Improving Audio-Driven Visual Dubbing Solutions using Self-Supervised Generative Adversarial Networks

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Mayur Ranchod
1601745

Supervised by Prof. Richard Klein

August 31, 2023

Ethics Clearance Number: H23/01/23

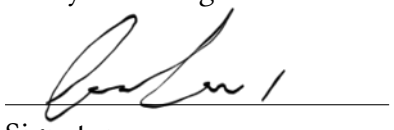
A dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science

Abstract

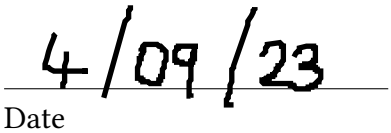
Audio-driven visual dubbing (ADVD) is the process of accepting a talking-face video, along with a dubbing audio segment, as inputs and producing a dubbed video such that the speaker appears to be uttering the dubbing audio. ADVD aims to address the language barrier inherent in the consumption of video-based content caused by the various languages in which videos may be presented. Specifically, a video may only be consumed by the audience that is familiar with the spoken language. Traditional solutions, such as subtitles and audio-dubbing, hinder the viewer's experience by either obstructing the on-screen content or introducing an unpleasant discrepancy between the speaker's mouth movements and the input dubbing audio, respectively. In contrast, ADVD strives to achieve a natural viewing experience by synchronizing the speaker's mouth movements with the dubbing audio. A comprehensive survey of several ADVD solutions revealed that most existing solutions achieve satisfactory visual quality and lip-sync accuracy but are limited to low-resolution videos with frontal or near-frontal faces. Since this is in sharp contrast to real-world videos, which are high-resolution and contain arbitrary head poses, we present one of the first ADVD solutions trained with high-resolution data and also introduce the first pose-invariant ADVD solution. Our results show that the presented solution achieves superior visual quality while also achieving high measures of lip-sync accuracy, consequently enabling the solution to achieve significantly improved results when applied to real-world videos.

Declaration

I, Mayur Ranchod, hereby declare the contents of this dissertation to be my own work. This dissertation is submitted for the degree of Master of Science in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.



Signature



Date

Acknowledgements

Conducting this research over the past 2 years has certainly been an insightful learning experience for me – undoubtedly, the greatest lesson that I have learned during this period is to never give up. This is in spite of the endless load-shedding and countless hardware issues (amongst others) that made this experience far more challenging than it was intended to be. This research would not have been possible, and I certainly would not be where I am today, if it was not for the support of my amazing circle of family and friends. I start off by expressing my gratitude to my parents, grandparents, aunts and uncles, sister, brother-in-law, and cousins for all they have done for me and for always making every effort to attend to my every need. A special thanks goes out to my 2 younger cousins, Ryu and Ishan, for always turning the most agonizing times for me into the most fun and joyful times. I would also like to thank my close circle of friends, especially Yukta, Wesley, and Progress amongst others for all their support and always holding me to a high standard. I express my gratitude to the Sybrin team, especially Kalin and Ali, for accommodating me to comfortably undertake the internship in tandem with my studies. I take this opportunity to also thank members of the RAIL research group for all their advice and assistance throughout this research. Last but not least, I thank my supervisor, Prof. Richard Klein, for granting me access to the CHPC cluster, as well as the PRIME machines.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Over the years, video-based content has proven to be an effective means of conveying information to a wide audience in a captivating manner compared to other forms of media such as images, audio, and text [Wilson *et al.* 2010; Krämer and Böhrs 2017; Schneiders 2020]. A distinguishing property of videos is their *bi-modal* nature; comprising of an audio and visual stream that simultaneously stimulates the viewer's auditory and visual systems [Molholm *et al.* 2002]. Due to their efficacy and versatility, videos may manifest themselves in numerous forms such as movies, television programs, and educational videos (e.g., online tutorials and documentaries) amongst many others. The production and consumption of video-based content has been further accelerated in recent years due to the popularity of video-based platforms such as YouTube, Netflix, and TikTok [Curry 2022]. To contextualize the consumption of video content, Goodrow [2017] state that more than one billion hours of video content are watched on YouTube daily.

Despite the widespread adoption of videos, their consumption is inherently limited due to the multiplicity of languages in which videos may be presented [Park *et al.* 2017]. Specifically, a video can only be fully understood by the audience familiar with the spoken language, thus, forming a *language barrier* for those who are not [Kottahachchi and Abeysinghe 2022]. This issue is particularly prevalent for viewers unfamiliar with English since 59.5% of the content on the Internet is presented in English [W3Techs 2022]; however, only a quarter of its users speak English as their first language [Interworldstats 2022].

1.2 Traditional Methods

Due to the importance of addressing the aforementioned language barrier, two predominant solutions have emerged i.e., *subtitling* [Cintas and Remael 2014] and *audio dubbing* [Chaume 2020b]. Subtitling entails displaying the translated script on-screen in synchrony with the spoken content. Unfortunately, subtitling tends to (1) divert the viewer's attention from the on-screen content to the subtitles, (2) obscure the viewer's view of the on-screen content, and (3) increase the cognitive requirements of the viewer, who is now required to read the subtitles

[Koolstra *et al.* 2002]. In contrast, audio dubbing is the process of replacing the (foreign) audio track of the video with one spoken in a target language understood by the local population [Martínez 2004]. Due to the extensive human involvement required, audio dubbing is a time-consuming and laborious process. The primary issue with audio dubbing is the discrepancy introduced between the speaker's mouth movements and the new audio track, resulting in an unpleasant viewing experience [Koolstra *et al.* 2002].

This research is motivated by the inability of subtitling and audio dubbing to achieve a natural viewing experience when localizing foreign content [Mailhac 2000; Díaz-Cintas 2013], as well as the need to address the language barrier inherent in the consumption of videos. Our objective is to broaden the audience to whom videos are consumed and, by extension, to expose viewers to the immense potential of foreign video production, as opposed to solely confining them to videos originally presented in their native language [Di Giovanni 2018; Limov 2020].

1.3 Motivation

In an attempt to address the aforementioned problem, we perform *audio-driven visual dubbing* (*ADVD*). ADVD is the process of accepting a talking-face video and a dubbing audio segment as inputs and producing a dubbed video in which the speaker's mouth movements are manipulated to appear as if the speaker is uttering the input dubbing audio. With this approach, the viewer's view of the on-screen content is no longer obstructed (unlike in the case of subtitles), and the unpleasant discrepancy between the speaker's mouth movements and the dubbing audio is eliminated (unlike in the case of audio dubbing). The goal is to achieve a seamless viewing experience such that the viewer is unable to distinguish a dubbed video from a video originally presented in their native language.

When assessing the efficacy of a visual dubbing solution, it is primarily the *visual quality* and *lip-sync accuracy* that is of interest. Visual quality refers to ensuring that the speaker's mouth region is morphed without appearing *uncanny* [Mori *et al.* 2012] or introducing artefacts such as blurring or colour distortion. Achieving accurate lip-sync is essential since the threshold for detecting sync errors by an average human viewer is approximately −125ms (the audio lags the video) to +45ms (the audio leads the video) [Chung and Zisserman 2016b; ITU 1998].

1.4 Prior Works

The concept of ADVD was first proposed in the seminal work of Bregler *et al.* [1997] that introduced an audio-driven solution employing a Hidden Markov Model (HMM) to construct an annotated database that maps *phonemes* (the smallest unit of sound) [Twaddell 1935] to *visemes* (the visual counterpart of phonemes) [Bear and Harvey 2017]. The HMM is then used to label the video's original audio to segment the video into frames. Subsequently, the input dubbing audio is labelled and used to index the dataset to retrieve the corresponding frames. The final result is produced by extracting the mouth region of the retrieved frames, which gets stitched into the frames of the original video. Despite pioneering the field, the solution is unable to achieve satisfactory results, which may be attributed to the challenges associated with ADVD, i.e., (1) using low-dimensional data (audio) to drive higher-dimensional data (video) [Suwajanakorn *et al.* 2017], and (2) the phoneme-to-viseme mapping being many-to-one (e.g., bilabial consonants (/b/, /m/, and /p/) all correspond to the same viseme) [Garrido *et al.* 2015].

As a consequence of the poor results achieved by Bregler *et al.* [1997], this led to a series of video-driven visual dubbing solutions [Garrido *et al.* 2015; Thies *et al.* 2016; Kim *et al.* 2018 2019b] being introduced, which take a considerably simplified approach towards dubbing. These solutions use the input video and a dubbing video to construct a 3D facial reconstruction i.e., a 3-Dimensional Morphable Model (3DMM) [Blanz and Vetter 1999] of the speaker and dubbing actor and simply melds the dubbing actor's mouth region onto the speaker's 3DMM. Due to the mediocre results achieved, as well as the unrealistic requirement for a video of the dubbing actor uttering the dubbing content, it was apparent that video-driven visual dubbing was not the optimal approach.

Following the advent of deep learning, researchers have gained improved tools and knowledge, leading to a series of ADVD solutions [Suwajanakorn *et al.* 2017; Chung *et al.* 2017; Thies *et al.* 2020; Prajwal *et al.* 2020]. Early solutions lacked temporal stability as they re-sequenced frames from the original video instead of synthesizing the appropriate mouth shape. Additionally, many solutions [Karras *et al.* 2017a; Thies *et al.* 2020; Song *et al.* 2022] require two to five minutes of video footage to construct an accurate 3DMM of the speaker for dubbing. However, this data requirement and speaker dependency of these solutions severely limit their applicability, as the data may not always be available, and retraining is needed for each new speaker.

It was only after the revolutionary work of Chung *et al.* [2017] that the need for 3DMM construction was obviated, thus, enabling *speaker-independent* ADVD. The solution employs an encoder-decoder network composed of an identity encoder, audio encoder, and face decoder, trained on footage of multiple speakers. Due to the significant advancement achieved, this solution went on to serve as the basis of numerous state-of-the-art solutions [KR *et al.* 2019; Prajwal *et al.* 2020; Yang *et al.* 2020]. Specifically, Prajwal *et al.* [2020] adapts this solution to a GAN framework [Goodfellow *et al.* 2014] and employs a pre-trained lip-sync discriminator to enforce accurate lip-sync. Yang *et al.* [2020] also adopts a GAN-based approach and incorporates an additional (reference) encoder and a temporal aggregation module to improve temporal stability.

