Deep Lip Reading

Usman Jalil, Chris Krenz, Cole Resurreccion, Thomas Simmons
{ujalil,ckrenz,coler,tsimmons}@bu.edu

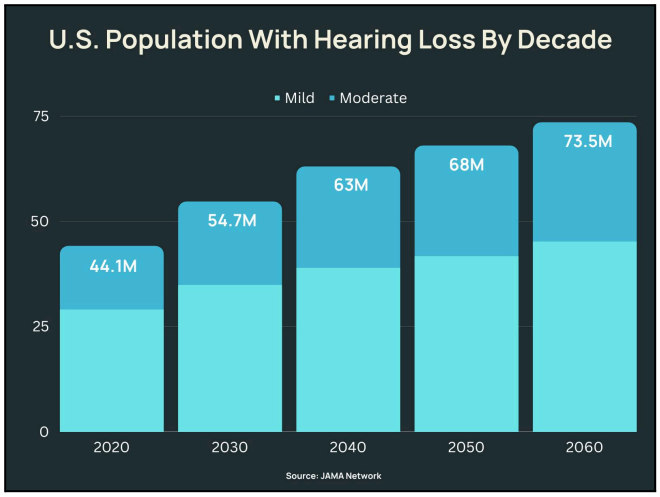

Figure 1. Bar chart of U.S. mild/moderate hearing loss by decade (from "*Important Hearing Loss Statistics and Studies 2023*" [7])

1. Task

About 1 in 8 people (13%) in the United States aged 12 years or older has hearing loss in both ears. That's approximately 30 million people, and the number is growing. As Figure 1 above shows, it is projected that by 2060 almost 75 million people in the United States will have mild or moderate hearing loss [7].

Given these concerning figures, we set out to use machine learning to help mitigate the problem of hearing loss. Our goal was to develop a deep learning model proficient in lip reading from video inputs. More specifically, we sought to **1)** develop the most accurate model we could, **2)** compare the accuracy of a variety of architectures across a variety of datasets, **3)** and develop an app that is able to use the model in real-time.

The task was challenging due to the diverse range of individuals' lip movements and shapes leading to different sounds. Indeed, unlike many other applications of machine learning, such as object identification/classification or image segmentation, the average human being has great difficulty performing the task of lip reading, at least without substantial training and experience. One study found the average human is able to achieve only ~10% accuracy with lip reading [11].

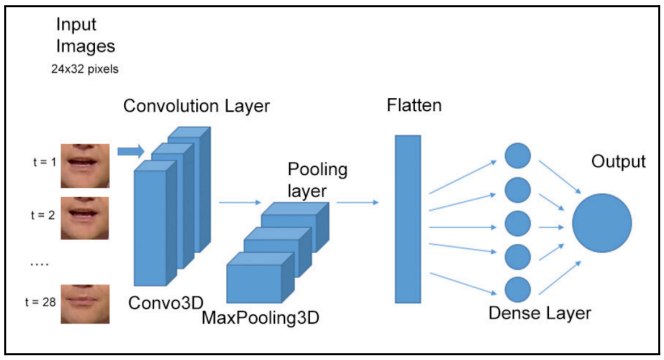

Figure 2. CNN Model Overview (from "*Lip Reading using Neural Network and Deep learning*" [1])

Another reported that even the hearing impaired, with lip-reading experience, achieve at most ~52% [12]. As such, achieving a high percentage accuracy is challenging, requiring well developed models capable of learning the often subtle connection between lip movements and words produced.

2. Related Work

2.1 CNNs and LRW Dataset

We used several articles for guidance and inspiration. For the CNN, one of our primary sources —*"Lip Reading using Neural Network and Deep Learning"*— served as a general guideline throughout the project (though we implemented our own models from scratch) [1,3,4]. The article provides clear steps for the design and implementation of a CNN model for lip-reading, including some of the details on pre-processing the data, setting hyperparameters like kernel size, and performing other training procedures like batch normalization.

The dataset used in this article is the Lip Reading in the Wild (LRW) dataset, which is discussed further in the Datasets section and is one of the datasets we chose to use in our project. The LRW dataset has clipped video segments corresponding to the utterance of individual words. In "*Lip Reading using Neural Network and Deep Learning"*, a Haar classifier (based on an OpenCV python framework) is then used to efficiently isolate the region of interest in each video: a rectangular region around the mouth. Once the region of interest is identified, a Median Flow tracking algorithm is used to track the location of that region across

frames. [1] The images were also converted to grayscale to reduce size and simplify training.

