Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

РОЗРОБКА АЛГОРИТМУ АВТОМАТИЧНОЇ СИНХРОНІЗАЦІЇ ГУБ ТА РИС ОБЛИЧЧЯ У ВІДЕОПОТОЦІ З АУДІО

Текстова частина до курсової роботи
за спеціальністю „Комп'ютерні науки" 122

Керівник курсової роботи
с.в. _Бучко О.А.__
(прізвище та ініціали)
____
(підпис)
"____" __________ 2021 р.

Виконав студент _________
___Андронік В.П._________
(прізвище та ініціали)
"____" __________ 2021 р.

Київ 2021

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Андроніку В.П. факультету інформатики четвертого курсу
ТЕМА РОЗРОБКА АЛГОРИТМУ АВТОМАТИЧНОЇ СИНХРОНІЗАЦІЇ ГУБ ТА РИС ОБЛИЧЧЯ У ВІДЕОПОТОЦІ З АУДІО

Вихідні дані:
-
-
Зміст ТЧ до курсової роботи:

Індивідуальне завдання
Вступ
1. Аналіз задачі
2. Огляд існуючих робіт
3. Опис розробленого алгоритму. Практичне застосування.
4. Експерименти та аналіз результатів

Висновки
Список літератури
Додатки (за необхідністю)

Дата видачі „___” _________ 2021 р. Керівник Бучко О.А.
(підпис)

Завдання отримав __________
(підпис)

Тема: Розробка алгоритму автоматичної синхронізації губ та рис обличчя у відеопотоці з аудіо

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	01.11.2020	
2.	Аналіз технічних матеріалів за темою.	01.01.2021	
3.	Розробка та програмування алгоритму.	01.02.2021	
4.	Виконання порівняльного аналізу представлених гіпотез та існуючих методів з синхронізації губ за допомогою нейронних мереж.	01.03.2021	
5.	Написання пояснювальної роботи.	15.03.2021	
6.	Коригування виконаної роботи.	20.03.2021	
7.	Написання курсової роботи та відповідної презентації для доповіді.	05.04.2021	
8.	Коригування курсової роботи.	10.04.2021	
8.	Остаточне оформлення роботи та слайдів.	12.04.2021	
9.	Захист курсової роботи.	20.04.2021	

Студент Андронік В.П.
Керівник Бучко О.А.
“______” ________________

Table of contents

Abstract

This material presents the solution to generate talking face images with the use of deep learning. We conduct the research of existing literature to compose more efficient network design. The final version has additional pre-trained discriminator network to reach superior lip synchronization performance with adversarial training to improve the visual quality of images.

We provide comparative analysis and ablation studies which show insights on how different components of the solution affect the result. This approach achieves comparable consistency in lip movements to other solutions in the field, but has higher visual quality.

Keywords: deep learning, face animation, lip synchronization, generative adversarial networks.

Introduction

There is a large amount of audio-visual content in the Internet with more than 500 hours of videos uploaded each minute [1]. And this naturally raises a problem of availability of this content to multilingual community. At the moment of writing this, nearly 60% of all the information is published in English[2] which creates an imbalance of information accessibility, especially for educational purposes.

That is why translation of all that content may make it available to millions of people across the world. The problem of audio-visual translation consists of two challenges: translation and lip sync. The former one is pretty straightforward and needs an editor or a machine to translate a piece of text to other language. Moreover, this problem became much easier with the current state of machine translation and services like Google Translate.

However, recent studies [3, 4] have stated that subtitles decrease the feeling of spatial presence and make content less sensible for non-native speakers, specifically beginner readers. By correcting face video to match the desired target speech one can make the video more understandable and perceptually feasible. Accomplishing this result is a hard task and it has received respectable amount of attention in the research community [5, 6, 7, 8, 9, 10, 11, 12].

Most of the solutions are using deep learning and are constrained to limited amount of speakers and words. For instance, prior works in the field [12, 13] are using several hours of Obama videos and learning the mappings between audio and lip landmarks. Next works are also constrained to the speakers they are trained on, but provide better quality results [7]. The solutions discussed upon have limitation to the speakers they are trained on and also an overhead to collect several hours of clean speech.

That is why next generation of research was aimed on providing the solution to work with unconstrained speakers with minimal amount of audio provided for each of them. First solutions have resulted in speaker-independent lip sync [8, 9] with good visual quality. Still, they had a drawback working only on static images. However, to be used in real world with unconstrained amount of videos taken from different perspectives and with different qualities one should have a model working not only on static images but videos.

The recent work in the field accomplishes just that with high-quality performance in-the-wild [5]. Their major contribution is in showing that the essential part of the task can be solved with the use of a strong discriminator – binary classifier which predicts if the lips movements on video are synced with the target speech. Their classifier reaches 91% accuracy at finding out-of-sync sequences using their evaluation dataset which is significantly higher that those from previous works.

In our work we are building upon this latter approach to construct the lip-sync system which will be working on arbitrary identities and voices. This is a challenging task due to versatile environment with many different poses and voices to support. Moreover, we have a little room for error given that humans are capable of detecting an out-of-sync video fragment ~0.05 – 0.1 seconds[12] in duration.

In our approach we will use the findings from the prior research with an emphasis on learning from a strong discriminator. We construct disentangled architecture for entities separation with concentration on more efficient training and more consistent performance on video.

The work consists of four sections.

The first one discusses previous approaches for speaker-independent lip-sync task. Our key findings are that prior works perform good at lip-syncing static photo with an arbitrary speech, however the quality drops when used on unconstrained videos. We hypothesize that the main reason for that is mainly because of using reconstruction losses and learning from weak discriminators which are not accurate enough to penalize for out-of-sync videos.

Next, in section two, we go into relevant concepts that are highly coupled with our work, specifically for the next more technical section.

Third part presents our method in details with different discussions on the specifics of the problem and training.

The fourth section shows experimental results and compares the presented solution with the related ones. Moreover, the ablation studies with analysis on the results are also provided.

Overall, the main tasks are:

1. Analyze the existing solutions.

2. Construct the speaker-independent solution for lip synchronization task using deep learning.

3. Analyze the results and compare with similar solutions in the field.

Main part

Section 1. Problem analysis and solutions overview

In this section, we are going to provide brief overview of the problem we are solving and describe the prior research in the field.

With the rise of consumption of audio-visual information the questions of its accessibility rises more often. And while the videos are mostly translated to different languages and the subtitles are made afterwards, the information may not be clear enough for the beginner readers. That is why making a lip-sync model to transform the face on the video to match the target speech is very helpful, especially for educational purposes.

Lip-sync is about correcting one’s lips movements on video to match the audio of target speech. Figure 1.1 illustrates the task in a more explicit way, so having an image and arbitrary speech audio we can generate multiple new images that reproduce the lips movements from the audio.

This is a challenging task because the model should work with different speakers, be applicable to multiple languages and work in high-dimensional image space while generating generating consistent result on videos. The task becomes more complicated because model also should work with in-the-wild captured videos (i.e. videos with unconstrained speaker poses, different quality of images, noisy speech etc).

The vast majority of early works were concentrated on solving lip-sync in constrained environment, mostly on single speaker or with the use of limit-sized vocabulary. For instance, one of the approaches [13] is using Obama videos to generate high-quality talking face videos. However, they are only working on Obama identity and do not generalize to unseen videos and voices. Moreover, this approach has an overhead in generating several hours of data for single speaker. Which is not a hard task for celebrity but may be quite expensive for other individuals. Next line of research aimed to reduce the data consumption for training. One of the recent works [14] is using a hybrid approach by training two models – one is speaker-independent and is trained on in-the-wild videos, while the other one is trained for each speaker in

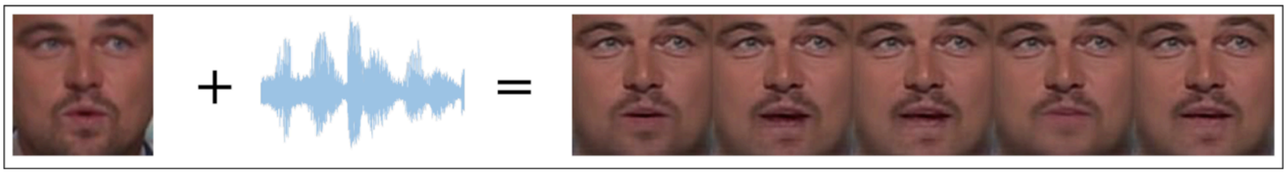

Figure 1.1 Lip sync task illustration from [17]

particular, but it demands significantly less data than previous approaches. Technically, the first stage is used for pre-training the model to learn and extract useful generic features, while the other one is fine-tuned to get better results for each speaker. Still, they have an overhead with collecting the additional data for each speaker and also the model was trained on small dataset which constrains the performance of the model.

Additionally, there is a limitation that comes from existing datasets which may be not big enough for the model to successfully generalize. For instance, LRW [15] and TIMIT [16] have only 1000 words vocabulary which may be limiting in the real world scenario.

That is why we aim to construct the speaker-independent solution which will learn from phoneme-rich dataset and generalize well to unseen speech and videos.

The solutions described above have one problem in common — they do not generalize to unseen data being limited to single speaker. So the next works aim to solve this more challenging problem.