1.5 Our Solution

Despite recent solutions [KR *et al.* 2019; Prajwal *et al.* 2020; Yang *et al.* 2020] achieving satisfactory measures of visual quality and lip-sync accuracy, we aim to significantly improve these measures. We present a GAN-based ADVD solution based on a deep residual U-Net generator [Zhang *et al.* 2018] trained in conjunction with a pre-trained lip-sync discriminator composed of R(2+1)D spatiotemporal blocks [Tran *et al.* 2018a]. Our results demonstrate that the presented pre-trained lip-sync discriminator achieves a 91% off-sync detection accuracy on the LRS2 test dataset [Afouras *et al.* 2018a], whereas Wav2Lip [Prajwal *et al.* 2020] achieves an accuracy of 81%. Furthermore, through quantitative, qualitative, and human subjective studies, we confirm that the presented solution achieves superior visual quality while maintaining high lip-sync accuracy.

While existing solutions have proven to achieve satisfactory measures of visual quality and lip-sync accuracy, these achievements are limited to low-resolution videos with frontal or near-frontal faces. However, real-world videos are typically high-resolution and contain ar-

bitrary head poses, making existing solutions unsuitable for widespread adoption. To fully realize the immense potential of ADVD in various applications such as video localization, multilingual advertising campaigns [De Ruiter 2021], teleconferencing [Zhou *et al.* 2020b; Doukas *et al.* 2021], and enhanced digital assistants, we aim to address the shortcomings of current solutions.

Firstly, we present one of the first ADVD solutions to utilize high-resolution training data by adapting our approach to handle inputs with a size of 192×192, instead of the typical 96×96 used by most existing solutions. This adjustment aims to investigate whether significantly improved visual quality can be achieved. Since there is no off-the-shelf dataset available to train such a solution, we employ the AVSpeech dataset [Ephrat *et al.* 2018] and design a comprehensive data-cleaning pipeline to transform the dataset into a suitable format for addressing the ADVD problem. Secondly, we present the first pose-invariant ADVD solution capable of dubbing talking faces with arbitrary head poses. Since a dataset containing a vast range of head poses does not exist, this research investigates whether improved results can be achieved by *pose-augmenting* [Cheng *et al.* 2020] the training data to rotate frontal and near-frontal footage to arbitrary head poses, which is subsequently used to train the solution. In addition to these advancements, the presented solution is speaker-independent, language-independent, and applicable in unconstrained (in-the-wild) conditions, which follows as a consequence of the solution's design and the training data used.

The contributions of this research are summarized as follows:

- A GAN-based ADVD solution based on a deep residual U-Net generator [Ronneberger *et al.* 2015; Zhang *et al.* 2018] trained in conjunction with a pre-trained lip-sync discriminator is presented. Results show that this solution achieves superior visual quality while maintaining high measures of lip-sync accuracy.
- A comprehensive data-cleaning pipeline to pre-process audio-visual talking-face datasets into a suitable form for addressing the ADVD problem.
- One of the first ADVD solutions to utilize high-resolution training data is presented, resulting in significantly improved visual quality.
- A first attempt at developing a pose-invariant ADVD solution is presented to enhance the quality of results when applied to non-frontal faces. We pose-augment [Cheng *et al.* 2020] the training dataset to expose the solution to a diverse range of head poses which allows the solution to learn pose-invariant features.

The structure of this dissertation is organized as follows: Chapter 2 provides an overview of the foundational concepts underlying our solution, along with a summary of previous approaches that highlight the rapid evolution within the field of visual dubbing. Chapter 3 offers an encompassing view of the entire solution, while the subsequent chapters delve into specific aspects. Chapter 4 explores the data landscape when addressing the ADVD problem, followed by a detailed presentation of our pre-trained lip-sync discriminator in Chapter 5. Chapter 6 elaborates on the generator network, while Chapters 7 and 8 delve into our high-resolution and pose-invariant solutions, respectively. Ethical considerations and real-world applications are discussed in Chapter 9, and Chapter 10 outlines potential future research directions along with a summary of the salient points presented in this study.

Chapter 2

Background & Related Work

2.1 Introduction

This chapter begins with an overview of the background concepts essential for understanding the presented solution. Thereafter, a summary of the solutions leveraged by our approach is provided, followed by a literature review of the visual dubbing field.

2.2 Background

This analysis commences with an overview of generative adversarial networks (GANs) [Goodfellow *et al.* 2014], which serve as the foundation of our solution. The mechanism behind GANs, as well as the numerous challenges associated with their training, are discussed. Subsequently, the notion of *contrastive learning* [Hadsell *et al.* 2006], which underpins the successful training of our lip-sync discriminator, is presented. Lastly, the building blocks of the lip-sync discriminator, *R(2+1)D spatiotemporal blocks* [Tran *et al.* 2018b], are explained.

2.2.1 Generative Adversarial Networks

A generative adversarial network (GAN) is a deep generative model introduced by Goodfellow *et al.* [2014] which attempts to learn the data distribution of training data. Doing so enables the synthesis of new (realistic) samples by sampling from the learned distribution. GANs are distinguished from other deep learning models by their composition of two sub-networks i.e., a generator network G and a discriminator network D.

At the core of the GAN framework, the generator network learns the data distribution of the training data, thus, enabling the synthesis of new samples through sampling [Donahue *et al.* 2016; Hoang *et al.* 2018]. Traditionally, the generator accepts a random noise vector $z \in \mathbb{R}^n$ sampled from a prior distribution e.g., a Gaussian distribution, as input which represents a point within the latent space. The generator's task is to produce a sample that is indistinguishable from real samples [Mogren 2016]. In practice, more convenient input representations, such as images, are preferred which get downsampled (*encoded*) into a latent

vector z which is then used to produce the final result [Isola *et al.* 2017; Wang *et al.* 2018; Jiang *et al.* 2021c].

In simple terms, the discriminator network functions as a binary classifier. It takes samples, either synthesized by the generator or real data, as input and is responsible for estimating the probability of the sample being real [Goodfellow *et al.* 2014; Salimans *et al.* 2016].

2.2.1.1 Training GANs

Despite its simplicity, the discriminator plays a crucial role in training the generator network. The two sub-networks of a GAN interact *adversarially*. This means that the generator is trained to minimize the probability that the discriminator correctly classifies synthesized samples as fake, whereas the discriminator is trained to maximize the probability that synthesized samples are correctly distinguished from real samples [Goodfellow *et al.* 2014; Frid-Adar *et al.* 2018]. This dynamic is captured by the *minmax* adversarial loss used to train GANs where V represents the value function, x denotes a real sample, and p_{data} represents the real data distribution which is given as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data(\boldsymbol{x})}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{2.1}$$

While the generator and discriminator networks attempt to optimize the objective function, they engage in a *zero-sum* game [Ge *et al.* 2018]. When the discriminator successfully discriminates synthesized samples from real samples, its parameters remain unchanged, while the generator receives large parameter updates as a penalty. Conversely, when the generator successfully fools the discriminator by synthesizing realistic samples, the generator's parameters stay unchanged, while the discriminator is penalized with large parameter updates. The generator network uses the feedback provided by the discriminator to update its weights [Wang *et al.* 2019]. The GAN converges when a *Nash Equilibrium* [Kreps 1989; Weng 2019] is reached, meaning that generated samples become indistinguishable from real samples, causing the discriminator to output a 0.5 probability for all inputs. Figure 2.1 illustrates a typical GAN setup.

An intuitive way to grasp the dynamics of GANs is by likening them to counterfeiting money [Goodfellow *et al.* 2014]. The generator represents the counterfeiter, and the discriminator acts as the police. Initially, the generator produces non-sensical outputs, leading to substantial parameter updates since the discriminator can easily distinguish fake specimens from real ones. With continued feedback from the discriminator, the generator improves the realism of the synthesized samples until the discriminator is unable to differentiate between real and synthesized samples [Durugkar *et al.* 2016; Chaudhari *et al.* 2020].

2.2.1.2 Challenges & Applications

In the early days of GANs, despite researchers being cognizant of their promise for several generative tasks, in practice, they were susceptible to various *failure modes* [Salimans *et al.* 2016; Lin *et al.* 2021], i.e.:

- *The Vanishing Gradient Problem*: Occurs in deep neural networks trained with gradient-based methods, where gradients diminish during training. In the context of GANs, this

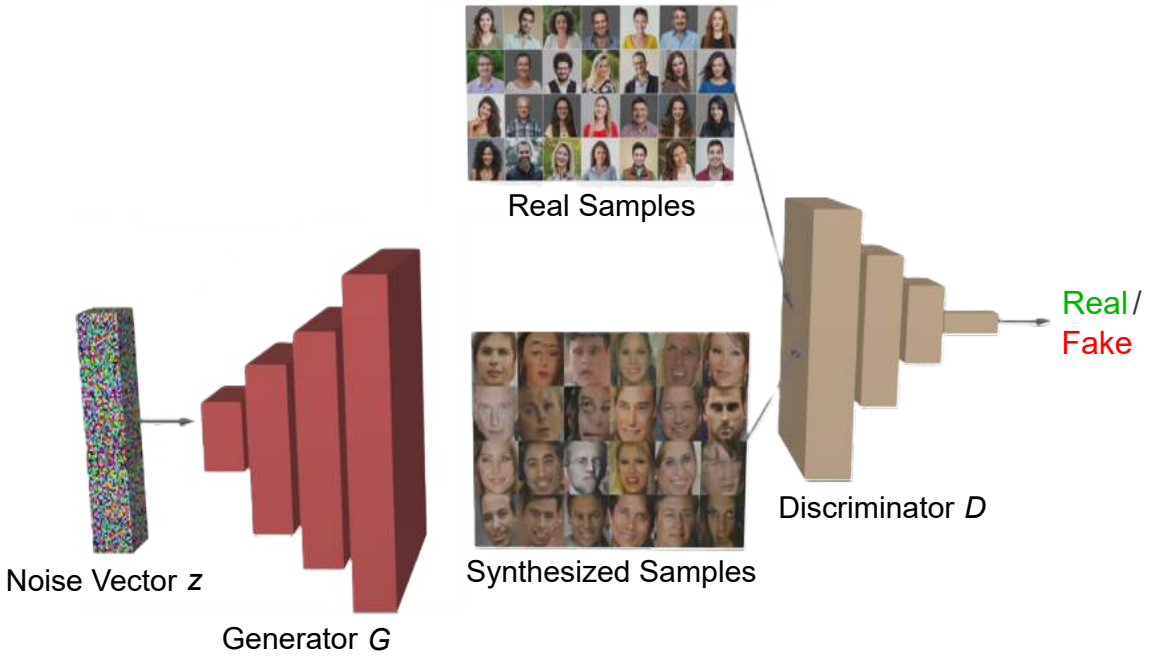


Figure 2.1: A typical GAN setup which illustrates the generator synthesizing samples whilst the discriminator classifies inputs as either real or fake.

problem prevents the generator from backpropagating the feedback provided by the discriminator effectively, leading to difficulties in performing meaningful parameter updates throughout the network [Weng 2019; Mansourifar *et al.* 2019].