The neural network itself is a 3D convolutional neural network (with the kernel translating across both spatial dimensions and across time). The authors provided an overview of the model as shown in Figure 2 [1]. After testing several model architectures, the authors found that one of the best performing models included a kernel size of 5x5x5, with three convolutional layers and ending in a flattening layer followed by the final dense layer. The convolutional layers are interspersed with two pooling layers, two batch normalization layers, and a dropout layer, and the final convolutional layer has a total of about two million parameters. In the publication, this model was able to achieve a test accuracy of >70%.

2.2 Grid Dataset

Another primary source was *"LipNet: End-to-End Sentence-level Lipreading,"* which discusses the Grid dataset used to implement the LipNet model [5]. This is the second dataset we chose to implement in our project. Grid is fundamentally different from LRW in that each video contains a sequence of several words, rather than a single word. However, the subset of the Grid dataset we used contained only 1,000 videos and 50 words. (We chose to use a subset of the Grid dataset because the massive size of the LRW dataset was presenting a challenge to training and testing our models; limiting the size of the Grid dataset allowed for more rapid iteration and testing.)

2.3 Transformers (and metrics)

Another article, *"Deep Lip Reading: a comparison of models and an online application,"* explores three different models designed for lip reading: a recurrent LSTM, a fully convolutional model, and a transformer model [2]. The article compares their implementations and accuracy. The primary metrics this article uses are Character Error Rates (CER) and Word Error Rates (WER). As discussed further below, we decided to measure the accuracy by word, rather than character or sentence, to evaluate and compare our models.

The study found that all three models performed well (better than any previous models at the time). Although the transformer had a better overall accuracy (~50% for one of the datasets), the fully convolutional model was much faster to train as well as having lower latency with inference/prediction, making it useful for developing a real-time lip reading tool. This article was one of our inspirations for attempting to develop an app capable of real-time lip reading.

Another article we used as a starting point for developing a Transformer was *"A Novel Machine Lip Reading Model"* [9]. The authors also used the Grid dataset and implemented a multi-head, encoder-decoder transformer. Their model was able to reach a word-level accuracy of ~46%.

Although we followed these (and other related) articles as guideposts, we implemented the models from scratch using Tensorflow (for the Grid dataset) and PyTorch (for the LRW dataset).

3. Approach

3.1 Overview

Our first step was to read in and preprocess the data. Given that we were working with video data, the datasets tended to be very large; as such, it took a long amount of time to train the models. So it was especially important that we streamline and simplify the data as much as possible to make training more manageable. The pre-processing steps included:

- Locating the region of interest: the speaker's mouth
- Cropping the image to the region of interest
- Converting the image to grayscale
- Resizing the images to a standard size
- Normalizing the image data

After pre-processing, we designed the models. For the LRW dataset, PyTorch was used; for the Grid dataset, TensorFlow (with Keras) was used. For Grid, we implemented both a CNN and Transformer; for LRW, we implemented a CNN.

After designing the models, we ran the training procedures and then graphed both the training and validation loss and accuracy, generally across 10 epochs.

Having generated this data, we were then able to make comparisons to identify the best models. Note that some parameters do differ between the LRW and Grid CNN implementations to account for differences in image size/orientation, video length, etc. We attempted to optimize each model separately for each dataset.

3.2 Pre-processing

For Grid, we utilized a Haar classifier from OpenCV to facilitate finding the region of interest. We generated the image in Figure 3 as an example of the classifier's ability to locate the speaker's lips without needing to manually annotate the video ourselves. After refining this process it became clear that the most effective approach was a

two-step process of first using a face-classifier to locate the speaker's face, after which a lips-classifier could be applied to narrow the bounding box down to the speaker's mouth.

Attempting to implement only the lips-classifier on the entire image seemed to result in an accuracy of approximately 85%. This was assessed by manually sampling a couple hundred images across the dataset and checking where the bounding box was located relative to the speaker's mouth. In contrast, the two step process resulted in an accuracy above 99%. (As a failsafe, the model would default to a static, central region in the image in the rare event the Haar classifier failed to locate a region of interest).