One of the early works at this frontier [17] constructed a pipeline with separate encoder for image and audio inputs with decoder taking extracted features to generate the final image. That is a good architecture choice however the results are blurry and do not perform well on hard mimic cases. They even trained deblurring network on top to get sharper results. Now, adversarial training is a standard for getting better image quality and we will use this in our work as well. One more thing to mention about this approach is that they did not use any external supervision by learning only from pixel wise losses which are not strong enough to penalize for out-of-sync in hard cases and tend to result in blurry results as well.

The next solution [18] got another way around by firstly learn the intermediate task of audio-visual speech recognition to learn the mapping between audio and visual representations. Then, they use learned representations to disentangle subject and speech related information inside. This approach lacks practicality as soon as one should collect separate dataset to train on each new language.

Next work [8] is actually the continuation of the first one we mentioned [17]. The approach stays the same, their main contributions were in data (700K training samples), speeding up architecture design and working on quality by adding multi-stream CNN for blending generated faces back in the frame. Still, they have a problem with the lip-sync quality, but the results are visually more appealing.

One of the recent works[9] goes along with encoder-decoder architecture, however with several key differences, they separate images into identity and reference where the former have their mouth region masked while the references are used to extract the visual features of the identity. Generated image then is compared with the identity one. This allows to extract relevant features about identity from reference image while the identity one is mostly used for pose identification. In our experiments such an approach helps to overcome the information leak and improve convergence speed. Another more important contribution is the introduction of lip-sync expert — binary classifier which outputs the probability of the audio-visual sequence being in-sync or not. This allows to get more sophisticated lip-sync results and we incorporate it in our approach. The drawback is that they train discriminator end-to-end which may give faulty gradient by overfitting on some visual artifacts present during training, thus such discriminator is a weak supervisor.

With all that in mind, one of the recent works[5] was published. They continued with the architectures from [8, 9] and concentrated mostly on training more accurate discriminator network with ~1.5 times improvement in accuracy compared to previous works. They reached 91% accuracy at detecting out-of-sync samples. They stated in their work that such expert is all you need to get a quality lip-sync. We got the same results in our experiments and can verify that accurate expert network makes huge different in terms of how natural the result is.

In our work we proceed with these findings and aim to get more efficient architecture with disentangled representations for identity and reference images, also working on skip-connections between encoder and decoder networks. We also change the way reference images are sampled and make research on information leak during training.

Section 2. Related work

2.1. Convolutional neural networks

In this work we extensively use convolutional neural networks and it should be useful to go into details and describe how they work together with explanation of common architectures and optimization techniques that will be used next.

The main component is a classic discrete convolution from computer graphics which is illustrated in Figure 2.1. Each layer of the networks consists of such kernels which are applied to the input as a sliding window.

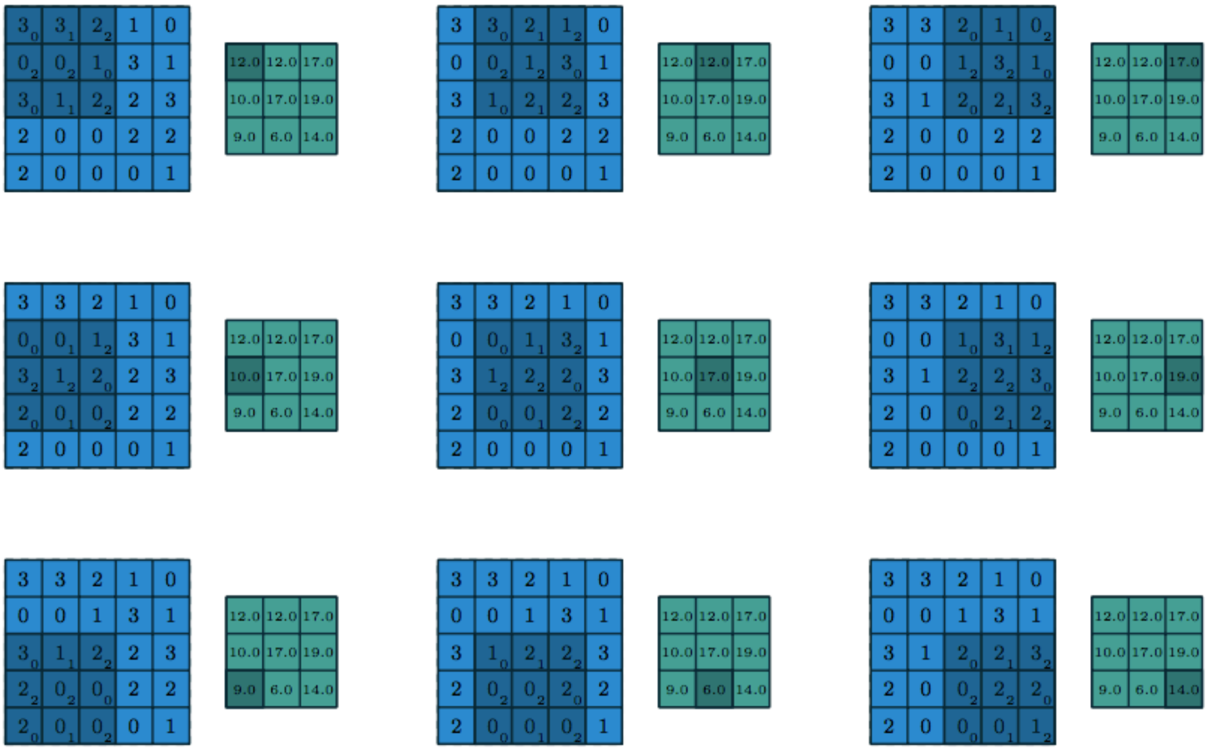

Figure 2.1. The discrete convolution overview. Blue matrix is an input, green matrix is an output. Kernel is applied in sliding window style. Picture from [46].

Deep convolutional networks are the essential component of AI revolution we now have, they are widely used for different task, specifically from computer vision task. This type of networks exists for a long time however their full potential has been fulfilled only recently in the 2012 work [47].

From there on, the extensive amount of research has been conducted and new network types were constructed.

Residual connections. Training very deep neural network from scratch is a hard task and it was very challenging at the early years of deep learning development. The main problem is vanishing gradients due to chain rule used during backward pass. Very important work[30] introduced ResNet architecture which employed residual connections as the building block. The main idea was very simple but yet elegant, they added the output of layer on step i with the output of the previous layer $i-1$. This solved the problem of vanishing gradients and allowed to train 1000 layer deep networks without any additional problems. For comparison, the deepest working architecture for those times was VGG-16[32] with 16 layers. Residual connections has been improved with newer versions proposed[48], you can refer to Figure 2.2 for comparison. However, the main idea keeps untouched and in our work we are using

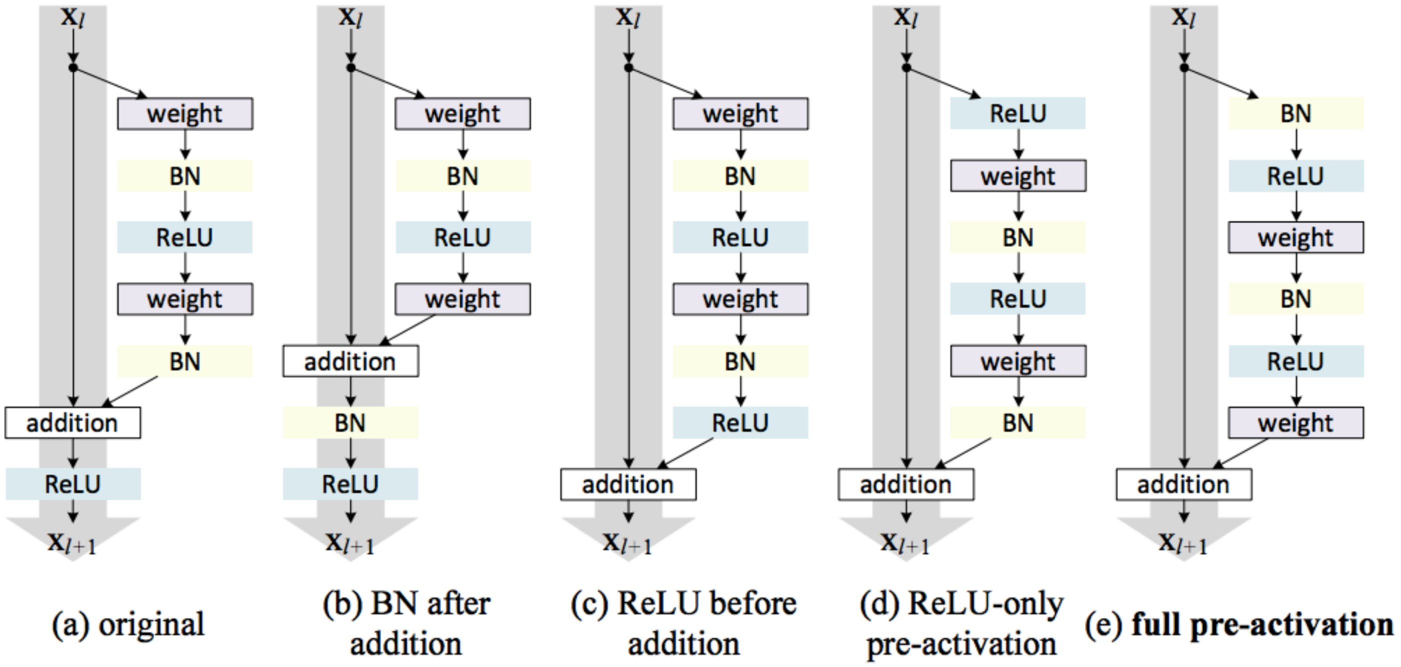

Figure 2.3. ResNets overview from [48]. These are different variation of origin residual connections from (a).

the originally proposed version. In our experiments, adding residual shortcuts improves quality and speedups convergence.

