Signal, Image and Video Processing (2024) 18:5433–5448
https://doi.org/10.1007/s11760-024-03245-7

ORIGINAL PAPER

A novel framework using 3D-CNN and BiLSTM model with dynamic learning rate scheduler for visual speech recognition

Vishnu Chandrabanshi[1] · **S. Domnic**[1]

Received: 7 March 2024 / Revised: 22 April 2024 / Accepted: 26 April 2024 / Published online: 18 May 2024

Abstract

Visual Speech Recognition (VSR) is an appealing technology for predicting and analyzing spoken language based on lip movements. Previous research in this area has primarily concentrated on leveraging both audio and visual cues to achieve enhanced accuracy in speech recognition. However, existing solutions encounter significant limitations, including inadequate training data, variations in speech patterns, and similar homophones, which need more comprehensive feature representation to improve accuracy. This article presents a novel deep learning model for performing word level VSR. In this study, we have introduced a dynamic learning rate scheduler to adapt the learning parameter during model training. Additionally, we employ an optimized Three-Dimensional Convolution Neural Network for extracting spatio-temporal features. To enhance context processing and ensure accurate mapping of input sequences to output sequences, we combine Bidirectional Long Short Term Memory with the CTC loss function. We have utilized the GRID dataset to assess word-level metrics, including Word Error Rate (WER) and Word Recognition Rate (WRR). The model achieves 1.11% WER and 98.89% WRR, respectively, for overlapped speakers. This result demonstrates that our strategy outperforms and is more effective than existing VSR methods. *Practical Implications* - The proposed work aims to elevate the accuracy of VSR, facilitating its seamless integration into real-time applications. The VSR model finds applications in liveness detection for person authentication, improving password security by not relying on written or spoken passcodes, underwater communications and aiding individuals with hearing and speech impairments in the medical field.

Keywords 3D-CNN · BiLSTM · Visual speech recognition · LRS · Deep learning

1 Introduction

People commonly interact through their senses of hearing and vision. In day-to-day life, speech recognition is a popular and effective method for understanding a person emotions and expressions. Speech recognition technology is remarkable in its ability to comprehend spoken language with precision. Previous research has primarily concentrated on the auditory mode of communication because of its outstanding ability to recognize spoken words. Unfortunately, this technique is not helpful for deaf and mute individuals, and even for normal people when the acoustic data is tainted. However, it becomes particularly challenging in noisy environments or when the acoustic signal is unavailable. Audio speech recognition faces challenges in accurately transcribing spoken language due to background noise, variations in speech patterns, speaker accents, and homophones that can result in different words being transcribed identically. To address these challenges, researchers have explored combining audio features with visual features techniques to improve the overall accuracy of speech recognition systems [1, 2]. Despite the numerous advantages of audio visual speech recognition, it encounters challenges that can lead to reduced accuracy. These challenges include operating in noisy environments, accounting for speaker variability, managing computational complexity, and addressing limited training data availability. Ensuring proper synchronization between audio and visual modalities is crucial for reliable speech recognition.

Vishnu Chandrabanshi and S. Domnic have contributed equally to this work.

✉ Vishnu Chandrabanshi
vishnukumar.nit@gmail.com

S. Domnic
domnic@nitt.edu

[1] Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

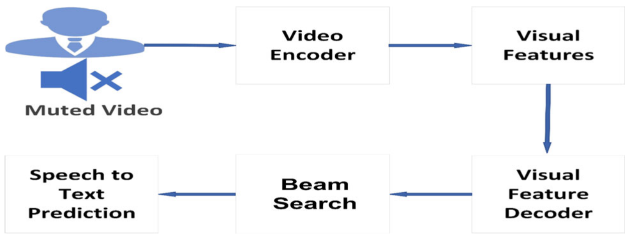

Fig. 1 General architecture of VSR system

To overcome these challenges, engaging in continuous research and advancements in AVSR algorithms, data collection techniques, noise robustness, speaker adaptation, and synchronization methods is imperative. Nowadays, researchers have moved towards the VSR technique, also known as lip reading technology, which identifies the spoken content by analyzing the lip movement characteristics of the speaker without relying on audio signals [3, 4]. Lipreading can identify a speaker speech in noisy environments, even without audio signals. Lip movements in visual-based speech recognition systems have proven effective in mitigating background noise and aiding individuals with auditory impairments [5]. This promising area of research has the ability to strengthen the precision and stability of automatic speech recognition systems. VSR is a fascinating study area in computer vision and image processing that has gained recent attention. The goal of VSR systems is to use visual information from lip movements to recognize speech content. In VSR, the data will be in videos converted into frames for further implementation. Each frame contains a wealth of information about the variability of visual cues across different speakers. VSR has emerged as an essential research topic with potential uses in areas like speech recognition [6, 7], facial bio-metric [8, 9], healthcare [10, 11], and security [12]. To enhance the accuracy of VSR, extraction of spatio-temporal characteristics from video recordings is essential in video-based speech recognition. In a broader sense, spatio-temporal pertains to a phenomenon where data is gathered simultaneously in space and time. VSR refers to video features that are considered indispensable in accomplishing the task. The general architecture of VSR system is given in Fig. 1.

The study conducted by [13] illuminates the pivotal role played by the morphology of the mouth and lips in the processing of speech. Particularly in scenarios where speech may be compromised due to environmental noise or other factors, lip-reading emerges as a valuable adjunct, augmenting our comprehension of spoken language. In this work, we have introduced a deep learning based architecture for VSR, which leverages spatio-temporal features and utilizes haar cascade [14] for face with lip localization and detection. Furthermore, we have optimized Three-Dimensional Convolution Neural Network (3D-CNN) to extract essential spatio-temporal features from video data and used Bidirectional Long Short Term Memory (BiLSTM) [15] to process the input video frame sequence bidirectionally for capturing dependencies between past and future elements for accurate speech recognition. Furthermore, the proposed framework analyses the visual characteristics of the users lip in the video to transcribe the spoken speech accurately. Spatial features encompass information about the shape and position of the lips, while temporal characteristics capture details such as the speed and direction of lip movements. This work also employs data shuffling during training to mitigate bias and promote model generalization. Additionally, the model training incorporates the variable Learning Rate Scheduler (LRS), as presented in algorithm 3, to optimize the learning process effectively. Learning rate tuning is essential for optimizing the performance of a machine learning model by striking the proper balance between convergence speed and avoiding overshooting the optimal parameter values. It affects how rapidly the model learns and adapts during training; therefore, determining the optimal learning rate is frequently a vital stage in the hyper-parameter tuning process. Furthermore, we employ the Connectionist Temporal Classification (CTC) loss function [16] to calculate the associated loss, followed by a CTC decoder to perform the final speech-to-text conversion. Researchers previously used Bidirectional Gated Recurrent Units (BiGRU) [17] to process data forward and backward to capture bidirectional context and dependencies in sequential data. BiGRU is computationally lighter and better suited for tasks with moderate dependencies and shorter sequences. BiLSTM has an advantage over BiGRU in capturing complex dependencies memory capacity through multiple gates and potentially performing better on large datasets. However, in the context of the gating mechanisms, Long short-term memory (LSTM) [18] units used in BiLSTM have three gates (input, forget, and output gates), which allow for more fine-grained control over the information flow within the cell state and hidden state. This intricate gating mechanism enables LSTM to capture and retain longer-term dependencies in sequences, making it better suited for tasks requiring complex sequence modeling. On the other hand, GRU units used in BiGRU have two gates (update and reset gates), which can lead to slightly fewer parameters and potentially more straightforward training dynamics. We monitored the LRS to observe its impact on the model output. After careful analysis, it was concluded that it significantly affects model training performance. Consequently, we proposed optimized LRS to adjust the learning rate better, aiming for improved model performance adaptation. This paper presents the significant contributions as follows:

1. This paper reviews state-of-the-art approaches and datasets for character, digit, word, and sentence-level VSR.

2. The proposed deep learning model exhibits versatility, making it suitable for character, word, sentence, and digit datasets, which rely only on visual information.
3. We have optimized the 3D-CNN architecture and implemented a dynamic Learning Rate Scheduler (LRS) to regulate the learning rate throughout the model training adaptively.
4. We showcased the enhanced accuracy and effectiveness of the proposed model compared to existing deep learning frameworks utilized in implementing VSR systems.

This article is structured as follows. Section 2 discusses relevant work on various techniques and datasets. Section 3 describes the proposed methodology in multi-phases ranging from lip dataset development to model architecture, experiment results, and evaluation details, followed by comparing the proposed methodology to state-of-the-art methods. Section 4 includes the conclusion, future endeavors, and acknowledgment.

