Visual Speech Recognition

Alexandra Gaál, Dorottya Nyárády, Dániel Berdó

Abstract

Lip reading also known as visual speech recognition recognizes the speech content from videos by processing the lip motion. It is definitely a difficult task and it depends on the speaker's articulation. In some cases they use the audio to help the recognition. In our project we use only the video frames for lip reading because first we want to examine the limitations of recognition based on only visual information. Nowadays this field has a growing attention but despite of the various, well-organized datasets (e.g. LRW, LRS, GRID) we didn't find a state of the art solution for the task because of its difficulties.

According to other previous experiments it seems a good way to use 3D Convolution with an effective model (e.g. VGG, ResNet) and recurrent layers (e.g. GRU, LSTM), so we created a network containing these elements. This network has numerous parameters so it is a very important task to use methods which help in avoiding overfitting during training. For training and validation we processed the LRW 1000 dataset. For testing the trained model we used the test set from the LRW 1000 dataset as well.

***Index Terms*—Visual Speech Recognition, Deep Learning, Lip Reading**

Introduction

Lip reading is the ability to understand what people are saying only based on visual information. It is an impressive skill and it is very difficult too even for a human. So what can we achieve with a machine in this task? The recent development of the deep learning techniques demonstrated that the problem can be solved with machines as well.

In this field there are audio speech recognition techniques but they are very sensitive to the audio noise, so lip reading seems to be a better way in several practical problems:

- helping aids of hearing-impaired people to understand other's speech without using the sign language
- helping people who would like to learn a new language and practice it with watching videos
- dictating messages to a phone in noisy environment
- creating subtitles for silent movies, videos
- resolving multi-talker simultaneous speech etc.

Lip reading is a challenging task because several factors make it difficult to recognize the speech e.g. lighting conditions, speaker's age, make-up, viewpoints etc. Moreover the differences among some of the phonemes or visemes are slightly noticeable (e.g., "b" and "p", "d" and "t") although their corresponding utterances can be easily distinguished. These factors determine the limits of lip reading.

Related work

Learn an effective lip reading model [5]

In this paper a popular pipeline is used considering lip reading. As a baseline ResNet-18 is used as the frontend module and the first convolutional layer implements 3D convolution with kernel size of 5×7×7. After that a global average pooling is performed on the output of ResNet. The output features are feeded to the backend network which is 3 layers of bidirectional GRU. After these a dropout layer is added to the network. Finally the last fully connected layer's output dimension is equal to the total number of word classes.

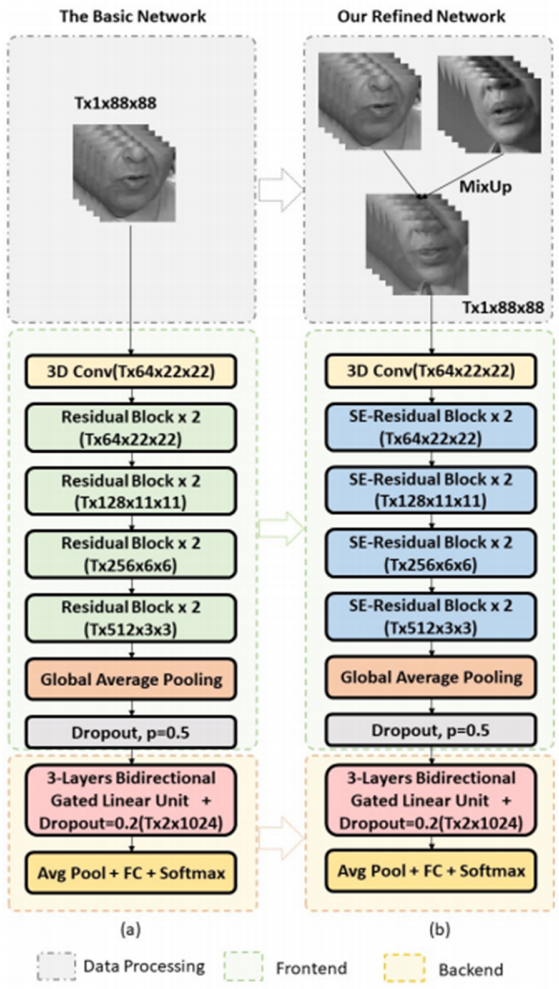

Figure 1. Network architectures from the paper [5]