UNet [29]. Skip connections concept and UNet architecture are important for understanding the next sections. UNet proposed fully convolutional architecture for segmentation. It has a structure of encoder-decoder architecture so the output dimensions match the input. Encoder reduces dimensions of the input while the decoder is increasing the resolution. Main idea behind the UNet was to add the skip-connections between the corresponding layers of encoder and decoder. Those intermediate features are concatenated to the input of each layer of the decoder. This effectively helps the decoder to use the information from different levels and receptive fields when generating the final image. For more details refer to Figure 2.4.

In our experiments we use the same ideas with multiple encoders for image and audio data which skip-connect their features to the corresponding layer of the

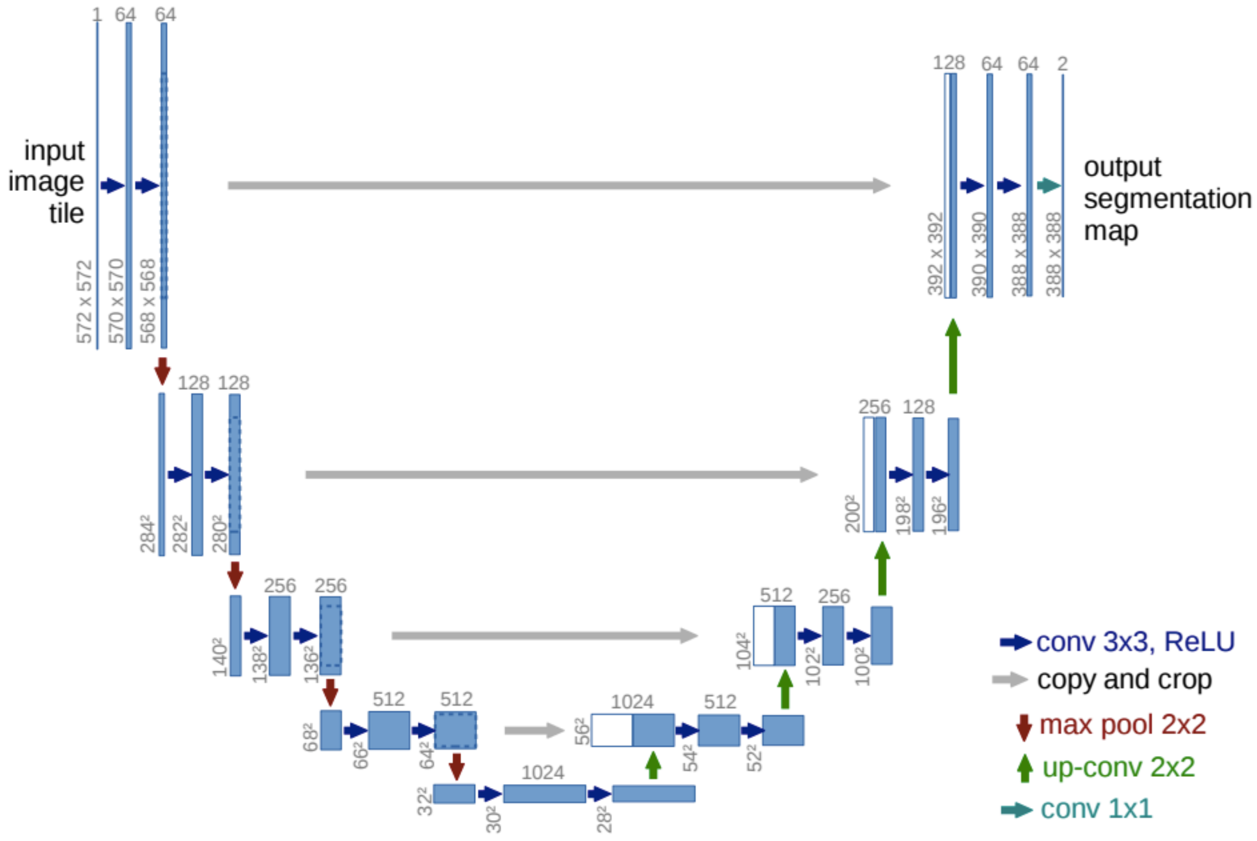

Figure 2.4. UNet architecture overview. Skip-connections are gray arrows which transfers the output from encoder to decoder input[29].

decoder. This provides good context awareness. For downsampling strided convolutions are used while bilinear upsampling + convolutional layer is used for upsampling.

Section 3. Method

Our lip-sync architecture is inspired by the early Lip-GAN[9] network design with changes made to improve visual and lip-sync quality. We also incorporate highly accurate discriminator network to learn from during training.

3.1. Data preparation

We overviewed different datasets to use during training, see Table 3.1 for more details. There has been a considerable rise in the number of large and quality datasets recently. For instance, AVSpeech[26] dataset from Google collected 4700 hours of audio-visual data from YouTube with good quality. However, maintaining dataset this large is troublesome and considering our work as a research-oriented we switched to smaller but versatile LRS datasets[19, 22, 25]. Specifically, LRS2 dataset with ~30 hours audio-visual content and ~18k vocabulary size is chosen. The whole dataset is constructed of utterances 2-3 seconds long.

After collecting the dataset we proceed with preparing it to training. We use face detection network[27] with available public implementation to crop the frames in the videos to the size of 96x96. We also extract landmarks from each face crop

Dataset	Year	# hours
GRID[20]	2006	43
LDC[21]	2009	8.3
LRW[22]	2016	167
LRS-2[19]	2016	180
LRS-3[25]	2018	176
VoxCeleb[23]	2017	352
VoxCeleb2[24]	2018	2442
AVSpeech[26]	2020	4700

Table 3.1. Lip-sync datasets overview

with the use of TDDFA[28] network which is state-of-the-art solution at the moment of writing.

3.2. Problem formulation

Let a sequence of face image crops be $X = \{X_1, ..., X_T\}$ with aligned to it waveform $A = \{A_1, ..., A_T\}$. The task of lip-sync is to train a model g to get the sequence of in-sync frames:

$$\hat{X} = g(X, A) \quad (1)$$

The result should have the lip movements consistent with those from A while the visual information should pass untouched relatively to X (i.e. light, pose, skin tone, identity attributes etc).

However, we cannot train g in a completely supervised manner because we only have synchronized samples. That is why our model could be called the reconstruction one, because instead of getting a negative sequence we mask out the mouth region of X. For that we use a constant rectangular mask M which is large enough to cover the mouth/chin region together with background information, the urge for this will be explained in the next section. We will assume X to be masked with M later in the paper.

But we lose very important identity information about the speaker by applying mask. Information about the lip color/shape, teeth etc is removed. To overcome this issue we use the set of additional frames from the same video of this speaker. We call them reference frames: $R = \{R_1, ..., R_N\}$. For X it is important to get a sequence of frames while references should be the most representative frames from the video. We go into details about picking those in the later sections. So, our function g now transforms into:

$$\hat{X} = g(X, A, R) \quad (2)$$

During inference we got (X_{test}, A_{test}) that can be arbitrary and out-of-sync. As a reference frame we can take the original one X_{test} given that we do not use the model with the original audio.

3.3 Lip-sync expert

Before moving on to discussing the model architecture and training techniques, the word on lip-sync expert is worth to mention. This is a key feature to reach state-of-the-art results for this task.

Previous works[8,9,17] use pixel wise losses which are not efficient for penalizing out-of-sync results. This is because they compare whole images and the mouth occupies the negligible amount of area. The network needs to reconstruct background and skin texture well enough, so that significant amount of signal starts go backwards from mouth region.

That is why for quality lip-sync we need more specific losses and one of the recent works[9] added discriminator to their pipeline to penalize sequences for not being synchronized. Still, this network was overfitting on visual artifacts of generator because it was trained in adversarial setup: by learning discriminator and generator networks at once.

The follow-up work[5] argued that for good lip-sync, discriminator network must be pre-trained without additional fine-tuning during training. This creates complementary overhead because one needs to train expert for each new dataset. However, this additional complexity is worth it.

We rely on the research from [5] and add pre-trained discriminator network into the pipeline. Next we will describe in more details the training scheme that was used for training.

