Lips Don't Lie

Mitchell Butovsky
butovsky.mitchell@gmail.com

Tom Bekor
tom.bekor@gmail.com

January 2022

1 Introduction

1.1 Previous Work

We got the inspiration for this project from the paper "Reading Sentences In The Wild"[2], in which they achieved state-of-the-art results on this task, as well as introduced a new visual speech recognition dataset which consists of over 100,000 natural sentences from British television. In order to collect the data they used a variety of BBC programs recorded between 2010 and 2016. Their processing pipeline is summarised in figure 1.

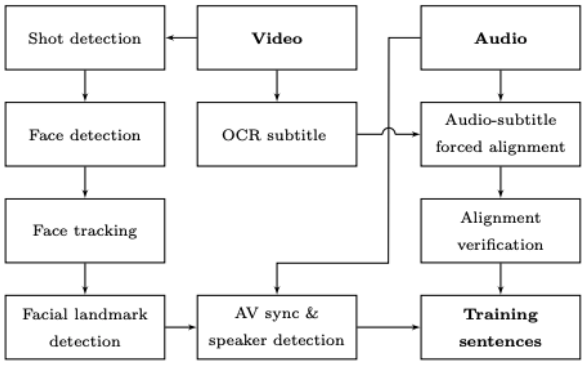


Figure 1: Pipeline to generate the 'Lip Reading Sentences' (LRS) dataset

Their model is LSTM-based model which auto-regressively produces characters. The model consists of three key components: the image encoder "Watch", the audio encoder "Listen" and the character decoder "Spell". The image encoder "watch" consists of a VGG-M based convolutional network which produces image features that are then passed to an LSTM encoder. The audio encoder "Listen" consists of solely an LSTM which gets as input 13-dimensional MFCC features. Finally, the "Spell" decoder is an LSTM decoder with attention with

respect to the hidden states generated by the former encoders. Their architecture is summarized in figure 2

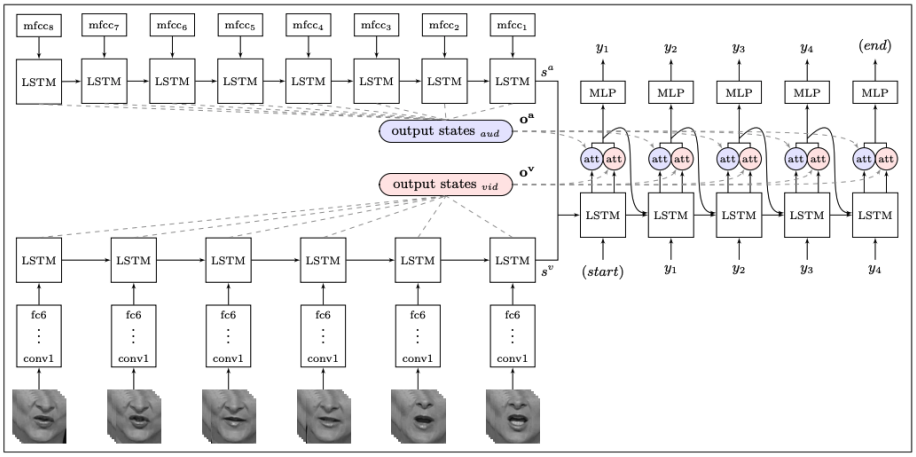


Figure 2: Watch, Listen, Attend and Spell architecture.

However, in our work we intend to use only the visual data, as if the audio of the videos wasn't given at all.

1.2 Dataset

In order to be able to train the model, we had to choose an easier dataset than the "Lip-reading sentences in the wild" dataset which is very hard (tons of real life data, a lot of words and sentences spoken, movement of the speakers in a varying backgrounds etc.). We noticed that in the "Lip-reading sentences in the wild" paper, they examined their model on an additional dataset named "GRID dataset". The GRID dataset consists of 34 subjects, each uttering 1000 phrases. The utterances are sequences of the form verb (4) + color (4) + preposition (4) + alphabet (25) + digit (10) + adverb (4). Note that the vocabulary is very small (magnitudes smaller than the LRS dataset), but the number of possible sentences is very high (as well as the number of the sentences that appear in the dataset). In addition, the speakers talk very clearly and are recorded in front of a blue screen.

