

LipNet’s Deep Neural Approach to Sentence-Level Lip Reading

Balavanthapu Sanjeev

Department of Computer Science, Rice University

sb200@rice.edu

Abstract

The task of generating textual descriptions from video content presents a significant challenge, particularly when audio is either absent or of poor quality. Traditional approaches have largely focused on lipreading techniques that decode speech on a word-by-word basis. However, these methods often fall short in accurately capturing the full scope of spoken language. In our project, we explore the implementation of lipnets, which employ a spatio-temporal convolutional neural network (CNN) to process the visual input derived from the mouth region’s movements. The CNN’s output is then fed into a recurrent neural network (RNN), which decodes the visual features into textual sentences, bridging the gap between visual cues and linguistic content. To achieve this task, we used an Extract of the original GRID dataset (high-quality audio and video (facial) recordings of 1000 sentences spoken by each of 34 talkers (18 male, 16 female), for a total of 34000 sentences. Sentences are of the form “put red at G9 now”). We train the model with the video and annotations and evaluate its performance relative to the original annotations using Connectionist Temporal Classification (CTC) loss for the particular video. We show that an end-to-end sentence-level lipreading solution is attainable by this approach.

1. Introduction

Speech-to-text communication has been greatly useful for people who have had hearing disabilities for a long time. However good they might be, hearing-impaired people achieve an accuracy of only $17 \pm 12\%$ and for compound words is $21 \pm 11\%$ for 30 compound words (Easton & Basala, 1982). This is where machine learning lipreading can be of great help. Machine lipreading is difficult because it requires extracting spatiotemporal features from the video (since both position and motion are important). For a long time, advancements in the field involved word classification, not sentence-level sequence prediction, which limited its applicability in real-world communication scenarios. This all changed with the introduction of “LIPNET: END-

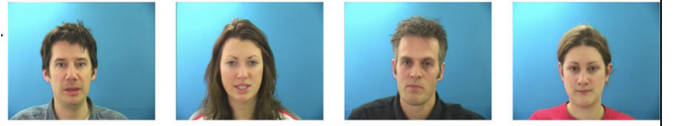


Figure 1: Here we show some sample images from the GRID corpus Dataset that we are using a subset from in our work. The original dataset comprises of 34 different speakers

TO-END SENTENCE-LEVEL LIPREADING” by Yannis M. Assael [1]. Using a subset of the original GRID corpus 1 (high quality audio and video recordings of 1000 sentences spoken by each of 34 talkers(18 male, 16 female) for total of 34000 sentences). I employed a convolutional neural network (CNN) to process sequential video frames, extracting spatial and temporal features that capture the dynamics of mouth movements. These extracted features were then utilized as inputs to a recurrent neural network (RNN), specifically leveraging LSTM (Long Short-Term Memory) units, which were trained to generate textual sentences by predicting sequences of words. Noise reduction for the project primarily focuses on the preprocessing of video frames . (Grayscale conversion, Frame cropping, Normalization). We aim to optimize the model to perform effectively on a subset of the GRID corpus, focusing on accurately predicting the sentences spoken by individuals in the videos. Our goal is to fine-tune the model using this specific dataset, enhancing its ability to produce precise transcriptions of spoken content. By assessing the model’s performance with Connectionist Temporal Classification (CTC) loss and measuring the Word Error Rate (WER), I intend to refine its capabilities for this targeted application. The ultimate objective is to develop an end-to-end system that reliably processes video input to predict user sentences with minimal training requirements, using CTC and WER as key metrics to gauge accuracy and effectiveness.