The network receives the sequence of frames X_s of shape $(B, 3 * T_v, H/2, W/2)$ concatenated by channel axis (only lower parts). Together with the corresponding audio sequence A_s for those $T_v = 5$ frames. To train the network one can sample different audio segments from other parts of the same audio for modeling out-of-sync fragment. The architecture is comprised of face and audio encoders both of which output the embeddings which are compared with cosine similarity binary cross-entropy loss. The intuition is that we can receive non-negative embeddings by applying ReLU activation function upon it. Then we can compute dot product between normalized versions of those vectors to get cosine similarity in the range

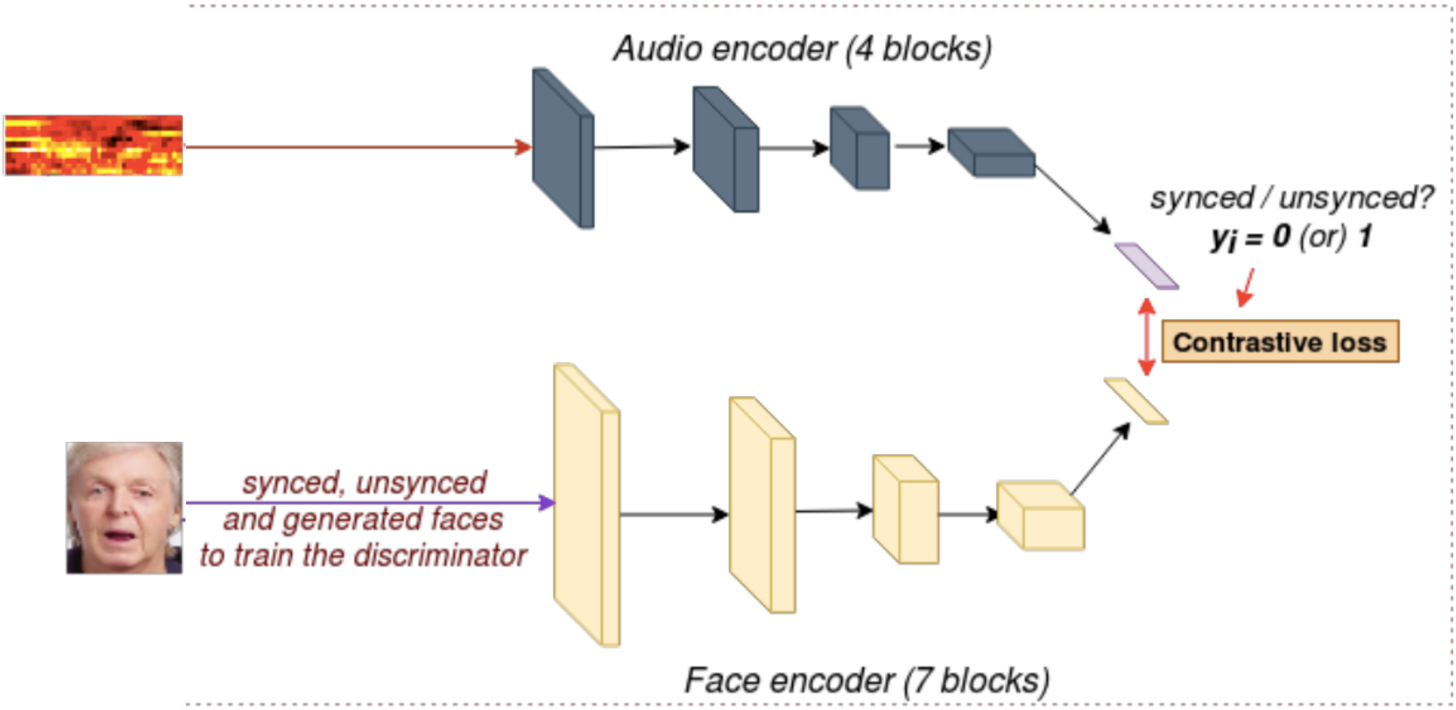

Figure 3.1. Lip-sync expert architecture overview[9]. Separate encoders for audio and images. Audio is encoded using MFCC features.

[0,1] which we can use to compute binary cross-entropy loss which is predominantly used for classification. Given v, a as video and audio embeddings correspondingly the distance(or probability in this case) between them is computed as:

$$d = \frac{av}{max(||a||_2||v||_2, \epsilon)} \quad (3)$$

And the target objective is:

$$L_{expert} = -\frac{1}{N}\sum y_i log(d_i) + (1 - y_i)log(1 - d_i) \quad (4)$$

where y_i is in-sync or out-sync audio fragment.

The network is trained on train portion of LRS2[19] dataset (appr. 30 hours) and reach 91% accuracy at detecting correctly synchronized sequences.

3.4. Model architecture

We construct encoder-decoder lip-sync architecture, see Figure 3.2 for more details. The essential architecture is U-Net[29] with residual connections which takes as input audio signal, masked identity and reference images. We have separate encoders for each input type, each of which outputs a feature embedding of shape 512. These are concatenated and passed to the decoder network, which mimics the structure of image encoders with bilinear upsampling + 2d convolution for upsampling the features. Multiple skip-connections are used to concatenate the intermediate features from reference and identity encoders to corresponding ones in decoder. In our experiments we noticed that skip-connections are important components for the network performance. Thus, we also add skip-connections of concatenated embeddings from each encoder, for this we use auxiliary blocks which transforms the dimensions of vectors to match those of intermediate features. Also, residual connections[30] are used for each layer in encoder and decoder which improve quality.

We will discuss in more details each component next.

Identity encoder. Stack of 2d convolutions with 10 ResNet Conv-BN-ReLU blocks and the downsampling implemented with strided convolutions. The network encodes input of shape 96x96 and outputs the embedding of size 1x512 with intermediate features cached for skip-connection to decoder.

Reference encoder. It has the same architecture as identity encoder. For each identity frame we have multiple references to encode the facial attributes of the speaker. Each of those frames is encoded independently by the encoder and we got the set of embeddings $(N,512)$. We aggregate these embeddings into one using the attention with audio embedding as the key. Main idea is to choose only those reference features which are relevant for current frame given audio which should encode the lips shape and chin position. The intermediate features are aggregated as well to pass to decoder network. Let $r = \{r_1, \ldots, r_N\}$ be a set of reference embeddings and a — audio embedding, then aggregated embedding $\hat{r}$ is computed as:

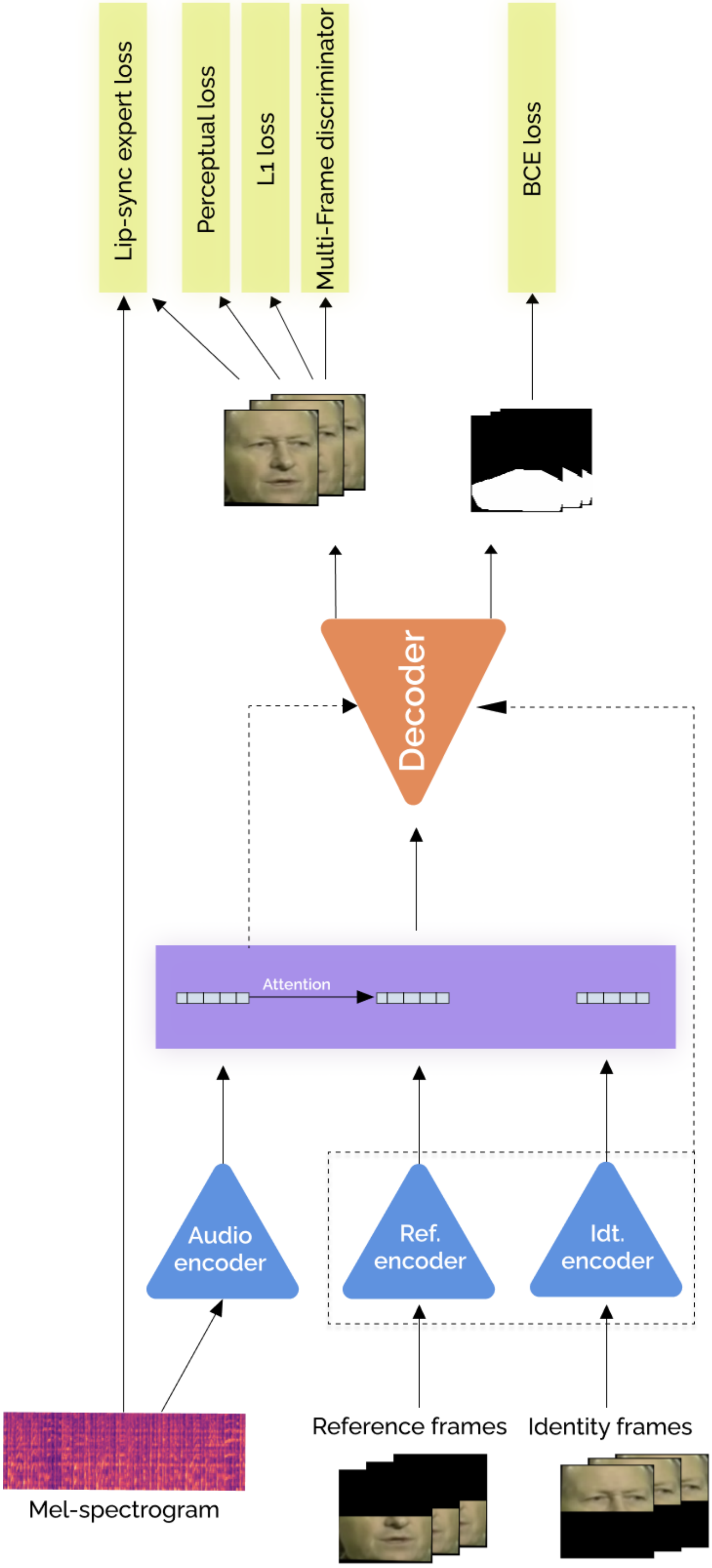
Lip-sync expert loss
Perceptual loss
L1 loss
Multi-Frame discriminator
BCE loss
Decoder
Attention
Audio encoder
Ref. encoder
Idt. encoder
Mel-spectrogram
Reference frames
Identity frames

Figure 3.2. Lip synchronization architecture overview. Using U-Net structure with residual connections. We employ separate encoders for each input modality: audio, reference and identity frames. Each encoder outputs embedding which are concatenated. Attention is used to aggregate reference frames with audio embedding. Decoder network receives multiple skip-connections from encoders and mel embedding. Model outputs the result and mask for blending. We use pre-trained lip-sync expert, as well as pixel wise losses. Adversarial training is also used for improving quality.

$$w_k = \frac{exp(-\sigma(ar_k))}{\sum_i exp(-\sigma(ar_i))}, \hat{r} = \sum_i w_i r_i. \quad (5).$$

Audio Encoder. We use mel-spectrogram with 80 filter banks to represent audio signal. Network is composed from stack of 2D convolutions for audio encoder with 7 ResNet Conv-BN-ReLU blocks and the downsampling implemented with strided convolutions. We also concatenate the input with two channels of Cartesian coordinates for better spatial adaptivity for convolution network.

