Lip Reading using Audio-Visual Data by using Deep Learning Methods

1st Himanshee
dept. MSDA
San Jose, CA
himanshee@sjsu.edu

2nd Mounica Ayalasomayajula
dept. MSDA
San Jose, CA
mounica.ayalasomayajula@sjsu.edu

3rd Mythri
dept. MSDA
San Jose, CA
mythri.muchmari@sjsu.edu

4th Richa Sharma
dept. MSDA
San Jose, CA
richa.sharma@sjsu.edu

***Abstract*—The purpose of this project is to generate audio for videos where the audio track is unavailable or not understandable clearly.**

***Index Terms*—component, Lipread, RNN Models, Attentions, Real Time pre-processing**

I. Motivation

As per recent research, Lip-reading is a critical application in the field of AI. However, traditional lip-reading systems that rely solely on visual information have limitations, especially in noisy environments. We investigate the task of lip to speech synthesis, or learning to generate natural speech from only a speaker's lip movements. Deep Lip reading is the method of utilizing deep neural networks to extract speech from a video of a silent talking face. Our dataset consists of a collection of image sequences, or low-quality films, where each one depicts a speaker uttering a single word or phrase. The objective is to categorize these sequences. The fact that sequence lengths and, consequently, the amount of characteristics per sequence, vary greatly is one of the key problems that prevents using previous approaches. As a result, different techniques for recording temporal information were employed to take into account all input aspects.Below fig (1) shows the project flow which we have followed.

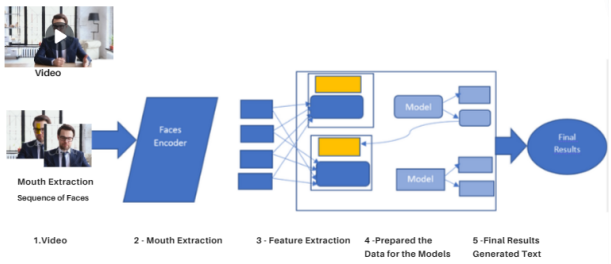

Fig. 1. Project Flow Diagram

II. Background and Literature Review

A recent study (Easton Basala, 1982) found that people generally have poor lip reading skills. Even on the limited subset of 30 monosyllabic words, the deaf person could only achieve approx 17 to 12 percent accuracy, and on 30 compound words she could only achieve approx 21 to 11 percent accuracy. It is important to automate the lip reading. Machine lip readers are practically large with applications such as enhanced hearing aids, silent dictation in public spaces, security, speech recognition in noisy environments, silent film processing and subtitling of silent films and videos etc.

In the paper 'Learning Individual Speaking Styles for Accurate Lip to Speech Synthesis ' authors have used Lip2Wav models to and specifically solved the problem by focusing on individual speakers for video data. They evaluated their model with extensive quantitative metrics and human studies.

In the paper 'AuthNet: A Deep Learning based Authentication Mechanism using Temporal Facial Feature Movements' a new authentication mechanism that combines facial recognition with the unique movements of a person's face while uttering a password, known as temporal facial feature movements. This approach aims to enhance security and overcome language barriers, as users can set a password in any language.

The study "Deep Learning for Visual Speech Analysis: A Survey" offers a thorough examination of the developments and difficulties in the field of visual speech analysis using deep learning techniques. The writers discuss a range of topics related to visual speech analysis, such as speaker identification, lip reading, emotion recognition, and speech recognition. In order to extract useful information from visual speech data, the survey investigates the application of deep learning models including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their derivatives.

III. Methodology

Our visual input is a short video sequence of face images. The model must learn to extract and process the fine grained sequence of lip movements. The main process consists of two steps: Step-1: Taking advantage of a quiet talking video's image frames to extract visual and temporal features. Step 2: Extract the speech by lip reading to generate more appropriate and meaningful audios for the given visuals. We methodically assess how well state-of-the-art data augmentation techniques, temporal models, and other training techniques, such as self-distillation and the use of word boundary indications, perform.Below are the detailed steps about all the methods and phases of this project.

A. Data Collection

Collected publicly available free dataset MIRACAL-VC1 and tried to understand the structure of the dataset such as, the format of data, size of the data, features, dimensions of data, etc. As collected dataset is publicly available hence there is no any permission or licence required.