Besides the baseline model they use a refined architecture as well, in which they replace the basic ResNet with a Squeeze-

and-Excitation based ResNet architecture. Instead of an equal representation of all channels in a given layer, this structure suggests developing a weighted representation. The weights of each channel can be learned in the SE-block. It introduces a hyperparameter, r (ratio) to be used in the SE-block. [6]

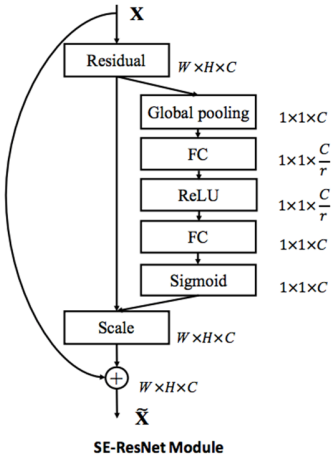

Figure 2. SE- ResNet Module

In this study they compare different backend and frontend networks according to the performance on the LRW and LRW-1000 dataset.

Frontend	Backend	LRW	LRW-1000
VGGM*	-	61.1%	25.7%
ResNet-18*	3 Layers GRU	83.0%	38.2%
ResNet-34*		83.5%	-
ResNet-18		83.7%	46.5%
SE-ResNet-18		84.1%	46.8%

Figure 3. Comparison of the backend modules

Frontend	Backend	LRW	LRW-1000
ResNet-18	3 Layers GRU	83.7%	46.5%
	GRU w/o droppout	83.1%	45.5%
	MS-TCN	83.4%	43.0%
	Transfomer*	76.2%	44.5%

Figure 4. Comparison of the frontend modules

In this paper other helping tricks are employed as well. We don't want to go into the details, so we just list the ideas. For example aligned lips, word boundary input, mixing up frames of different videos of the same word, label smoothing (changing qi in cross-entropy loss) and cosine or exponential learning rate scheduler.

Implication and Utilization of various Lip Reading Techniques [7]

This paper talks about the different types of preprocessing methods of the data, and when and how should be used. When it comes to lip reading, the appearance or the shape that the lip makes while talking is the key to understanding the letters or the words that are being spoke.The paper also shows us the studies that were made in this topic. It also calls the attention to the fact that most likely the environment has a very big impact on the results of the automatic speech recognition, moreover if this environment is noisier than usual.

The one of the paper's authors propose was to use RASTA, which is a type of inter-filtering method that helps in reducing stationary and convolutional noise that maybe present, along with Image Transform Lip Reading Algorithm (ITLR) with the aim of enhance the overall performance of automatic speech recognition systems in general.

The authors also wanted to achieve the integration of interframe filtering with the lip-reading algorithm. In order to reach this goal, they needed to bring together two phases namely PRE-I (pre-integration) and POST-I (postintegration). As the name suggests, in the PRE-I phase the RASTA and inter-frame filtering is performed prior to the image transform whereas in POST-I phase, inter-frame filtering is done after the image transform process.

On their observation of the experimental results, one can easily pass a verdict in favor of the proposed technique. The paper [8], they mentioned, works on active appearance model. This appearance model works on the idea of getting support from cavity features, like appearance of teeth and tongue, when the movement of the lip can't be precisely tracked or even if after doing so it doesn't contribute to a better understanding. In lip reading techniques, lip detection or segmentation is the most basic step and a challenging one too. Most researchers take the help of the red coloration of the lips to segment the lips. But this may not provide them with accurate results in case of poor choice of background lights, red blobs in the speaker's clothing or any other of such unforeseen factors. To counter this problem, the author Hamza Mirza made them the Adaboost algorithm for face and mouth detection. Adaboost classifier cascades Haar like features, hence it is fast and accurate for face detection.

Transformations are often accompanied with a dimension reduction phase like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) requiring a finite number of computations for training as well as testing. PCA focuses on reducing the data size by diminishing the difference between the actual and the recovered data. LDA aims for better segmentation and keeping the discriminating features intact.

The paper gives a good summary of the actual techniques that can be used, and about the type of data it is operated on. In the first column the numbers represent another study, which are equivalent of the following references, we use. [1] -> [10], [2] -> [11], [4] -> [12].