2 Related work

The major problem in the VSR system is visual ambiguity that arises due to the similar homophones of words or characters because they generate identical lip movements (e.g., 'm', 'b', 'p', 'pat', 'mat', 'bat'). The phoneme is the smallest standard unit capable of distinguishing the meaning of a word in speech processing. Similarly, viseme is the standard unit many researchers use for analyzing the visual information from the video domain. Our research primarily focuses on word-level performance, even though we are utilizing a sentence-level dataset for VSR. We have identified several existing datasets that are suitable for performing VSR tasks. The datasets available for this task are as follows: LRW and LRS2-BBC [19], LRW-1000 [20], AVLetters [21], AVLetters2 [22], LRS3-TED [23], MIRACLE-VC1 [24], LIPAR [25], AV Digits [26], OuluVS2 [27], GRID [28].

A deep learning methodology has been proposed in [29] for recognizing complete words. This study trained LSTM network with discrete cosine transform, and deep bottleneck features for word recognition. Similarly, [30] employed an LSTM with Visual Geometry Group Network (VGGNet) to recognize the complete word. Distinguishing different words or characters with similar phonemes in lipreading poses a challenge due to the similarity in lip movements associated with those phonemes. The model aims to overcome phoneme similarity challenges by training a recurrent neural network on spatio-temporal feature patterns, resulting in state-of-the-art performance for the challenging lip reading task. Similarly, [31] introduced a lip-reading model incorporating a multi-grained spati-temporal approach to capture the distinctions between words and the individual speaking styles of different speakers. The model was trained for Word Recognition Rate using Dense-Net3D and BiLSTM. A more complex model than LipNet [3] has been introduced in the research work [32], which introduces residual networks with 3D convolutions to extract more robust features.

Existing datasets contain recordings with a limited number of participants and a limited lexicon, which further impedes progress in the field of VSR. In order to tackle this issue, [19] proposed a methodology to tackle the challenge of a small lexicon by contributing the LRW and LRS2-BBC dataset, encompassing a vocabulary of more words. Their approach introduced the Watch, Listen, Attend, and Spell (WLAS) model, which effectively converts mouth movements into words. The WLAS model transcribes spoken sentences into characters and can handle input solely from the visual stream. It achieved a WRR of 76.20% on the LRW dataset. Likewise, [20] proposed a work in which they provided a comprehensive and naturally distributed benchmark dataset, LRW-1000, for the lip-reading task. A non-autoregressive lipreading model proposed by [33] for fast lip reading and generating target word. It is designed to recognize silent source videos and generate all target text tokens simultaneously.

Wang et al. [11] created an HMM-based lip-reading software for the speech impairment people. The software employs a pre-trained VGGNet model with a user-friendly graphical user interface. However, the software disadvantage is its lower testing accuracy. Similarly, [34] focus on appearance-based visual features for people with learning disabilities. Based on the test results, the proposed system obtains a visual speech accuracy of 76.60%. In 2022, Huang et al. [35] proposed a lip-reading model, which utilizes a pre-trained neural network for feature extraction and processing through a transformer network. The overall accuracy assessment was based on word-level accuracy, achieving a result of 45.81%. Vayadande et al. [48] recognized the significance of lip reading for individuals with hearing impairments, leading to the development of advanced deep learning model, LipReadNet. This model has used 3D-CNN with LSTM networks and demonstrated substantial efficacy by achieving a 93% WRR on the GRID corpus. Time complexity is essential in analyzing the model performance; therefore, a lightweight deep learning model has been introduced in [36] based on isolated word-level recognition. This system investigates efficient models for VSR, achieving the exhibit similarity to the available model, but computational costs are reduced by eight times. Recognizing the slow training pace of conventional lipreading models, He et al. [37] proposed a new approach called the batch group training strategy. Their architecture combines 3D-CNN, MouthNet, and Bi-LSTM networks with a CTC loss function, resulting in a 93.8% accuracy on the GRID corpus. The performance of existing models with the GRID dataset as well as other datasets for VSR is provided in Tables 1 and 2 , respectively. Lip reading poses significant

Table 1 VSR comparison on different existing dataset and model

References	Methodology	Database	Work	WRR%
[30]	LSTM, VGG+CFI	MIRACLVC1	Words, Phrases	76.00
[29]	LSTM, DBNF+DCT	AVLetters	Alphabet	58.10
[38]	DNN-HMM, DBNF	CUAVE	Digits	64.90
[19]	CNN+LSTM, Attention+LSTM	LRW	Words	76.20
[26]	Bi-LSTM, Autoencoder	AVDigits	Digits	68.00
[31]	Bi-LSTM, DenseNet3D+ResNet	LRW	Words	83.34
[39]	Bi-GRU, ResNet+3D-CNN	LRW	Words	84.41
[9]	Bi-GRU, Resnet+3D-CNN	LRW	Words	85.20
[40]	HMM, Haar Cascade	Arabic Figures(0-9)	Digits	96.20
[41]	LDT+GPT Transformer, 3D-CNN+ResNet	LRS2	Sentences	64.60
[42]	Temporal CNN, ResNet	LRW	Words	85.30
[36]	MS-TCN+ResNet	LRW-1000, LRW	Words	46.60, 88.50
[25]	3D-CNN, Dlib	LIPAR	Words	77.90
[34]	ANN, SVM, NB, LBP+GLCM	CUAVE	Digits	76.60
[2]	TM-CTC, SSD+CNN	LRS2-BBC, LRS3-TED	Sentences	65.00, 73.00
	TM-seq2seq, SSD+CNN	LRS2-BBC, LRS3-TED		51.20, 59.30

Table 2 VSR comparison on GRID dataset

References	Methodology	Database	Work	WRR%
[43]	LSTM, Feed-Forward	GRID	Words	79.60
[3]	Bi-GRU+CTC, 3D-CNN	GRID	Words	95.20
[44]	LSTM, Feed-Forward	GRID	Words	83.30
[1]	LSTM, Feed-Forward	GRID	Words	84.70
[19]	LSTM+Attention, CNN+LSTM	GRID	Words	97.00
[4]	Bi-GRU+Attention+CTC, 3D-CNN	GRID	Words	97.10
[33]	Transformer + CTC, STCNN	GRID	Words	95.50
[45]	BiGRU + Attention CTC, STCNN	GRID	Words	97.30
[46]	GRU + 3D-CNN, ResNet	GRID	Words	88.80
[47]	BiGRU + CTC, STCNN	GRID	Words	90.30
[37]	BiLSTM, 3D-CNN+MouthNet	GRID	Words	93.8
[35]	Transformer, VGG16	GRID	Words	45.81
[48]	LSTM, 3D-CNN	GRID	Words	93

challenges due to similar lip movements observed across various consonant sounds.

In response to the inherent complexities associated with the VSR task, Rastogi et al. [45] proposed a deep learning approach involving a sequential model. The model incorporates the context of preceding and subsequent words to better interpret lip movements. To address the limitations stemming from the assumption of conditional independence in the CTC framework, they introduced a hybrid CTC/Attention model, effectively integrating the strengths of both approaches. Typical seq2seq models face two main challenges: exposure bias and a mismatch between the optimization target and the evaluation metric. To tackle these issues, [46] introduced a GRU based deep learning model known as pseudo-convolutional policy gradient. This model incorporates a pseudo-convolutional operation on the reward and loss dimensions, allowing it to consider more context around each time step. This approach produces a more robust reward and loss for the overall optimization process. To enhance lip-reading accuracy [41] introduced a neural network based model for a lip reading task that system relies only on visual information and operates without a lexicon. The system effectively reads sentences with a diverse vocabulary, even including words absent from its training data. 3D-CNN architecture has been customized by [25] for VSR that extracts spatio-temporal features to recognize the words. A thirteen-layer CNN integrated with batch normalization methodology has been proposed in the work [49] for lip-reading. However, the model exhibited a lower testing accuracy. Sarhan et al. [47] proposed a hybrid lip-reading model that uses encoding at the front end and decoding at the back end. The front end contains inception, gradient preservation, and a BiGRU layer, while the rear end has an attention layer, a fully connected layer, and a CTC layer. The model achieved 1.4% CER and 3.3% WER for overlapping speakers. LCSNet [50], an architecture designed by Xue et al., utilizes the channel attention mechanism to capture essential lip movement features. These extracted features are subsequently fed into a BiGRU to acquire long-term spatio-temporal features.