And architecture of the data is that it includes information about 15 speakers. 15 folders have been created for all the 15 speakers.Under each speaker's folders there are two separate components phrases and words.Figure() shows final architecture of our dataset.

The characteristics of the MIRACL-VC1 dataset that it was produced using the speech of 15 individuals.Each speakers spoke ten words and ten sentences ten times, for a total of 3000 instances (15 x 20 x 10). Each instance consists of a series of 640 x 480 pixel color and depth images.To comply with the pre-trained VGGNet model, we just use the colored portion of the image and ignore the depth component.The words and phrases in the dataset are shown in fig(2).

Fig. 2. Dataset Architecture

B. Data Pre-processing

We have done our data preprocessing in two steps, one for the input data -sequence of frames and step two is to preprocess the target data – Labels.

- Loading the face detetector: Used the Dlib face detector.Dlib offers a face detector model that has already been trained to find faces in pictures.
- Face Detection: We use the face detector model to find faces in image or video frames once it has loaded. After analyzing the image, the face detector outputs the bounding box coordinates of each face it finds.
- Mouth Extraction: The coordinates of the mouth region can be obtained from the faces that have been detected. A number of distinct landmarks, such as the corners of the lips, serve as the traditional delineators of the mouth region. These facial landmarks can be located using the pre-trained facial landmark detector that Dlib offers.
- Cropping the mouth area: Using the mouth landmarks, we crop the picture or frame to simply show the area around the mouth. Using the locations of the landmarks, a rectangle zone around the mouth is defined and extracted in this stage.
- Make image sequences: Using video, we extract the mouth region from each frame of the film by repeating the procedures above for each frame. The mouth movement over time will be represented by a series of photos as a result of this.
- Data Augumentation: We employed data augmentation to fictitiously enhance the data size in order to address this issue. The original image has been changed in the following two ways as part of our data augmentation:(i) When cropping, nudged the crop area randomly in both the horizontal and vertical directions. (ii) By fluctuating the image's brightness or contrast at random.
- Phrases and words: Added our target phrases and words.
- Image Feature Extraction using VGG16: Used VGG net to extract features from these image sequences. The image sequences can be passed through the network to extract features after the VGG16 model has been loaded and the image sequences have undergone preprocessing. Multiple convolutional layers make up the VGG16 model, which gradually learns and extracts visual information from the input images. The output of these convolutional layers acts as a learnt feature-based representation of the input image.The output of these convolutional layers acts as a learnt feature-based representation of th input image with the dimension of N*16*4096.
- Preprocessing Target labels: we performed word embeddings using a tokenizer. This step allowed us to preprocess the textual data by converting it into tokens. Each token represents a specific word in the text. Consequently, these tokens will be converted into numerical representations, enabling computational analysis. In order to generate tokenized sequences, we incorporated the ¡start¿ and ¡end¿ tokens into the tokenizer. These additional tokens serve as markers for the beginning and end of each sequence, Moreover, we ensured that the sequences were padded with a vector length of 50, enabling consistent dimensions for further processing and modeling.
- Data preparation before Modeling: Prepared the extracted features as input features and the target text as input for the encoders and decoders.
- Real Time Processing:We did real time data processing using FPS, it is nothing but "frame per second". Simply means FPS value indicates how many frames per second

a lip-reading system can process or analyze. In general, a higher FPS score indicates faster processing, smoother real-time performance, and the ability of the system to deliver lip-reading results with minimal latency.

C. Models Implemented

This section describes the numerous approaches we used to employ i.e. RNN with attention mechanism, LSTM, GRU and self-attention with fully connected layersm to try to solve the lip reading challenge.We have used same paramerters for all the models model parameters such as batch size, epochs, learnable parameters etc.

1) RNN+Attention: The below architecture as shown in figure(3) is a combination of RNN with having attention layers.It has few learnable parameters to map the extracted features from using pre trained VGGnet model as a cropped image of mouth from the sequence of faces image and spoken words to train or learn the simple RNN models with adding attention mechanism and dropout, which has improved the performance of the model. This model has used less number of learnable parameters 1,223,266 compared to other models.

Input encoding:Convert the input text to a numeric representation using techniques such as word embedding (sequence to sequence , GloVe, etc.) and character embedding. This step obtains the semantic meaning of the words or characters in the input.

Simple RNN:Pass the encoded input to the SimpleRNN layer. SimpleRNN is a type of iterative neural network that processes continuous data by maintaining hidden states that are updated at each time step. Get contextual information from previous inputs.

