multimedia Tools Appl.*, vol. 82, no. 1, pp. 551–571, Jan. 2023, doi: 10.1007/s11042-022-13321-0.

[8] S. Fenghour, D. Chen, K. Guo, B. Li, and P. Xiao, ''Deep learning-based automated lip-reading: A survey,'' *IEEE Access*, vol. 9, pp. 121184–121205, 2021, doi: 10.1109/ACCESS.2021.3107946.

[9] R. A. Ramadan, ''Detecting adversarial attacks on audio-visual speech recognition using deep learning method,'' *Int. J. Speech Technol.*, vol. 25, pp. 625–631, Jun. 2021, doi: 10.1007/s10772-021-09859-3.

[10] S. Fenghour, D. Chen, K. Guo, B. Li, and P. Xiao, ''An effective conversion of visemes to words for high-performance automatic lipreading,'' *Sensors*, vol. 21, no. 23, p. 7890, Nov. 2021, doi: 10.3390/s21237890.

[11] A. M. Sarhan, N. M. Elshennawy, and D. M. Ibrahim, ''HLR-net: A hybrid lip-reading model based on deep convolutional neural networks,'' *Comput., Mater. Continua*, vol. 68, no. 2, pp. 1531–1549, 2021, doi: 10.32604/cmc.2021.016509.

[12] S. Fenghour, D. Chen, K. Guo, and P. Xiao, ''Lip reading sentences using deep learning with only visual cues,'' *IEEE Access*, vol. 8, pp. 215516–215530, 2020, doi: 10.1109/ACCESS.2020.3040906.

[13] C. Guan, S. Wang, and A. W. Liew, ''Lip image segmentation based on a fuzzy convolutional neural network,'' *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1242–1251, Jul. 2020, doi: 10.1109/TFUZZ.2019.2957708.

[14] M. Hao, M. Mamut, N. Yadikar, A. Aysa, and K. Ubul, ''A survey of research on lipreading technology,'' *IEEE Access*, vol. 8, pp. 204518–204544, 2020, doi: 10.1109/ACCESS.2020.3036865.

[15] B. Martinez, P. Ma, S. Petridis, and M. Pantic, ''Lipreading using temporal convolutional networks,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6319–6323, doi: 10.1109/ICASSP40776.2020.9053841.

[16] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, ''Deep audio-visual speech recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8717–8727, Dec. 2022, doi: 10.1109/TPAMI.2018.2889052.

[17] N. K. Mudaliar, K. Hegde, A. Ramesh, and V. Patil, ''Visual speech recognition: A deep learning approach,'' in *Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, Jun. 2020, pp. 1218–1221, doi: 10.1109/ICCES48766.2020.9137926.

[18] P. Ma, Y. Wang, J. Shen, S. Petridis, and M. Pantic, ''Lip-reading with densely connected temporal convolutional networks,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2856–2865, doi: 10.1109/WACV48630.2021.00290.

[19] A. Adeel, M. Gogate, A. Hussain, and W. M. Whitmer, ''Lip-reading driven deep learning approach for speech enhancement,'' *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 3, pp. 481–490, Jun. 2021, doi: 10.1109/TETCI.2019.2917039.

[20] D. Howell, S. Cox, and B. Theobald, ''Visual units and confusion modelling for automatic lip-reading,'' *Image Vis. Comput.*, vol. 51, pp. 1–12, Jul. 2016, doi: 10.1016/j.imavis.2016.03.003.

[21] S. Petridis, Z. Li, and M. Pantic, ''End-to-end visual speech recognition with LSTMS,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2592–2596.

[22] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, ''LipNet: End-to-end sentence-level lipreading,'' 2016, *arXiv:1611.01599*.

[23] M. Wand, J. Koutník, and J. Schmidhuber, ''Lipreading with long short-term memory,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 6115–6119.

[24] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, ''Lip reading sentences in the wild,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3444–3453.

[25] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, ''Lipreading using convolutional neural network,'' in *Proc. Interspeech*, 2014, p. 3.

[26] S. Hilder, R. W. Harvey, and B. J. Theobald, ''Comparison of human and machine-based lip-reading,'' in *Proc. Interspeech*, 2009, pp. 1–4.

[27] M. Cooke, J. Barker, S. Cunningham, and X. Shao, ''An audio-visual corpus for speech perception and automatic speech recognition,'' *J. Acoust. Soc. Amer.*, vol. 120, no. 5, pp. 2421–2424, 2006.

[28] B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, M. Mulville, B. Coppin, B. Laurie, A. Senior, and N. de Freitas, ''Large-scale visual speech recognition,'' 2018, *arXiv:1807.05162*.

[29] K. Xu, D. Li, N. Cassimatis, and X. Wang, ''LCANet: End-to-end lipreading with cascaded attention-CTC,'' in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 548–555, doi: 10.1109/FG.2018.00088.

[30] D. K. Margam, R. Aralikatti, T. Sharma, A. Thanda, S. Roy, and S. M. Venkatesan, ''LipReading with 3D-2D-CNN BLSTM-HMM and word-CTC models,'' 2019, *arXiv:1906.12170*.

[31] S. Gergen, S. Zeiler, A. Abdelaziz, R. Nickel, and D. Kolossa, ''Dynamic stream weighting for turbo-decoding-based audiovisual ASR,'' in *Proc. Interspeech*, 2016, pp. 2135–2139, doi: 10.21437/Interspeech.2016-166.

[32] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*.

[33] N. Boyko, O. Basystiuk, and N. Shakhovska, ''Performance evaluation and comparison of software for face recognition, based on dlib and opencv library,'' in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2018, pp. 478–482, doi: 10.1109/DSMP.2018.8478556.

S. A. AMUTHA JEEVAKUMARI received the B.E. degree in electronics and communication engineering from Vellore Engineering College, Vellore, in 1994, the M.Tech. degree in information technology from Punjabi University, Patiala, in 2003, and the M.E. degree in communication systems and the Ph.D. degree from Anna University, Chennai, in 2013 and 2018, respectively. She has more than two decades of teaching and research experience in various universities. She is currently an Assistant Professor (Sr.) with the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai. Her research interests include wireless networks, reconfigurable antenna, artificial intelligence, machine learning, and deep learning.

KOUSHIK DEY received the Bachelor of Computer Applications degree from the Maulana Abul Kalam Azad University of Technology, West Bengal, India, in 2021. He is currently pursuing the Master of Computer Applications degree with Vellore Institute of Technology, Chennai, India. He completed his schooling in 2018 at Haldia Government Sponsored Vivekananda Vidyabhawan under West Bengal Council of Higher Secondary Education. His research interests include image processing, pattern recognition, and deep learning.