- *Mode Collapse*: Occurs when the generator produces an especially plausible output which consistently fools the discriminator. This issue causes the generator to map all inputs to this output, therefore, depleting the variability of the samples produced [Tran *et al.* 2018d; Bau *et al.* 2019].

- *Failure to Converge*: Since the generator and discriminator are trained in tandem, balancing the strength of each network is crucial to achieving stability. When either becomes overpowering, the network is unable to provide any valuable feedback to the other with regards to how the complementary network should be updated. When such instability occurs, the GAN experiences oscillatory behaviour and eventually fails to converge [Li *et al.* 2018; Weng 2019].

Following the work of Radford *et al.* [2015], which proposes several design guidelines to mitigate failure modes, GANs have been widely adopted for various tasks, primarily in the image domain. These tasks include image-to-image translation [Yi *et al.* 2017; Zhu *et al.* 2017a; Isola *et al.* 2017], image super-resolution [Ledig *et al.* 2017; Wang *et al.* 2018], and image inpainting [Demir and Unal 2018; Jiang *et al.* 2020], among others. The unique training paradigm of GANs allows for more complex data distributions to be modelled, resulting in a closer approximation to the real data distribution compared to other models such as Convolutional Neural Networks (CNNs) [Fard *et al.* 2021]. In recent years, GANs have emerged as the dominant

generative model, particularly showcased by flagship models such as *StyleGAN* [Karras *et al.* 2019] which demonstrates unmatched photorealism.

2.2.2 Contrastive Learning

Contrastive learning [Hadsell *et al.* 2006] is a framework commonly used for self-supervised representation learning [Chen *et al.* 2020a; Kotar *et al.* 2021; Wickstrøm *et al.* 2022]. As the name suggests, a latent space is learned by contrasting (comparing) samples, typically in a pairwise manner. This process entails forming *positive pairs* (x, x^+) of latent vector representations of two semantically related/similar samples, and *negative pairs* (x, x^-) of latent vector representations of two semantically unrelated/dissimilar samples [Chuang *et al.* 2020]. By leveraging the inherent property of latent spaces where distance (such as Euclidean distance) is a measure of semantic similarity, the objective of contrastive learning is to attract the latent points of positive pairs to reside close by and to repel the latent points of negative pairs. For instance, consider images of a cat, lion, and dog as illustrated in Figure 2.2 — the latent vectors of the cat image and lion image would form a positive pair since both species belong to the feline family [Stains 2022] and should therefore be attracted in the latent space. In contrast, the latent vectors of the lion image and the dog image would form a negative pair since a lion is deemed dissimilar to a dog relative to a cat. In practice, the attraction and repulsion of latent points is achieved through a series of contrastive losses such as max-margin, InfoNCE, or triplet loss [Oord *et al.* 2018; Zolfaghari *et al.* 2021].

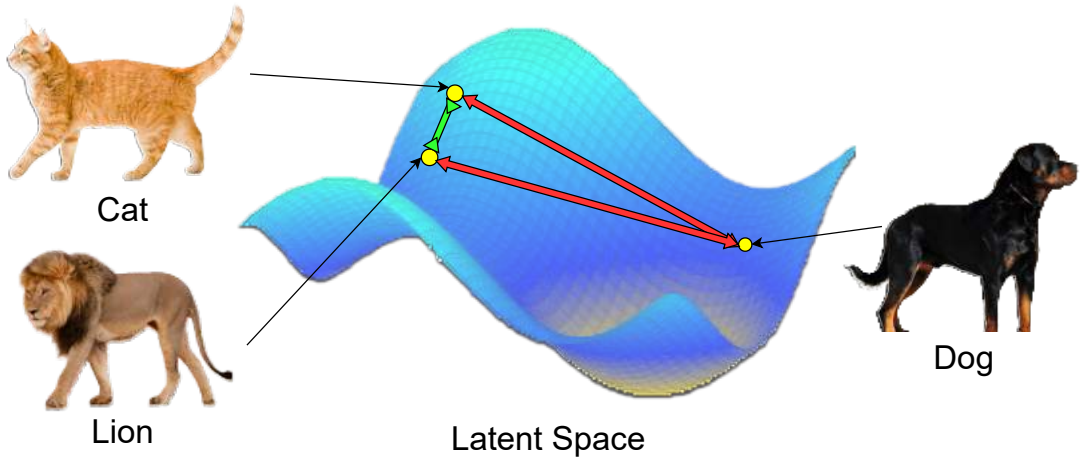


Figure 2.2: Illustration of the repulsion between (cat, dog) and (lion, dog) latent points and the attraction between (cat, lion) latent points.

2.2.3 R(2+1)D Spatiotemporal Blocks

One of the main characteristics distinguishing videos from images is the temporal information embedded within adjacent video frames, which enables temporal reasoning in addition to accounting for spatial dimensions [Huang *et al.* 2018]. Therefore, to fully exploit the unique nature of videos, *spatiotemporal* features should be learned. This is not possible with 2D convolutions which disregard the temporal dimension. In contrast, employing spatiotemporal

convolutions, such as 3D convolutions, is preferred, however, their adoption is hindered by their high computational cost [Tran *et al.* 2018a].

To improve how spatiotemporal features are learned, Tran *et al.* [2018a] proposed to factorize 3D convolutions into two independent and successive operations i.e., a 2D convolution in the spatial domain followed by a 1D convolution in the temporal domain. Since these convolutions are implemented within a ResNet-like architecture, they are referred to as *R(2+1)D spatiotemporal blocks*. The benefits of this decomposition are two-fold: Firstly, the number of non-linearities is doubled without significantly increasing the number of parameters. The additional non-linearity between the two convolution operations allows for more complex functions to be modelled. Secondly, this decomposition simplifies optimization, leading to lower training and testing errors. By carefully controlling the dimensionality of the subspace between spatial and temporal convolutions, the number of parameters can be made to be less than or equal to that of 3D convolutions. Due to its compact design, R(2+1)D spatiotemporal blocks serve as a drop-in replacement for traditional 3D convolutions. They have also proven to achieve an improved trade-off between accuracy and computational cost for video understanding tasks, such as action recognition [Tran *et al.* 2018a].

Consider the decomposition of a traditional 3D convolution with a kernel size of $t \times w \times h$, where t denotes the temporal extent, and w and h represent the width and height of the spatial dimension, respectively. Despite theoretically performing 2D and 1D convolutions, in practice, these operations are achieved using 3D convolutions by carefully calibrating the kernel sizes to facilitate the implementation. Specifically, the 2D spatial convolution is performed using a kernel size of $1 \times w \times h$, and similarly, the 1D temporal convolution is performed using a kernel size of $t \times 1 \times 1$, as shown in Figure 2.3.

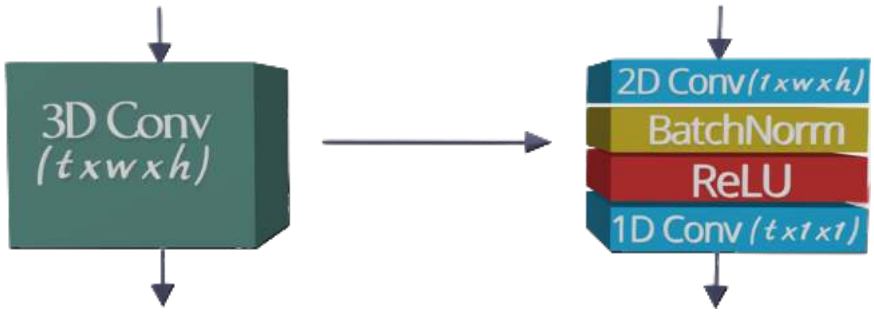


Figure 2.3: A 3D convolution decomposed into an R(2+1)D spatiotemporal block.

2.3 Related Work

This section commences with a succinct overview of the *SyncNet* network [Chung and Zisserman 2016b] which serves as the basis of our lip-sync discriminator. In addition, a brief summary of the *pose augmentation* solution [Cheng *et al.* 2020] used to synthesize non-frontal talking-face footage from frontal and near-frontal footage for our pose-invariant solution is presented. Lastly, an overview of a few video-driven and audio-driven visual dubbing solutions is provided, which highlights the rapid evolution that the field has undergone.

2.3.1 SyncNet

SyncNet [Chung and Zisserman 2016b] is a speaker and language-independent network designed to measure audio-visual synchronization between mouth motion and speech in videos. The network is trained in a self-supervised manner, utilizing the audio and visual channels of talking-face videos through *cross-modal supervision* [Sayed *et al.* 2018; Chung *et al.* 2019]. SyncNet is composed of two streams i.e., a video stream that takes five grayscale images of the mouth region as input, and an audio stream that takes audio as Mel-Frequency Cepstral Coefficients (MFCCs) [Bridle and Brown 1974; Mermelstein 1976]. The visual stream's architecture, based on the work of Chung and Zisserman [2016a], employs 3D convolutions which allow for spatiotemporal features to be learned. The audio stream is derived from VGG-M [Chatfield *et al.* 2014]. Both streams are trained simultaneously, aiming for the audio and visual embeddings a and v (produced by the Multi-Layer Perceptron (MLP) at each stream's end) to be similar for in-sync videos and different for out-of-sync videos. The network's training employs the max-margin contrastive loss, given as:

$$E = \frac{1}{2N}\sum_{n=1}^{N}(y_n)d_n^2 + (1-y_n)max(margin - d_n, 0)^2, \tag{2.2}$$

$$d_n = \|v_n - a_n\|_2, \tag{2.3}$$

where $y \in [0, 1]$ denotes the binary similarity between the audio and visual inputs.

