Lip Reading Using Computer Vision

Authors:
- Sachin Shivanna – A-20552795
- Harshith Deshalli Ravi – A-20552830

Table of Contents

Abstract

With the rise of digital editing tools, it has become increasingly easy to manipulate videos by altering audio content. This manipulation leads to misinformation by replacing the original audio with different speech, creating a mismatch that may go unnoticed by viewers. This project proposes an enhancement of the LipNet model using techniques from LCANet to develop an automated system that can detect discrepancies between lip movements and audio, focusing on sentence-level synchronization. Our implementation demonstrates improved performance in Continuous Error Rate (CER) and Word Error Rate (WER) using the GRID Corpus dataset, paving the way for practical applications in areas like video forensics, surveillance, and accessibility for the hearing impaired.

Keywords

LipNet, LCANet, lip reading, computer vision, deep learning, GRID corpus, audio-visual synchronization, deepfake detection

Introduction

Background

Lip reading is inherently challenging for humans, with average accuracy rates only reaching around 20% under optimal conditions. With advancements in digital editing, the potential for manipulating video content to spread misinformation has increased significantly. Video manipulation techniques, such as swapping original audio with mismatched sounds, make it difficult to verify the authenticity of digital media.

Problem Statement

Until recent research, systems often focus on word-by-word lip reading rather than understanding entire sentences. This limits the ability to detect discrepancies between spoken words and visual lip movements over time. Our project aims to develop an enhanced version of the LipNet model, using techniques from LCANet, to perform sentence-level lip reading and achieve robust synchronisation analysis. This system has potential applications in content authenticity verification, aiding in detecting manipulated content in digital media.

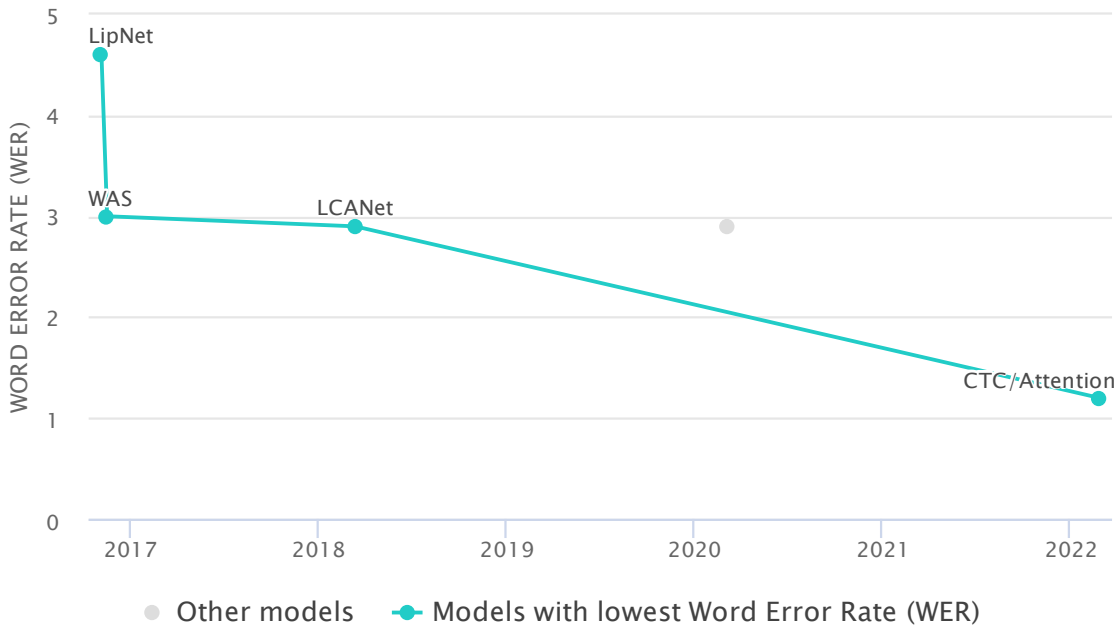
Objectives

- Develop an automated lip-reading system using an improved LipNet architecture.
- Integrate cascaded attention mechanisms from LCANet to enhance sentence-level recognition.
- Improve CER and WER metrics on the GRID Corpus dataset.

