CV 498 – FINAL REPORT

Nicole ElChaar & Joshua Krinsky
(nje222 & jpk3222) @lehigh.edu

Lehigh University
Computer Science & Engineering
113 Research Drive, Bethlehem, PA 18015

ABSTRACT

The recent advancements in automated lip reading have recently made the technology applicable to real scenarios such as recovering old and damaged videos or effective speech-to-text for the hearing impaired. Using a standard 3D CNN and 2D CNN lip reading model our paper proposes a new model utilizing a stacked hourglass as the 2D CNN. This model can capture both smaller and larger features from image frames to better understand the words being said. Although ultimately our model suffers from some failures and does not perform to the state-of-the-art we do think the model shows promise.

***Index Terms*—** Computer Vision, Automated Lip Reading, 3D CNNs, Hourglass Network

1. INTRODUCTION

Sometimes the audio of a speaker within a video can be become unintelligible, manipulated, or lost. In these cases, we require technology that can interpret the lip movement of a speaker and return it as text.

Recovering text through solely or mostly visual means is referred to as automated lip reading. It has important applications in media, communications, and visual forensics. Understanding how to extract the proper features from a video of a speaker becomes challenging as it is important to capture both large and small scale details within the frame without clouding the networks judgement with needless features. In this project, we are hoping to develop a system that can reconstruct speech from speakers in a video rather than rely exclusively on audio. Current methods for automated lip reading use a combination of CNN models, as well as multi-sensory input such as sound and heat maps. In this project, we propose an updated method that uses a stacked hourglass framework 2D CNN to extract both large and small scale features from videos. We will weight frames by their volume and how close they are to the center of the spoken word.

2. LITERATURE REVIEW

Lip reading has advanced quickly over the past several years. In 2019, the optimal method for automated lip reading consisted of a 3D convolutional layer followed by an 18 layer Residual Network (ResNet) and a softmax layer. These methods produced the highest accuracy on the LRW (English) and LRW1000 (Mandarin) datasets until Temporal Convolutional Networks (TCN).[1] TCNs instead use dilated, causal 1D convolutional layers, each layer of the same dimensions, to track movements from tensors over time. The TCN is causal in that each next layer of the CNN relies only upon the previous layer and dilated in that the kernel size is widened by skipping kernel elements at a specified step. By January 2020, the state-of-the-art model for recognizing isolated words from faces was a residual network with Bidirectional Gated Recurrent Unit (BiGRU) layers [1]. This paper proposes that TCNs can be better because they greatly simplify training and produce an absolute improvement of 1.2% in the Lip Reading in the Wild English dataset. The training protocol first involves training with a single-layer TCN, returning to the BiGRU with parameters randomly initialized, and finally fine-tuning the parameters of the BiGRU. The authors follow this procedure to create stronger feature encoding layers, although the paper notes that the method requires 3 weeks of GPU time. [1]

Other systems use a combination of audio and visual features for automated lip reading. [2] The traditional approach is to use hidden Markov models (HMMs) to extract temporal information of speech and Gaussian mixture models (GMMs) to discriminate between different HMMs states. Using 3D CNNs to map both modalities into the same representation space is recommended by the Coupled 3D Convolutional Neural Networks for Audio-Visual Recognition paper. By incorporating spatial, temporal, and audio streams, we can expect to enhance the performance of our system. [2]

Criticisms of 3D CNNs for automated lip reading are in the the lack of fine-grained adjustments that are required to detect lip movements. In 2021, a new proposal for improving lip reading with hierarchal pyramidal convolution (HPConv) was developed to replace conventional convolution features.

[3] This approach integrates local lip movements with the original feature set. We intend to adopt part of this approach by isolating the mouth, as is consistent across much of the literature, along with the stacked hourglass 2D CNN to extract local features. [4] Boosting Lip Reading with a Multi-View Fusion Network (MVFN) takes a similar approach [5]. Combining TCNs with an Adaptive Spatial Graph Model (ASGM) to incorporate appearance and shape information of the lips significantly outperforms the baseline. The MVFN separates out cropped sections of the lips from the rest of the image, runs the cropped section through 2D and 3D convolutions and an 18-layer ResNet to extract a Spatial Topology and Spatial Feature map. The image of the whole face is then run through a pre-trained fully convolution network to extract a heatmap of motions and is integrated with the spatial map feature to be used in a preliminary prediction. This preliminary prediction is combined with the 3D convolution's preliminary prediction to make a final prediction. [5]