Given a talking-face video as input, SyncNet outputs three values: an AV offset, which measures the offset between the audio and visual streams (measured in frames), a distance measure, and a confidence measure [Chung and Zisserman 2016b]. The distance measure is computed by shifting the video (relative to the audio) by a range of offsets. For each shift (offset), the windowed L_2 distance between the audio and video features is computed and averaged to produce a synchrony score. The distance measure reported is taken to be the minimum of all synchrony scores computed (lower is better). On the other hand, the confidence measure is computed by determining whether the minimum distance offset obtained has a much lower distance measure compared to adjacent offsets [Nayak *et al.* 2022]. A larger difference indicates that the model is highly confident that the reported offset is correct. Chung and Zisserman [2016b] also show that SyncNet may be adopted for other audio-visual tasks, such as active speaker detection and lip-reading [Chung and Zisserman 2017; Fernandez-Lopez and Sukno 2018].

2.3.2 Pose Augmentation

Cheng *et al.* [2020] propose a pose-invariant lip-reading solution, which stemmed from the observation that the majority of existing solutions at the time were mainly designed for frontal and near-frontal faces [Ninomiya *et al.* 2015; Petridis and Pantic 2016; Chung and Zisserman 2016a]. Their approach aims to broaden the applicability of lip-reading solutions to a wider range of head poses by introducing an innovative *pose augmentation* operation. Notably, instead of redesigning the solution itself, the pose augmentation utilizes the abundance of frontal talking-face footage to synthesize talking-face footage at arbitrary head poses by rotating the speaker's head in 3D space. The non-frontal talking-face footage produced can then be used to train other existing solutions to be pose invariant.

To perform pose augmentation, a 3D representation of the speaker's face is first obtained. This approach is chosen because head rotations occur in 3D space, and self-occlusion can be naturally modelled within this space [Zhu *et al.* 2017b]. A 3D Morphable Model (3DMM) [Blanz and Vetter 1999] of the speaker's face is then constructed, which serves as a powerful statistical model and offers a convenient parametric representation. Since 3DMMs are typically constructed by performing Principal Component Analysis (PCA) on a collection of high-resolution laser scans of human faces [Booth *et al.* 2018], they are often expressed as:

$$S = \bar{S} + A_{id}\alpha_{id} + A_{exp}\alpha_{exp}. \tag{2.4}$$

Here, S represents the reconstructed 3D face, $\bar{S}$ represents the mean 3D face, A_{id} and A_{exp} denote the eigenbases corresponding to facial identity and expression respectively, and α_{id} and α_{exp} denote the corresponding identity and expression parameters respectively. Inspired by the literature [Zhu *et al.* 2015 2016], Cheng *et al.* [2020] defines the 3DMM as a combination of the Basel Face Model (BFM) [Paysan *et al.* 2009] and FaceWareHouse model [Cao *et al.* 2013]. Specifically, A_{id} and A_{exp} are obtained from the BFM and FaceWareHouse models, respectively.

To fit the reconstructed 3DMM to an input facial image, a weak perspective projection is performed to project the 3D model to the image plane as follows:

$$s_{2d} = fPR(\alpha, \beta, \gamma)(S + t_{3d}). \tag{2.5}$$

Here, s_{2d} is the 2D positions of the 3D points on the image plane, f is the scale factor, P is the orthographic projection matrix $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $R(\alpha, \beta, \gamma)$ is the 3×3 matrix constructed with pitch (α), yaw (β) and roll (γ) angles, and t_{3d} is the translation vector. By coalescing the parameters from Equation (2.4) and Equation (2.5) into $p = \{f, R, t_{3d}, \alpha_{id}, \alpha_{exp}\}$, the 3DMM fitting process may be defined as utilizing the 2D projection s_{2d} and input facial image s_{2dt} to recover p such that the error between the 2D projection and input facial image is minimized, i.e.:

$$\underset{f,R,t_{3d},\alpha_{id},\alpha_{exp}}{\arg\min} \; \|s_{2dt} - s_{2d}\|. \tag{2.6}$$

Their solution employs the state-of-the-art 3D Dense Face Alignment (3DDFA) solution [Zhu *et al.* 2017b] which trains a fully convolutional network to regress the parameters in a cascaded manner. Choosing 3DDFA is reasonable because it offers accuracy and robustness even for head poses up to profile views. Additionally, they use the Facial Alignment Network (FAN) [Bulat and Tzimiropoulos 2017b] to detect 68 facial landmarks to provide a better initialization for 3DMM fitting. FAN is a pioneering solution that uses a CNN for facial alignment, offering accuracy, robustness to self-occlusion, and the ability to perform 2D and 3D facial alignment. The aspect that makes this possible is network architecture, composed of four stacked hourglass networks [Newell *et al.* 2016] with the hierarchical, parallel, and multi-scale block of Bulat and Tzimiropoulos [2017a].

Unlike other pose augmentation solutions [Masi *et al.* 2016; Zhao *et al.* 2017; Deng *et al.* 2018], the pipeline of Cheng *et al.* [2020] preserves the background surrounding the speaker's face, significantly improving the naturalness of the synthesized result. This result is achieved by

using the fitted 3DMM to help estimate the depth information of the entire image by defining a set of *anchor points* i.e., points at which the depth is estimated. The process occurs in three stages i.e., (1) anchor points are defined along the periphery of the face, (2) anchor points enclose a larger region surrounding the speaker's face, including the head back, ears, and neck, and (3) anchor points along the image boundary are defined to include depth information for the background. Figure 2.4 illustrates the process of defining anchor points. The collection of anchor points is then triangulated using the Delaunay algorithm [Lee and Schachter 1980] to produce a 3D mesh, which is rotated in 3D space to generate the non-frontal result, including the speaker's head and surrounding background, as shown in Figure 2.5.

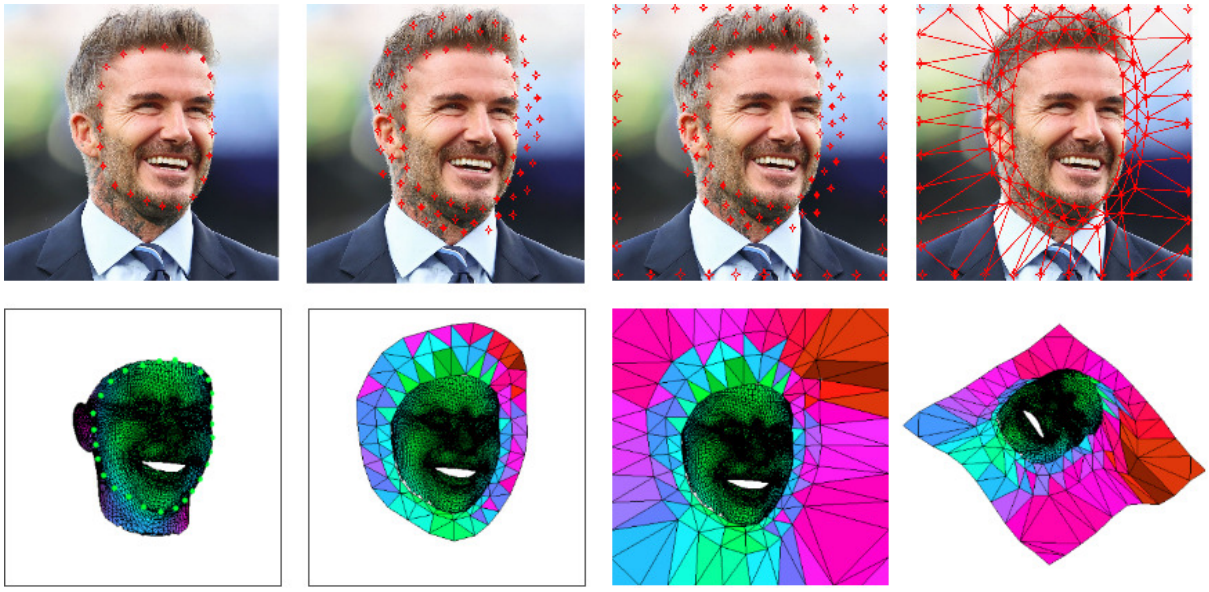


Figure 2.4: The process of forming the 3D mesh that is rotated to produce the non-frontal result.

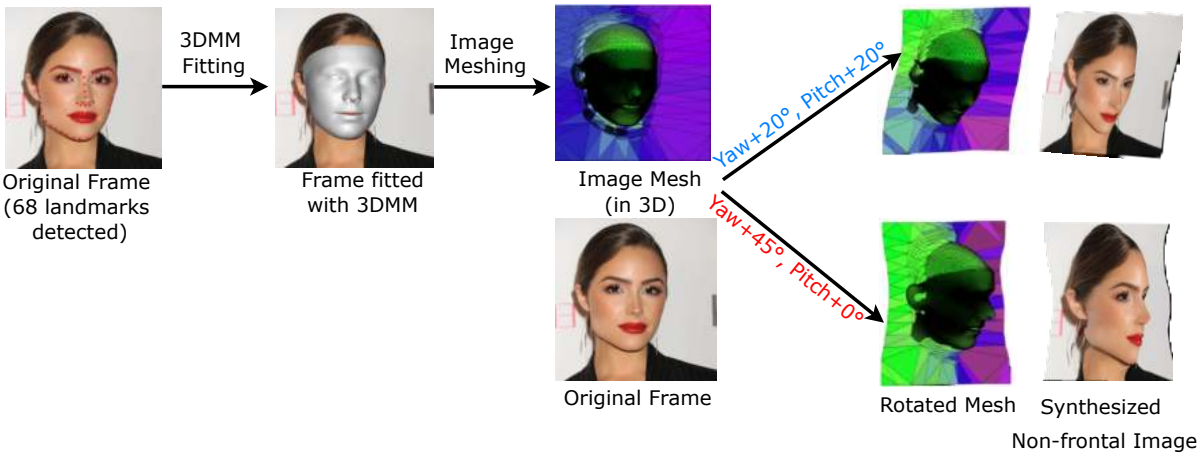


Figure 2.5: The pose augmentation pipeline.