Literature Review

1. ***LipNet Model (Assael et al., 2016):*** This foundational paper introduced an end-to-end sentence-level lip-reading model, achieving 95.2% accuracy on the GRID corpus. LipNet combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to model both spatial and temporal aspects of lip movements, offering a pioneering approach to automated lip reading.

2. ***LCANet Model (Xu et al., 2018):*** LCANet builds on LipNet with cascaded attention mechanisms and Connectionist Temporal Classification (CTC) for sequence modeling. This improvement provides better temporal alignment, which we aim to adapt for enhanced performance in continuous speech recognition tasks.

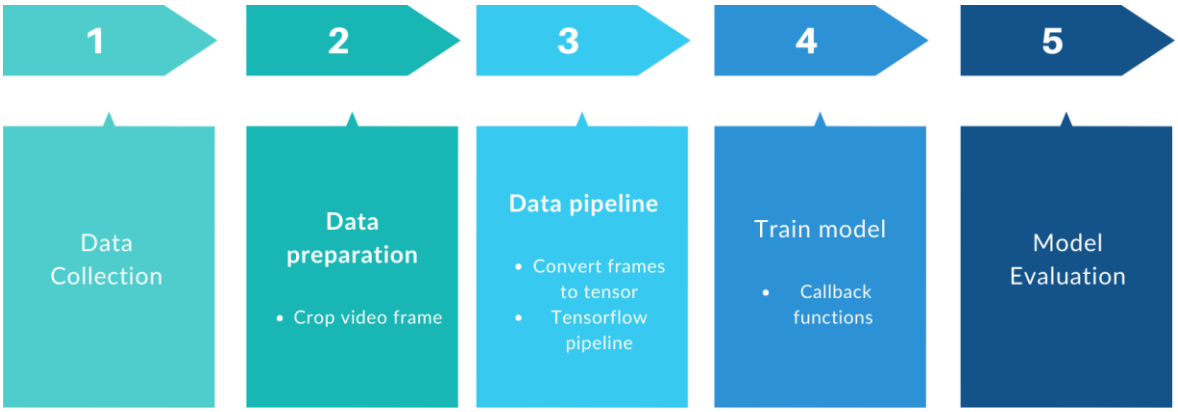


WER Stats

3. ***Challenges in Lip Reading:*** Studies by Wand et al. (2016) and Chung & Zisserman (2016) highlight the difficulties in distinguishing subtle mouth movements, especially for sentence-level tasks. These studies underscore the limitations of traditional word-level classification models and support the need for context-aware architectures like LipNet and LCANet.

Methodology

System Flow Chart



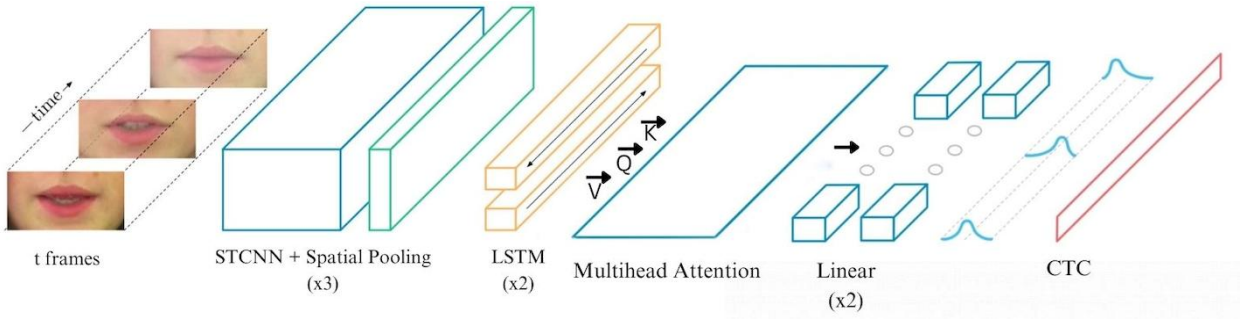
Flow Chart

Data Collection and Preprocessing

- ***Dataset:*** We used the GRID Corpus, consisting of 450 video clips for training and 50 clips for testing, with corresponding audio for synchronization.
- ***Preprocessing:*** Video frames were cropped and converted to tensors for processing in a TensorFlow pipeline. The data was standardized to maintain consistent input across the model.