Attention Mechanism:Applies an attention mechanism to the output of the SimpleRNN layer. Attention mechanisms assign different weights to different parts of the input sequence, allowing the model to focus on the most relevant information. This allows the model to understand the context better and produce more accurate answers.

Context Vector:Compute the weighted sum of the SimpleRNN outputs using the attention weights. This aggregation produces a context vector representing the most important parts of the input sequence based on the attention mechanism.

Output Layer:Connect the context vector to the dense (fully connected) layer, followed by the output layer. The output layer maps the context vector to the desired output format, such as probability distributions or continuous values across different answer options. Fig(3) shows the architecture of RNN+Attention model.

Training The Model:Based on your specific task and output format, train your model with the appropriate optimization algorithm (Adam, RMSprop, etc.) and loss function (categorical cross entropy, mean squared error, etc.).

2) GRU: Another RNN architecture with fewer parameters than LSTM is the GRU (Gated Recurrent Unit) model. The mapping between lip area features and voice output is learned with the help of the GRU (Gated Recurrent Unit) model.

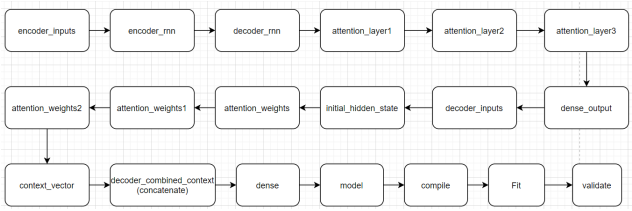

Fig. 3. Rnn+Attention Model Architecture

The GRU model functions as an encoder that progressively processes and records the temporal relationships of the input lip features. The model's GRU layer uses an input sequence to produce a sequence of hidden states. Each hidden state is a representation of the input that has been learned at a specific time step. To encode the temporal context, the encoder's goal is to extract pertinent information from the lip characteristics and condense it into a fixed-length representation (state-h).

The decoder component of the GRU model creates the corresponding voice output sequence using the encoded representation (state-h). The model's decoder RNN layer generates the output for the current time step using the anticipated output from the previous time step as input. Until the whole output sequence is generated, this process is repeated. The attention mechanism is also used by the decoder RNN layer to concentrate on pertinent segments of the input sequence while producing the output. This aids the model in matching the proper speech components with the appropriate lip features. The model's architecture is shown as below in fig(4):

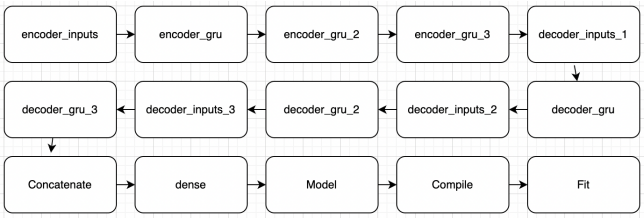

Fig. 4. GRU Model Architecture

The GRU model learns to capture the temporal connections and match the visual cues with the proper speech components by being trained on a sizable dataset of lip area features and their related speech outputs. The model's objective is to produce precise and understandable speech using the input's observed lip movements.

3) LSTM: The LSTM (Long Short-Term Memory) model is a different kind of recurrent neural network that can be utilized in the context of a lip-to-speech synthesis project to learn the mapping between lip area characteristics and voice output.

The LSTM model functions as an encoder that successively processes the input lip features and records their temporal repercussions. The model's LSTM layer uses an input sequence to produce a sequence of hidden states. Each hidden state is a representation of the input that has been learned at a specific time step. To encode the temporal context,

the encoder's goal is to extract pertinent information from the lip characteristics and condense it into a fixed-length representation (state-h).

The LSTM model's decoder section uses the encoded representation (state-h) to produce the appropriate voice output sequence. Similar to the encoder, the decoder LSTM layer generates the output for the current time step using the predicted output from the previous time step as input. Until the whole output sequence is generated, this process is repeated. The attention mechanism may be included in the decoder LSTM layer to concentrate on important segments of the input sequence while producing the output. This aids the model in matching the proper speech components with the appropriate lip features.