2.3.3 Literature Review

This section provides a brief overview of video-driven visual dubbing solutions [Garrido *et al.* 2015; Thies *et al.* 2016; Kim *et al.* 2018 2019b] and discusses the rationale behind this class of solutions. We highlight their pitfalls and how the presented solution attempts to address them. These shortcomings naturally emphasize the need for audio-driven visual dubbing solutions [Suwajanakorn *et al.* 2017; KR *et al.* 2019; Thies *et al.* 2020; Prajwal *et al.* 2020].

2.3.3.1 Video-driven Visual Dubbing Solutions

The premise behind video-driven visual dubbing solutions is to take a video of the target actor (the actor to be dubbed) and a video of the dubbing actor uttering the content as inputs, and to construct a 3D Morphable Model (3DMM) [Blanz and Vetter 1999] of both actors. Dubbing is then performed through *deformation transfer*, where the mouth region of the dubbing actor's 3DMM is transferred to the target actor's 3DMM, as shown in Figure 2.6. Thies *et al.* [2016] introduced the first real-time solution, which relies on live footage from a commodity RGB camera as the dubbing source, along with an innovative facial performance capture method using non-rigid model-based bundling for accurate facial reconstruction.

When performing deformation transfer, Garrido *et al.* [2015] attempts to add fine-scale detail to the amalgamated 3DMM by transferring skin details from the frame in the target sequence that most closely corresponds to the resulting 3DMM. Additionally, an *audio analysis* is performed to enforce mouth closure for bilabial consonants, improving the realism of the result. Upon assessing the achieved results, a lack of temporal coherence is evident, and the results inadvertently exhibit the talking style of the dubbing actor instead of that of the target actor. We hypothesize that these issues follow as a consequence of producing the result by resequencing frames (instead of synthesizing the appropriate mouth shape), as well as deformation transfer being a naïve approach to visual dubbing.

Kim *et al.* [2018] presented the first solution that enabled full control of the target actor's head (i.e., rigid head pose, facial expression, and eye motion) based on the behaviour of the dubbing actor. To perform dubbing, the illumination and identity parameters of the target actor's 3DMM are augmented with the pose, expression, and eye coefficients of the dubbing actor to produce a new 3DMM. At its core, a conditional rendering-to-video translation GAN [Goodfellow *et al.* 2014] based on U-Net [Ronneberger *et al.* 2015] is used. Specifically, the network is conditioned on a short sequence of colour renderings of the new 3DMM, correspondence maps, and eye gaze images which improve the temporal stability of the results achieved. Kim *et al.* [2019b] deviate from the naïve deformation transfer method and attempts to preserve the talking idiosyncrasies of the target actor which they believe forms part of a person's *brand*, particularly for prominent figures such as actors and politicians. This solution uses the 3DMM coefficients of the target actor as input to a style-preserving recurrent GAN consisting of LSTM [Hochreiter and Schmidhuber 1997] units and is trained in a self-supervised fashion using cycle-consistency constraints [Zhu *et al.* 2017a]. The final result is produced using a layered neural face renderer.

Except for the solutions proposed by Kim *et al.* [2018 2019b], the majority of video-driven video dubbing solutions fail to achieve a natural viewing experience due to unsatisfactory

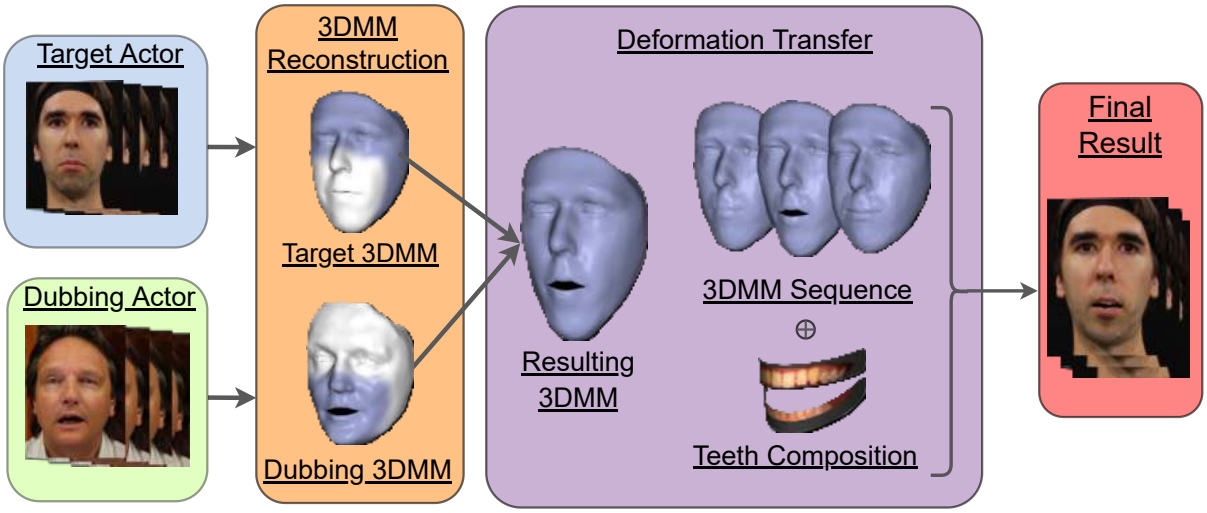


Figure 2.6: Illustration showing the process of deformation transfer followed by video-driven visual dubbing solutions. Illustration adapted from Garrido *et al.* [2015].

visual quality and lip-sync accuracy. Moreover, we consider these solutions unsuitable for real-world applications for the following reasons:

- A video of the dubbing actor uttering the dubbing content is required which is an inconvenient data requirement. This process entails conducting several recordings in a studio setting such that the dubbing actor is clearly visible and modulates their mouth movements according to the semantics (emotion) of the spoken content [Garrido *et al.* 2015].
- The majority of solutions are *model-based* and therefore:
  - require at least two to five of video footage to construct an accurate 3DMM for each actor and this data requirement may not be easily satisfied;
  - are required to resort to inelegant methods, such as teeth proxies, to model the speaker's mouth interior which is not modelled by most 3DMMs [Garrido *et al.* 2015; Suwajanakorn *et al.* 2017]; and
  - need to be re-trained for each new target actor since the facial reconstruction process for each actor renders the solution to be person-specific. This consequence significantly hinders the adoption of these solutions, particularly for real-world purposes such as dubbing movies which typically contain several actors.

2.3.3.2 Audio-Driven Visual Dubbing Solutions

Audio-driven visual dubbing (ADVD) solutions take a more natural approach to dubbing since dubbing audio is significantly more attainable compared to dubbing videos. The notion of ADVD was pioneered by Bregler *et al.* [1997], who employed a Hidden Markov Model (HMM) to label the original audio based on phonemes, which is used to segment the video frames into the corresponding visemes. An annotated database is then constructed to map phonemes to their corresponding visemes. The resultant dubbing video is produced by labelling the

dubbing audio, indexing the database, and retrieving the corresponding video frames with the appropriate mouth shapes. Unfortunately, due to the difficulty of performing ADVD, this naïve approach was unable to achieve acceptable results.

With the advent of deep learning, researchers have gained the necessary computational resources and knowledge to tackle more complex problems. Nakashima *et al.* [2020] first label the original and dubbing audio segments with their corresponding phoneme labels. They then use a volumetric regression network [Jackson *et al.* 2017] to produce a depth map, which is fitted to a 3DMM for each frame. Similar to most video-driven solutions, they retrieve the mouth region of the 3DMM whose audio label most closely corresponds to the current phoneme label (from the dubbing audio) and superimpose it onto the 3DMM of the current frame to perform dubbing. Due to the poor levels of visual quality and lip-sync accuracy achieved by solutions that perform frame resequencing, resulting in a lack of temporal stability, the majority of solutions that followed began to synthesize the corresponding mouth shapes with respect to the dubbing audio.

A prominent solution that attracted attention to the field of visual dubbing is Suwajanakorn *et al.* [2017] due to the photorealistic results achieved. This solution performs dubbing in two steps i.e., (1) utilizing an LSTM [Hochreiter and Schmidhuber 1997] to map MFCC audio features to a sparse time-varying mouth shape and (2) using the mouth shape to guide the synthesis of the mouth texture. Despite its impressive results, the effectiveness of this solution requires further consideration. Specifically, it is only applicable to speakers with an abundance of publicly available talking-face footage. As a result, the authors trained their solution on 17 hours of footage of Barack Obama, which is not readily available for most speakers. Secondly, the solution achieves a poor runtime performance of five seconds to render a single frame.

The approach by Thies *et al.* [2020] also consists of two phases i.e., a generalized phase and a specialized phase. The generalized phase, shared by multiple speakers, includes *Audio2ExpressionNet*, a network with a per-frame estimation network and a temporal network that maps audio to expression (3DMM) coefficients. In contrast, the specialized phase constructs a speaker-specific 3DMM and learns the speaker's idiosyncrasies from a two to three-minute input video. The coefficients produced by the generalized phase are then used to drive the speaker-specific 3DMM in the specialized phase to perform dubbing. A similar approach is taken by Wen *et al.* [2020] and Song *et al.* [2022], where the input dubbing audio is mapped to expression parameters which get augmented with the remainder of the original 3DMM coefficients. A neural renderer is then used to render the resulting 3DMM and produce the final result. Despite achieving satisfactory results, these solutions require constructing a 3DMM of the speaker, which subjects them to the same limitations as the aforementioned model-based video-driven solutions and, therefore, makes them unsuitable for general purposes.