Image Decoder. The network mimics the image encoder architectures with bilinear upsampling + 2d convolutions used for upsampling. Decoder concatenates its features with intermediate ones from each encoder. Also, the additional blocks are used to transform mel embedding to the size of the current feature map. We do not use skip-connections for final layers because model learns to copy reference features which degrades the convergence. Network has two heads where one outputs the result for masked sequence while the other predicts mouth region mask that is used for blending the result into.

3.5. More details on inputs and outputs

Identity images and audio input. Given the pair of matched video and audio we sample random fragment of 5 frames from video. Audio has sample rate of 16kHz so we transform it to mel-spectrogram with 80 filter banks and the corresponding sequence is taken. Our models are working with single images so we further cut the spectrogram to be centered for each video frame. In the result, each video frame has 16 frames of spectrogram of shape (1, 80, 16) . We use only the upper half of each image to mask out the mouth region.

Reference frames. LipGAN[9] and Wav2Lip[5] use single reference frame randomly sampled from the whole video. In our experiments we noticed that reference frames may distract training by choosing frames which are significantly different from the identity frame. So we experimented with different sampling methods, such as: sampling random K frames, sampling from the neighborhood of the identity sequence etc. This step is important because right choice of reference frames will give better visual quality by better encoding the speaker appearance and different lightning condition present in the video. Thus, choosing diverse input from the video is a point.

We decided to use the flexibility of having the separate encoder for reference images to encode multiple reference frames and to use the audio embedding to reweigh the importance of each sample in every occasion. We want the selection algorithm to produce different poses present in the video and to not incorporate any information leak. Hence, we extracted the landmarks for each frame and applied clustering method KMeans to separate the frames into K clusters. This ensures the diversity of selected frames. We used lips and chin landmarks for clustering , because these parts are the most relevant for the task. In the end, number of clusters K=10 was chosen with good diversity in poses and mouth openness. During training we use the same K frames for each identity frame.

However, given short videos in the train set we do not always get frames from each cluster, that is why to reduce redundancy of the data only those frames with present clusters are taken. To fill the free space we randomly augment chosen frames with affine transformations.

Binary masks for mouth region. Our model predicts binary masks for mouth region to blend the generated results into. We generate masks using extracted landmarks from bottom chin part and the nose tip. The convex hull algorithm from OpenCV is used and mask is further processed with dilation with kernel size 11. This ensures that masks cover the face all the time.

3.6 Information leak and generalization

After first iterations of research we noticed that model can severely overfit due to the training scheme explained in section 3.2. Model can learn some features which are present during training but absent during inference.

For example, we used the mask M to locate a chin position because for some videos we encountered a problem that mask does not cover up the whole mouth region. During training phase everything worked normal but on inference we got bad result with no lip-sync at all. We also had an experiment where we used mouth mask computed from landmarks instead of constant one, but this had the same drawback. After making mask larger and fixating it, the result became much more better. The problem with this approach is that model just learns how to open the mouth depending on the mouth mask region, i.e. if the mouth mask is translated to the bottom then the mouth should be opened and otherwise.

Interestingly, we discovered a few more problems like this which distract training, independent of other training techniques used. These are main ones:

1. Learning correlations between mask position and lip shape.
2. Relying too much on reference frames or copying the reference if it is the same as target or very similar.

The solutions for these:

1. Make mask large enough, it should cover the mouth/chin region as well as background and not to be dependent on landmarks. In our experiments we use the whole bottom half of the image.
2. Do not use as reference frames those from target ones in X. Use Gaussian blur of reference images and remove skip-connections at final layer. This stimulates the network to generate features and not to copy. Get only mouth region as reference frames, this helps with generalization and also improves convergence as capacity of the network is not used for the image parts that are left untacked during inference.

These steps were important for our lip-sync to work better and they were inferior to all architectural insights discussed earlier.

3.7 Model training

We train our model using separate sets of different losses: reconstruction, lip synchronization quality and visual quality losses. Reconstruction losses are computed as comparison of the output with ground truth frames. These are used to penalize the generator network for wrong result. Lip synchronization loss is used to penalize the out-of-sync lip movements, as we already mentioned, the pre-trained discriminator network is used for this. This network is frozen during training of the generator network. Also, we have visual quality losses which are computed using adversarial training with additional discriminator which takes as input sequence of frames and outputs the probability that this sequence comes from true data distribution. In our experiments, the discriminator which uses a sequence of frames tend to get better results as it also can penalize for visually inconsistent lip movements.

Next, we will go into more details of each component but for now it is worth noting how those losses are computed. The generator network has additional head which predicts the mask of the mouth region of identity images. This is used, then, to blend the resulting image back into the original one. In our experiments, it improved the convergence and visual quality because the capacity of generator is not consumed on reconstructing the invariant features such as: background information, neck, hair etc. To train such network we are estimating the ground-truth mask from extracted face landmarks and learn the network to reconstruct this mask. We intentionally do not use the ground-truth mask instead but predict our own to avoid the additional overhead of computing the landmarks for each new frame. Also, using an independent mask for blending may also be not accurate, because it does not reflect the result of the generator network. While the generated mask is conditioned on audio itself and is aligned with the generated image.

In next subsections we are going to discuss each training loss in more details. For quick overview of the whole architecture and losses, refer to Figure 3.2.

L1 loss. For generated image $\hat{X} \in R^{3\times H\times W}$, original frame $X \in R^{3\times H\times W}$ and predicted blending mask $M \in R^{1\times H\times W}$ we can formulate the reconstruction loss as L1 norm of the masked differences between those frames:

$$L_{L1} = \frac{1}{\sum M} \sum M \,|\hat{X} - X| \quad (6).$$

This loss was predominantly used in previous works[8, 17] on its own to both penalize for visual quality and lip synchronization consistency. In our experiments we noticed that large weight on this loss is useful at the start of training but it slows down the training progress in the end because it tends to produce blurry results.

Perceptual or VGG loss. Another reconstruction loss which is predominantly used for the task of style transfer, super resolution and related image generation tasks[31]. This loss computes the perceptual difference between images by comparing the extracted high-level features from the images. We use network ϕ which is a 16-layer VGG network[32] trained on ImageNet[33].

Instead of using the pixel wise difference between the images we compute the distance between feature representations of those images by comparing the feature maps of VGG network. So, let ϕ_i be the activation from i-th layer of the network ϕ. Then we can use the function $\phi(x)_i$ to get the feature representation of the input x. This has shape of $(C_i \times H_i \times W_i)$. So the final loss is formulated as:

$$L^i_{perceptual}(X, \hat{X}) = \frac{1}{C_i H_i W_i} \,||\phi(X)_i - \phi(\hat{X})_i||^2_2 \quad (7).$$

This loss can be classified both as the reconstruction or visual quality depending on the number of layer used to extract feature representation. If one uses the low level features then this loss is very similar to the L1 loss described above, given that low-level features have small receptive field. In our experiments, we use features from the mid-to-high level which do not guarantee that content of images are identical but rather motivates the network to generate images which are identical perceptually. This has a good effect on quality of the images and we can get quality results using only perceptual loss. But still it does not strongly penalize generator for blurry results.

Lip synchronization. We use a pre-trained highly accurate network to penalize for inconsistent lip movements. This is the only strong penalizer for out-of-sync lip movements, so it is especially relevant for our pipeline and the generation quality.

As we already described in section 3.3 the expert network is trained using sequences of frames X. Given that our generator network is trained with single images, the data pipeline is organized in a way that it provides the sequences of frames which are concatenated in batch dimension. The sequences are of length $T_v = 5$. That is why the shape of the input to generator network is now: $(B * T_v \times 3 \times H \times W)$ which will be reshaped to get the input to expert network $X_{seq} \in R^{(B,3*T_v,H,W)}$. Expert network also needs a mel spectrogram which is aligned with those T_v frames. Given our training data we computed that the corresponding length of mel-spectrogram features is $T_a = 16$. Thus, mel input is $A_{seq} \in R^{(B,1,80,T_a)}$ where 80 is a number of mel filter banks.