The performance of the VSR task is also degraded due to pose variations, lighting, and speaking speed. To address these challenges, Speaker Adaptive Training (SAT) is used in the research [51] to train Deep Neural Networks, which allows them to recognise words efficiently regardless of the speakers viewing angles. Similarly, [39] employed two constraints: local mutual information maximization for fine-grained lip movement detection and global mutual information maximization for essential frame identification to capture the lip movements. Three categories of speech data have been utilized by [26], including normal, whispered, and silent speech, to facilitate precise transcription of spoken words. Their methodology involved training an LSTM classifier while extracting DCT features. The model demonstrated a commendable 68% WRR on the AVDigits dataset. A comprehensive analysis of three pivotal works for VSR, as proposed by [1, 43, 44], is given in Table 2. Their proposed deep learning approach trained an LSTM classifier with a feed-forward network and obtained a maximum 84.70% WRR on the GRID dataset.

3 Proposed methodology

This section describes the architecture of the proposed approach to deal the GRID dataset. The approach begins with a detailed dataset description, followed by video frame extraction and conversion. The next steps proceed to extract Regions of Interest (ROI) and normalization of ROI. These normalized frames are given to the model to extract features and use them to forecast frame sequence prediction. Finally, the model target is model evaluation, with the primary goal of rigorously assessing the proposed framework performance, particularly regarding WER and WRR.

3.1 Dataset, image pre-processing and feature concatenation

In the following subsections, we will cover the dataset description, image pre-processing techniques employed for localizing the subject face and lip regions, cropping the lip areas, and further image processing techniques for processing the cropped lip region.

3.1.1 Dataset description

Our study utilizes the GRID dataset publicly available [52], widely recognized and extensively employed within speech recognition. The dataset consists of.mpg video files and corresponding.align files that contain a series of time intervals paired with related labels. These files describe a sequence of events or activities, each with its start and end times, representing the duration of the specific event.

The dataset offers a comprehensive collection of videos and alignments of 34 speakers (s1 to s34), each delivering 1000 sentences. These sentences follow a fixed structure, consisting of a command (4 options), a colour (4 options), a preposition (4 options), a letter (25 options), a digit (10 options), and an adverb (4 options). There are 51 exclusive words represented in the dataset, and the random word alternatives are used to avoid dependency on contextual cues for classification. The chosen dataset offers valuable temporal insights with a comprehensive collection of diverse recordings. It includes speakers from various backgrounds, ethnicities, and age groups, making it suitable for training and evaluating VSR models that can handle variations in speech and visual cues. Significantly, in this particular dataset, our research has been solely centered around video content without audio. It is worth noting that many earlier researchers have also chosen to focus exclusively on the video without relying on the audio portion of this dataset for the VSR task (refer Table 2). Each sentence within the corpus lasts 3 s, capturing 25 frames per second. As a result, the data for each speaker encompasses 3000 s (50 min) in total. In this paper, we have utilized 7200 videos, a subset of the corpus comprising 30 subjects, each with 240 videos. The subjects available in the dataset are given in Fig. 2.

3.1.2 Face localization and cropping region of interest (ROI)

The first stage of this pipeline entails using a video in which the subject pronounces a sentence from the provided corpus. We divide the video into separate frames, convert them into gray scale, and treat each frame as a sub-input for the pipeline. The importance of converting color frame to grayscale lies in its capacity to decrease the dimensions of the input vector. Consequently, this results in fewer parameters for training the model. The algorithm 1 outlines the entire process from face localization to ROI cropping, while Fig. 3 provides a visual representation of this procedure. Figure 3a is an original color frame, (b) is a converted grayscale frame, (c), (d) represents the face localization and lip detection, (e) represents the final phase of lip region (ROI) extraction for lip data preparation.

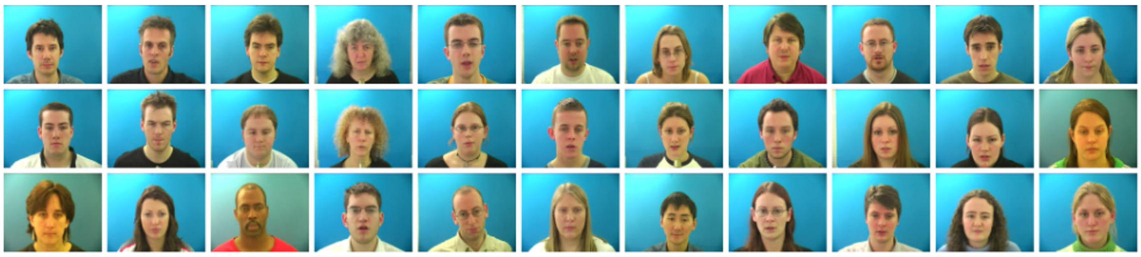

Fig. 2 Subjects in GRID dataset

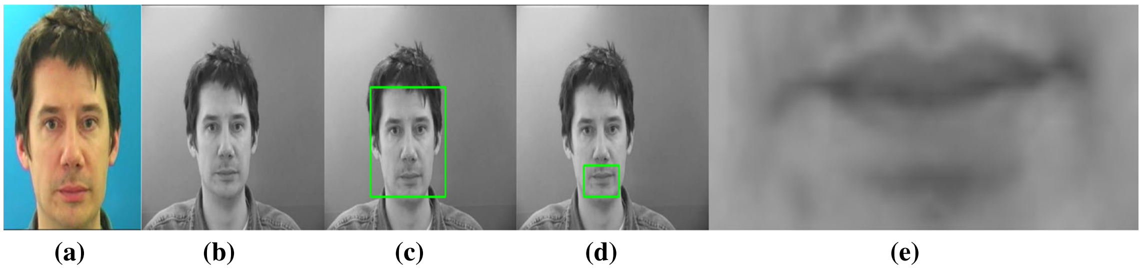

Fig. 3 **a** Original frame **b** Gray scale frame **c** Face localization **d** Lip localization **e** Lip extraction

In this process, ROI is a lip region, which will be the model input for tracking the lip movement for word recognition.

3.1.3 Extraction and normalization of ROI

As described in the previous section, the image processing technique depicted in Fig. 3 is utilized to precisely crop the ROI of the subject.

Algorithm 1 Video Pre-processing and Extraction of ROI

```
Require: Video
Ensure: Array of Cropped ROI
 1: Initialize Roi_frames as an empty list
 2: Convert input frame to grayscale
 3: Load pre-trained model for face detection
 4: faces = Detect_Face()
 5: for each $(x, y, w, h)$ in faces do
 6:     Draw_Rectangle_Region()
 7:     Localize mouth region
 8:     lips = Extract_Lips()
 9:     for each detected lip do
10:         Draw rectangle around the detected lip
11:         Calculate and append Roi size to Roi_frames list
12:     end for
13: end for
14: Return Roi_frames
```

The subsequent part of this step is ROI normalization. The following stage of this pipeline involves image Contrast Normalization (CN). During this process, the converted image pixel values are adjusted to operate within the limit of [0 to 128 intensity values]. It defines image contrast and intensity, rendering them more uniform and suited for later processing or analysis. The CN method is applied to mitigate the contrast disparity between light and dark pixels in an image (refer to Algorithm 2).

Algorithm 2 Contrast Normalization (CN)

```
Require: ROI, $\mu$, $\epsilon$
Ensure: Normalized ROI
1: if ROI $\neq$ None then
2:     img = Read(ROI)
3:     a = Convert img into array
4:     Contrast = $\sqrt{\mu + \text{mean}(a^2)}$
5:     CN = $sf \times a / \max(\text{Contrast}, \epsilon)$
6: end if
7: return Normalized ROI
```

In algorithm 2, μ represents a parameter that controls the strength of contrast normalization, sf is a scaling factor, and ϵ (epsilon) is a small positive constant used in algorithms and mathematical operations to prevent division by zero and ensure numerical stability.

3.1.4 ROI concatenation

The last stage of the image pre-processing pipeline involves combining normalized ROI frames for the preparation of feature matrix. Due to variations in the number of features obtained from each sentence, the feature vector can become uneven. The reason for this is that the number of frames captured for each sentence varies, causing imbalanced data features that can pose challenges in training the model. To address this issue, we have concatenated some silent frame denoted as 'sil'. The silent frame is a video frame captured

Table 3 Alignment of a video

Start time	End time	Label
0	23,750	sil
23,750	29,500	bin
29,500	34,000	blue
34,000	35,500	at
35,500	41,000	f
41,000	47,250	two
47,250	53,000	now
53,000	74,500	sil

when the subject remains silent and does not articulate any words from the sentence. In the alignment of a video, 'sil' is added at the beginning and ending of the alignments. The alignment of a video file with the spoken words"bin blue at f two now"is illustrated in Table 3.

