Individual Final Report

Hala Abualtayeb

1. Introduction.

Our project focuses on developing a lipreading model that takes short videos of human speech as input and outputs the corresponding text predictions of what the speaker said. The model analyzes video frames to generate character-level predictions rather than word-level outputs, and it does not use audio at any stage. This work was inspired by the 2016 paper *LipNet: End-to-End Sentence-level Lipreading*, by Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas, which served as the foundation for our approach.

Our dataset was sourced from the GRID audiovisual sentence corpus, the same dataset used in the LipNet paper. We preprocessed the data using two methods: cropping based on the mouth region dimensions provided by the GRID dataset and detecting the mouth location using MediaPipe's Face Mesh.

We trained three separate custom models to challenge ourselves with increasing complexity. The first model was trained on one speaker using a sample of 1,000 videos and followed an architecture inspired by the LipNet research paper. The second model was trained on five speakers using a sample of 5,000 videos and introduced a more complex architecture with a BiLSTM and an attention module, based on prior research. The final model was trained on three speakers using a sample of 3,000 videos and employed an even more advanced architecture. We also developed evaluation metrics to assess model performance and built an application to test the models using sample inputs.

2. Description of Work Distribution

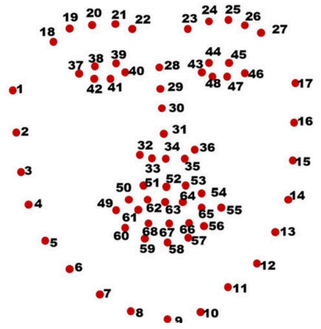
We began this project by researching existing lipreading methodologies and prior applications in the field. From there, we experimented with both data preprocessing techniques and model design. My work focused on preprocessing the dataset using approaches beyond the standard GRID dimensions, including dlib and MediaPipe's Face Mesh. I also built a complex custom model, performed hyperparameter tuning, implemented CTC, and evaluated the model using relevant performance metrics (WER, CER). In addition, I set up a basic version of the application with the preprocessing and data download pipeline. Once we determined that Henry's final model slightly outperformed mine, he took over the final model integration.

3. Description of Individual Work

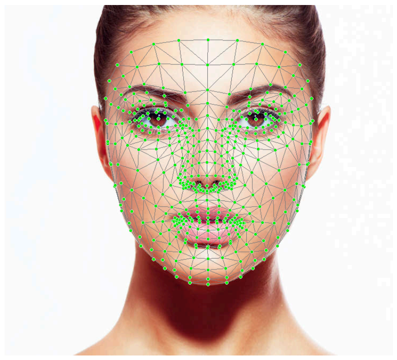
3.1 Pre-Processing

- **dlib**
  - **dlib** is a popular computer vision library that provides tools for face detection and facial landmark localization. I experimented with dlib on the GRID dataset to define and crop the mouth area for use as model input. I was able to run it on two speakers, but it was too CPU-intensive. The library often crashed and couldn't reliably process the full dataset. Cropping the lips is important because it isolates the region of interest, reduces background noise, and allows the model to focus on the subtle movements that correspond to speech. However, dlib's 68 coarse landmarks were

insufficient to capture these detailed lip movements, making the extracted features too limited for accurate lip-reading.

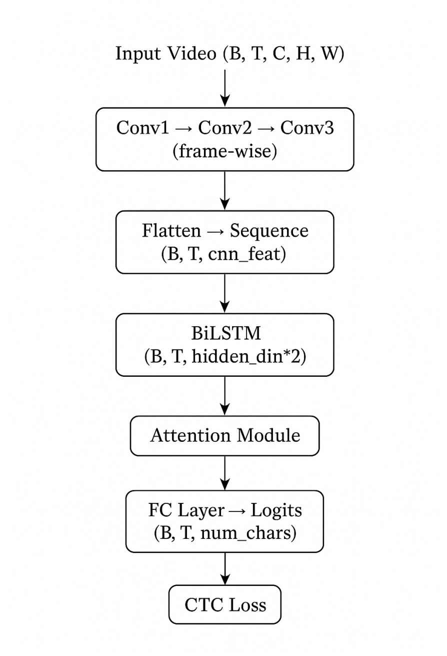


- **Media Pipe's Mesh Face**
  - **MediaPipe Face Mesh** is a Google framework that provides real-time, dense 3D facial landmark detection. We experimented with MediaPipe Face Mesh on the GRID dataset to define and crop the mouth area for use as model input. Unlike dlib, it was able to handle multiple speakers, although the computation was still intensive, limiting processing to about five speakers. With 468 dense 3D landmarks, MediaPipe Face Mesh captured detailed lip shapes and subtle movements, producing rich features that significantly improved the performance of our lip-reading models. This makes it better than dlib because it provides higher-resolution, 3D lip information, enabling the model to learn fine-grained visual speech cues that dlib's 68 coarse landmarks cannot capture.