Model Architecture

- ***LipNet Modifications:*** We modified the LipNet architecture to include Bi-Directional Long Short-Term Memory (LSTM) layers in place of the original GRU layers. This change aims to improve the model's ability to understand long-sequence dependencies.
- ***Cascaded Attention Mechanism:*** Drawing from LCANet, we implemented cascaded attention to enhance temporal alignment. This mechanism allows the model to focus on critical lip movements across frames, which is essential for sentence-level recognition.
- ***Loss Function:*** We used CTC loss, optimized to penalize temporal misalignments between predicted sequences and actual audio.



Model Architecture

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
=================================================================================================
input_1 (InputLayer)         [(None, 75, 46, 140       0         []
                             , 1)]

conv3d (Conv3D)              (None, 75, 46, 140,       3584      ['input_1[0][0]']
                             128)

activation (Activation)      (None, 75, 46, 140,       0         ['conv3d[0][0]']
                             128)

max_pooling3d (MaxPooling3D)  (None, 75, 23, 70,       0         ['activation[0][0]']
                             128)

conv3d_1 (Conv3D)            (None, 75, 23, 70,        884992    ['max_pooling3d[0][0]']
                             256)

activation_1 (Activation)    (None, 75, 23, 70,        0         ['conv3d_1[0][0]']
                             256)

max_pooling3d_1 (MaxPooling3D)  (None, 75, 11, 35,     0         ['activation_1[0][0]']
                             256)

conv3d_2 (Conv3D)            (None, 75, 11, 35,        518475    ['max_pooling3d_1[0][0]']
                             75)

activation_2 (Activation)    (None, 75, 11, 35,        0         ['conv3d_2[0][0]']
                             75)

max_pooling3d_2 (MaxPooling3D)  (None, 75, 5, 17, 7    0         ['activation_2[0][0]']
                             5)

reshape (Reshape)            (None, 75, 6375)          0         ['max_pooling3d_2[0][0]']

time_distributed (TimeDistribu  (None, 75, 6375)       0         ['reshape[0][0]']
ted)

bidirectional (Bidirectional)  (None, 75, 256)         6660096   ['time_distributed[0][0]']

dropout (Dropout)            (None, 75, 256)           0         ['bidirectional[0][0]']

bidirectional_1 (Bidirectional  (None, 75, 256)        394240    ['dropout[0][0]']
)

dropout_1 (Dropout)          (None, 75, 256)           0         ['bidirectional_1[0][0]']

multi_head_attention (MultiHea  (None, 75, 256)        1051904   ['dropout_1[0][0]',
dAttention)                                                      'dropout_1[0][0]']

add (Add)                    (None, 75, 256)           0         ['dropout_1[0][0]',
                                                                 'multi_head_attention[0][0]']

layer_normalization (LayerNorm  (None, 75, 256)        512       ['add[0][0]']
alization)

dense (Dense)                (None, 75, 512)           131584    ['layer_normalization[0][0]']

dense_1 (Dense)              (None, 75, 256)           131328    ['dense[0][0]']

add_1 (Add)                  (None, 75, 256)           0         ['layer_normalization[0][0]',
                                                                 'dense_1[0][0]']

layer_normalization_1 (LayerNo  (None, 75, 256)        512       ['add_1[0][0]']
rmalization)

dense_2 (Dense)              (None, 75, 41)            10537     ['layer_normalization_1[0][0]']