Consider max is the maximum number of frames utilized as input for 3D-CNN.The maximum frame count (max) is configured at 75, and the specially designated augmented frame is labeled as 'sil' represented by b. For a person p with n frames, we define the qth frame as f_{pq}. The concatenated frames F_c of one person can be defined by Eq. 1. The visual representation of F_c is shown in Fig. 4.

$$F_c = \left(\sum_{q=1}^{n} f_{pq} + \sum_{k=1}^{75-\text{n}} b_k \right) \tag{1}$$

3.2 Model architecture

This section will explore various components of the proposed model architecture given in Fig. 5. Firstly, the feature extraction process is achieved by utilizing optimized 3D-CNN, effectively capturing spatial and temporal information. Activation functions are pivotal, introducing non-linearity to enhance the model capacity to learn intricate patterns. In addition, pooling layers are responsible for dimensionality reduction and preserving essential information. Furthermore, BiLSTM layers are employed to enable bidirectional context processing. To mitigate over-fitting, dropout layers are integrated. A dense layer with softmax activation is incorporated to facilitate class prediction. An appropriate loss function is selected to quantify the model performance effectively. Finally, a learning rate scheduler is implemented to adjust hyper-parameters during training dynamically, optimizing the models convergence and performance.

3.2.1 2D CNN

Two-dimensional convolution (2D-CNN) is employed at the convolutional layers to extract features from the immediate neighborhood on feature maps in the preceding layer. In 2D-CNN, the convolution layers primarily extract spatial features from the input data. The formal representation of the output of a neuron located at coordinates (p, q) in the n^{th} feature map in the m^{th} layer, denoted as $S_{m,n}^{p,q}$ is given by:

$$S_{m,n}^{p,q} = ReLU\left(\sum_{g} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} K_{m,n,g}^{h,w} S_{(m-1)g}^{(p+h)(q+w)} \right) \tag{2}$$

Where ReLU is an activation function and $S_{m,n}^{p,q}$ signifies the value of the unit associated with the feature map at point (p, q) within the previous $(m-1)th$ layers of nth feature map. H and W are the kernels height and width, respectively, and g is an index over the set of feature maps in the previous $(m-1)th$ layer corresponding to the current feature map. $K_{m,n,g}^{h,w}$ is the kernel weight value at position (p, q). In this analysis, bias terms are omitted.

3.2.2 3D CNN

In 2D-CNN, convolution only focus on capturing the spatial feature, while to capture the motion information from video, it is desirable to use 3D-CNN to capture both spatial and temporal feature map. The proposed model has three 3D-convolution layers with 128, 256, and 75 filters of size 3 × 3 × 3 for each layer, expecting input data in the shape of a sequence of 3D frames with dimensions 75 × 81 × 140 denoting the number of frames for each video with frame size. The feature maps within the convolution layer establish connections with several adjacent frames from the preceding layer, effectively capturing motion details. Given a 3D-CNN with multiple layers, the function computes the value at a specific position (p, q, r) in the m^{th} feature map in the n^{th} layer is given by:

$$S_{m,n}^{p,q,r} = ReLU\left(\sum_{g} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \sum_{t=0}^{T-1} K_{m,n,g}^{h,w,t} S_{(m-1)g}^{(p+h),(q+w),(r+t)} \right) \tag{3}$$

where, $S_{m,n}^{p,q,r}$ denotes the value of the unit connected to the current feature map located at position (p, q, r) within the n^{th} feature map of the preceding $(m-1)^{th}$ layer. H and W are the height and width of kernel, T represents the dimensions of a 3D kernel along the temporal axis. g is indexed over the set of features map in the $(m-1)^{th}$ layer connected to the current feature map. $K_{m,n,g}^{h,w,t}$ is the kernel weight value at position (p, q, r). A ReLU activation function processes the output of a CNN before passing it to subsequent layers to introduce non-linearity, allowing the model to learn complex patterns and relationships in the data. It replaces all negative values in the CNN output with zeros and leaves positive values unchanged.

3.3 ReLU

A good activation function improves CNN performance significantly. The proposed architecture adds three layers of ReLU after each convolutional layer. ReLU is a widely recognized and frequently used activation function in neural networks [53] and is a notable choice due to its non-saturating nature. In this work, Fig. 6 is adapted from [54] to represent the mathematical architecture of ReLU. The definition of the ReLU activation function is given in Eq. 4.

$$a_{m,n,k} = \max(S_{u,v,k}, 0) \tag{4}$$

where $S_{u,v,k}$ represents the input to the activation function at the index (u, v) in the k^{th} channel. ReLU is a piecewise linear function that zeroes off negative numbers while keeping positive values. The max operation employed by ReLU enables faster computation compared to sigmoid or tanh activation functions. Additionally, it encourages sparsity within the hidden units and facilitates the networks acquisition of sparse representations. Deep networks can be trained efficiently with ReLU even if no prior training has been given [55]. Even though the disruption of ReLU at 0 may degrade performance, prior research has indicated its empirical superiority over sigmoid and tanh activation functions [56]. The ReLU activation and max-pooling combination helps CNNs learn complex features, reduce computational complexity, and improve the network ability to recognize patterns regardless of their exact location in the input data.

3.4 Max pooling

In CNN, pooling reduces computational complexity by minimizing connections between convolution layers. The proposed model provides three 3D Max-Pooling layers with a pool size of (1, 2, 2) for reducing the feature map. This layer performs down-sampling, reducing the spatial dimensions of the data while preserving important features. The pooling procedure entails sliding a three dimensional filter over each channel of the feature map and aggregating the features within the filters covered region. For an image with a feature map of dimensions n_{hi} x n_{wi} x n_{ch}, the resulting dimensions after applying a max pooling layer can be calculated as follows:

$$[\left(\frac{n_{hi} - a + 1}{b}\right) \times \left(\frac{n_{wi} - a + 1}{b}\right) \times n_{ch}] \tag{5}$$

Where n_{hi}, n_{wi}, n_{ch} is the height, width and channel of the feature map and a, b is the size of filters and stride. Here, n_{ch} is one because the analysis is conducted on gray scale frames. A one-time distributed flattened layer is added before giving the down-sampled features to the BiLSTM. TimeDistributed wrapper with a flattened layer combination is applied to each temporal slice of the input tensor. It independently flattens

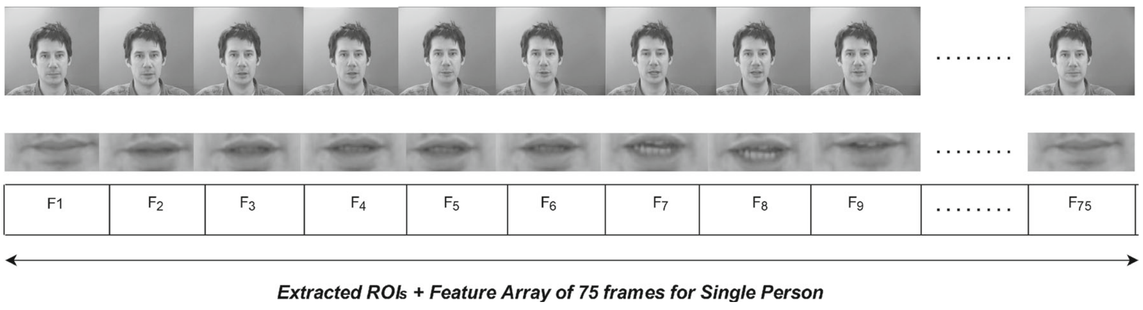
....
F1
F2
F3
F4
F5
F6
F7
F8
F9
F75
Extracted ROIs + Feature Array of 75 frames for Single Person

Fig. 4 Concatenated ROI frames including silent frames

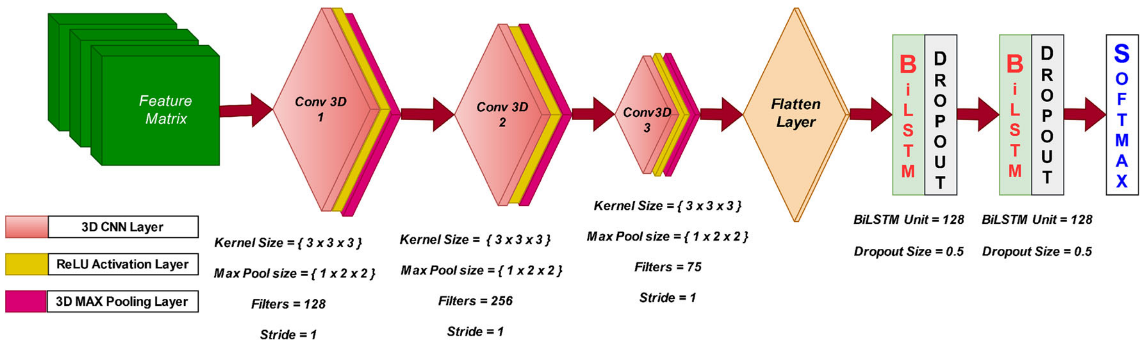

Fig. 5 Proposed model architecture

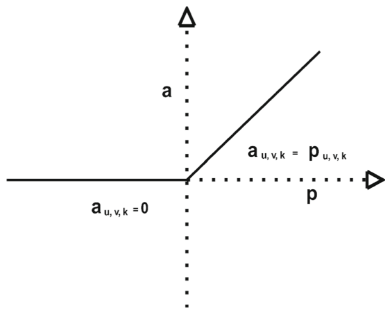

Fig. 6 ReLU [54]

the features at each timestep, allowing the model to capture temporal patterns. This layer is beneficial when dealing with sequential data, such as text sequences.