	Feature Extraction	No. of frames selected	Media Used	Language Used	No. of Test Subjects
[1]	Lip folding, Principal component analysis	30	Video	Korean	70
[2]	2D DWT with three decomposition levels, PCA, LDA and LSDA	16	Video	English	7
[4]	Geometric features are normalized, Active Shape	25 or 30	Video	English	8

Figure 5. Example of lip-reading techniques

DATABASE

Several large-scale lip reading datasets have been released in recent years: BBC LRW - lip reading words in the wild or LRS - lip reading sentences, LRW-1000, GRID and MIRACL-VC1. As you can see there are datasets containing words or phrases, sentences. The general solution would be recognizing words, because it can be used in a large-scale of applications maybe with a little finetuning on the new words. But it is difficult to separate the words in speech. That's why it is better to train on phrases and sentences in some specific cases. We wanted to create a general solution even if it is the harder way.

We used the LRW dataset [4] which consists of up to 1000 utterances of 500 different English words. The videos derive from BBC news reports, so the background, the lightning and the speaker's appearance is very diverse. That's why training on this dataset is difficult but the model will have a good generalization ability. In other datasets there are fewer speakers and they use the same background and lightning in every video. (MIRACL-V1) Training on these kinds of video frames can lead to a better performance on test examples that are similar to the training data. So that is the reason why we used the LRW dataset. We expect that if the model fits on these hard examples, it can perform well on easier ones.

About hundreds of different speakers are talking in the videos. All videos are 29 frame length (1.16 seconds), and the word occurs in the middle of the video. The word duration is given in the metadata, from which you can determine the start and end frames. The dataset has train, validation and test part as well. The train set consists of 800-1000 videos per class. In the validation set we can find 50 videos per class. There are 50 examples per class for testing as well.

We have to add that the dataset is only available for non-commercial, academic research. For getting the password to the dataset we requested a data sharing agreement from Rob Cooper, who is responsible for the permissions.

Figure 6. Example frames from the videos in the dataset. [4]

DATA PROCESSING

The size of the LRW dataset is about 80G. This size and the number of the classes (500) are too big for the first experiments so we started with 10 classes with 100 training videos per class. After that we exported 100 classes. So the first step is to create a stable model that achieves a good accuracy on the validation dataset at least. Then we can recognize more words by finetuning the previous model with more classes.

The word is spelled in the middle of every video. The word duration is given, so we can determine the start and end frames. Each video frame is a colour image (3 channels) 256x256 pixels in size. For lip reading the mouth is the most important part, it contains the information we need for the training. The following tasks are executed on every video frame to extract the mouth region:

1. detect face using dlib face detector
2. detect the points of the mouth with shape predictor according to the facial landmarks
3. crop the mouth with the bounding box around the found points
4. make square bounding box and resize it to 120x120 images

The training data is so big that it would be memory wasting to store the whole set in the memory. The best way is to create a specific data generator which reads only a batch of video frames in every step. The input shape of the neural network structure is [batch size, frame number, width, height, channel]. In lip reading the color information is not a decisive factor, so we use grayscale images during training. The datagenerator converts the color images to grayscale and resizes them to the given size. We have to use a fixed frame number (20 video frame) for the processing so the frames of the short videos are padded with zero in the end. Moreover we can apply data augmentation methods with the given ImageDataGenerator instance. During training our data generator executes randomly the given transformations on the video frames. After every epoch it shuffles the training data as well.

Figure 7. The detection of the mouth region. [4]

NETWORK EXAMPLES ON THE DATASET

In the documentation of the dataset [4] there are some suggestions and experiments considering the network structure:

Early Fusion (EF): The input of the network is a T-channel image, where each of the channels is an individual grayscale frame. The subsequent layers are identical to the layers of the regular VGG-M network.

3D Convolution with Early Fusion (EF-3): This architecture is similar to the EF, but it uses 3D Convolution. It takes [Height x Width x Time x 3] shape input.

Multiple Towers (MT): There are T=25 (frame number) "towers" with convolutional layers which have shared weights. The activations are concatenated channel-wise creating an activation with 1200 channels. After that an 1D convolution reduces the channel number. The rest of the network is the same as the VGG-M.