Figure 3. Example of Haar classifier's ability to locate the speaker's mouth (Grid dataset)

(We ultimately decided there was no need to implement the Median Flow Tracking from [1], as once the Haar classifier was optimized, it proved sufficiently accurate.)

For LRW, use of a Haar classifier was not necessary as the speaker's mouth was always centrally located, even as their mouth moved. As such, a region of a standard size and positioning could be used across all images.

After having located the speaker's mouth, we then performed the additional pre-processing steps: cropping the image to the region of interest, converting it to grayscale, resizing the image (80x40 for Grid and 80x130 for LRW), and normalizing the data. This allowed us to both train faster and remove unnecessary features, like color. This resulted in tens of thousands of images for Grid (75 images per video/sequence) and millions for LRW (29 images per video/sequence) like the grayscale image in Figure 4. Note, the larger region for the LRW dataset was chosen to accommodate the greater variability in speaker orientation, mouth-size, etc., as compared to the more highly curated Grid dataset.

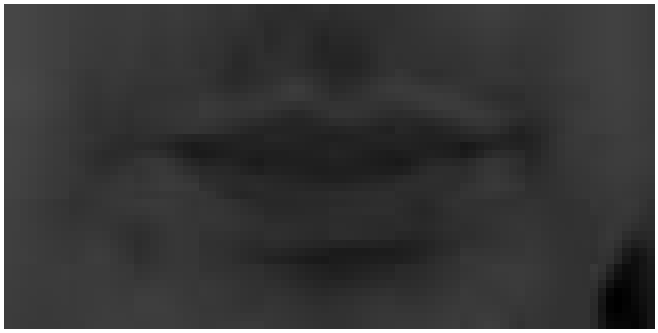

Figure 4. Cropped, grayscale image from the bounding box in the image from Figure 3 (Grid dataset)

Normalizing and reducing the size of the images in this way allows them to be processed far more efficiently than the full-scale, color images with which we started.

It is also worth noting that not all videos in the LRW dataset are face on like the Grid dataset. Some videos are near-frontal, meaning it is angled on one part of the face. We believe that the large size of the dataset accounted for these variations in speaker orientation and likely even made the model more robust.

3.3 CNN

After pre-processing, we began experimenting with the model architectures and pipelines. For the CNN, we first modeled the architecture after that used in "*Lip Reading using Neural Network and Deep Learning"* [1]. We included the model summary and a diagram taken from the article.

```
Model: "sequential_22"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv3d_66 (Conv3D)          (None, 75, 40, 80, 32)    2624

 max_pooling3d_66 (MaxPooli  (None, 37, 20, 40, 32)    0
 ng3D)

 conv3d_67 (Conv3D)          (None, 37, 20, 40, 64)    55360

 max_pooling3d_67 (MaxPooli  (None, 18, 10, 20, 64)    0
 ng3D)

 conv3d_68 (Conv3D)          (None, 18, 10, 20, 128)   221312

 max_pooling3d_68 (MaxPooli  (None, 18, 5, 10, 128)    0
 ng3D)

 flatten_22 (Flatten)        (None, 115200)            0

 dense_44 (Dense)            (None, 128)               14745728

 dense_45 (Dense)            (None, 41)                5289
```

Figure 5. CNN implemented in TensorFlow (based on "*Lip Reading using Neural Network and Deep learning*" [1])

This model includes the following layers:

1. A few Conv3D layers
2. Max pooling layers (1 per convolution layer)
3. Flattening layer
4. Dense layers

The max pooling layers are to reduce the dimensionality after each convolutional layer (and likely help to control for certain kinds of noise like rotation of the mouth), and the flattening and dense layers further reduce dimensionality at the end to prepare for classification.