3.5 BiLSTM

The architecture of LSTM is based on three gates and cells, which help to store, forget, and retain the information bidirectionally during data processing. The proposed architecture contains two layers of BiLSTM for better context embedding. Each Layer of BiLSTM contains 128 memory units and an orthogonal weight initializer, and the returns sequence is true as a parameter. To elaborate, it involves an input sequence vector denoted as X, which can be represented as $(x_1, x_2, \ldots, x_n)$, where n represents the length of the input sentence.The LSTM structure is made up of three distinct gates: an input gate, an output gate, and a forget gate. These gates are essential components for controlling the flow of information within the LSTM cell. The hardware level symbolic circuit representation of the LSTM network structure is derived from [57] and illustrated in Fig. 7. The initial phase in LSTM revolves around determining which information of the cell state should forget, facilitated by the forgetting gate.

3.5.1 Forget gate

Typically, a sigmoid function within this gate determines what information to eliminate from the LSTM memory. This determination primarily relies on h_{t-1} and x_t. The result of this gate is denoted as f_t, a value ranging from 0 to 1. A value of 0 indicates that the learnt information has been completely removed, whereas a value of 1 indicates that the entire value has been retained. The calculation for this output is as follows:

$$\begin{cases} f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \\ c_{tf} = f_t.c_{t-1} \end{cases} \tag{6}$$

Where σ is the sigmoid activation function, $(W_{fh}h_{t-1}$ represents the weight value of the previous hidden state that is h_{t-1}, $W_{fx}x_t$ denotes the current weight value of the current input state x_t at particular time step t. Additionally, b_f represents the bias value at the forget gate.

3.5.2 Input gate

This gate is responsible for determining whether or not the incoming data should be stored in the LSTM memory. It comprises two parts: a sigmoid segment and a tanh segment. The sigmoid segment decides which values must be updated, whereas the tanh component generates a vector of potential new values for LSTM memory integration. The outputs of these two components are computed as follows:

$$\begin{cases} i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \\ g_t = \tanh(W_{gh}h_{t-1} + W_{gx}x_t + b_g) \\ c_{ti} = i_t.g_t \\ c_t = c_{ti} + c_{tf} \end{cases} \tag{7}$$

Where $(W_{ih}h_{t-1})$ represents the weight value of the previous hidden state h_{t-1}, $(W_{ix}x_t)$ denotes the weight value of the current input state x_t at a particular time step t. Additionally, b_i represents the bias value for the input gate. A similar representation is used for the weight matrix of the previous hidden state, the current input state, and for the bias in the tanh activation function.

3.5.3 Output gate

The process begins with a sigmoid layer determining the influence of a segment of LSTM memory on the output. The values are then adjusted with tanh functions to fall inside the range of -1 to 1. Finally, the result is multiplied by the output of the sigmoid layer. The equations below illustrate the computation process:

$$\begin{cases} o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \\ h_t = \tanh(c_t) \cdot o_t \end{cases} \tag{8}$$

The sigmoid activation function (σ) is applied to the output gate. In this context, $(W_{oh}h_{t-1})$ represents the weight value associated with the previous hidden state, while $(W_{ox}x_t)$ denotes the weight value corresponding to the current input state at time step t. Additionally, b_o represents the bias value at the output gate. A single LSTM cell, which is limited to capturing the preceding context and lacks the ability to incorporate future information, is enhanced by the introduction of a bidirectional recurrent neural network proposed by [15]. A BiLSTM processes the input sequence $X = (x_1, x_2, \ldots, x_n)$ in both the forward and backward directions, generating

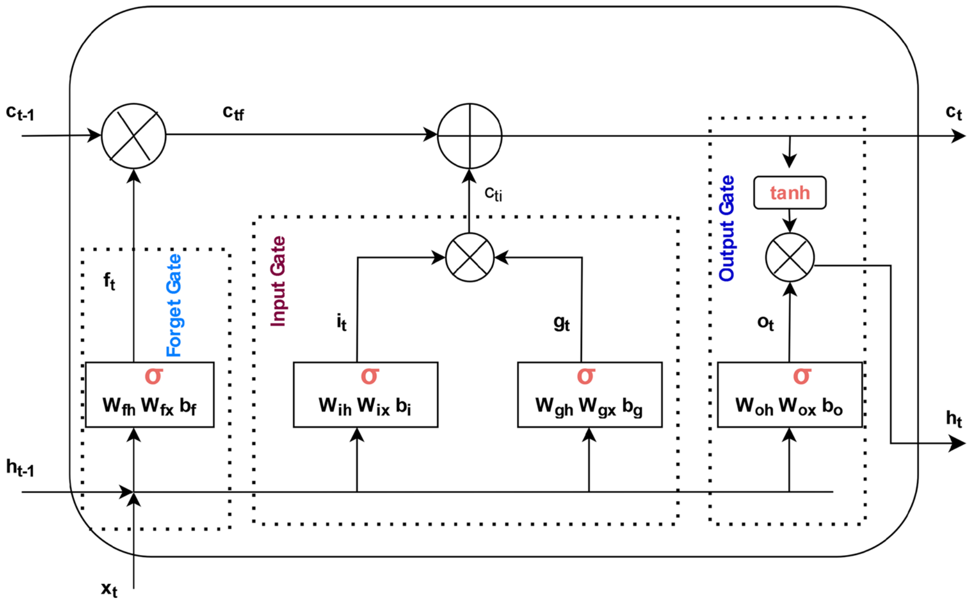

Fig. 7 Architecture of LSTM cell [57]

forward hidden states $\overrightarrow{h}_t = (\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_n)$ and backward hidden states $\overleftarrow{h}_t = (\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_n)$. The resulting encoded vector is constructed by concatenating the final forward and backward outputs, denoted as $Y = [\overrightarrow{h}_t, \overleftarrow{h}_t]$.

$$\begin{cases} \overrightarrow{h}_t = \sigma(W_{\overrightarrow{h}x} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \\ \overleftarrow{h}_t = \sigma(W_{\overleftarrow{h}x} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\ Y_t = W_{y\overrightarrow{h}} \overrightarrow{h}_t + W_{y\overleftarrow{h}} \overleftarrow{h}_t + b_y \end{cases} \tag{9}$$

Y denotes the output sequence of the initial hidden layer, expressed as $(y_1, y_2, \ldots, y_t, \ldots, y_n)$. The BiLSTM layer output is transferred to the dropout layer to improve model resilience and prevent overfitting. This dropout layer randomly deactivates neurons during training, encouraging more generalized representations and reducing reliance on specific patterns.

3.6 Dropout layer

Dropout is a strategy that prevents neural networks from relying too heavily on individual neurons or groups of neurons, enabling the network to maintain accuracy even when specific input is missing. The proposed architecture includes two dropout layers, each with a dropout rate of 0.5, after each BiLSTM layer. This strategic integration seeks to reduce overfitting and improve the model generalization abilities while training. Dropout was first applied to fully connected layers by [58], demonstrating its effectiveness in reducing overfitting. The output of the BiLSTM layer (Y) is applied to dropout, and then it is defined as follows:

$$Z = r{\star}b(W^A Y) \tag{10}$$

where $Y = [y_1, y_2, \ldots, y_n]^A$ is input to the dense layer, $W \in \mathbb{R}^{u \times v}$ is a weight matrix, and r is a binary vector whose elements are independently drawn from a bernoulli distribution. The BiLSTM output, enhanced by a dropout layer by mitigating overfitting, is subsequently fed into a dense layer. The dense layer predicts class probabilities using softmax function.