And the expert loss which is used for training can be formulated using the computed similarity between video and audio representations d from (3). Thus the final loss is:

$$L_{expert} = \frac{1}{B}\sum_{i=1}^{B} - log(d_i) \quad (8)$$

Note that d is further clipped to the range of $[1e^{-3},1]$ due to the extremely low values that may come from expert network which tends to destabilize the training due to explosion in the loss function because of the logarithm.

This network is frozen during training and used only for inference. We scheduled the training in the way that for the first iterations expert loss is not evaluated. Network needs to firstly converge to visually adequate results and then lip sync loss is used. This technique improves the speed of convergence. We used the values of reconstruction losses on validation set for this, in our experiments we use $L_{L1} = 1.1, L_{perceptual} = 27$ as reference values to enable the discriminator loss.

Mask loss. For predicting the mask for mouth region we are using the separate head in generator network, specifically convolutional block with sigmoid activation to obtain the correct range of values. Mask is a 1-channel image in range [0,1] $M \in R^{(B \times 1 \times H \times W)}$. This task is not demanding and the network successfully converges with the use binary cross-entropy loss between the ground-truth mask M_{true} obtained from landmarks and the generated one. We experimented with additional losses on entropy of the mask but did not obtain better results.

The loss for mask can be formulated as:

$$L_{mask} = -\frac{1}{B}\sum M_{true} log(M) + (1 - M_{true}) log(1 - M) \ \ (9).$$

Then we are using this mask to blend the generated result $\hat{X}$ back into the original frame X. We can formulate this operation as:

$$\hat{X} = blur(sg(M)) * \hat{X} + (1 - blur(sg(M)) * X \ \ (10).$$

where $sg()$ is a stop-gradient operation which removes the variable from the computation graph and $blur()$ is a Gaussian blur with $k = 5, \sigma = 3$ parameters. We use stop-gradient so that network do not use the blending operation to update its parameters, which may result into mask being trained in the way that it starts to create the additional holes to optimize other losses, for instance it can skip the lip region to obtain the ideal results for lip sync loss. We also use stop-gradient for computation of other losses which demand masking. $blur()$ is used to get smoother edges of the mask on the blended image.

Adversarial training. All previous losses should provide the acceptable result with good lip synchronization quality, especially it is possible with perceptual loss training. However, the results are still blurry and there is a room for improvement.

That is why we experimented with adding a quality discriminator network to gain visual quality by approximating the images from our video data distribution.

These type of networks are called Generative Adversarial Networks (GAN) [37] and they reach state-of-the-art results in generating high-quality images indistinguishable from real data[34, 35, 36]. In our settings we have a generator

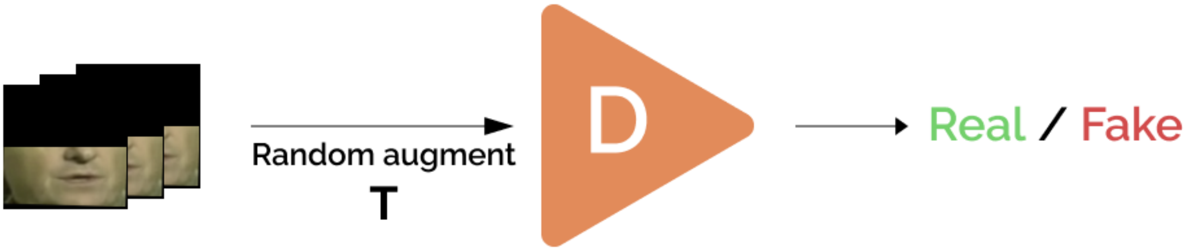

Figure 3.3. Overview of multi-frame discriminator which is obtained to get more fine-grained quality. We employed several stabilization techniques as: R1 regularization, non-saturating loss and DiffAug[39]. Each sample is transformed with random augmentation T which consists of affine and erosion augmentations.

network which generates the lip movements, thus we add the additional discriminator network which is trained to output the probability of the image being real. The overview of the framework can be seen at Figure 3.3.

The original objective function looks like this:

$$L_G = E_{x \sim \hat{X}}[1 - logD(x)] \ \ (11).$$

$$L_D = E_{x \sim X}[logD(x)] + L_G \ \ (12).$$

In our experiments we tried different settings by training the discriminator on single frames and the sequences. The performance of those are quite similar, but multi-frame discriminator tends to produce more consistent result on video because it can penalize the lip sync movements that are not «natural» enough.

Furthermore, we invested time in finding the right GAN training setting. Training of those networks is erroneous and can be quite tricky due to stability issues which received considerable amount of attention from research community[37, 38, 39]. For that reason, we changed the original losses (11) and (12) to non-saturating losses with R1 regularization[40] which were successfully applied for StyleGAN2 architecture[34]. Thus the losses are transformed to:

$$L_G = E_{x \sim \hat{X}}[log(1 + e^{-x})] \ \ (13)$$

$$L_D = E_{x \sim X}[log(1 + e^{-x})] + E_{x \sim \hat{X}}[log(1 + e^{x})] + \frac{\gamma}{2} E_{x \sim \hat{X}}[||\nabla D(x)||_2^2] \quad (14)$$

In our experiments, R1 regularization successfully penalized the discriminator for overfitting, so training could last longer. We used $\gamma = 10$ in all our experiments.

On the other hand, we used convolutional discriminator architecture with strided convolutions for downsampling the input and LeakyReLU as activation function. The input is a sequence of images with masked upper region of the image, so that discriminator does not spend its capacity for invariant image regions. We enabled spectral normalization for each layer which is a standard for regularizing discriminator[38].

Still, we suffered from overfitting due to discriminator simply remembering most of the samples. To tackle this we used the recent technique with differentiable augmentations for adversarial training — DiffAug[39]. The main idea is to randomly augment all samples which come to discriminator with differentiable operations. This enables training the discriminator without overfitting even on the dataset with couple of hundreds of images. We used random affine and erosion augmentations. This technique successfully helped with the problem of overfitting and we reached stable training on our dataset.

Overall loss. To sum up we combine all the losses together and describe training and optimization techniques. The final loss is:

$$L = \alpha_{L1} * L_{L1} + \alpha_{perceptual} * L_{perceptual} + \alpha_{expert} * L_{expert} + \alpha_{mask} * L_{mask} + \alpha_{adv} * (L_G + L_D), (15)$$

where coefficients change depending on the epoch of training. It is best to separate the training into two stages. First for training lip synchronization and the second one with fine-tuning the visual quality of the result.

For the first stage, we are using this set of hyper parameters: $\alpha_{L1} = 1, \alpha_{perceptual} = 0.5, \alpha_{expert} = 0.01, \alpha_{mask} = 1$, where $\alpha_{expert} = 0$ until the validation losses would indicate mild visual quality of the generated images.

For the second stage, it is better to change the balance of the losses to: $\alpha_{L1} = 1, \alpha_{perceptual} = 0.5, \alpha_{expert} = 0.1, \alpha_{mask} = 1, \alpha_{adv} = 0.01$. Getting more for expert loss is because adversarial loss contradicts the expert in some way.

We are using Adam to optimize the networks using learning rate of $2e^{-4}$ for generator and discriminator networks. All the code is written using PyTorch[41] framework and trained on single Nvidia 2080 RTX GPU with batch size of 4. It takes approximately 3 days to converge for stage 1 and 18-24 hours for stage 2. We employ linear learning rate annealing at the end of the training for both first and second stages.

Section 4. Analysis

We provide the quantitative comparison of our result comparing it to the other solutions in the field and running the ablation studies to understand the effect of each component in our pipeline.

4.1. Quantitative evaluation

We are using test set from LRS2 dataset which consists of 1285 videos. For inference the same settings of sampling the identity and reference frames are used so that true frame does not leak to the reference.

The examination of the solution consists of visual quality investigation and the quality of lip synchronization. For quality comparison we selected three different metrics: perceptual metric, peak signal to noise ratio (PSNR) and the Laplacian variance (vL). While for lip synchronization quality examination we used the expert network that was used for training the whole pipeline.

We will go into details of each metric to gain better understanding of what each of them checks.

Perceptual metric. This is the important visual quality metric which compares two images feature representations extracted using VGG16[32] trained on ImageNet[33]. One of the important characteristics of it is its correspondence with humans' judgment and perception[42]. Different variations of perceptual metric are widely used to verify the results of generative models as: GANs[37], VAEs[43] etc.

This metric should provide insights on high-level image similarities and the value of it should be minimized. More details on this metrics is provided in section 3.7, formula (7). We use this metric upon the classic structural similarity index (SSIM) which is also strongly bonded with human perception because it used hand-constructed features which are less versatile than the ones extracted with high-capacity neural network, though it would also be acceptable to use.

PSNR. Peak signal to noise ratio is a classic metric to compute the similarity between the ground-truth image and the reconstructed version of it. It is widely used to evaluate the quality of compression of images and audios. The main advantage of using this metric is that it is relative — being dependent on how much signal is inside the image. It is advised to compute the metric over the converted luminance channel or by converting the color image to grayscale given that human eyes are 4 times less susceptible to changes in chrominance rather than in luminance. However, in our implementation we compared each channel in RGB space and average the final value.