Other literature also found that full facial expressions are key in the performance of lip-reading models.[6] When adding in a dynamic face deformation model to track expressions, the performance of visual speech detection increases significantly. While we do not need to use face frontalization transformations on the LRW dataset to deform faces, we can expand our exploration by adding additional facial features to track expressions, rather than just isolating the mouth. This is consistent with higher accuracy metrics found in Boosting Lip Reading with a Multi-View Fusion Network in that using multiple methods for prediction (as opposed to more layers of the convolution) can improve performance without overfitting the data.[5]

Multi-modal approaches are common among well-established papers, with some debate on how to combine predictions. Rather than using averaging, it was found that other consensus methods and weighing are helpful in model performance.[4] In our analysis, we hope to achieve something similar by weighing the relative importance of frames.

3. DATA SET

The Lip Reading in the Wild (LRW) dataset is used by all but one paper referenced. The dataset consists of 29-frame video clips (mp4s) of 500 spoken words. There are 800-1000 instances of training data for each spoken word (488,776 training videos in total) with separate sets for testing and validation. Words are centered in the clips so that the 15th frame is the middle frame of the word spoken. The cleanliness of this dataset and wide use make it ideal for comparing our performance against existing approaches'. [7]

Videos are named by the word and instance number. Each folder consists of a training, validation, and testing folder with files named {WORD}_{INSTANCE}.mp4 where WORD is the word centered in the clip and INSTANCE is the value from 0-999 indicating the ID of the particular file.

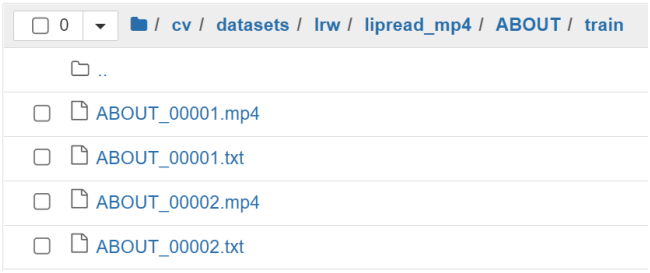

Fig. 1. Structure of unzipped LRW data

Fig. 2. Frame 14 from ABOUT_00001.mp4 in the training set

The structure of each directory is shown in Figure 1, where "train" can be replaced by "val" or "test", depending on the type of data stored.

A sample frame is shown in Figure 2.

Alternatively, we considered using the Lip Reading Sentences in the Wild (LRS3) dataset which consists of thousands of spoken sentences from TED and TEDx videos. While we originally downloaded this set we found that its size of almost quadruple the size of the LRW dataset and the requirement of facial frontalization (most faces angled away from the camera during the clips) made it more difficult to get started with.

There is also the LRW1000 dataset which consists of Mandarin words and the LRW-R set of Russian words. As English speakers, we would have more difficulty translating this data and understanding differences in the language that would lead to confusion of the model, so we opted to stick with the English LRW set instead.

4. APPROACH

4.1. Preprocessing

Before we are able to build, test and train our model we have to preprocess our data. We chose to preprocess our audio frames by weighing each according to their average absolute amplitude and distance from the center frame. For visual data, we extracted each frame from the video and then extracted each mouth from each frame. When faces could not be detected in a particular frame, we copied the data of the mouth from the frame before (as would occur in a buffering video stream). If more than 4 frames of a video in the training set need to be replaced, we discard the video.

4.1.1. Pyspark

We chose to batch and pipeline our process using user-defined functions in Apache Spark. By using transformers, we can take input paths to particular images, output the paths to each individual frame, and extract the mouths into separate image files. In a separate pipeline, we create a wave file and determine the relative amplitudes.

This is helpful when working with this large of a dataset, as Spark automatically batches and parallelizes the computation. When our server times out, we can use Spark to lazily identify data that has yet to be processed as it attempts to extract new images.

4.1.2. Mouth Extraction

To focus on lip shape in our model, we want to extract the mouths from each frame. We followed a tutorial [8] that used shape detector trained on the iBUG 300-W data set. Then we took this trained model to label the mouth of each frame and created a bounding box around. Finally we took the bounding box, made it an image, resized it (so all images are constant) and saved it, so we have the extracted mouth from every frame. These extracted mouths look similar to figure 4.