3D Convolution with Multiple Towers (MT-3): The basic design principles of the architecture are the same as in the EF-3. But the frontend of the network consists of the "tower" layers mentioned in case of MT model. (That's why there is no explicit time-connectivity between frames at the frontend.) So after the concatenation the activation layer goes to the 3D convolutions. This is the results of the different models:

Net	500-class			333-class	
	Top-1	Top-10	ED	Top-1	Top-10
EF-3	43.9%	81.0%	3.13	55.7%	87.9%
MT-3	46.2%	82.4%	2.97	56.8%	88.7%
EF	57.0%	88.8%	2.32	63.2%	91.8%
MT	**61.1%**	**90.4%**	**2.06**	**65.4%**	**92.3%**

Figure 8. Accuracy results of the models suggested in the paper of the dataset. [4]

As you can see in the table the best model reaches only 61,1% top-1 accuracy on the dataset with 500 classes and 65,4% with 333 classes. This means that it is a very difficult task to train a good lip reading model.

During our experiments we considered using VGG network but the big amount of parameters can easily lead to overfitting, so we decided to use ResNet architecture instead, because the residual block used in the network helps the convergence. Furthermore the ResNet has fewer parameters than the VGG-M, so it is less susceptible to overfit on the training dataset.

NETWORK

Modern deep lip reading models usually consist of two modules: a frontend module and a backend module. The frontend's task is to find local motion patterns, including clip-level and frame-level features, whereas the backend module's part is to discover sequence-level patterns and to learn the temporal dynamics of the sequence based on the frontend module's output features.

Although the architectures of most models can be divided into these two parts, we never found a consensus on which strategies could bring effective learning of the lip reading model. Different works always have their own strategies to obtain effective lip reading. We constructed our network similarly to these state of the art solutions.

A standard ResNet18 network is used as the frontend, except that the first convolution is a 3D convolution with 5 × 7 × 7 kernel size, as proposed in [2]. These first set of layers applies spatiotemporal convolution to the preprocessed frames, as spatiotemporal convolutional layers are capable of capturing the short-term dynamics of the mouth region, [1]. They consist of a convolutional layer with 32 3-dimensional (3D) kernels of 5×7×7 size (time/width/height), followed by Batch Normalization (BN, [3]) and Rectified Linear Units (ReLU). The extracted feature maps are passed through a spatiotemporal maxpooling layer, which compacts the features in the spatial domain.

The backend of the network is a 2-layer Bidirectional GRU (Gated Recurrent Unit) network followed by a dense softmax layer. A Bi-GRU contains two independent single directional GRUs. The input sequence is fed into one GRU in the normal order, and into another GRU in the reverse order. The outputs of the two GRUs are concatenated together at each time step to represent the whole sequence. The output of the Bi-GRU will be finally sent to a linear layer for classification.

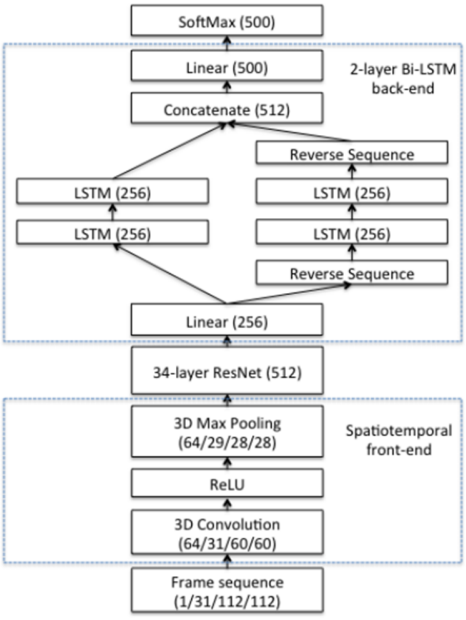

Figure 9. Block diagram similar for the proposed network[2]

TRAINING

Developing a general model for solving a problem is never an easy challenge. It is a good start to experiment with the small part of the classes above all if we have so many classes like in our case (500).

First we exported 10 classes from the database with 100 training, 10 validation and 5 test videos per class. After some manual and automated hyperparameter optimization we achieved 60% validation accuracy. We used Adam optimizer with 0,0008 starting learning rate. The activation used in the layers was ELU (Elastic ReLu). The filter number of the 3D convolution was 32. The dimension of the GRU layers was 20. We used l2 regularization on the kernels of the 3D convolution layer and on the parameters of the FC layer after the ResNet. We used dropout with 20% probability after the GRU layers block and the ResNet. The batch size was 32 in this case. The training stopped after 71 epoch.

The Figure 5. represents two trainings with 10 classes. It is obvious that both trainings overfit on the dataset, because the validation accuracy is much lower than the train accuracy. So the model doesn't perform well on unknown data. The experiment called 'resnet18_backengru_adam_elu' used Adam optimizer and ELU activation and set the dropout probability to 11,5%. In the other case the difference is that we used l2 regularizer, 20% dropout probability and data augmentation (random horizontal flip, height shift) as well.