2. Related Work

Automated lipreading has advanced greatly since Goldschen et al. [2] first tried visual-only sentence-level lipreading with hidden Markov models on a small dataset with manually segmented phonemes. Following up on these early work, Neti et al. [3] investigated sentence-level audio-visual voice recognition using HMMs and hand-engineered features from the IBM ViaVoice dataset. Prior to the advent of deep learning, Gergen et al. [4] set the standard, reaching 86.4% speaker-dependent accuracy on the GRID corpus using an HMM/GMM system informed by LDA-transformed Discrete Cosine Transforms of mouth regions. The introduction of deep learning resulted in a paradigm shift, as Ngiam et al. [5] investigated multimodal audio-visual representations. Wand et al. [6] made a significant step towards current approaches by applying LSTM recurrent neural networks to lipreading; nevertheless, their work did not address sentence-level sequence prediction or speaker independence. Chung and Zisserman [7] made a substantial contribution by proposing spatial and spatiotemporal convolutional neural networks based on the VGG model for word classification. When tested on the BBC TV dataset at the word level, their spatial models outscored spatiotemporal ones by an average of 14%. Despite these advances, the problem of handling variable sequence lengths and attaining sentence-level sequence prediction remained unsolved at the time. Generalisation across speakers and extraction of motion features is considered an open problem, as noted in Zhou et al. [8]. LipNet addresses both of these issues.

3. Model

The preprocessing steps start with converting the videos into grayscale, which are then cropped to our area of interest (Mouth). This step isolates the mouth region, removing irrelevant background information that could confuse the model

Cropping: The frames are then cropped to focus solely on the mouth region. This step isolates the region of interest and eliminates background noise that could potentially mislead the model. By concentrating on the mouth, the model is trained to recognize subtle movements essential for accurate lip reading.

Normalization: The pixel values of the cropped frames undergo normalization. This process involves calculating the mean and standard deviation of the pixel values across the frames. The formulas for the mean (μ) and standard deviation (σ) are given by:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

where x_i represents the pixel values, and n is the total number of pixels in the frames.

The normalized pixel values (x'_i) are computed as follows:

$$x'_i = \frac{x_i - \mu}{\sigma}$$

This normalization ensures that the input data to the neural network has zero mean and unit variance. It helps in stabilizing the learning process by keeping the activation outputs and gradients at a consistent scale, thereby improving both the stability and speed of the learning process. 2.

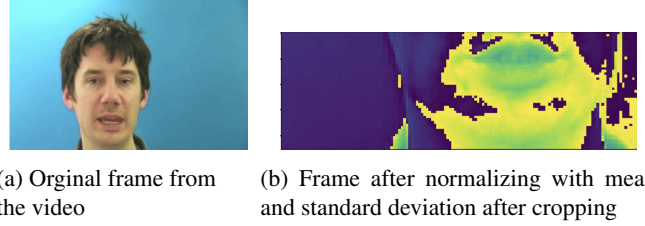


Figure 2: Original frame and pre-processed frame

The preprocessed video files are then sent into a series of Convolutional 3D Layers to extract spatial features such as the shape of the mouth and temporal features such as movements over time inspired by official implementation [9]. Each Conv3D layer is followed by a LeakyReLU activation function to enhance non-linearity and a MaxPool3D layer to reduce dimensionality, with the layers configured to progressively increase the number of filters from 128 to 256, and finally 75, each initialized with HeNormal. After feature extraction through these convolutional layers, the model incorporates Bidirectional LSTM layers with 128 units each, initialized with GlorotUniform, to capture both forward and backward dependencies in the data, enhancing the model's ability to understand contextual information across the entire sequence. Dropout of 0.6 is applied after each LSTM layer to prevent overfitting. The output is handled by a Dense layer with softmax activation, converting logits into a probability distribution over possible classes, using Connectionist Temporal Classification (CTC) loss during training, which is ideal for tasks like lip-reading where the alignment between video frames and spoken words is not explicitly defined 3

4. Experiments and Results

4.1. Data Processing Pipeline Overview

I took inspiration from official implementations of methods to handle data, along with writing my own custom data

Method	Test-data	Average WER
CNN+LSTM	GRID	0.12
CNN+LSTM	Customdata	1.2

Table 1: Preliminary experimental results. XX marks pending results on planned experiments.