The revolutionary solution of Chung *et al.* [2017] was the first to enable speaker-independent visual dubbing i.e., capable of dubbing speakers not present in the training data. The authors achieved this by employing an encoder-decoder network comprised of an identity encoder, audio encoder, and face decoder. Despite the solution achieving unsatisfactory visual quality, the significant advancement that this solution achieved resulted in the solution serving as the basis for several state-of-the-art solutions. KR *et al.* [2019] propose *LipGAN*, which attempts to address the complete dubbing problem. In addition to dubbing the speaker's mouth movements, the solution obviates the need for dubbing audio and instead automatically translates

the video to the desired language. For the visual dubbing component, the solution adapts the work of Chung *et al.* [2017] to a GAN framework and begins by masking the mouth region as a pose-prior and performs dubbing by inpainting the masked region with the appropriate mouth shape.

Due to the potential of LipGAN, Prajwal *et al.* [2020] introduced *Wav2Lip*, which integrates several design enhancements, including a pre-trained lip-sync discriminator. Additionally, the generator and discriminator ingest five contiguous frames instead of a single frame, thus, allowing for motion modelling. Wav2Lip shows promising results in visual quality and lip-sync accuracy. However, the results occasionally exhibit an unpleasant *box effect* in the mouth region, despite the use of a visual quality discriminator. Wang *et al.* [2022] integrated spatial and channel attention modules into Wav2Lip, but the resulting improvement were negligible. Yang *et al.* [2020] utilized an additional (reference) encoder and a temporal aggregation module in the generator network, along with the discriminator setup of Clark *et al.* [2019]. After assessing the three representative videos released, the solution appears to achieve photorealistic results. However, this may not be a true reflection of the solution's capabilities since the videos are recorded under ideal (studio) conditions, and the solution is not made publicly available.

2.4 Conclusion

This chapter presented an overview of the fundamental concepts that underpin the presented solution. Moreover, a summary of the works on which the presented solution is based, i.e., SyncNet [Chung and Zisserman 2016b] and the pose augmentation solution of Cheng *et al.* [2020], was presented. Lastly, a survey of previous visual dubbing solutions was provided, highlighting the rapid progress of the field. Specifically, the unnatural results achieved by video-driven visual dubbing solutions, as well as the inconvenient requirement for a video of the dubbing actor uttering the dubbing content, has led to a series of model-based ADVD solutions being introduced. The following chapter presents a holistic overview of our entire solution.

Chapter 3

Solution Overview

3.1 Introduction

This chapter expands on the discussion presented in Chapter 1, emphasizing the significance of addressing the visual dubbing problem, outlining the challenges associated, and motivating the approach taken. Additionally, we briefly explain our strategies to enhance visual quality and lip-sync accuracy while extending ADVD solutions to support high-resolution videos and videos with arbitrary head poses.

3.2 Traditional Solutions

When analyzing the two most widely adopted video localization techniques, i.e., *subtitling* [Cintas and Remael 2014] and *audio dubbing* [Heiss 2004], it is evident that both solutions are outdated and achieve sub-optimal viewing experiences. Subtitles tend to obstruct the viewer's view of the on-screen content, divert their attention from the visuals to the subtitles, and increase the viewer's cognitive load [Mailhac 2000]. Audio dubbing, on the other hand, addresses the shortcomings of subtitling but introduces its own set of challenges. The most notable issue viewers encounter with dubbed videos is the discrepancy between the dubbed audio and the speaker's lip movements [Koolstra *et al.* 2002]. We argue that both solutions will soon become obsolete due to the rapid advancements made in generative modelling.

3.3 Why Audio-Driven Visual Dubbing?

Audio-driven visual dubbing (ADVD) involves taking a talking-face video and a dubbing audio segment as inputs to produce a dubbed video, where the speaker's mouth movements appear as if they are uttering the input dubbing audio. Opting for an audio-driven approach is advantageous as dubbing audio is more accessible compared to dubbing videos. ADVD ensures the preservation of timing and semantics of the original content. Since this method only manipulates the speaker's mouth movements relative to the dubbing audio, it eliminates the obstruction of the viewer's view (unlike subtitles) and the discrepancy between mouth movements and audio (unlike audio dubbing), creating a seamless viewing experience. The

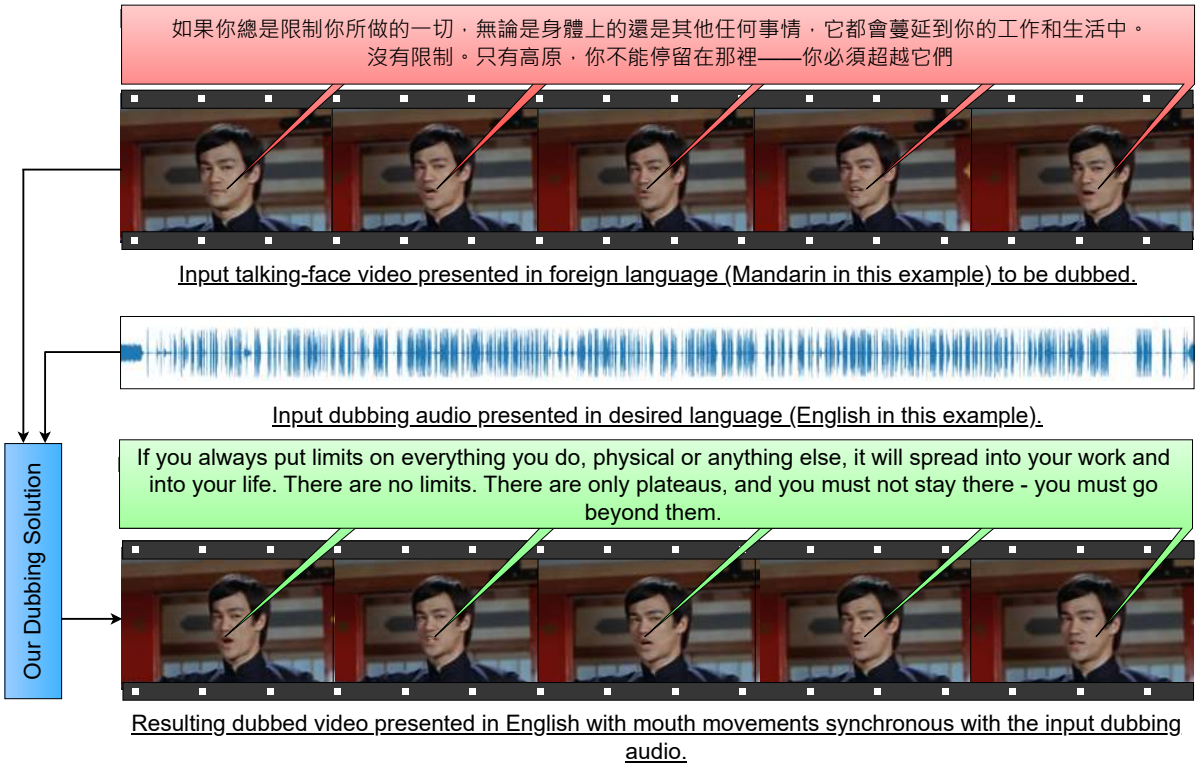


Figure 3.1: Example showing how the presented ADVD solution works when visually dubbing a video of Bruce Lee, originally presented in Mandarin, to English.

goal is to make the dubbed videos indistinguishable from those originally presented in the viewer's native language. Finally, ADVD offers the advantage of retrospective dubbing for videos that were previously dubbed using audio dubbing. This capability is possible because the dubbing audio is already available, and the only requirement is to adjust the speaker's mouth movements to synchronize with the dubbing audio. Figure 3.1 provides an illustrative example of this process for the movie *Fist of Fury*, released in 1972.

3.4 Challenges

Despite the concept of ADVD being straightforward in theory, there are several challenges to overcome to achieve a natural viewing experience in practice. Firstly, this process involves using low-dimensional data (i.e., dubbing audio) to drive higher-dimensional data (i.e., the input video) [Suwajanakorn *et al.* 2017]. Secondly, since the input dubbing audio is used to determine the appropriate mouth shape, this requires establishing a mapping between *phonemes* (the smallest unit of speech) and *visemes* (the visual counterpart of phonemes, in our case, mouth shapes). Unfortunately, this mapping is not one-to-one but is many-to-one instead [Kim *et al.* 2019b]. For instance, bilabial consonants (/b/, /m/, and /p/) all map to the same viseme, as shown in Figure 3.2. Lastly, achieving a natural viewing experience requires high

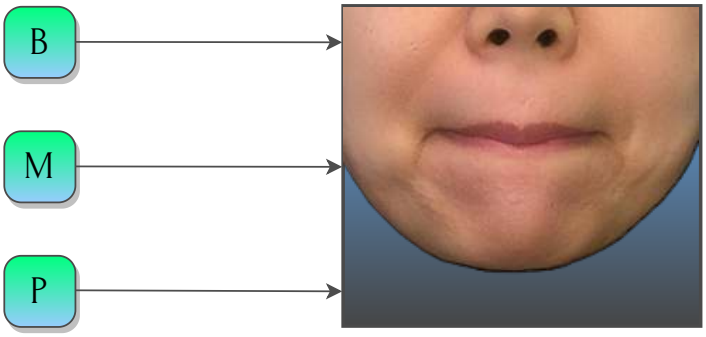


Figure 3.2: Bilabial consonants (/b/, /m/ and /p/) all map to the same viseme.

measures of visual quality and lip-sync accuracy; however, in practice, these two properties are in contention, where optimizing one often compromises the other [Prajwal *et al.* 2020].

3.5 Visual Dubbing vs One-shot Talking-Face Generation Methods

An essential requirement when addressing the visual dubbing problem is to ensure that the speaker's lip movements are morphed in a manner that preserves all other elements of the video as much as possible, such as head pose, lighting conditions, and background. In practice, this can be achieved by acting on each frame of the video and producing the dubbed result on a frame-by-frame basis. This constraint emphasizes the need for temporal consistency when dealing with the ADVD problem. It is important to note that there is a crucial distinction between visual dubbing and *one-shot talking-face generation* [Doukas *et al.* 2021; Zhou *et al.* 2021; Wang *et al.* 2021a]. The latter involves extracting a single identity reference frame from the video and generating the dubbed result based on this single frame. However, this paradigm is impractical for real-world purposes, as these solutions are unable to take scene dynamics (e.g., variations in head pose, lighting conditions, and background) into account.