4.1.3. Audio Weighting

Not all frames are created equal. Gaps between words or within a word (like the small pause when saying the portmanteau "something") should not be considered as heavily in the network as those that contain more useful lip movement. To factor this into our model, we extract the wave signal from every video and then get the average. After getting the average, we break the wave form into 29 equal partitions (1 for each frame) and get the absolute average. Frames where the corresponding partition's average signal strength is lower are scaled down and sections where the frame's corresponding partition is higher scaled upwards. These weights will be factored into the model that we use, hopefully giving more clarity as to which frames of video are more worth attention than others.

4.2. Model

As discussed within section 2 many lip reading models are built with a general model of a 3D CNN followed by a 2D CNN, followed by a GRU. The 3D CNN extrapolates the importance between frames, the 2D CNN finds the key features in each individual frame and the final section factors in the importance of the order of each frame. For an attempt at an improved lip reading model we looked toward improving the feature extraction from each individual frame. This is because although ResNet-18 is an exceptionally versatile model and used as a backbone model for many applications, the model has a weakness against picking up small features from frames.

Before discussing our changes to the 2D CNN we should discuss the parts of the model that remain consistent. The 3D CNN comes from Feng et al. Learn An Effective Lip Reading Model Without Pains[9]. This code uses a 3D convolution with 64 output frames and compresses the original frames from 88x88 to 22x22. The model also does batch normalization, ReLU activation and a final 3D max pool. From the output of the model, the BiGRU temporal convolution also remains the same. One GRU is used in the forward direction input and another in the backward direction, each with a reset and update gate. It is key when passing values through to the BiGRU that the values maintain their time dimension even while being flattened.

The stacked hourglass model has shown great performance with regard to feature extraction in the problem of pose estimation [10]. Since the repeated down- and up-sampling of a stacked hourglass model allows for the network to extract key features at many scales within the image. In the stacked hourglass model the highest scale features are detected in lowest levels of the hourglass when the image is squashed down to a small size and the smallest features are detected at highest levels of the hourglass when the frame is expanded up to its original size. Our hope was that this would allow for large scale features of the lips as well as small scale lip features that are important for differentiating words that sound very similar like 'Country' and 'Countries'. Something ResNet cannot does not specifically attempt to capture. For the purposes of our lip reading model we use the hourglass model to replace the ResNet model. Due to the size of the images (22x22) multiple hourglasses did not improve the accuracy but instead seemed to only confuse and worsen the output. Instead, we use one hourglass as can be seen in figure 3 the hourglass first extracts the important features from 64x22x22 then pools the image to 64x11x11 and gets the extracted features, finally up-sampling back to 22x22 and adding together the extracted feature outputs from up-sampled 11x11 and extracted features from original 22x22 for a final output of 22x22x64, which ultimately gets flattened

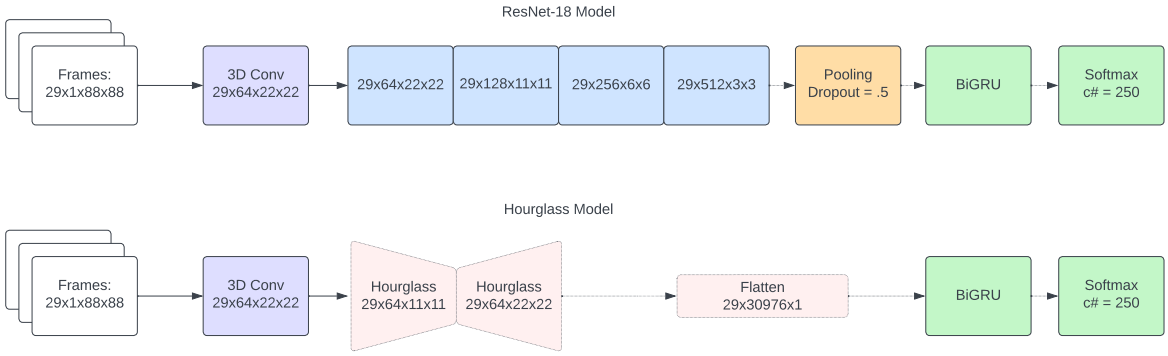

Fig. 3. Depiction of the 2 models used in testing for the lip reading. First, the baseline ResNet-18 model which after putting the 29 image frames through a 3D CNN put it through a ResNet model to extract the features from each individual frame. Second is the Hourglass model, our attempted change is to replace the ResNet model with an hourglass model.