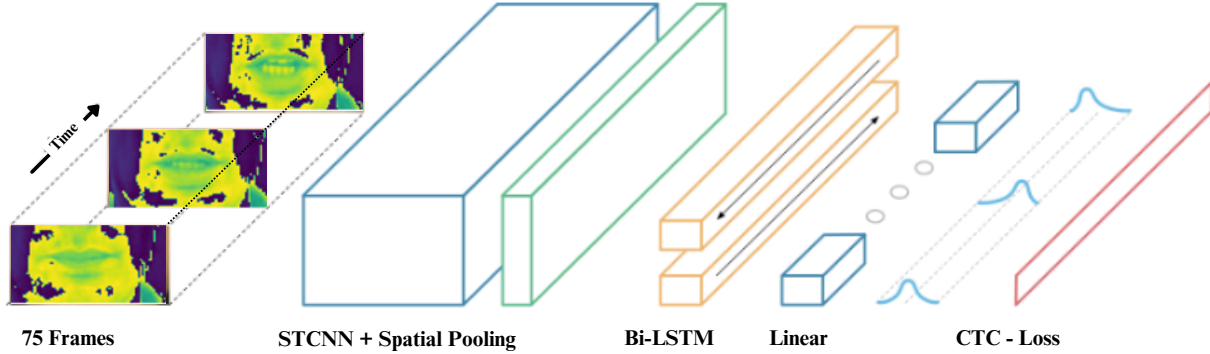


Figure 3: Model architecture. A sequence of preprocessed 75 frames are used as input, and is processed by 3 layers of STCNN with LeakyReLU activation, each followed by a spatial max-pooling layer. The features extracted are processed by 2 Bi-directional LSTM layers; each time-step of the GRU output is processed by a linear layer and a softmax. This end-to-end model is trained with CTC.

loading and callback functions. [10] [11] [12] [13]. The process begins by converting video frames to grayscale to simplify visual content, focusing on pixel intensity changes essential for accurate lip-reading. These frames are then normalized to zero mean and unit variance, ensuring consistent and stable inputs for the learning model. Textual data handling employs charنون and numtochar layers from the StringLookup class, which facilitate efficient character-to-numeric conversions and vice versa.

During the training phase, techniques such as shuffling, mapping with the mappablefunction, batching, and prefetching are utilized to optimize data flow through the model. The architecture comprises several Conv3D layers followed by bidirectional LSTM layers with dropout to enhance learning and prevent overfitting. A learning rate scheduler and Connectionist Temporal Classification (CTC) loss are integrated to fine-tune the training process. Initially, the model’s predictions were notably inaccurate, but with continual adjustments and dedicated training, there has been significant improvement.

4.2. Experiments with the model

Initial experiments with the model were to see how many CNN layers would be better with LipNet architecture as basis. Using LeakyReLU instead of ReLU activation helped in performance with 0.01 being the optimal alpha. Hypoth-

esis is using LeakyReLU helped with the vanishing gradients in the model. Orthogonal and Glorhouniform kernel initializers were experimented with for LSTM layers for which glorhouniform performed well due to the fact that help keep the gradient uniform. In the original papers they used bi-directional GRU (Gated Recurrent Units) on experimentation we found LSTM (Long Short-Term Memory) performed better for our conditions, additional parameters and the structure of LSTMs helped capture deeper nuances compared to GRUs. Adding dropout layers after the LSTM layers helped in achieving better results while finding out perfect dropout rate took some experimentation with 0.6 being the ideal rate.

4.3. Quantitative Analysis

In evaluating our lip-reading model, we employed two main metrics: **Word Error Rate (WER)** and **Connectionist Temporal Classification (CTC) Loss**, each offering insights into different aspects of model performance.

Word Error Rate (WER) WER measures the proportion of word-level errors in the model’s predictions, calculated as the minimum number of word insertions, deletions, and substitutions required to match the target sequence. It directly assesses the accuracy of speech recognition.







No	Frames	Actual Text	Predictions	WER
1		lay blue by e two please	lay blue by e two please	0
2		place green by k six now,	place green by six now	0.16
3		Set blue at h one again,	set blue at h one again	0
4		set blue at a seven again,	set blue at a seven again	0
5		place white by y one again,	place wite y y one again	0.33
6		bin blue seven six,	bin re win n ie naon	1.25

Table 2: Predictions for the sample frames compared to the actual sentence read by the speaker, WER is used to compare the model performance

In the experiment, I achieved an average WER of 0.12 for the GRID subset dataset. Some of them were 0 because the model anticipated the precise sentences spoken by the speaker. However, when tested on custom data, I was able to reach 1.2, which could be attributed to insufficient training.