Figure 3: The structure of the sentences in the GRID dataset

Figure 4: A frame token from a video in the dataset. The speaker appears very clearly and is recorded in front of a blue screen.

2 Methods

2.1 Face and Landmarks

Following the preprocess in [2], we focused on the speaker lips in order to improve our learning process. To do so, we cropped a bounding box of the lips for each frame of the video. This was made using face recognition (Facial HOG Pyramid [3, 4]) and facial-landmarks predictor (BlazeFace[1]) pretrained models. In our first idea, we used the landmarks we found to get a bounding box of the lips. The BlazeFace model returns 68 (x, y) facial landmarks, where landmarks 48-68 indicates the lips, and in order to get the lips' bounding box we took a fixed size rectangle, which centered over the centroid of these 20 landmarks. Later we changed our attitude, and instead of using the lips' bounding box, we fed our model with the lips' facial landmarks themselves.

2.2 Architecture

2.2.1 Initial Model

As our first model, we used a combination of a pretrained VGGNet, as backbone, and a classical Transformer model. This model worked with the first version of the preprocessing as follows: after the preprocessing was done, we got the lips' bounding box for each video frame. These bounding boxes is in gray scale, they where up scaled with 1×1 conv and loaded into the VGG. We took from the VGG only its backbone, which was trained to capture more local and general features in the input image, and we replaced its linear layers by new ones, so they will precept the unique global features in our inputs. The VGG created from the bounding boxes feature vectors that creates our vector sequence, which was loaded into the transformer. The transformer outputs were loaded into a

fully connected neural network, and created our final output sequence, that is, the transcribed sentence. See figure 5.

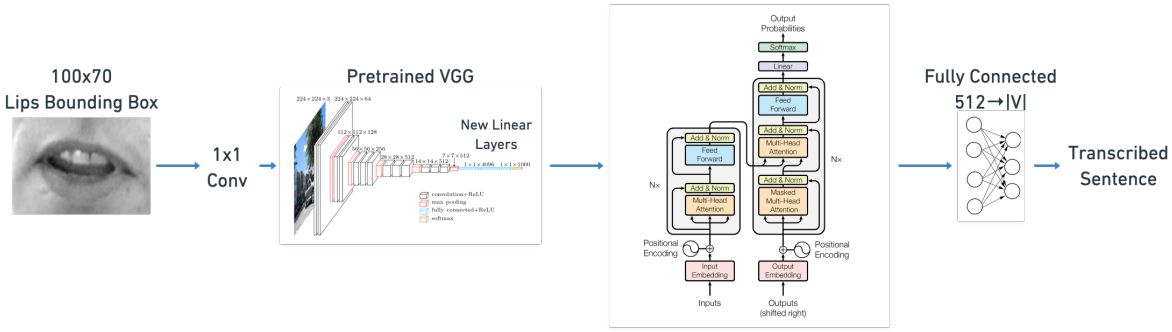


Figure 5: First architecture - mouth bounding box into 1x1 convolution to increase the number of channels, then into pretrained VGG, where the linear layers are replaced with new ones, each sequence fed into the transformer, and using a fully-connected neural network the final predictions are created.

2.2.2 Landmarks Model

After we thought about how can we improve the model, we came to the idea of using only the landmarks, without the need for the lips' bounding box. Logically, the landmarks are exactly the features we would want to extract from the bounding box, so instead of using the pretrained VGG, we fed the landmarks into a 3-layer MLP , with ReLU activations and dropout of 0.3, to create more complex relations between them. The output of the MLP has the same shape as the transformer inputs shape.

2.2.3 Enhancing the learning process with Temporal information

After some thought about the model we designed, we noticed that the absolute location of the landmarks is not important, but how these landmarks change during time is really the interesting feature in our case. Moreover, if one speaker moves from one point to another in different videos this would make the learning harder. Aiming to solve this issue we applied a very simple trick which yielded a GREAT improvement in the results! instead of considering the raw landmarks of a video we considered the element-wise shift of the landmarks. Namely, instead of feeding a raw sequence of landmarks: $(\bar{l}_1, ..., \bar{l}_n)$ we fed to the model the sequence: $(\bar{d}_1, ..., \bar{d}_{n-1})$ (where $\bar{d}_i = \bar{l}_{i+1} - \bar{l}_i$). Afterwards, we took this idea even further; intuitively, if we would take a glance on few frames before the current one, they would add an important information - "information from the past" which helps to understand the current frame. Following this intuition, we concatenated to every frame landmarks (or, to be more precise, landmark shifts)