3.7 Dense layer

The proposed model includes a fully connected neural network with a softmax activation function. This layer establishes connections between all neurons from the preceding layer to the current one. Utilizing the softmax activation function, the model transforms a real-numbered vector into a probability distribution spanning multiple classes. The output of dropout layer passed to the dense layer containing softmax function. The input vector Z is $Z = (z_1, z_2, \ldots, z_n)$, where total number of categories deonoted as n, the softmax function calculates the probability p_i for each class i through the following formula:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{11}$$

where e^{z_i} represents the exponential of the i^{th} element of the input vector. This exponential transformation ensures that all values are positive. p_i' is the probability that the input belongs to class i, and the denominator $\sum_{j=1}^{n} e^{z_j}$ calculates the sum of exponentials for all classes. The softmax function generates a higher likelihood of categories with higher scores and a lower possibility of categories with lower scores, transforming raw scores into probability distributions over the classes. The softmax layer produces class probabilities, while the CTC loss ensures that the predicted sequence aligns correctly with the ground truth, making it a crucial component in sequence-to-sequence tasks like speech recognition.

3.8 CTC LOSS

CTC loss is used as an objective function to train the proposed model, which permits end-to-end training without the necessity for frame-level alignment between input and target labels. A single set of label tokens at each time step can be denoted as χ through the utilization of CTC, where the sequence of size-T produced by the temporal module constitutes the output marked with the blank symbols ϕ and consecutive symbols are repeated. We can define a function to remove the adjacent character and the blank symbol denoted as function $F : (\chi \cup \{\phi\})^* \rightarrow \chi^*$ because blank symbol may come in the processed string. The probability of observing a labeled sequence α can be computed by summing over this label as $\gamma(\alpha|\beta) = \sum_{u \in \mathcal{F}^{-1}(\alpha)} \gamma(u_1|\beta) \dots \gamma(u_T|\beta)$, considering all possible alignments. The conventional CTC loss, denoted as L_{CTC}, is defined as follows:

$$\begin{cases} \gamma^{\text{ctc}}(\alpha|\beta) = \sum\limits_{\eta \in \mathcal{F}^{-1}(\alpha)} \gamma^{\text{ctc}}(\eta|\alpha) \\ = \sum\limits_{\eta \in \mathcal{F}^{-1}(\alpha)} \prod\limits_{t=1}^{T} \tau^{t}_{\rho_t} \\ L_{\text{ctc}} = -\ln \gamma^{\text{ctc}}(\alpha|\beta) \end{cases} \tag{12}$$

T_i represents the input time duration of the frame sequence, while ρ_i is the output label produced by softmax probability $\tau^{i}_{\rho_i}$, where ρ_i is chosen from the set {la, le, bn, bl, …, pl, blank} at frame t. The sequence path of CTC is defined as $\rho = (\rho_1, \rho_2, \dots, \rho_T)$, and α denotes the sentence label (ground truth). The set of all viable paths within CTC that can be mapped to the ground truth α is represented by $\mathcal{F}^{-1}(\alpha)$. CTC restricts auto-regressive connections to model dependencies between time steps in a label sequence. This conditional independence is obtained by ensuring that the model is unaffected by the marginal distributions created at each successive phase. Consequently, CTC often decoded using a beam search method to reintroduce label temporal dependencies, combining probabilities with a language model to achieve this.

3.9 Learning rate scheduler

This article present a dynamic learning rate scheduler for adjusting the learning rate during the training process. The learning rate is a hyper-parameter that defines the step size during each iteration. Reducing the learning rate as the training advances to improve model performance is advantageous. To adopt the variable learning rate, first, we find the maximum number of epochs to be required for training the model, then, the range of the maximum number of epochs is divided into three parts($epoch < EH1, EH1 > epoch < EH1, epoch > EH1$), a fixed learning rate of 0.0001 is employed for the first EH1 epochs initially.

Algorithm 3 Learning Rate Scheduler

```
Require: $initial\_lr(0.0001), max\_epochs(80)$
Ensure: $optimized\_learning\_rate$
 1: Initialize $current\_lr$ to $initial\_lr$
 2: Initialize $epoch$ to 0
 3: Create an empty list $lrs$ to store learning rates
 4: while $epoch < max\_epochs$ do
 5:     if $epoch \le$ EH1 then
 6:         Append $current\_lr$ to the $lrs$ list
 7:         return $current\_lr$
 8:     else if EH1 $< epoch <$ EH2 then
 9:         Update $current\_lr$ by multiplying it by $e^{\lambda_1}$
10:         Append $current\_lr$ to the $lrs$ list
11:         return $current\_lr$
12:     else
13:         Update $current\_lr$ by multiplying it by $e^{\lambda_2}$
14:         Append $current\_lr$ to the $lrs$ list
15:         return $current\_lr$
16:     end if
17: end while
18: return $lrs$
```

Subsequently, an exponential decay technique is applied, causing the learning rate to decrease exponentially with each epoch. This gradual reduction in the learning rate allows the model to adjust its weights more precisely. So, the learning rate is reduced exponentially by λ_1 for the epochs greater than $EH1$ to less than $EH2$ epoch. Then, learning rate is reset to the initial value of 0.0001 and decays exponentially by λ_2 for the epochs greater than EH2 to max_epochs. These adaptive adjustments have resulted in achieving a training accuracy of 98.85%. In our experiment, we observe that the proposed model requires max_epochs is 80 to achieve better performance. We have also set $EH1 = 15$, $EH2 = 30$, $\lambda_1 = -0.2$ and $\lambda_2 = -0.1$ experimentally (refer Algorithm 3). After completing max_epochs, the training loss and accuracy of the model reach a point where model have remained constant and did not show significant improvement or degradation. The graph for model evaluations with respect to the learning rate includes data points at 21 intervals, with sampling occurring every 4th epoch, including the last epoch, within the range from 1 to max_epochs. The visualization of graph is given in Fig. 8, which helps to understand how the learning rate influences the model convergence and performance as the training progress.

Figure 8 depicts the relationship between loss, accuracy, and learning rate throughout training epochs. The graph presents data at intervals of every 4th epoch, up to a maximum of 80 epochs. Initially set at 0.0001, the learning rate remains constant for some epochs before gradually decreasing through exponential decay. The implementation of dynamic learning rate scheduler is given in algorithm 3. As the number of epochs increases, the learning rate decreases, and training and validation losses decrease as well, converging towards zero. Simultaneously, accuracy trends towards 1.00, indicating optimal model performance. Ultimately, when the learning rate approaches zero, the model achieves peak performance, with optimized loss and accuracy values.

3.10 Model evaluation

An exhaustive evaluation of the model performance is being carried out using the GRID dataset. This assessment encompasses various metrics, including WER, WRR, and monitoring loss and accuracy of the model throughout the training and validation phases.

3.10.1 Training and validation

Training is the process of teaching a model to make predictions by adjusting its parameters based on labeled data, while validation is the evaluation of the model performance on a separate dataset to ensure its accuracy and generalization capabilities. Our model undergoes an extensive training process spanning 80 epochs, employing the highly effective Adam optimizer to maximize its performance. Experimental dataset is partitioned into distinct segments to ensure a robust evaluation, allocating 80% of the data for rigorous training. The remaining 10% each is designated for the crucial tasks of validation and testing, allowing us to scrutinize the model capabilities and effectiveness meticulously. The model obtained 99.5% training accurcay, 98.8% validation accuracy while 1% training loss and 1.5% validation loss. The total number of examined frames can be computed as the product of the number of videos (V_d), frames per video (N_f), and subjects (T_s): total analyzed frames = $V_d \times N_f \times T_s$.

The Table 4 offers a comprehensive breakdown of the video dataset, encompassing the distribution ratio for the total data sample (Data Size), as well as metrics such as model training loss (Train Loss), training accuracy (Train Acc.), validation loss (Valid Loss), and validation accuracy (Valid Acc.), expressed in percentages. The last three columns of Table 4 indicates the percentage of data allocated for model training (Train), model validation (Valid), and model testing (Test). We have conducted model testing on a subset of 720 videos, representing 10% of the whole dataset (7200 videos), for assessing both WER and WRR. In contrast, Fig. 9 represents the loss and accuracy of model training and validation, enhancing visualization and understanding. The model demonstrates high training and validation accuracy and low loss, indicating its effectiveness in capturing patterns and enhancing its potential for accurate predictions.

3.10.2 WER and WRR

The WER and WRR is essential measures for determining the predictive power of the VSR model. These metrics measure the dissimilarity between the words recognized by the VSR model and spoken words in the visual input, which serves as the reference or ground truth. The WER is evaluated for the overlapping speakers, given in Eq. 13. The WER is calculated by the sum of the minimum number of insertions, substitutions, and deletions denoted as (M_i, M_s, M_d), respectively and divided by the total number of words in the ground truth. WRR serves as a complementary metric to WER, computed as 1-WER.

$$\mathrm{WER} = \frac{(M_i + M_s + M_d)}{N} \tag{13}$$

The Table 5 illustrates the comparison of WER and WRR between the proposed model with the existing models, which contains significant observations on the overall effectiveness of the proposed model. Specifically, when tested on overlapped speakers, the model achieves a WER of 1.11% and a WRR of 98.89%. These results highlight the model effectiveness in differentiating spoken words in challenging scenarios involving overlapping. The results revealed that the proposed model outperforms the existing VSR models, demonstrating better performance in terms of both WER and WRR.