CTC Loss CTC loss is used for training models where the alignment between inputs and targets is unknown, crucial for sequence prediction tasks like lip reading. It quantifies the model’s effectiveness in learning the correct alignment between video frames and spoken words.

The CTC loss for the 450 training points was 36.78, while the validation loss for the 50 testing points was 32.42 suggesting the validation set might be easier for the model to predict.

4.4. Success and Failure modes

4.4.1 Success mode

The model performs best with clear visual patterns where it is very clear visually what the speaker is reading out. It worked great for the trained dataset cause it had clear speech patterns which the model could catch on. Because the lips are integral to the model, a frontal camera viewpoint is optimal. Other perspectives may not be as effective. The bidirectional nature of LSTMs allows model to pickup the past and future context within a sequence making it work best where temporal relationships in speech are crucial.

4.4.2 Failure mode

Several environmental factors could significantly impact performance, such as substantial background noise, poor lighting, or physical obstructions like hands covering the mouth or rapid facial movements. These conditions could degrade the visual quality of input frames, leading to less accurate lip-reading results. Furthermore, the model’s effectiveness across different accents or dialects might be limited since it is primarily trained on a single individual. However, the primary goal of the project is to achieve reliable lip-reading for one person, so these broader linguistic variations were not the main focus of the training process. This specialization intentionally narrows the model’s utility across varied linguistic contexts, aligning with the project’s specific objectives.

In the experimentation I was able to successfully implement LipNet architecture with changes in the model from my end. I was able to extract the sentence read by the speaker with an average WER of 0.12 for a subset of the GRID dataset. While the model performed admirably on a subset of the dataset, it failed to deliver the same results with custom data from my video. Mainly because time constraints prevented it from being trained on my data, and my custom data did not have the same ideal conditions as the original subset. This could be a potential scope for future projects to develop and expand on.

References

- [1] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. Lipnet: End-to-end sentence-level lipreading. In *arXiv preprint arXiv:1611.01599*, 2016.
- [2] A.J. Goldschen, O.N. Garcia, and E. Petajan. Continuous automatic speech recognition by lipreading. In *Motion-Based Recognition*, pages 321–343. Springer, 1997.
- [3] Chalapathy Neti, Gerasimos Potamianos, Juergen Luetttin, Iain Matthews, Hervé Glotin, Dimitra Vergyri, Jay Sison, Ashutosh Mashari, and J. Zhou. Audio-visual speech recognition. In *Workshop 2000 Final Report*, volume 2002, pages 26–29, 2000.
- [4] Sébastien Gergen, Steffen Zeiler, and Hervé Glotin. Dynamic stream weighting for turbo-decoding-based audiovisual asr. In *Proceedings of the Interspeech*, 2016.
- [5] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011.
- [6] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. Lipreading with long short-term memory. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6115–6119, 2016.
- [7] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, pages 87–103. Springer, 2016.
- [8] Ziheng Zhou, Guoying Zhao, and Matti Pietikäinen. Review of visual speech recognition techniques. *Journal of Machine Learning Research*, 15:3349–3398, 2014.
- [9] Lipnet: Model. <https://github.com/rizkiarm/LipNet/blob/master/lipnet/model.py>.
- [10] Lipnet: Videos. <https://github.com/rizkiarm/LipNet/blob/master/lipnet/lipreading/videos.py>.
- [11] Lipnet: Alignments. <https://github.com/rizkiarm/LipNet/blob/master/lipnet/lipreading/aligns.py>.
- [12] Lipnet: Callbacks. <https://github.com/rizkiarm/LipNet/blob/master/lipnet/lipreading/callbacks.py>.
- [13] Lipnet: Training script. https://github.com/rizkiarm/LipNet/blob/master/training/random_split/train.py.