The hyperparameters for the CNNs were as follows:

- Kernel-size: 3x3x3
- Learning rate: 0.0001
- Batch size: 32

3.3 ResNet TCN Hybrid

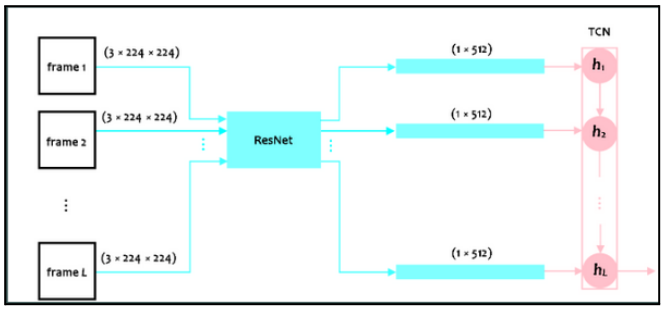

Figure 6. Model Overview (from "*Improving state-of-the-art in Detecting Student Engagement with ResNet and TCN Hybrid Network*" [6])

The next model we experimented with was a ResNet and TCN Hybrid Network as proposed in [6]. While this model originally was used for viewing student attentiveness, we altered it to work on a video classification task. This model first passed the video frame by frame through a ResNet to capture spatial features. Residual networks are able to stack CNNs without vanishing gradients by adding a previous instance of the input. The ResNet used was the SE-ResNet-50, which was pre-trained to extract those features from faces. The TCN then analyzes the changes in those features over time. The model ends with a linear layer.

Hyperparameters for the RestNet TCN were as follows

- Spatial Features Dimensions: 32
- Number Hidden Layers: 128
- Levels: 8
- Kernel Size: 7
- Dropout: 0.1

These were chosen after extensive experimentation, based on both accuracy and feasibility (given time and memory constraints). The kernel sizes both approximated the 5x5x5 found in the *"Lip Reading using Neural Network and Deep Learning"* article, with the LRW dataset optimizing at a larger kernel size, perhaps due to the larger image size [1].

3.4 Transformer

As mentioned above, we used a couple of articles for designing the Transformer: *"A Novel Machine Lip Reading Model"* and *"Transformer-based lip-reading with regularized dropout and relaxed attention"* [2,9]. However, finding that the training times for the transformers were taking prohibitively long to complete (or were using a prohibitive amount of memory), we simplified the models for the sake of feasibility. While the articles discuss encoder-decoder models, our implementation was an encoder-only model, which proved sufficient. The transformer architecture included the following layers:

1. Dense layer
2. Transformer (encoder) layer
3. Time distributed layer

(We tried a variety of models but found this simple model allowed us to complete the training in a feasible amount of time while still generating decent results. The first models we tried would have taken multiple days to complete.)

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 75, 128)           19267712

 reshape (Reshape)           (None, 75, 128)           0

 transformer_encoder (Trans  (None, 75, 128)           561024
 formerEncoder)