In our experiments we used the masked version of this metric given that generated part of the image is blended into the results and most of the image is untouched.

$$PSNR(f,g) = 20log_{10}(\frac{MAX_f}{\sqrt{MSE(f,g)}}) \ (15)$$

$$MSE(f,g) = \frac{1}{\sum_{i,j} M(i,j)} \sum_i \sum_j M(i,j)\,||f(i,j) - g(i,j)||^2 \ (16)$$

where f is a clean image, g — reconstructed image, MAX_f is a maximum value of our true distribution of images. It equals to 255 in most occasions. This metric should be maximized for better reconstructed images.

Laplacian variance (vL). We use this metric to evaluate how sharp the generated images are. The blurriness of generated images was an issue for past research[8, 17] that is why image discriminators for adversarial training are widely employed to get more fine-grained results. Thus, we decided that it would be relevant to compute such metric and to compare it with other works in the field.

This function is computed using the Laplacian of an image which is a second order derivative calculated by applying Sobel filter twice. Then, variance of the result is computed. We also use the masked version of this metric by computing the variance only for the values inside the mask.

There are several preliminary steps to preprocess the images before getting a Laplacian. First, they should be gray scaled and processed with Gaussian filter to suppress the noise. However, we do not use the Gaussian Filter because our images are already quite blurry and the additional transformation would harm the true estimates.

This metric should be maximized which would indicate that image is sharper.

Lip synchronization quality. We use trained lip-sync expert to evaluate the consistency of lips movements on generated results. The lip-sync expert is trained to high accuracy of 91% of finding the out-of-sync sequences. That is why we decided that it would be relevant to use, especially given that it was trained on LRS2 dataset.

We experimented with different losses obtained from the networks final layers, but most accurate and interpretable for us was the binary cross entropy function (8) which is also used for training. Using the cosine similarity between audio and face embeddings directly is misleading given that the values of this metrics are not uniform and are distributed near its bounds.

To compute the loss we sample video and audio frames with step 5 and compute the expert loss on those sequences. The final value is obtained by averaging.

Comparison with others. To compare the performance of our solution with existing in the field we selected Wav2Lip[5] solution given its similarity to previous solutions and its superiority upon them. You can refer to Table 4.1 for more details.

During qualitative comparison we spotted that Wav2Lip solution has more natural lips movements while our has more acceptable visual quality with higher sharpness (vL) and perceptual VGG metric. We also provide visual comparison between generated samples by sampling a sequence of 5 frames, see Figure 4.1.

Refs.	**Architecture**			**Losses**		**Metrics**			
Sampling method	**ResNet**	**Attn.**	**Audio skips**	**Mask**	**GAN**	**Expert**	**Perc.**	**PSNR**	**vL**
Random 1	No	No	No	No	No	0.833	35.5	13.7	66.2
Random 1 in neigh. of 30	No	No	No	No	No	0.820	35.1	13.5	66.8
KMeans	No	No	No	No	No	0.818	34.4	13.8	66.4
KMeans + Aug	No	No	No	No	No	0.789	33.6	13.6	69.1
KMeans + Aug	Yes	Yes	No	No	No	0.753	30.4	14.6	78.3
KMeans + Aug	Yes	Yes	Yes	No	No	0.724	29.7	14.2	79.5
KMeans + Aug	Yes	Yes	Yes	Yes	No	**0.693**	28.1	15.2	86.1
KMeans + Aug	Yes	Yes	Yes	Yes	Yes	0.716	**24.28**	15.44	**103.2**
Real						***0.651***	-	-	***198.3***
Wav2Lip[5]						**0.679**	26.89	**15.75**	73.42

Table 4.1. Ablation studies and comparison with other solutions in the field. **Real** indicates the metrics on real images from the dataset. Attn. is attention for reference frames, ResNet stands for adding the residual connections, audio skips — skip-connections of audio embedding to decoder, mask stands for predicting the mask and blending. Expert loss is lip-sync quality metric. **Bold** values indicates best results.

4.2 Ablation studies

During our experiments we evaluated different hypothesis including reference sampling, architectural components and different losses. For those different stages of research we recorded the target metrics to show the effect of each one. Please refer to table 4.1 for more information.

Different techniques are relevant for visual quality and lip synchronization. We experimented with several architectural designs and spotted that residual connections are important for the visual quality and convergence. As can be seen from the table, ResNet provided gain in visual quality as well as in lip sync consistency. We also added the attention which together with ResNet gave considerable improvement in lip sync quality.

On the other hand, we also evaluated the effect of skip-connections and giving the decoder information about lip movements and shape, which is encoded in audio embedding. This also resulted in considerable gain in lip sync quality, though it was not significant for visual quality.

We also researched the value of different sampling methods for reference frames. As a baseline we trained the random frame generation sampling technique when reference frame is taken in random from the whole video, excluding the identity frame. This technique was volatile and we spotted the significant ambiguity between poses and lighting conditions between identity and reference frames during training. This difference may be more erroneous if trained on other dataset with longer training videos. Our heuristic decision was to choose the frames from the neighborhood around the target frame assuming that frames are less different in this setting. This worked better, but we continued and experimented with more advanced technique by sampling multiple frames using KMeans to cluster all frames and get complete representations of an identity. By combining this technique with random augmentations of those frames, we got more robust solution with better convergence and all metrics. Please note that when KMeans was used without attention we just averaged the embeddings and intermediate feature representations.

After we added mask predictions with blending into original frame afterwards the visual quality improved, partly because there were less errors in background images. This also helped with training convergence.

Finally, visual quality has been significantly improved after adding adversarial training with nearly 25% gain in sharpness and 15% in perceptual. However, adding adversarial training resulted in increasing the expert loss which is quite surprising given that multi-frame discriminator should help with making lips movements more natural. However, from qualitative point of view, there were not any significant changes and such difference could be due to the expert network artifacts. We did not research this further.

4.3. Future work

There are several ideas which might help with making our decision better and which we have not covered in this work.

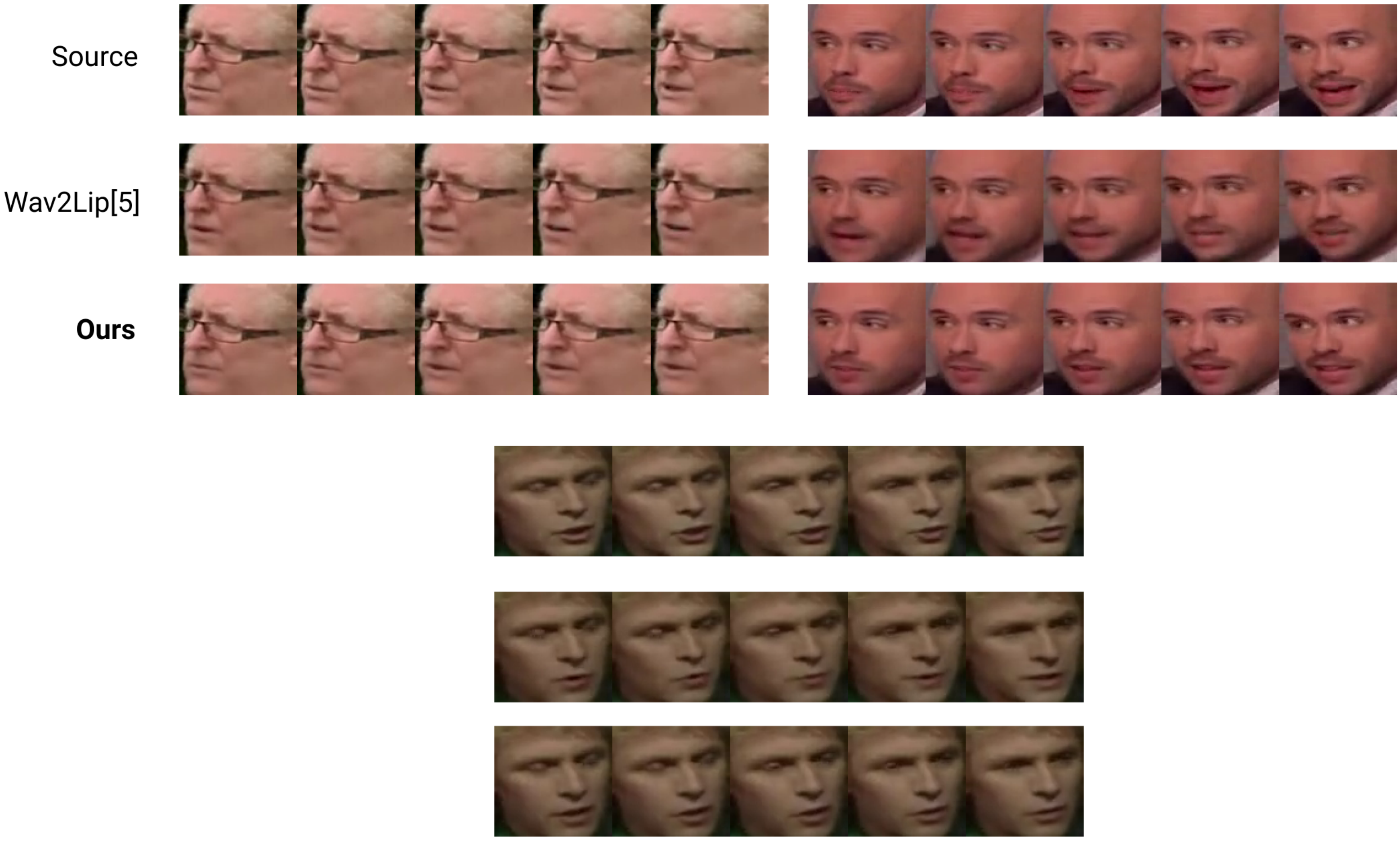

Figure 4.1. Overview of the results. Comparison between Wav2Lip[5] model and ours given random sequences of generated videos. Source indicates true images from the dataset. Our results are sharper than those produced with Wav2Lip while the lip sync quality is comparable.