4 Conclusion and future scope

This study has presented a comprehensive deep-learning approach for word-level VSR on the GRID dataset. The proposed methodology showcases the development of a more robust system, leading to significant enhancements compared to existing VSR models. This work optimized the 3D-CNN architecture to extract detailed local information from the lip region. The incorporation of BiLSTM introduced temporal relationship, enriching context embeddings that are crucial for VSR. The proposed dynamic LRS plays a vital role in adjusting the learning rate during model training, considerably improving its performance. Additionally, CTC loss measures the difference between actual and predicted out-

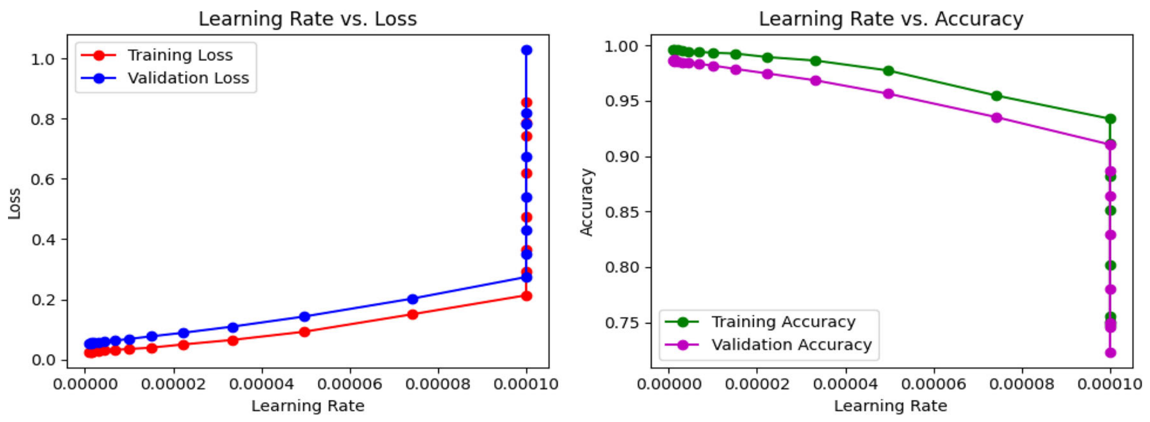

Fig. 8 Training, validation accuracy and loss with learning rate

Table 4 Metrics for loss and accuracy based on data split ratio

Data size	Train loss (%)	Train Acc. (%)	Valid loss (%)	Valid Acc. (%)	Train (%)	Valid (%)	Test (%)
7200	1	99.5	1.5	98.8	80	10	10

Fig. 9 Model loss and accuracy with training, validation

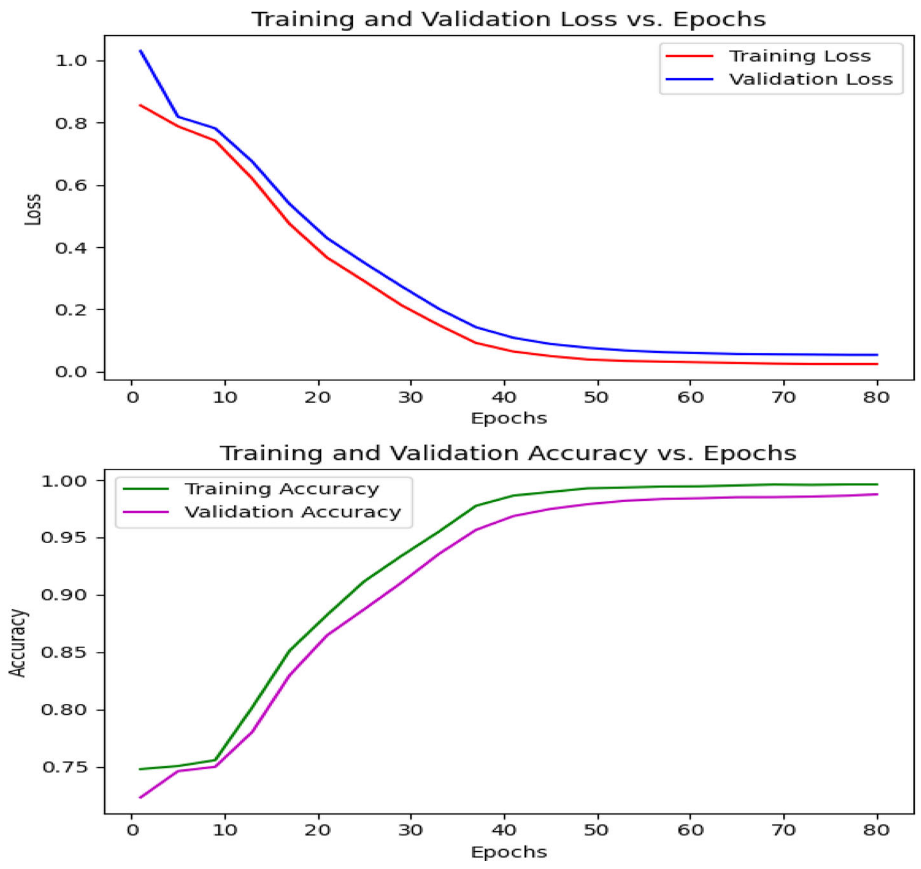

Table 5 WER and WRR comparison with existing methods

Dataset	Methodology proposed by authors	Overlapped speakers	
		WER%	WRR%
GRID	Assael et al. [3]	4.80	95.20
GRID	Chung et al. [19]	3.00	97.00
GRID	Wand et al. [44]	16.70	83.30
GRID	Xu et al. [4]	2.90	97.10
GRID	Wand et al. [1]	15.30	84.70
GRID	Rastogi et al. [45]	2.70	97.30
GRID	Liu et al. [33]	4.50	95.50
GRID	Sarhan et al. [47]	3.30	96.70
GRID	Huang et al. [35]	54.19	45.81
GRID	He et al. [37]	6.20	93.8
GRID	Vayadande et al. [48]	7	93
GRID	Xue et al. [50]	2.30	97.70
GRID	Proposed Model	1.11	98.89

puts, which improves the model training and performance. The model is verified through the experiment and compared with existing techniques, obtained 1.11% WER and 98.89% WRR for overlapped speakers, which outperforms than the existing state-of-the-art methods. The future endeavor involves leveraging multi-modal features to articulate lip motion, prioritizing speaker independence by incorporating a self-attention mechanism.

Acknowledgements This research received institutional support from the National Institute of Technology Tiruchirappalli. We thank the institute for providing access to the high-performance supercomputer (PARAM PORUL) for this research work.

Author contributions In the development of this article, V.C. authored the manuscript, designed the figures and tables, and executed the crucial experiments. Concurrently, S.D. diligently reviewed the paper, ensuring its accuracy, and provided valuable guidance, contributing insightful suggestions to enhance the overall quality of the work.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Wand, M., Schmidhuber, J., Vu, N.T.: Investigations on end-to-end audiovisual fusion, In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3041–3045, IEEE, (2018)
2. Afouras, T., Chung, J.S., Senior, A., Vinyals, O., Zisserman, A.: Deep audio-visual speech recognition. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 8717–8727 (2018)
3. Assael,Y.M., Shillingford, B., Whiteson, S., De Freitas, N.: Lipnet: End-to-end sentence-level lipreading, arXiv preprint arXiv:1611.01599, (2016)
4. Xu, K., Li, D., Cassimatis, N., Wang, X.: Lcanet: End-to-end lipreading with cascaded attention-ctc, in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 548–555, IEEE, (2018)
5. Yousaf, K., Mehmood, Z., Saba, T. , Rehman, A., Rashid, M., Altaf, M., Shuguang, Z.: A novel technique for speech recognition and visualization based mobile application to support two-way communication between deaf-mute and normal peoples. Wirel. Commun. Mob. Comput., **2018**, (2018)
6. Thanda, A., Venkatesan, S.M.: Multi-task learning of deep neural networks for audio visual automatic speech recognition, arXiv preprint arXiv:1701.02477, (2017)
7. Kumar, L.A., Renuka, D.K., Rose, S.L., Shunmugapriya, M.: Attention based multi modal learning for audio visual speech recognition. In: 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST), 1–4, IEEE, (2022)
8. Liu, Y.-F., Lin, C.-Y., Guo, J.-M.: Impact of the lips for biometrics. IEEE Trans. Image Process. **21**(6), 3092–3101 (2012)
9. Liu, M., Wang, L., Lee, K.A., Zhang, H., Zeng, C., Dang, J.: Deeplip: A benchmark for deep learning-based audio-visual lip biometrics. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 122–129, IEEE, (2021)
10. Sankar, S., Beautemps, D., Hueber, T.: Multistream neural architectures for cued speech recognition using a pre-trained visual feature extractor and constrained ctc decoding. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 8477–8481, IEEE, (2022)
11. Wang, X., Han, Z., Wang, J., Guo, M.: Speech recognition system based on visual feature for the hearing impaired. In 2008 Fourth International Conference on Natural Computation, **2**, 543–546, IEEE, (2008)
12. Hassanat, A.B.: Visual passwords using automatic lip reading, arXiv preprint arXiv:1409.0924, (2014)
13. Sadeghi, M., Leglaive, S., Alameda-Pineda, X., Girin, L., Horaud, R.: Audio-visual speech enhancement using conditional variational auto-encoders. IEEE/ACM Trans. Audio, Speech, Lang. Process. **28**, 1788–1800 (2020)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer

society conference on computer vision and pattern recognition. CVPR 2001, **1**, I–I, Ieee, (2001)
15. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
16. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning, 369–376, (2006)
17. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, (2014)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
19. Chung, J.S., Senior, A., Vinyals, O., Zisserman, A.: Lip reading sentences in the wild. In 2017 IEEE conference on computer vision and pattern recognition (CVPR), 3444–3453, IEEE, (2017)
20. Yang, S., Zhang, Y., Feng, D., Yang, M., Wang, C., Xiao, J., Long, K., Shan, S., Chen, X.: Lrw-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. In: 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019), 1–8, IEEE, (2019)
21. Matthews, I., Cootes, T.F., Bangham, J.A., Cox, S., Harvey, R.: Extraction of visual features for lipreading. IEEE Trans. Pattern Anal. Mach. Intell. **24**(2), 198–213 (2002)
22. Liu, H., Zhang, X., Wu, P.: Regression based landmark estimation and multi-feature fusion for visual speech recognition. In: 2015 IEEE International Conference on Image Processing (ICIP), 808–812, IEEE, (2015)
23. Afouras, T., Chung, J.S., Zisserman, A.: Lrs3-ted: a large-scale dataset for visual speech recognition, arXiv preprint arXiv:1809.00496, (2018)
24. Rekik, A., Ben-Hamadou, A., Mahdi, W.: A new visual speech recognition approach for rgb-d cameras. In: Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part II 11, 21–28, Springer, (2014)
25. Nemani, P., Krishna, G.S., Ramisetty, N., Sai, B.D.S., Kumar, S.: Deep learning based holistic speaker independent visual speech recognition, IEEE Transactions on Artificial Intelligence, (2022)
26. Petridis, S., Shen, J., Cetin, D., Pantic, M.: Visual-only recognition of normal, whispered and silent speech. In: 2018 ieee international conference on acoustics, speech and signal processing (icassp), 6219–6223, IEEE, (2018)
27. Anina, I., Zhou, Z., Zhao, G., Pietikäinen, M.: Ouluvs2: A multi-view audiovisual database for non-rigid mouth motion analysis. In 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG), **1**, 1–5, IEEE, (2015)
28. Cooke, M., Barker, J., Cunningham, S., Shao, X.: An audio-visual corpus for speech perception and automatic speech recognition. The J. Acoust. Soc. Am. **120**(5), 2421–2424 (2006)
29. Petridis, S., Pantic, M.: Deep complementary bottleneck features for visual speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2304–2308, IEEE, (2016)
30. Garg, A., Noyola, J., Bagadia, S.: Lip reading using cnn and lstm, Technical report, Stanford University, CS231 n project report, (2016)
31. Wang, C.: Multi-grained spatio-temporal modeling for lip-reading, arXiv preprint arXiv:1908.11618, (2019)
32. Stafylakis, T., Tzimiropoulos, G.: Deep word embeddings for visual speech recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4974–4978, IEEE, (2018)
33. Liu, J., Ren, Y., Zhao, Z., Zhang, C., Huai, B., Yuan, J.: Fastlr: Non-autoregressive lipreading model with integrate-and-fire. In: Proceedings of the 28th ACM International Conference on Multimedia, 4328–4336, (2020)
34. Debnath, S., Roy, P., Namasudra, S., Crespo, R.G.: Audio-visual automatic speech recognition towards education for disabilities. J. Autism Dev. Disord., 1–14, (2022)
35. Huang, H., Song, C., Ting, J., Tian, T., Hong, C., Di, Z., Gao, D.: A novel machine lip reading model. Proced. Comput. Sci. **199**, 1432–1437 (2022)
36. Ma, P., Martinez, ., Petridis, S., Pantic, M.: Towards practical lipreading with distilled and efficient models. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7608–7612, IEEE, (2021)
37. He, L., Ding, B., Wang, H., Zhang, T.: An optimal 3d convolutional neural network based lipreading method. IET Image Process. **16**(1), 113–122 (2022)
38. Rahmani, M.H., Almasganj, F.: Lip-reading via a dnn-hmm hybrid system using combination of the image-based and model-based features. In: 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), 195–199, IEEE, (2017)
39. Zhao, X., Yang, S., Shan, S., Chen, X.: Mutual information maximization for effective lip reading. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), 420–427, IEEE, (2020)
40. Ezz, M., Mostafa, A.M., Nasr, A.A.: A silent password recognition framework based on lip analysis. IEEE Access **8**, 55354–55371 (2020)
41. Fenghour, S., Chen, D., Guo, K., Xiao, P.: Lip reading sentences using deep learning with only visual cues. IEEE Access **8**, 215516–215530 (2020)
42. Martinez, B., Ma, P., Petridis, S., Pantic, M.: Lipreading using temporal convolutional networks. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6319–6323, IEEE, (2020)
43. Wand, M., Koutník, J., Schmidhuber, J.: Lipreading with long short-term memory. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6115–6119, IEEE, (2016)
44. Wand, M., Schmidhuber, J.: Improving speaker-independent lipreading with domain-adversarial training, arXiv preprint arXiv:1708.01565, (2017)
45. Rastogi,A. Agarwal,R., Gupta,V., Dhar,J., Bhattacharya,M.: Lrne-unet: An attention based deep architecture for lipreading from multitudinous sized videos, in 2019 International Conference on Computing, Power and Communication Technologies (GUCON), 1001–1007, IEEE, 2019
46. Luo, M., Yang, S., Shan, S., Chen, X.: Pseudo-convolutional policy gradient for sequence-to-sequence lip-reading. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), 273–280, IEEE, (2020)
47. Sarhan, A.M., Elshennawy, N.M., Ibrahim, D.M.: Hlr-net: a hybrid lip-reading model based on deep convolutional neural networks. Comput. Mater. & Contin. **68**(2), 1531–1549 (2021)
48. Vayadande, K., Adsare, T., Agrawal, N., Dharmik, T., Patil, A., Zod, S.: Lipreadnet: A deep learning approach to lip reading. In: 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), 1–6, IEEE, (2023)
49. NadeemHashmi, S., Gupta, H., Mittal, D., Kumar, K., Nanda, A., Gupta, S.: A lip reading model using cnn with batch normalization. In: 2018 eleventh international conference on contemporary computing (IC3), 1–6, IEEE, (2018)
50. Xue, F., Yang, T., Liu, K., Hong, Z., Cao, M., Guo, D., Hong, R.: Lcsnet: End-to-end lipreading with channel-aware feature selection. ACM Trans. Multimed. Comput., Commun. Appl. **19**(1s), 1–21 (2023)
51. Almajai, I., Cox, S., Harvey, R., Lan, Y.: Improved speaker independent lip reading using speaker adaptive training and deep neural

networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2722–2726, IEEE, (2016)
52. Cooke, M., Barker, J., Cunningham, S., Shao, X.: The grid audio-visual sentence corpus. https://spandh.dcs.shef.ac.uk/gridcorpus/, (2006)
53. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), 807–814, (2010)
54. Agarap, A.F.: Deep learning using rectified linear units (relu), arXiv preprint arXiv:1803.08375, (2018)
55. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision **115**, 211–252 (2015)
56. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml, **30**, 3, Atlanta, GA, (2013)
57. Khalil, K., Dey, B., Kumar, A., Bayoumi, M.: A reversible-logic based architecture for long short-term memory (lstm) network. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 1–5, IEEE, (2021)
58. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, (2012)