In order to reduce the category cross-entropy loss between the predicted output and the ground truth speech labels, the model learns to modify its parameters during training.LSTM shown in fig(5) below:

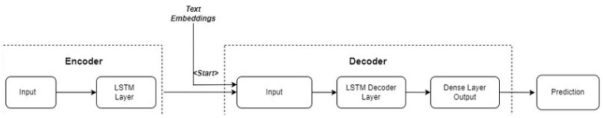

Fig. 5. LSTM Model Architecture

4) Self-Attention+FC: A feed-forward network and self-attention layers are combined in the Self-Attention+FC (Fully Connected) model to figure out how to transfer lip region information to speech output.

A process called self-attention enables the model to weigh the significance of various input sequence components when producing the output. The associations between various lip features throughout the sequence are used by the self-attention layers to construct attention weights. Self-attention can assist the model in focusing on particular lip motions that are essential for producing appropriate speech in the setting of lip-to-speech synthesis.

Multiple parallel self-attention layers are frequently utilized in the Self-Attention+FC model to capture various facets or perspectives on the features of the lip region. The input sequence is attended to differently by each self-attention head, enabling the model to extract a variety of complementing information. In a layer normalization step, the outputs of the multi-head self-attention layers are mixed. By averaging the results across various attention heads, layer normalization aids in the stabilization of the training process.

The output is then transmitted through a feed-forward network after the self-attention layers. The model may learn intricate non-linear mappings between the lip features and the speech output thanks to the feed-forward network's one or more fully connected (dense) layers. The feed-forward network's hidden layers may use activation functions like ReLUs (Rectified Linear Units) to induce nonlinearity. When producing speech output, the Self-Attention+FC model makes use of the self-attention mechanism to record dependencies between various lip features and pay attention to significant visual cues. Self Attention with FC model architecture shown below in fig(6).

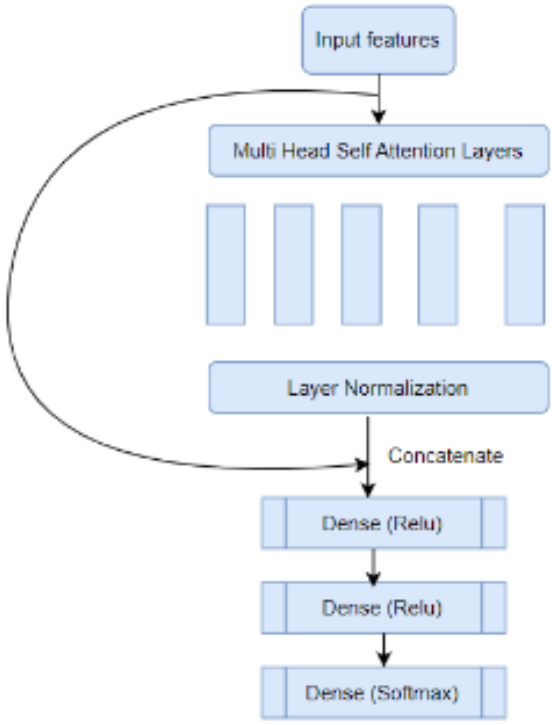

Fig. 6. SelfAttention+FC Model Architecture

Using identical model parameters shown as fig(7), we ensured a fair and objective comparison between the four models. This approach allowed us to identify the best performing model for a given dataset.

RNN+ATTENTION MODEL	
Accuracy	94.3 %
Loss	0.178
No. of learnable Paramters	1,223,266
Dropout	0.5
Epoches	100
Learning Rate	0.01
Optimizer	adam
Btach size	64
Validation slpit	20%

MULTI-HEAD SELF ATTENTION +FC MODEL	
Accuracy	80.02%
Loss	0.6096
No. of learnable Paramters	44,808,754
Epoches	100
Learning Rate	0.01
Optimizer	adam
Btach size	64
Validation slpit	20%

STACKED GRU MODEL	
Accuracy	87.95 %
Loss	0.6136
No. of learnable Paramters	4,881,458
Epoches	100
Learning Rate	0.01
Optimizer	adam
Btach size	64
Validation slpit	20%

LSTM MODEL	
Accuracy	84.99 %
Loss	0.461
No. of learnable Paramters	10,617,906
Epoches	100
Learning Rate	0.01
Optimizer	adam
Btach size	64
Validation slpit	20%

Fig. 7. Models Parameters

D. Train and Validate the Model

To divide the data into train and test sets, we use the train test split function from the sklearn.model selection module is utilized. Separate inputs are split up for the encoder, decoder, and target (ground truth). A test size of 0.2 is used for the

split, allocating 20 percent of the data to the test set. For reproducibility, the random state parameter is set to 42.