An example of a one-shot talking-face generation solution is ATVGnet [Chen *et al.* 2019], which employs landmarks as an intermediary representation between the driving audio and the resulting video. Instead of establishing a direct mapping from audio to video, the solution first estimates facial landmarks from the input audio and uses these estimated landmarks to guide the synthesis process. Moreover, the solution incorporates an attention mechanism to enhance the quality of generation. However, as illustrated in Figure 3.3, this approach fails to capture scene dynamics and, consequently, lacks naturalness when compared to visual dubbing solutions.

3.6 Prior Solutions

From the survey of visual dubbing solutions presented in Section 2.3.3, it became evident that the majority of solutions were speaker-specific, primarily due to their reliance on 3DMMs (3D Morphable Models). In theory, these solutions possess the advantage of "memorizing" intricate visual attributes, such as lip shape, skin colour, and speaking style, thereby enabling them

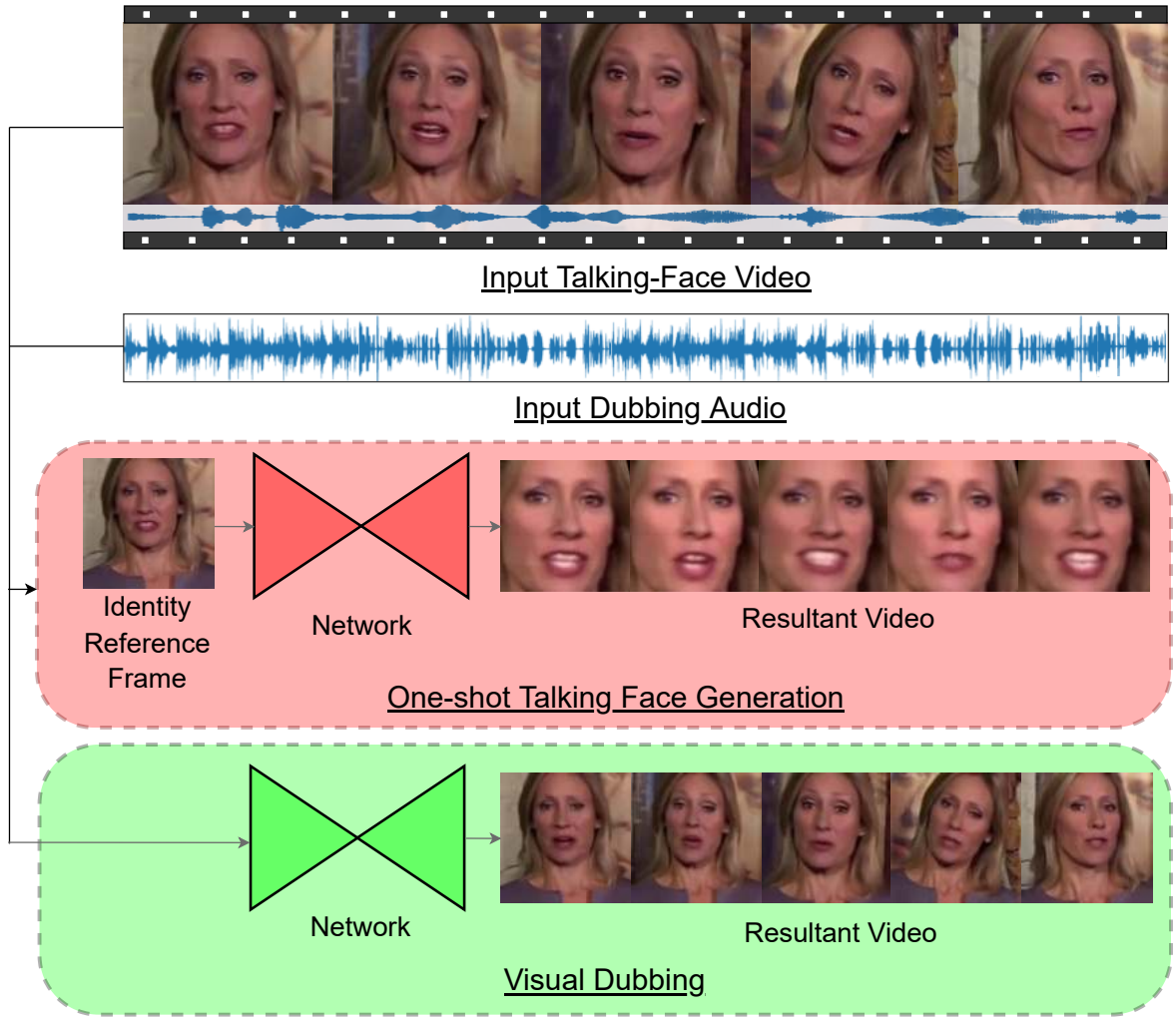


Figure 3.3: One-shot talking-face generation compared to visual dubbing. Notice that, unlike one-shot talking-face generation solutions, visual dubbing preserves as much of the original scene as possible such as background, head pose, lighting, etc.

to achieve remarkable levels of visual quality. Unfortunately, the applicability of these solutions [Karras *et al.* 2017a; Nakashima *et al.* 2020; Thies *et al.* 2020] is significantly hampered due to the necessity of retraining the solution for each new speaker, and the requirement of two to five minutes of video footage to construct an accurate 3DMM of the speaker, which may not always be possible.

3.7 Our Presented Solution

In pursuit of extending ADVD solutions to account for high-resolution videos and videos containing arbitrary head poses, we draw inspiration from solutions [KR *et al.* 2019; Prajwal *et al.* 2020; Yang *et al.* 2020] based on Chung *et al.* [2017]. These solutions involve training a large-scale speaker-independent model, i.e., a model that is invariant to the speaker being dubbed, as well as to the audio used to drive the dubbing process, and is trained using footage of multiple speakers. The key aspect that makes these solutions speaker-independent is the process of

producing the dubbed video by *inpainting* the speaker's mouth region with the appropriate mouth shape, as determined by the dubbing audio. This approach eliminates the need for any intermediate representations such as 3DMMs or landmarks and directly utilizes the dubbing audio to drive the dubbing process. When designing a speaker-independent solution, it is crucial to preserve the speaker's identity while achieving high levels of visual quality and lip-sync accuracy.

Inspired by the effectiveness of recent state-of-the-art ADVD solutions [Prajwal *et al.* 2020; Yang *et al.* 2020] and the superior performance of GANs [Goodfellow *et al.* 2014] at various generative tasks [Isola *et al.* 2017; Brock *et al.* 2018; Karras *et al.* 2019], we present a GAN-based approach. The solution is trained in a self-supervised setting, leveraging the cross-modal supervision present in the audio and visual streams of a video. This approach allows us to harness the abundance of talking-face videos available, as discussed in Chapter 4. Due to the innate ability of GANs to achieve high measures of visual quality, we begin by considering how to enforce accurate lip-sync. Prajwal *et al.* [2020] showed that standard loss functions, such as L_1 loss, are incapable of enforcing accurate lip-sync due to being a weak and non-specific measure of lip-sync. Inspired by their findings, we employ a *pre-trained lip-sync discriminator*, which is discussed in Chapter 5. Prajwal *et al.* [2020] justified their design choice of pre-training their lip-sync discriminator by achieving a claimed off-sync detection accuracy of 91% on the LRS2 test dataset [Afouras *et al.* 2018a], compared to the 73% accuracy achieved when trained adversarially, resulting in a 19.7% increase in accuracy. Chapter 6 presents our deep residual U-Net generator, trained on the LRS2 dataset, which strives to improve the visual quality and lip-sync accuracy achieved. An overview of the solution is presented in Figure 3.4.

Our first endeavour towards extending the capabilities of ADVD solutions begins with the observation that, except for Yang *et al.* [2020], all existing solutions are trained using a reso-

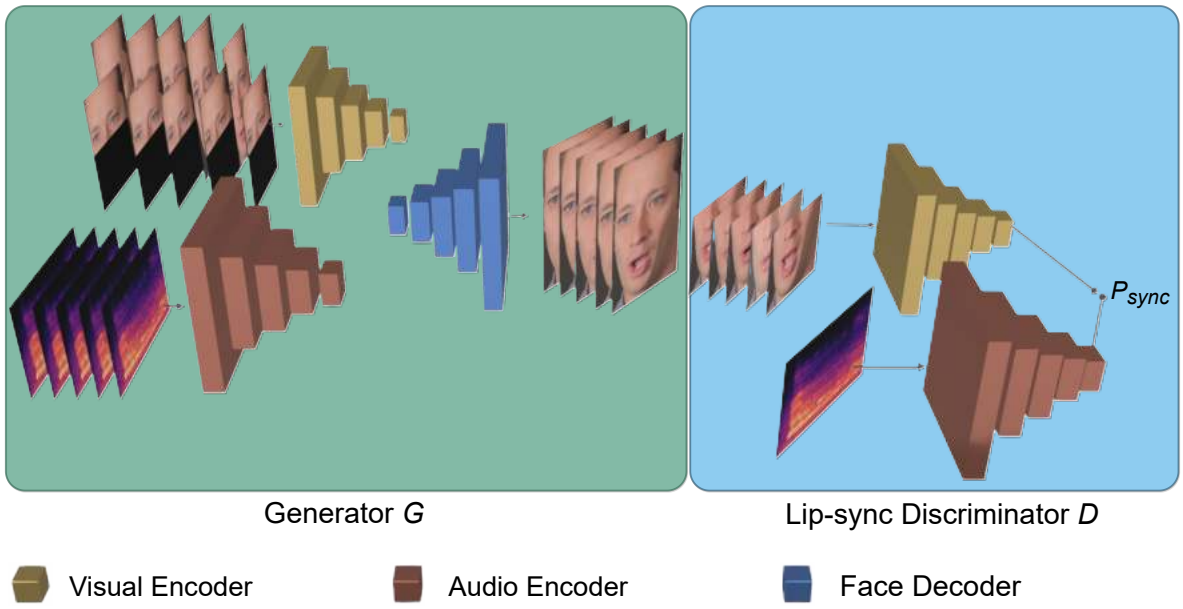


Figure 3.4: Overview of the presented solution.

lution smaller than 120×120. This resolution is considerably smaller than that of real-world videos and much smaller compared to resolutions used for other generative tasks, such as image-to-image translation [Isola *et al.* 2017] and text-to-image synthesis [Xu *et al.* 2018; Zhu *et al.* 2019], where resolutions such as 256×256 are considered the norm. The deficiency of high-resolution ADVD solutions may be attributed to the lack of large-scale, high-quality data suitable for addressing the visual dubbing problem. In response, we present one of the first attempts to employ high-resolution training data for the visual dubbing problem by extending our solution to ingest inputs with a resolution of 192×192. Chapter 7 elaborates on this high-resolution implementation, and Section 4.7 presents the comprehensive *data-cleaning pipeline* developed to pre-process the AVSpeech dataset [Ephrat *et al.* 2018] for the high-resolution implementation.