=================================================================================================
Total params: 9,787,764
Trainable params: 9,787,764
Non-trainable params: 0
_________________________________________________________________________________________________
```

Model Summary

Implementation

- ***Software and Tools:*** TensorFlow was used for model training and evaluation. Supporting libraries included OpenCV for image processing, NumPy for data manipulation, and Matplotlib for visualizing results.
- ***Training Configuration:*** The model was trained with a batch size of 2 due to memory constraints, and the dataset was increased to 450 clips with additional epochs to improve loss.

Results

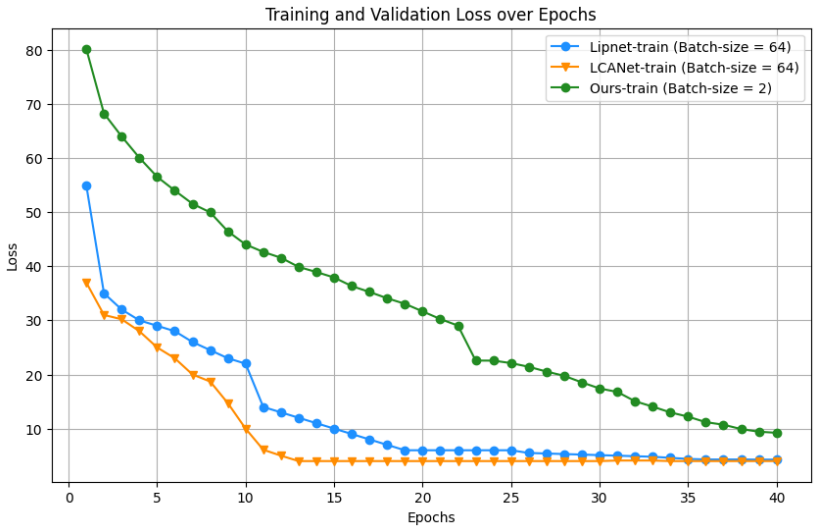
The model demonstrated significant improvement in both CER and WER, particularly on overlapping data, indicating robustness in sentence-level lip reading. Doubling the number of epochs allowed for a notable reduction in loss.

```
Epoch 100/100
450/450 [==============================] - ETA: 0s - loss: 3.9059
```

Model training final epoch

Model	Loss	Batch-size	Dataset size	Epochs
LipNet	4.27	64	25,330	50
LCANet	4.16	64	25,330	50
Our Model	3.90	2	450	100

Loss comparison table



Loss comparison

#	Method	CER %	WER %
1	**LipNet**	1.9	4.8
2	**LCANet**	1.3	2.9
3	**Our Model**	**0.29**	**1.38**
4	**CTC/Attention**	0.05	1.2

Error rate table

- **Continuous Error Rate (CER):**

$$\text{CER} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where:

- y_i is the observed value,
- $\hat{y}_i$ is the predicted value, and
- N is the total number of observations.

- **Word Error Rate (WER):**

$$\text{WER} = \frac{S + D + I}{N}$$

where:

- S is the number of substitutions (words incorrectly predicted),
- D is the number of deletions (words that were missed in the prediction),
- I is the number of insertions (extra words in the prediction), and
- N is the total number of words in the reference text.

Discussion

Our model improvements over the original LipNet model show promise in detecting audio-visual discrepancies. The integration of Bi-Directional LSTM layers and cascaded attention mechanisms has enhanced the model's ability to interpret long sequences, making it more suitable for continuous speech recognition. Despite the improvements, our model faced limitations with memory constraints that restricted batch size, which could be addressed with more computational resources. Future work could explore integrating more sophisticated attention mechanisms and testing on a wider variety of datasets.

Limitations

- ***Memory Constraints:*** Limited batch size and training dataset size due to computational constraints.
- ***Dataset Scope:*** The GRID Corpus is a controlled dataset; performance on more diverse, real-world data remains to be evaluated.

Future Work

- Implement attention-based transformers to enhance sequence modeling further.

- Evaluate model performance on real-world datasets and explore alternative architectures for greater scalability.

Conclusion

This project successfully extended the LipNet model with enhanced temporal alignment and sentence-level recognition capabilities. By integrating cascaded attention, our model improved on standard lip-reading metrics, making it a valuable tool for detecting audio-visual inconsistencies. These advancements have potential applications in video forensics, content authentication, and assistive technology for the hearing impaired.

References

1. Assael, Y., Shillingford, B., Whiteson, S., & de Freitas, N. (2016). LipNet: End-to-End Sentence-Level Lipreading. arXiv preprint. Retrieved from https://arxiv.org/abs/1611.01599

2. Xu, K., Li, D., Cassimatis, N., & Wang, X. (2018). LCANet: End-to-End Lipreading with Cascaded Attention-CTC. IEEE. Retrieved from https://ieeexplore.ieee.org/document/8373881

3. Repository: https://github.com/rizkiarm/LipNet

4. Data Source: https://spandh.dcs.shef.ac.uk/gridcorpus/

5. Other Resources: https://spandh.dcs.shef.ac.uk/gridcorpus/

Appendix

- ***Team Responsibilities***
  Sachin Shivanna: Responsible for data cleaning, pre-processing, and creating the data pipeline.
  Harshith Deshalli Ravi: Led model training, testing, and performance evaluation.