The original split's training data is further separated into train and validation sets. On the training data, the train test split function is applied once more. This time, 20 percent of the training data is distributed to the validation set due to the split being carried out with a test size of 0.2. For consistency with the initial split, the random state parameter is set to 42.

IV. Model Evaluation and Results

Comparing the LSTM model to the other models, the LSTM model has a marginally superior precision score and F1-score than stacked GRU. The RNN+attention model has a lower F1-score but a better validation accuracy, suggesting that it may perform well for majority classes but poorly for minority classes.

When analyzing these results and deciding which model is most appropriate for a given use case, it is crucial to take into account the precise needs of the work at hand.

Although precision and recall are both crucial, the importance of each depends on the application. A balance must be struck based on the unique requirements of the task. Precision puts an emphasis on reducing false positives, which is crucial when false positives can have serious repercussions (like in a medical diagnosis). When false negatives are expensive (as they are when spotting fraud, for example), recall places a strong emphasis on obtaining as many real positives as possible. By categorizing predictions, the confusion matrix gives a more detailed view of the model's performance, allowing for additional analysis and insights.Fig(8) shows the models performance comparison.

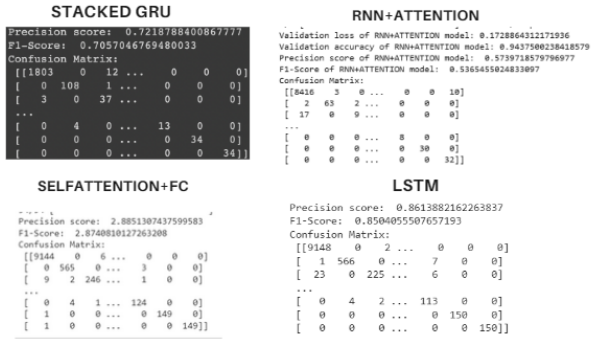

Fig. 8. Model Performance Comparison

V. Challenges and Key Learnings

Data preprocessing: Processing image frames and reshaping the data. Sequence to Sequence: Working with temporal sequence data and working with self attention models. Handling overfitting and underfitting issues. Real time Lip Reading Model: Working with real time data / Unseen data.

VI. Discussion and Future Improvements

Our project have opportunity for future scope such as making this application robust to real time data.

Real-Time Integration: Merge our lip-reading application with real-time video streaming platforms or communication applications to enable live lip reading. This will allow users to receive instant transcription or translation of spoken words during video calls or presentations.

Speech Synthesis: Combine our lip-reading technology with text-to-speech synthesis. By generating spoken output from the lip-reading data, this application must be able to assist individuals who have speech disabilities, enabling them to communicate more easily.

Performance: RNN with attention model has given the best accuracy as compared to other models.

Memory: RNN with attention model has the least number of parameters.

Exploring more models: Working with transformers , GCN etc.

Increasing training: Training the model for more epochs and with more data. Increasing the vocabulory.

Acknowledgment

We would like to express our sincere appreciation and gratitude to everyone who contributed to the completion of this report.

References

[1] Prajwal, K. R., Mukhopadhyay, R., Namboodiri, V. P., Jawahar, C. V. (2020). Learning Individual Speaking Styles for Accurate Lip to Speech Synthesis. https://doi.org/10.1109/cvpr42600.2020.01381.

[2] Faisal, M. (2018, February 15). Deep Learning for Lip Reading using Audio-Visual Information for Urdu Language. arXiv.org. https://arxiv.org/abs/1802.05521.

[3] Sheng, C. (2022, May 22). Deep Learning for Visual Speech Analysis: A Survey. arXiv.org. https://arxiv.org/abs/2205.10839.

[4] Raghavendra, M. (2020b, December 4). AuthNet: A Deep Learning based Authentication Mechanism using Temporal Facial Feature Movements. arXiv.org. https://arxiv.org/abs/2012.02515v2.

[5] Zhang, X. (2022, October 18). Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages. arXiv.org. https://arxiv.org/abs/2210.09984v1

[6] Torfi, A. (2018). SpeechPy - A Library for Speech Processing and Recognition. Journal of Open Source Software. https://doi.org/10.21105/joss.00749