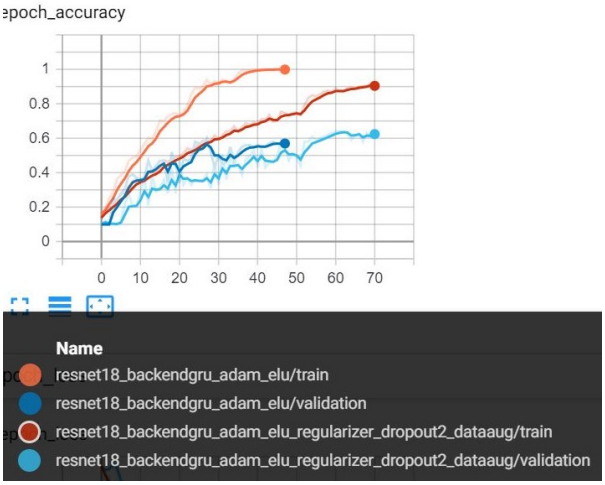

Figure 10. Effects of using methods against overfitting or not

You can see that in the second experiment the difference between the train and validation accuracy is smaller than in the first case. We decreased the impact of the overfitting, but it is still not the best.

On the test data (50 video) the second model achieves 58%. On Figure 6. you can see that the confusion matrix is close to a diagonal matrix, so that means that we go on the right way.

Figure 11. Confusion matrix for 10 classes.

We also have experiments on training a network for recognizing 100 different words according to the motions of the lips on the video frames. So far we haven't achieved a good performance with 100 classes, because it requires more time. We applied a bigger model for the task as well. Instead of ResNet18 we used ResNet34. If we could train a good model for 100 classes, we could use it when we want to add other words to recognize, we only need to finetune the frontend and backend module and retrain the classifier head for the models. All in all it is advisable to train the model in smaller steps for more classes, because end-to-end training is slow and it won't give the best results.

HYPERPARAMETERS AND OPTIMIZATION

We have several hyperparameters like the input dimensions, the activation function or the dropout. The best way to check the variety of them is to use the Hyperas module. It allows us to evaluate different hyperparameter values, and to find the best settings for our model. The parameters we used for this purpose were: the dropout size, the type of the activation function, the type of the optimizer, and the size of the hidden and the input dimension. The dropout value could change between 0 and 0.4. For activation function, we were supposed to use among the following possibilities: ReLU, ELU and Tanh. The optimizer could be RMSprop, Adam or SGD. The hidden dimension and the input dimension values were discrete numbers. In case of the hidden dimension those were, 20, 50 and 80. The input dimension could be 32, 64 or 128.

The best result we got, with this hyper parameter optimization was, when we used Adam optimizer, the activation function was ELU, the dropout was set to 20 % with 20 hidden dimensions and the input size was 32. As a parameter we set the number of epochs to 40 but usually, the optimizer didn't reach this number. The maximum evaluation number was set to 100.

VISUALIZING THE RESULTS OF THE OPTIMIZER

We also made some graphs so we could evaluate the data which was saved into a log file. The CSV file contained the above mentioned hyperparameters and also the validation accuracy.

First we listed out the 10 best models, in this case the batch size and the number of hidden dimensions were locked, as we experienced, that with these numbers we got the best results so far.

	dropout	activation	optimizer	hiddenDim	inputDim	n_batch	acc
33	0.231863	elu	adam	20	128	32	0.51
10	0.073973	tanh	adam	20	128	32	0.49
28	0.123371	elu	adam	20	32	32	0.48
11	0.122027	elu	adam	20	32	32	0.47
9	0.048361	relu	adam	20	32	32	0.44
21	0.130838	tanh	adam	20	32	32	0.43
34	0.239914	tanh	sgd	20	128	32	0.39
19	0.004680	tanh	adam	20	128	32	0.39
38	0.389552	tanh	adam	20	128	32	0.33
32	0.317938	elu	adam	20	64	32	0.33

Figure 12. The detailed description of the 10 best results.

As we could see, the best optimizer for our purpose is the Adam optimizer, and the best value for input dimension is 128 or 32.

We also evaluated the relations between the validation accuracy and the input dimension and the activation function. In the above figure we can see that the most effective result was given when the hyperas set the activation function to ELU.