 reshape_1 (Reshape)         (None, 8, 1200)           0

 time_distributed (TimeDist  (None, 8, 54)             64854
 ributed)
```

Figure 7. Architecture of Transformer in TensorFlow.

The dense layer is used for mapping the dimensionality of the input to the embedding dimensions of the transformer encoder, and the time-distributed layer applies this across all frames of the videos.

Hyperparameters for the transformer were as follows:

- Embedding dimensions: 128
- Feedforward dimensions: 128
- Dropout rate: 0.1
- Learning rate: 0.001
- Batch size: 32

All of our hyperparameter tuning started with the parameters set to the values we found in existing literature. The values were then methodically adjusted up and down until optimal results were achieved.

Our models all used a cross entropy loss:

$$H(p,q) = -\sum p(x) \log q(x)$$

We experimented with using regular cross entropy loss with one-hot encoding vs. sparse categorical cross entropy without one-hot encoding, though this did not have an effect on the results.

4. Datasets

4.1 LRW

We utilized both the Lip Reading in the Wild (LRW) and Grid datasets. LRW contains 800-1000 utterances of 500 different words from people of different groups. The metadata of the data point contains timestamps of when in the video the word starts and for how many frames. The validation and test datasets contain 50 data points per class.

The LRW dataset also contains metadata for each video. The metadata states the duration of the word being spoken. While each video is 1.16 seconds long, the duration of the word can be anywhere from 0.1 to 0.9 seconds long. For this reason, the beginning or end of other words can sometimes be present in the video. However, in the aggregate, this should have little to no effect.

4.2 Grid

Access to the LRW dataset was initially restricted, requiring express written permission from the owners. Though this did cause a delay in our accessing the LRW data, we were able to receive approval and begin utilizing the data. As a stepping stone, before we acquired access to the LRW dataset, we started the implementation using a slightly different and smaller dataset–a subset of the Grid dataset, which is discussed in the "*LipNet: End-to-End Sentence-Level Lipreading*" article [5]. This dataset was readily accessible and took up much less space (LRW is on the order of ~100Gb, which makes downloading and processing the data difficult).

The Grid dataset we used includes 1000 videos, each with 6-8 words, with the annotation for each video consisting of a list of words spoken in the clip. For this dataset, 800 videos were used for testing, 150 for validation and 50 for testing.

The Grid dataset we used was a subset of the larger dataset that researchers used to implement the LipNet dataset, which was one of the most advanced models we found in the literature, achieving a 95.2% testing accuracy [5].

4.3 Contrasting Datasets

We tested the CNN model on both the LRW and Grid datasets. The datasets are meaningfully different in that LRW contains one-second, 1-word videos, and Grid contains longer, multi-word videos. This presented a useful opportunity to compare the accuracies achieved with each dataset in order to evaluate the model.

5. Evaluation Metrics

To evaluate success, we based our metric on the Word Error Rate, which is defined as the sum of the substitutions, insertions, and deletions needed to match our prediction to the actual sentence, divided by the number of words spoken.

Although the Grid videos contain entire sentences, (with each prediction consisting of an entire sequence of words), for the purposes of comparing results with the LRW dataset, we are reporting word accuracy (number of words correct out of all words in the video) consistently across all models.

6. Results

Our primary goals were to develop the best lip reading model possible as well as compare different model architectures, across different datasets. One of the primary comparisons we were able to make was between the CNN and Transformer applied to the Grid dataset.

6.1 Grid Dataset - CNN vs. Transformer

For this comparison we wanted to see how a purely convolutional model compared to a pure transformer model. Although there was not a dramatic difference in performance between the two models, the Transformer did slightly outperform the CNN, with the Transformer achieving a validation accuracy of 41% and the CNN achieving 38%.