the landmarks shifts from the previous $w-1$ frames. Thus, the final input to our model was of the form: $(\bar{t}_1, ..., \bar{t}_{n-w})$ where $\bar{t}_i = concat(\bar{d}_i, \bar{d}_{i+1}, ..., \bar{d}_{i+w-1})$.

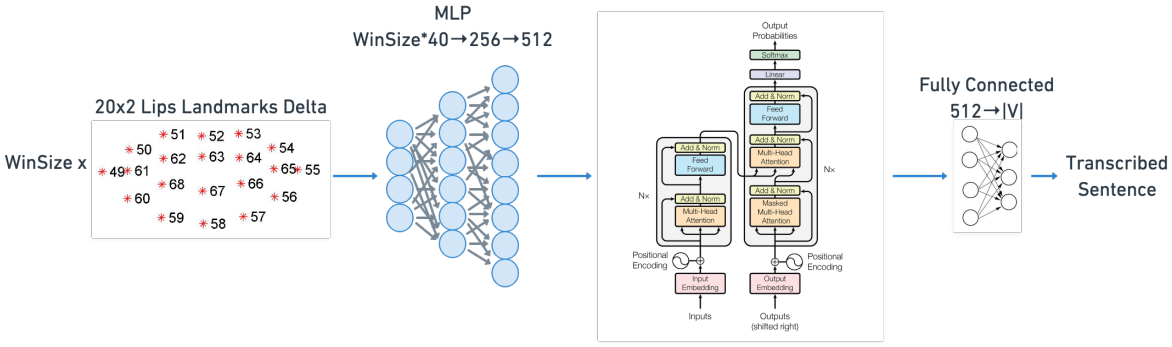


Figure 6: Final Model - lips landmarks delta are concatenated through WinSize number of frames, fed to MLP, which creates sequences that are fed into the transformer, and using a fully-connected neural network the final predictions are created.

3 Experiments and Results

3.1 Training Process

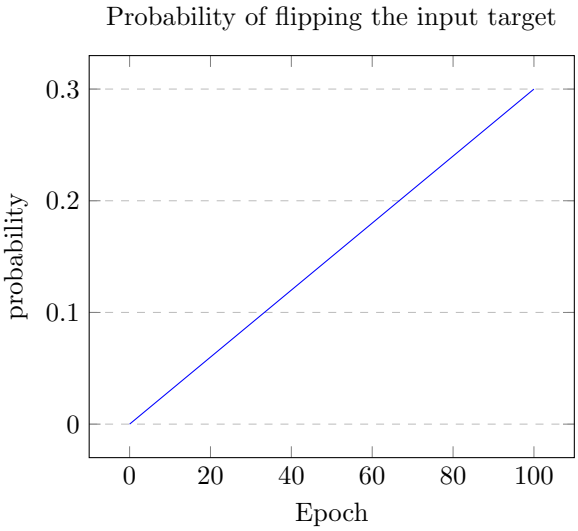
We trained the model using CE Loss, Stochastic Gradient Descent (with $lr = 1 \times 10^{-1}$) and Adam optimizer with the default parameters. We used the "reduce learning rate on plateau" scheduler on the loss with $patience = 2$ and $factor = 0.1$ and updated it every 50 iterations (namely, we reduced the learning rate by 90% each time the loss didn't decrease for $2 * 50 = 100$ times).

In addition, in order to improve generalization, when training the transformer we changed some of the targets which were passed as an input to the transformer to be a random token from the vocabulary (except the sos, eos and pad tokens). Each target that was passed as as input to the transformer was flipped to a random target independently, with probability p_{epoch} which was increased gradually during training;

This improved the validation accuracy with about 3%, by forcing the model to focus on the information it gains from the frames, but still lets it exploit this information. We clarify that the reference targets that were used in the loss functions remained untouched.