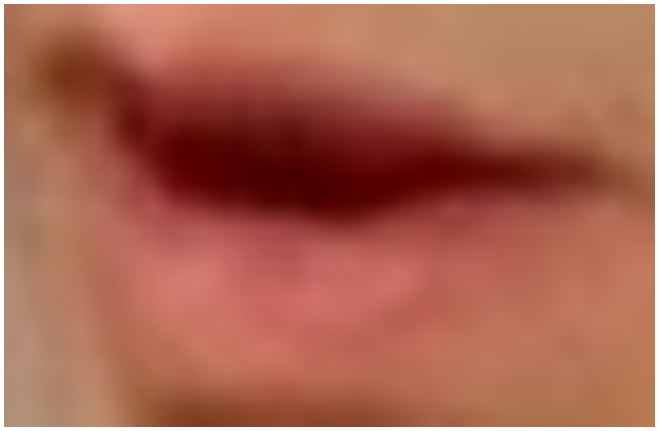

Fig. 4. Example of an extracted mouth from the preprocessing code. (2nd frame of the 13th sample of "money" in the validation set)

to 30976x1 because the GRU is expecting a 1 dimensional tensor for processing. The largest difference of changing this model is certainly the number of parameters we now pass into the GRU, before with ResNet we were passing in 512 parameters to the Gated Linear Unit (GRU) but now we have over 60 times the parameters (30976). As well, the pooling step in the original model is replaced by the flattening of the parameters.

The final step of our model is the Gated Linear Unit which is a type of Gated Reccurent Unit (GRU). A GRU functions much like an LSTM, however it has been shown to perform better in certain language learning tasks [11]. For our model the importance of the GRU is to determine the temporal importance of each frame to determine which class the word falls into. Since the 3D CNN can extract the value between concurrent frames and our 2D CNN network can extract the feature of each individual frame the last factor to consider in our network is the order of the frames as a whole. Essentially with this GRU we are trying to both retain information about current frames while attempting to remeber the information we saw in previous ones. [12]

5. EXPERIMENTATION

5.1. Baseline

The model we borrowed from found that ResNet18 was most effective on the LRW set.[9] This model has 512 learnable parameters, and we used it as a baseline to compare against our hourglass network.

5.2. Hyperparameters

Regardless of whether the model was the baseline or hourglass, during training, we kept mixup, lr, num_workers, and label_smooth stable. We use mixup to shuffle the data because we do not yet know which words are most easily learned. Learning rate, lr, is set to 0.0003 as higher learning rates led to a more biased model and lower rates slowed learning beyond an achievable number of epochs. Because we ended up using so few epochs, a higher learning rate may have been helpful. Number of workers, num_workers, refers to the number of threads to use in loading in the data. As we are not using any other threads during data loading for the model, we found 32 to be a good balance. Label smoothing, label_smooth, refers to how the model uses predictions to compute loss in training. When True, it uses q_{is} from Equation 1 instead of q_i from Equation 2 in the loss function. This makes the model generalize better on unseen videos.[9] Because of limited time, we chose to train and test on only half our samples by selecting every other word in the original LRW dataset, leaving 250 words.

$$q_{is} = \begin{cases} \epsilon/N, y \neq i \\ 1 - \frac{N-1}{N}\epsilon, y = i \end{cases} \quad (1)$$

$$q_i = \begin{cases} 0, y \neq i \\ 1, y = i \end{cases} \quad (2)$$

The hyperparameters that changed depending on the model type are batch_size, save_prefix, and baseline. Because of the high number of learnable parameters that came from using the hourglass, we had to reduce batch size from 64 to 4 when using the hourglass model. The checkpoint path also changed depending on the model type. When running the baseline, we set baseline to True.

The last hyperparameter used in training was to load weights from the last successful checkpoint. On timeout, we used the most recent weights to continue training by setting the weights parameter to its path.

On testing, we switched the test parameter from False to True.