Fist proposal is data-centric, given our training dataset LRS2 is collected from different British TV shows of 2010-2017 there is large variance in quality of these samples. And we could not get good quality even after training with adversarial training. That is why using newer and bigger dataset may make a significant difference, we believe our solution can scale to the new datasets quite well.

Secondly, we did not evaluate our algorithm in multilingual setup using English-only dataset, though it is quite extensive. This problem can be solved with collection more data or taking the existing dataset. For instance, AVSpeech[26] may be a good fit to continue the research with.

Thirdly, the information leak problem was not solved completely. Though our training techniques helped to solve the most of the problems the face detection network we used leaks the information about the chin position mildly, that is why our solution is not as natural as it could be.

Finally, we have hypothesis that the model design may be redundant and the whole task could be reformulated in predicting the deformation field to warp the reference image rather than generating the whole image. This approach is quite promising with recent works in image animation[44, 45] going for it. We have worked out the possible architecture with encoder-decoder design where the bottleneck would gradually predict the deformation field to warp reference image. All information will be incorporated and shared during the generation rather that using independent encoders for each. We believe that such architectural design may apply its capacity in a more efficient way. Also, this design will give lighter and faster network. However, this may create the additional difficulties for training with additional restraints on deformation field.

Conclusions

In this work we constructed a new algorithm for lip synchronization task with the use of deep learning. We analyzed previous works to construct the efficient neural network architecture and provided the insights on training such system. Our solution produces the comparable results with those existing in the field with improved visual quality. The ablation studies show that learning from additional expert network has significant effect on lip movements consistency. We also showed that skip-connections and residual blocks are superior for the performance and together with adversarial training one can reach visually appealing talking face results.

In future work we want to research the applicability of our solution to more versatile, multilingual datasets which should improve the quality. Furthermore, we believe that our architecture can be improved and propose more efficient design.

References

[1] - J. Clement. Hours of video uploaded to youtube every minute as of may 2019. https://www.statista.com/ statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/, August 2019.

[2] - W3Techs. Usage statistics of content languages for websites. https://w3techs.com/technologies/overview/ content_language, July 2020.

[3] C. M. Koolstra, A. L. Peeters, and H. Spinhof. The pros and cons of dubbing and subtitling. European Journal of Communication, 17(3):325–354, 2002.

[4] B. Wissmath, D. Weibel, and R. Groner. Dubbing or subtitling? effects on spatial presence, transportation, flow, and enjoyment. Journal of Media Psychology, 21(3):114–125, 2009.

[5] K R Prajwal, Rudrabha Mukhopadhyay. Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild — https://arxiv.org/abs/2008.10010

[6] Lele Chen Haitian Zheng. Sound to Visual: Hierarchical Cross-Modal Talking Face Video Generation. — https://www.cs.rochester.edu/~cxu22/p/cvprw2019_facegen_paper.pdf

[7] Ohad Fried, Ayusg Tewari. Text-based Editing of Talking-head Video. — https://arxiv.org/pdf/1906.01524.pdf

[8] Amir Jamaludin, Joon Son Chung. You Said That? Synthesising Talking Faces from Audio — https://www.researchgate.net/publication/331086769_You_Said_That_Synthesising_Talking_Faces_from_Audio

[9] Prajwal K R, Rudrabha Mukhopadhyay. Towards Automatic Face-to-Face Translation — https://arxiv.org/pdf/2003.00418.pdf

[10] Yudong Guo, Keyu Chen. AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis — https://arxiv.org/abs/2103.11078

[11] Yang Zhou , Xintong Han. MakeItTalk: Speaker-Aware Talking-Head Animation — https://people.umass.edu/~yangzhou/MakeItTalk/

[12] Joon Son Chung, Andrew Zisserman. Out of time: automated lip sync in the wild. https://www.robots.ox.ac.uk/~vgg/publications/2016/Chung16a/chung16a.pdf

[13] ObamaNet: Photo-realistic lip-sync from text. Rithesh Kumar, Jose Sotelo — https://arxiv.org/pdf/1801.01442.pdf

[14] Justus Thies, Mohamed Elgharib. Neural Voice Puppetry: Audio-driven Facial Reenactment — https://arxiv.org/abs/1912.05566

[15] - Joon Son Chung and Andrew Zisserman. 2016. Lip reading in the wild. In Asian Conference on Computer Vision. Springer, 87–103

[16] - Naomi Harte and Eoin Gillen. 2015. TCD-TIMIT: An audio-visual corpus of continuous speech. IEEE Transactions on Multimedia 17, 5 (2015), 603–615

[17] Joon Son Chung, Amir Jamaludin. You said that? https://arxiv.org/abs/1705.02966

[18] Hang Zhou, Yu Liu. Talking Face Generation by Adversarially Disentangled Audio-Visual Representation — https://arxiv.org/abs/1807.07860

[19] - T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. 2018. Deep Audio-Visual Speech Recognition. In arXiv:1809.02108. - LRS2 dataset

[20] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. The Journal of the Acoustical Society of America, 120(5):2421–2424, 2006.

[21] C. Richie, S. Warburton, and M. Carter. Audiovisual database of spoken American English. Linguistic Data Consortium, 2009.

[22] - J. S. Chung and A. Zisserman. Lip reading in the wild. In Asian Conference on Computer Vision, pages 87–103. Springer, 2016.

[23] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612, 2017.

[24] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622, 2018.

[25] Triantafyllos Afouras, Joon Son Chung. LRS3-TED: a large-scale dataset for visual speech recognition — https://arxiv.org/pdf/1809.00496.pdf

[26] AVSpeech Large-scale Audio-Visual Speech Dataset — https://looking-to-listen.github.io/avspeech/index.html

[27] Adrian Bulat, Georgios Tzimiropoulos. How far are we from solving the 2D \& 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks) — https://arxiv.org/abs/1703.07332

[28] Jianzhu Guo, Xiangyu Zhu. Towards Fast, Accurate and Stable 3D Dense Face Alignment — https://arxiv.org/abs/2009.09960

[29] Olaf Ronneberger, Philipp Fischer. U-Net: Convolutional Networks for Biomedical Image Segmentation. https://arxiv.org/abs/1505.04597

[30] Kaiming He, Xiangyu Zhang. Deep Residual Learning for Image Recognition https://arxiv.org/abs/1512.03385

[31] Justin Johnson, Alexandre Alahi. Perceptual Losses for Real-Time Style Transfer and Super-Resolution — https://cs.stanford.edu/people/jcjohns/papers/eccv16/JohnsonECCV16.pdf

[32] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)

[33] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3) (2015) 211–252

[34] Tero Karras, Samuli Laine. Analyzing and Improving the Image Quality of StyleGAN — https://arxiv.org/abs/1912.04958

[35] Tero Karras, Samuli Laine. A Style-Based Generator Architecture for Generative Adversarial Networks — https://arxiv.org/abs/1812.04948

[36] Andrew Brock, Jeff Donahue. Large Scale GAN Training for High Fidelity Natural Image Synthesis — https://arxiv.org/abs/1809.11096

[37] Ian J. Goodfellow, Jean Pouget-Abadie. Generative Adversarial Networks. https://arxiv.org/abs/1406.2661

[37] Tim Salimans, Ian Goodfellow. Improved Techniques for Training GANs — https://arxiv.org/abs/1606.03498

[38] Takeru Miyato, Toshiki Kataoka. Spectral Normalization for Generative Adversarial Networks — https://arxiv.org/abs/1802.05957

[39] Shengyu Zhao, Zhijian Liu. Differentiable Augmentation for Data-Efficient GAN Training — https://arxiv.org/abs/2006.10738

[40] Lars Mescheder, Andreas Geiger. Which Training Methods for GANs do actually Converge? — https://arxiv.org/abs/1801.04406

[41] Adam Paszke, Sam Gross. PyTorch: An Imperative Style, High-Performance Deep Learning Library — https://arxiv.org/abs/1912.01703

[42] Martin Heusel, Hubert Ramsauer. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium — https://arxiv.org/abs/1706.08500

[43] Diederik P Kingma, Max Welling. Auto-Encoding Variational Bayes — https://arxiv.org/abs/1312.6114

[44] Aliaksandr Siarohin, Stéphane Lathuilière. First Order Motion Model for Image Animation — https://arxiv.org/abs/2003.00196

[45] Linsen Song, Wayne Wu. Everything's Talkin': Pareidolia Face Reenactment — https://arxiv.org/abs/2104.03061

[46] Vincent Dumoulin, Francesco Visin. A guide to convolution arithmetic for deep learning — https://arxiv.org/abs/1603.07285

[47] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks — https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[48] Kaiming He, Xiangyu Zhang. Identity Mappings in Deep Residual Networks — https://arxiv.org/pdf/1603.05027.pdf