3.2 Evaluation

Note that the task of lip reading is a sequence-to-sequence task. In general, evaluating the performance on such tasks is difficult because of the fact that



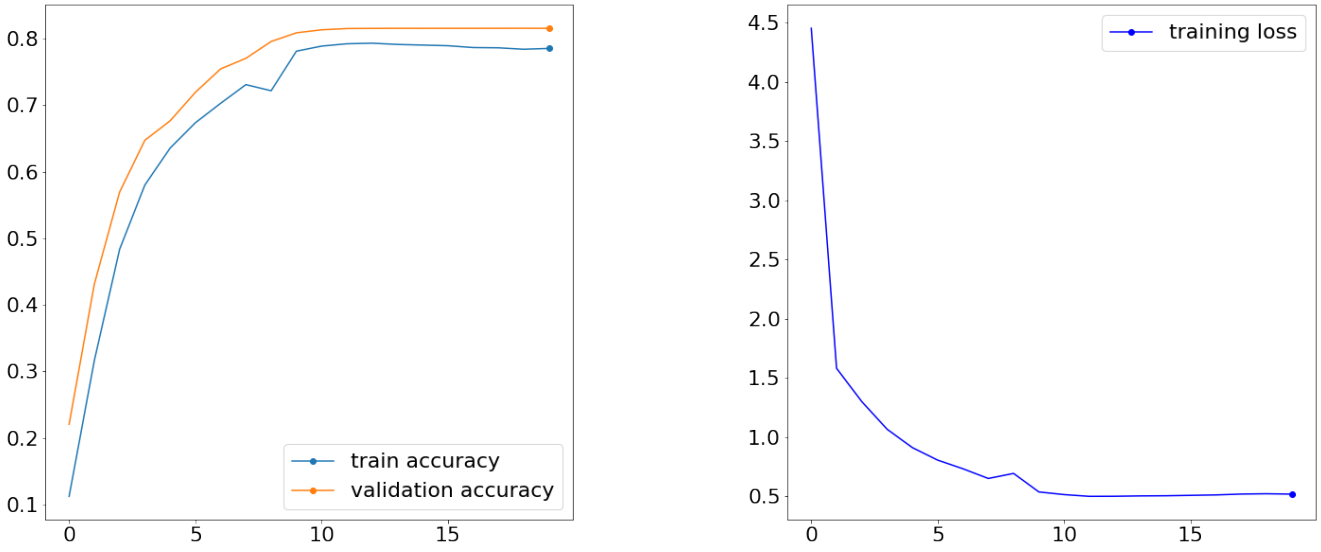
the recognized word sequence can have a different length from the true word sequence. One simple and common approach for evaluating such tasks is using the WER (word error rate) metric. This metric is calculated as follows:

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

where S, D, I are the numbers of word substitutions, insertions, and deletions made in the generated sequence with respect to the ground truth sequence. It is also common to define the $W_{acc} := 1 - WER$. We noticed that in our case, all of our purposed models always generated sentences which match the structure of the spoken sentences. This is due to the fact that all the dataset sentences are fixed structured, as mentioned in section 1.2. Note that in this case, the W_{acc} metric exactly equals to the per-word accuracy.

3.3 Results

After training for only 15 minutes on a microsoft Azuer's platform we achieved a final test W_{acc} of 83.2%, compared to 97% W_{acc}. We present here the plots for the loss and the accuracy 7



(a) accuracy

(b) loss

Figure 7: Final models results - accuracy on training and validation sets, and loss on training set.

As we can observe the validation accuracy exceeded the train accuracy. This might seem surprising at first glance, but recall the random flipping mentioned in section 3.1.

3.4 Future Work

Future work could be to try to match current SOTA results and to extend our project to work on the LRS dataset.

References

[1] Valentin Bazarevsky et al. "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs". In: *CoRR* abs/1907.05047 (2019). arXiv: 1907.05047. URL: http://arxiv.org/abs/1907.05047.

[2] Joon Son Chung et al. "Lip Reading Sentences in the Wild". In: *CoRR* abs/1611.05358 (2016). arXiv: 1611.05358. URL: http://arxiv.org/abs/1611.05358.

[3] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[4] Pedro F. Felzenszwalb et al. "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9 (2010), pp. 1627–1645. DOI: 10.1109/TPAMI.2009.167.