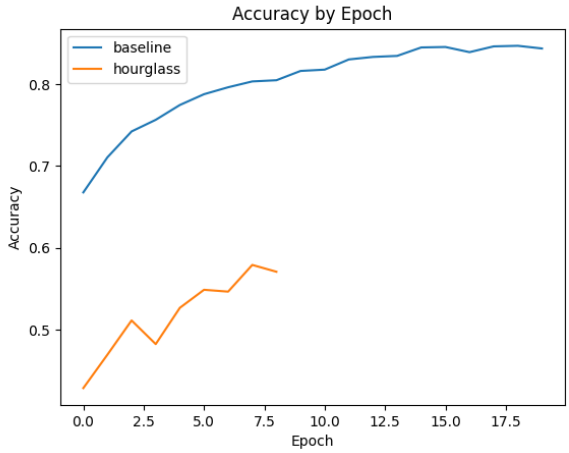

Fig. 5. Accuracy plot of both the baseline ResNet model and our modified hourglass model. Baseline model had 19 epochs with ultimate accuracy of 88.4%, hourglass had 9 epochs with highest accuracy of 60%

6. RESULTS & DISCUSSION

The results of every plot seems fairly conclusive that our approach is not better than the state-of-the-art models. Not only is our process ultimately slower to train and test due to the increased number of parameters, which resulted in us not being able to complete all 19 epochs we intended to, but our model also performs much worse than the current model. The ResNet model after 19 epochs of testing hit a maximum accuracy of 88.4% reaching almost 80% after just 10 epochs. The high accuracy apposes our model which only performed at about 60% accuracy after 9 epochs. The accuracy values for all epochs can be seen in figure 5. We had hoped that even with worse accuracy maybe the hourglass method could perform more consistently, maybe failing in fewer instances than the ResNet model but, as can be seen in figure 7 the box and whiskey plot this is not the case. The baseline model is not only more accurate but more robust, having very little variation in the accuracy for each batch as the box and line are nearly identical and there are few outliers where as our hourglass model varies in accuracy by almost 15%. Our lowest outlier being a batch with 50% accuracy and our highest outlier being 65%. Some of this variation may come from the fact that due to the requirements of the hourglass model we needed to use smaller batch sizes than for the baseline model. Both hourglass and baseline error plots are very close, given then constant and controlled decent towards zero this implies that both models are in fact learning. We can predict that the hourglass model could continue to decrease in loss, so while our hourglass model is learning we believe that the increase in parameters may lead to massive confusion inside the GRU, over training, and focusing on unimportant features resulting in lower accuracy.

Our confusion matrix does indicate some themes in mistakes of the model. The word ”thought” was mistaken for the word ”water” 11 times, likely because the first syllable of ”water” forces a similar mouth shape to ”thought”. Other mistakes are understandable as we did not crop the section of the frames containing the word. Mispredictions like ”capital” mispredicted as ”Russia” (9 times), ”legal” and ”England” (8 times), and ”justice” and ”central” (9 times). Other predictions, like that ”human” was predicted as ”great” 14 times, are not logically understood without our feature set.

7. FUTURE WORK

Our main limitations in this project were hardware and time based. With 538,776 videos (including the 25,000 in the validation set and 25,000 in the testing set), it was difficult to run computationally-expensive operations over and over to test different hyperparameters. As the MAGIC cluster times out after 30 minutes of inactivity, we had to repeatedly update and re-run steps of preprocessing and find ways to store and load our model mid-way through training. As a result, there is much more exploration to do in terms of the network itself, hyperparameters, and inputs.

Firstly, we would like to explore why the stacked hourglass network did not perform well. The downsampling and upsampling of stacked hourglass networks makes them useful at extracting features regardless of their relative size in the image. Because our preprocessed images only showed mouths, it is possible that there was not enough detail for the stacked hourglass to pick up on. Using the image of the whole face in training and testing, as supported by the benefit of using full facial features in lip reading [6], would be a likely next step.

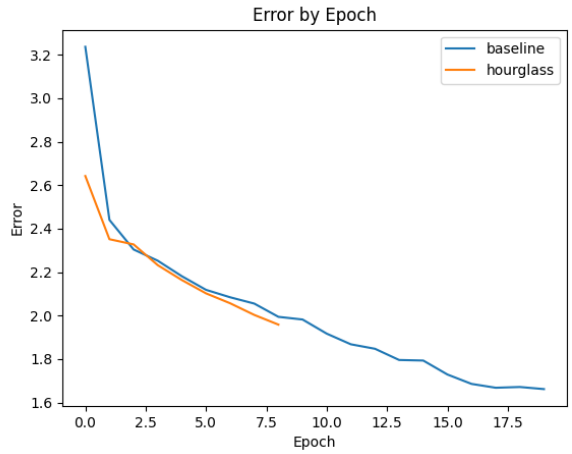

Fig. 6. Error plot of both ResNet and hourglass model both have a strong consistent drop in loss, explaining their increases in accuracy. Each value is the average loss per batch for the epoch.