For both models, we see only minor divergence between the training and validation sets, suggesting the models were not overfitting. While the accuracies are not as high as we had hoped, they would likely be much higher were we to train on a larger dataset over a longer period of time. Furthermore, a more sophisticated transformer with an encoder-decoder structure and more embedded dimensions would undoubtedly have performed better (though would have taken substantially longer to train).

Notably, our transformer performed almost as well as the transformer described in "*A Novel Machine Lip Reading*

Model," which reached ~46%, though not as well as the model in "*Lip Reading Using Neural Networks and Deep Learning*," which reached up to ~76% [1,9].

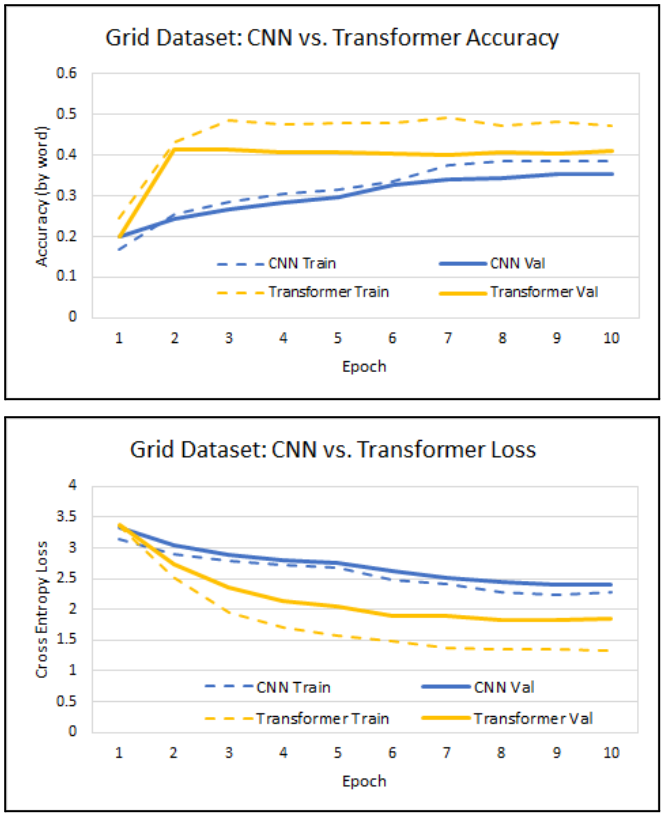

Figure 8. Comparison of Loss and Accuracy for the CNN and Transformer models in the Grid dataset

6.2 Other Architectures/Designs

We also tested models using the LRW dataset. This dataset was substantially larger and so was not as conducive to the more rapid iteration we could perform with the Grid dataset. However, that size is also a significant advantage. The LRW dataset is notably different from Grid in that it has a single word per video and uses video clips from the real-world (TV broadcasts, etc., as opposed to Grid, which is a dataset generated specifically for machine learning).

This comparison was notable in part because of fundamental differences between the video data in each dataset. LRW has the advantage of being a larger dataset with greater variety in the video clips. The Grid dataset is distinct in that it contains videos with entire (albeit short) sentences, which could potentially allow the model to capture additional parameters associated with how the mouth transitions from one word to the next.

Ultimately, testing the ResNet-TCN model on the LRW dataset achieved the highest validation accuracy of 52%, likely due to the size of the dataset and perhaps due to the sophistication of the ResNet-TCN model.

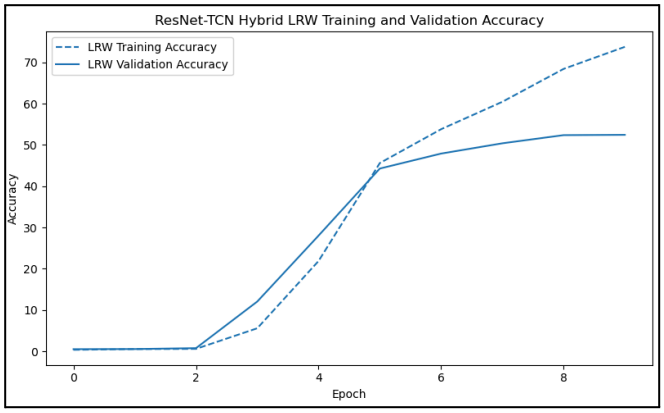

Figure 9. ResNet-TCN training and validation accuracy on the LRW dataset

Because the transformer performed better than the CNN on the Grid dataset, we also tried testing a transformer on this dataset, however, the combination of the size of the dataset and the large compute requirements of the transformer made this infeasible within the given time constraints.

6.3 Real-Time Lip-Reading App

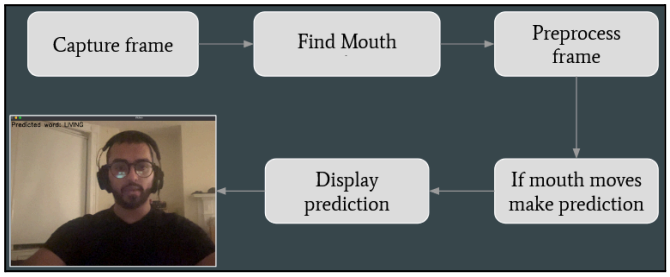

Figure 10. Diagram of application process and example output (correctly identifying the word 'Living')

In addition to training and testing the models as shown above, we developed an app using Opencv and the optimal LRW ResNet-TCN model to perform lip-reading in real time. Figure 10 shows how the app functions:

1. Captures a frame from the live feed of the user's webcam
2. Finds and tracks the mouth region using background subtraction
3. Preprocesses frame (cropping, grayscaling, resizing)
4. Checks whether the mouth has moved since last frame
5. If it has moved, it sends the processed image to the model to begin processing
6. Receives prediction from model and displays it on the screen

The app does have some limitations: it still makes a lot of mistakes and makes multiple/different overlapping

predictions. In fact, sometimes when the user simply moves his head, the model will make a prediction. Thus, we believe that if we were to use a Haar's classifier to specifically extract the mouth and send those frames to the model, it would get better results. Also, we're hopeful that a better model would yield better results. However, given the challenges inherent in performing this complex process in real-time, we are pleased to have a working application–an excellent foundation from which to develop a more robust app.

7. Conclusion

This report aimed to compare different model architectures and find what would be the best to use for lip reading by testing them on different lip reading datasets. Through a comparison of different transformer and CNN models, we found that the transformer is slightly more accurate for the video classification task of lip reading, with an edge of 3%. This edge, however, may be even larger, but we were not able to test this more due to a lack of compute capabilities. The transformer often required large amounts of VRAM to train, requiring us to use a simplified model.