- **Data Pre-processing and Tokenization**
  - After the lip region was cropped from each frame we then resized it to 64×64 pixels. Frames were converted to grayscale and pixel values were normalized to standardize the input distribution. Video sequences were standardized to a maximum of 75 frames, with shorter sequences padded to maintain consistent input dimensions for the model.
  - For tokenization, a character-to-index mapping was created for the transcripts. This mapping enables CTC (Connectionist Temporal Classification) loss, which allows the model to align variable-length video sequences with text transcripts without requiring explicit frame-level labels.

3.2 Model Architecture


Input Video (B, T, C, H, W)
Conv1 → Conv2 → Conv3 (frame-wise)
Flatten → Sequence (B, T, cnn_feat)
BiLSTM (B, T, hidden_din*2)
Attention Module
FC Layer → Logits (B, T, num_chars)
CTC Loss

For my model architecture, I started with a CNN backbone to extract spatial features from each frame of the video. I knew that lip shapes carry the key visual information, so I stacked multiple convolutional layers with batch normalization, ReLU, pooling, and dropout to capture subtle mouth movements while addressing overfitting. This way, each frame is transformed into a rich feature representation that highlights the important visual cues for speech.

For the temporal aspect, I added a bidirectional LSTM to capture how the lips move over time, producing context-aware features at each frame. I then included an attention module so the model could focus on the most informative frames, emphasizing critical movements that signal specific characters. Finally, a linear layer outputs character probabilities, and I used CTC loss with label smoothing to handle the variable-length nature of lip sequences. This design combines spatial, temporal, and attention mechanisms in a way that aligns with how humans visually interpret speech.

3.3 Hyperparameters

- Batch size: 8
- Learning rate: 3e-4
- Optimizer: AdamW with Weight Decay 1e-4
- LSTM hidden size: 512, 3 layers, dropout 0.4
- Label smoothing: 0.05 applied with CTC loss
- Epochs: 100, with early stopping (patience 30 epochs)

4. Results

Observed Performance

Metric	**Value**	**Interpretation**

Train Loss	1.5697	Stable, moderately high; model underfits.
Validation Loss	1.5968	Close to training loss, indicating no overfitting.
CER	0.5655	Partial character-level learning; ~55% of characters wrong.
WER	0.9354	Most words predicted incorrectly; poor word-level coherence.

Performance evaluation used Character Error Rate (CER) and Word Error Rate (WER) to assess both fine-grained character-level accuracy and broader sequence-level coherence.

Character Error Rate (CER): Fraction of incorrectly predicted characters in sequence.

$$\text{CER} = \frac{\text{Number of character edits (insertions + deletions + substitutions)}}{\text{Total number of characters in reference}}$$

Word Error Rate (WER): Fraction of incorrectly predicted words.

$$\text{WER} = \frac{\text{Number of word edits (insertions + deletions + substitutions)}}{\text{Total number of words in reference}}$$

Our evaluation shows that training and validation losses are close but relatively high, indicating underfitting. While the model captures some character-level patterns, it struggles with word-level sequence coherence, resulting in high CER and WER. Loss plateauing around 1.57–1.60 suggests the current CNN-LSTM capacity and dataset size are insufficient for full sequence modeling. Despite these weaknesses, the attention module functions effectively, and the architecture provides a good starting point that could benefit from more data, enhanced temporal modeling, or increased model capacity to improve end-to-end lip-reading performance.

5. Summary and conclusions.

Working on this lipreading project gave me an understanding of character-level video-to-text prediction using spatiotemporal neural networks. Inspired by the 2016 LipNet paper by Assael et al., we designed models to capture both spatial and temporal patterns in video sequences. Through preprocessing, we learned the importance of face detection, lip landmark extraction, cropping the mouth region, grayscale conversion, pixel normalization, and sequence standardization, steps to support stabilized training. Training the models showed how CRNNs can capture basic mouth movement patterns, but underfitting was a challenge, resulting in high character and word error rates and limited word-level coherence. My final CRNN model, combining CNNs with Bidirectional LSTMs and attention, performed best at extracting spatiotemporal features and modeling temporal dependencies, yet also highlighted the limitations imposed by dataset size, model capacity, and decoding strategies.

Looking forward, several improvements could improve performance. A larger dataset would help the model learn a richer mapping from lip movements to text. More advanced decoding strategies beyond greedy decoding, such as beam search with CTC constraints, could improve word-level accuracy by considering multiple character sequences rather than picking the most likely character at each frame. The current approach also misses modeling the CTC blank token effectively, which represents silent frames or transitions, limiting the model's ability to handle pauses and variable-length sequences. Additionally, exploring more powerful architectures or deeper temporal modeling, along with hyperparameter optimization, could reduce underfitting and better capture complex viseme-to-character patterns.

6. Code Percent

~40-50%, with the rest sourced from various sources on the internet to support the implementation of the pre-processing models and other aspects of our project.

7. References

https://arxiv.org/pdf/1611.01599
https://spandh.dcs.shef.ac.uk/gridcorpus/

8. Appendix

Deep_Learning_6303_GROUP4