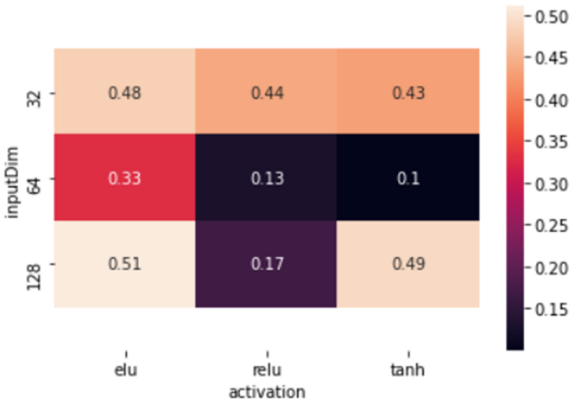

Figure 13. Relation between the input dimensions and the activation functions.

We made the same evaluation with the optimizer function. The result was the following:

Figure 14. Relation between the number of input dimensions and the optimizer type

Here we could see that the best results were made with the Adam optimizer, no matter the number of the input dimensions.

RESULTS

All in all we achieved adequate accuracy on recognizing 10 different words with our network. In this case we struggled with overfitting because of the high amount of the parameters, but applied different methods to solve this problem. We used dropout layers, L2 regularization, different data augmentation methods, decreased the parameters of the network as much as we could and we used batch normalization as well.

In the further experiments we want to focus on creating a network, which recognizes at least 100 different words, but it is a complicated task considering the difficulties in lip reading.

CONCLUSION

Summing up our experiences we had to admit this visual speech recognition task is not as trivial to do as we first imagined. It is very important to make a good model that can determine the spelled word in the whole sequence of the video frames.

We could also use the audio as a helping hand and make some predictions, so we suppose that the model could detect a word easier by its spectrum. Furthermore, it is another difficult task to connect the words detected with lip reading solutions. In this case we can use a language model from NLP to bound the detected words with the right conjunction word to create a grammarly proper sentence, because we can't perceive the short words like 'a', 'an' 'the', 'to', 'as'.

Considering everything, the field of lip reading has a lots of challenging problems, that's why it is difficult to create a state of the art solution. We have to apply tricks and examine the root cause of the problems.

REFERENCES

[1] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "Lipnet: Sentence-level lipreading," arXiv preprint arXiv:1611.01599, 2016. https://arxiv.org/pdf/1611.01599.pdf

[2] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in Interspeech, 2017. https://arxiv.org/pdf/1703.04105v4.pdf

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015, pp. 448–456. https://arxiv.org/pdf/1502.03167.pdf

[4] Joon Son Chung and Andrew Zisserman "Lip Reading in the Wild" https://www.robots.ox.ac.uk/~vgg/publications/2016/Chung16/chung16.pdf

[5] Dalu Feng, Shuang Yang , Shiguang Shan , Xilin Chen - LEARN AN EFFECTIVE LIP READING MODEL WITHOUT PAINS - https://arxiv.org/pdf/2011.07557.pdf

[6] Jie Hu, Li Shen, Gang Sun - Squeeze and Excitation Nteworks - https://arxiv.org/pdf/1709.01507v2.pdf

[7] Saiqa Khan, Hamza Azmi, Ajay Nair, Hamza Mirza – Implication and Utilization of various Lip Reading Techniques - https://www.researchgate.net/publication/317608327_Implication_and_Utilization_of_various_Lip_Reading_Techniques

[8] Lip Reading Using DWT and LSDA Sunil Sudam Morade and Suprava Patnaik Professor - Advance Computing Conference (IACC), 2014 IEEE International 27 March 2014.

[9] S. Zhao, R. Kricke, and R.-R. Grigat, "Tunir: A multimodal database for person authentication under near infrared illumination," in Proceedings of 6th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA 2007), Corfu, Greece, February 2007.

[10] Jinyoung Kim, Suengho Choi, Seongmo Park "Performance Analysis of Automatic Lip Reading Based on Inter-Frame Filtering".

[11] Lip Reading Using DWT and LSDA Sunil Sudam Morade and Suprava Patnaik Professor, Advance Computing Conference (IACC), 2014 IEEE International 27 March 2014.

[12] Automatic Lipreading With Limited Training Data , The 18th International Conference on Pattern Recognition (ICPR'06), S.L.Wang, W.H.Lau, A. W. C. Liew and S.H.Leun.