Were we to continue this project, one thing we would like to explore further is a combined CNN + transformer. At least one paper we found achieved an impressive ~89% using this method [13]. We started to implement one of these models, but we did not have time to develop it and were not able to achieve superior accuracy. With more time we could develop this further.

The lip reading task has much more variance compared to an NLP task. Even with only 500 words used, the accuracy only reaches 92.1% in popular implementations [10]. Regional accents, angle of the face, and slang make these classification tasks challenging. In the future, methods like zero-shot learning should be used in order to better classify words.

Ultimately, we were able to develop a model that achieved greater than 50% accuracy (comparable to some of the models we found in our research) and create an app that uses that model to perform lip reading in real time, accomplishing our primary objectives.

References

1) Krishna J. N, "Lip Reading Using Neural Networks and Deep Learning," Indian Scientific Journal Of Research In Engineering And Management, Apr. 2023, Available: https://www.academia.edu/109271764/Lip_Reading_Using_Neural_Networks_and_Deep_Learning
2) T. Afouras, J. S. Chung, and A. Zisserman, "Deep Lip Reading: a comparison of models and an online application." arXiv, Jun. 15, 2018. doi: 10.48550/arXiv.1806.06053.
3) J. S. Chung and A. Zisserman, "Lip Reading in the Wild," in Computer Vision – ACCV 2016, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 87–103. doi: 10.1007/978-3-319-54184-6_6.
4) J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip Reading Sentences in the Wild," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 3444–3453. doi: 10.1109/CVPR.2017.367.
5) Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-End Sentence-level Lipreading." arXiv, Dec. 16, 2016. doi: 10.48550/arXiv.1611.01599.
6) A. Abedi and S. S. Khan, "Improving state-of-the-art in Detecting Student Engagement with Resnet and TCN Hybrid Network," 2021 18th Conference on Robots and Vision (CRV), pp. 151–157, May 2021, doi: 10.1109/CRV52889.2021.00028.
7) "Important Hearing Loss Statistics and Studies 2023." Accessed: May 02, 2024. [Online]. Available: https://www.soundly.com/blog/hearing-loss-statistics
8) Z. Li, T. Lohrenz, M. Dunkelberg, and T. Fingscheidt, "Transformer-Based Lip-Reading with Regularized Dropout and Relaxed Attention," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, Doha, Qatar: IEEE, Jan. 2023, pp. 723–730. doi: 10.1109/SLT54892.2023.10023442.
9) H. Huang *et al.*, "A Novel Machine Lip Reading Model," *Procedia Computer Science*, vol. 199, pp. 1432–1437, Jan. 2022, doi: 10.1016/j.procs.2022.01.181
10) Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic, "Lipreading using Temporal Convolutional Networks," arXiv:2001.08702, 2020.
11) N. A. Altieri, D. B. Pisoni, and J. T. Townsend, "Some normative data on lip-reading skills (L)," *J Acoust Soc Am*, vol. 130, no. 1, pp. 1–4, Jul. 2011, doi: 10.1121/1.3593376.
12) "Oxford's lip-reading AI outperforms humans," New Atlas. Accessed: May 03, 2024. [Online]. Available: https://newatlas.com/oxford-lip-reading-ai/48460/
13) H. Wang, G. Pu, and T. Chen, "A Lip Reading Method Based on 3D Convolutional Vision Transformer," *IEEE Access*, vol. 10, pp. 77205–77212, 2022, doi: 10.1109/ACCESS.2022.3193231.

Appendix A: Detailed Roles

Table 1. Team member contributions

Name	**Task**	**File names**	**Code lines**
Usman	● Gathered Grid dataset ● Started implementation of the CNN for Grid dataset ● Built basic transformer for Grid dataset ● Built app for real time prediction	● CNN_Grid.ipynb ● Transformer_Grid.ipynb ● app.py	~ 1000
Chris	● Developed Grid Transformer ● Refined CNN Grid model	● Transformer_Grid.ipynb ● CNN_Grid.ipynb	~ 800
Cole	● Developed LRW Preprocessing ● Developed ResNet TCN ● Developed LRW Transformer ● Gathered LRW dataset	● LRW_CNN.ipynb ● transformer.ipynb	~ 800
Thomas	● Developed Grid CNN	● CNN_Grid.ipynb	~600

Appendix B: Code Repository

- https://github.com/chris-krenz/ec523-lip-reading-project