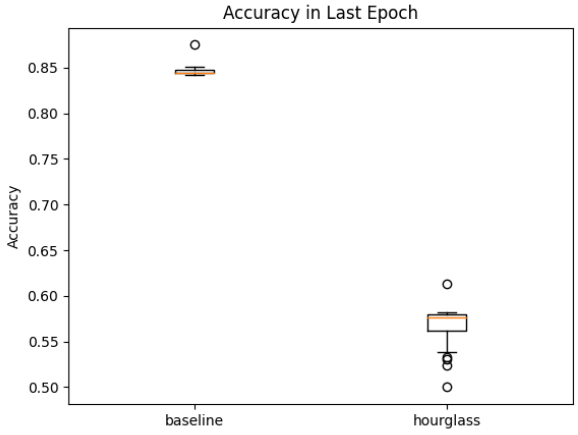

Fig. 7. Box and whisker plot for the final accuracy scores of both the ResNet and Hourglass model.

Hardware limitations also prevented the use of more hourglass layers. The multiplicative nature of learnable parameters made it impossible to run, unless batch sizes were only one or two frames.

Other steps we explored and did not commit to in this project included Gaussian scaling for additional frame weights and color correction. Weighing frames closer to the center of the video higher than those towards the the beginning and end of the video could be helpful in generating a more performative model. The more identifiable lip shapes for a word tend to be toward the center of the word [5], and the beginning and end of each video may contain the previous and following word which we do not want the model to recognize. We also created code to preprocess our images with Gamma correction but skipped this step due to the high cost of time.

8. REFERENCES

[1] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic, "Lip reading using temporal convolutional networks," 2020.

[2] Amirsina Torfi, Seyed Mehdi Iranmanesh, Nasser Nasrabadi, and Jeremy Dawson, "3d convolutional neural networks for cross audio-visual matching recognition," *IEEE Access*, vol. 5, pp. 22081–22091, 2017.

[3] Amirsina Torfi, Seyed Mehdi Iranmanesh, Nasser M. Nasrabadi, and Jeremy Dawson, "3d convolutional neural networks for cross audio-visual matching recognition," 2017.

[4] Hang Chen, Jun Du, Yu Hu, Li-Rong Dai, Bao-Cai Yin, and Chin-Hui Lee, "Automatic Lip-Reading with Hierarchical Pyramidal Convolution and Self-Attention for Image Sequences with No Word Boundaries," in *Proc. Interspeech 2021*, 2021, pp. 3001–3005.

[5] Xueyi Zhang, Chengwei Zhang, Jinping Sui, Changchong Sheng, Wanxia Deng, and Li Liu, "Boosting lip reading with a multi-view fusion network," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022, pp. 1–6.

[6] Zhiqi Kang, Mostafa Sadeghi, Radu Horaud, and Xavier Alameda-Pineda, "Expression-preserving face frontalization improves visually assisted speech processing," 2022.

[7] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*, 2016.

[8] Adrian Rosebrock, "Detect eyes, nose, lips, and jaw with dlib, opencv, and python," 2017.

[9] Dalu Feng, Shuang Yang, Shiguang Shan, and Xilin Chen, "Learn an effective lip reading model without pains," *arXiv preprint arXiv:2011.07557*, 2020.

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds., Cham, 2016, pp. 483–499, Springer International Publishing.

[11] Yuanhang Su, Yuzhong Huang, and C.-C. Jay Kuo, "On extended long short-term memory and dependent bidirectional recurrent neural network," *CoRR*, vol. abs/1803.01686, 2018.

[12] Dina Ibrahim, Dina Hussein, Amany Sarhan, and N. Elshennawy, "Hlr-net: A hybrid lip-reading model based on deep convolutional neural networks," *Cmc - Tech Science Press-*, vol. 68, pp. 1531–1549, 04 2021.