Our second endeavour towards enhancing the capabilities of ADVD solutions arises from the observation that existing solutions have primarily been designed to account for frontal and near-frontal faces. Similar to other facial applications, such as facial recognition, emotion recognition, and lip-reading, the effectiveness of these solutions diminishes as the head pose becomes increasingly non-frontal (rotated along the yaw axis) [Blanz *et al.* 2005; Zheng *et al.* 2010; Cheng *et al.* 2020]. This limitation stems from the inherent bias in the datasets typically employed to tackle the visual dubbing problem which predominantly contain frontal and near-frontal faces. Instead of the resource-intensive process of curating a non-frontal talking-face dataset, we perform *pose augmentation* [Cheng *et al.* 2020]. This innovative approach involves leveraging existing frontal and near-frontal footage to synthesize footage at arbitrary non-frontal head poses. The novelty of this approach is that it avoids any modifications to the solution itself, thus enabling prior solutions to also become pose-invariant by applying the aforementioned pose augmentation, as detailed in Chapter 8.

3.8 Conclusion

This chapter provided a concise overview of the presented solution, emphasized the importance of addressing the ADVD problem, and mentioned the challenges involved. The subsequent chapters elaborate on each major idea mentioned above. Chapter 4 offers a comprehensive survey of the data landscape. Chapters 5 and 6 provide detailed analyses of the proposed pre-trained lip-sync discriminator and generator network, respectively. Chapter 7 discusses the extension of the solution to handle high-resolution training data, while Chapter 8 explains how the solution is made to be pose-invariant. Lastly, Chapter 9 discusses the applications and ethics of the solution, whereas Chapter 10 concludes the document by outlining potential future directions, as well as a summary of salient points presented in this study.

Chapter 4

Data

4.1 Introduction

The preceding chapters have provided a comprehensive representation of the ADVD field, along with an overview of the presented solution and the ways it aims to advance the field. We now delve into each major aspect of our solution, starting with a critical component of any deep-learning approach – the data. First, the criteria used to evaluate the appropriateness of existing datasets is presented, followed by the data-cleaning pipeline developed to pre-process the AVSpeech dataset [Ephrat *et al.* 2018] into a suitable form for addressing the visual dubbing problem.

4.2 Ideal Data Properties

Below, we list some of the desired data properties used to evaluate the appropriateness of audio-visual datasets. This synopsis guides our decision on which dataset(s) to use for training the presented solution:

- *Large speaker diversity*: The presented solution takes inspiration from recent dubbing solutions [Chung *et al.* 2017; KR *et al.* 2019; Prajwal *et al.* 2020; Yang *et al.* 2020] that enhance the capabilities of ADVD by training a speaker-independent model i.e., a large-scale unified model trained on footage of multiple speakers that is capable of dubbing any speaker. Consequently, it is essential to utilize a dataset that contains footage of a vast array of speakers with varying ages, genders, and ethnicities among other aspects.
- *Assorted environment & recording conditions*: Since talking-face videos can be recorded in any environment, the dataset should include footage of *in-the-wild* (*unconstrained*) conditions with variations in lighting, head pose, and background [Afouras *et al.* 2018a; Chung *et al.* 2018]. This is in contrast to datasets recorded in *constrained* environments, such as recording studios, where the aforementioned factors are carefully controlled [Cooke *et al.* 2006; Harte and Gillen 2015]. Furthermore, since videos can take on various forms, such as interviews, instructional (how-to) videos, and sitcoms among others, it is desirable to further increase the data variability by including videos recorded under various recording contexts.

- *Multilingual*: Since visual dubbing is a solution for localizing foreign videos to a desired language [Chaume 2020b], this process inherently involves transitioning between various languages [Heiss 2004]. Several solutions [KR *et al.* 2019; Prajwal *et al.* 2020] are considered to be language-independent, despite being trained on an English-only dataset. This follows due to the overlap of phonemes and visemes that the English language shares with other languages. However, we argue that this approach does not make a solution language-independent in the true sense, as there may exist phonemes and/or visemes in other languages that do not exist in English (e.g., the *ge* sound of the Afrikaans language). Consequently, we expect that utilizing a multilingual dataset would improve the lip-sync accuracy achieved since this would expose the solution to a broader range of phonemes and visemes. Additionally, collecting a multilingual dataset would entail gathering footage from multiple speakers of various nationalities, inadvertently encompassing a wide variety of accents and talking styles.
- *High-Quality Data*: The above-mentioned properties all pertain to improving the robustness of the solution, however, as with any machine-learning problem, the use of high-quality data is essential. In the context of visual dubbing, this entails the dataset being large-scale, the videos contained being of sufficiently high resolution, and the audio being sufficiently clean from background degradation, noise, and echo. Furthermore, Prajwal *et al.* [2020] advise that training data should be *sync-corrected* when addressing the ADVD problem. This involves passing each talking face video as input to the SyncNet network [Chung and Zisserman 2016b] as a pre-processing step to attain an AV offset measure used to eliminate any minor offset between the audio and visual streams. This offset may be caused due to the recording equipment used during acquisition or introduced during data transmission.

4.3 Existing Audio-Visual Datasets

The GRID corpus [Cooke *et al.* 2006] was acquired in a controlled environment in which a fixed six-word sentence structure (of the form <command:4> <color:4> <preposition:4> <letter:25> <digit:10> <adverb:4>, where the number of choices for each component is indicated) was imposed. For each of the 34 speakers, 1000 recordings were made from a frontal view. The TCD-TIMIT dataset [Harte and Gillen 2015] contains footage of 62 speakers, yielding 6913 phonetically rich sentences recorded from a frontal and 30° angle. Since both datasets are small-scale, recorded in a studio setting, and impose a fixed vocabulary, neither of these datasets was suitable for our purposes.

The LRS2 dataset [Afouras *et al.* 2018a] is widely adopted when addressing the visual dubbing problem due to its variability of speakers and environment conditions. It contains over 200 hours of footage extracted from (English) BBC television programs, encompassing various genres such as dramas, comedies, and talk shows, increasing the dataset's diversity. However, the resolution of all videos is limited to 160×160, which might be considered too low for practical purposes. On the other hand, the LRS3 dataset [Afouras *et al.* 2018b] consists of over 400 hours of English TED and TEDx videos, offering a wide range of identities. Despite its appeal for various audio-visual tasks, the dataset proves sub-optimal for our objectives due to the limited variability in environment, recording conditions, and a vocabulary that is biased towards typical public-speaking topics featured in TED talks.

Dataset	# of Identities	# of Videos	# of Hours	Resolution	Videos Unconstrained?	Vocabulary Unconstrained?	Multilingual?	Sync Corrected?
GRID [Cooke *et al.* 2006]	34	34K	27.5	360×288	✗	✗	✗	✗
CREMA-D [Cao *et al.* 2014]	91	7442	11.1	960×720	✗	✗	✗	✗
TCD-TIMIT [Harte and Gillen 2015]	62	6913	11.1	–	✗	✗	✗	✗
LRW [Chung and Zisserman 2017]	–	539K	173	256×256	✓	✗	✗	✓
LRS2 [Afouras *et al.* 2018a]	–	144K	224.5	160×160	✓	✓	✗	✓
LRS3 [Afouras *et al.* 2018b]	9545	165K	438	224×224	✓	✓	✗	✓
VoxCeleb2 [Chung *et al.* 2018]	6112	150K	2400	720p~1080p	✓	✓	✓	✗
Talking-Head-1KH[1] [Wang *et al.* 2021b]	–	180K	1000	720p~1080p	✓	✓	✓	✗
DeepMind[2] [Yang *et al.* 2020]	464K	3M	3130	720p~1080p	✓	✓	✓	✗
AVSpeech [Ephrat *et al.* 2018]	150K	2.8M	4700	720p~1080p	✓	✓	✓	✗

Table 4.1: Summary of statistics of several audio-visual datasets that may be used when addressing the visual dubbing problem.

Table 4.1 presents a summary of key statistics for potential audio-visual datasets applicable to the visual dubbing problem. Based on our analysis, our goal is to select a dataset that strikes

[1]Was not made publicly available at the time of our development period.
[2]Not made publicly available.

the best balance among the desired data properties mentioned earlier. Among the datasets previously used for visual dubbing, both LRS2 [Afouras *et al.* 2018a] and LRS3 [Afouras *et al.* 2018b] stand out as promising candidates. Both datasets are large-scale, sync-corrected, and offer diverse speakers, environment/recording conditions, and vocabulary (encompassing various phonemes and visemes). The LRS2 dataset is selected for our study due to its extensive size, diverse video types (in contrast to the LRS3 dataset which is primarily composed of public-speaking videos), and an unbiased vocabulary. A brief overview of the LRS2 dataset is presented below.

4.4 The LRS2 Dataset

The LRS2 dataset [Afouras *et al.* 2018a] is divided into four partitions: *pre-train*, *train*, *validation*, and *test*. The *pre-train* partition consists of talking-face videos with partial and multiple sentences, while the *train* partition contains only single full sentences. Videos in the *pre-train* partition have durations ranging from 1.12 to 15.76 seconds, with an average duration of 5.34 seconds, whereas videos in the *train* partition range from 0.56 to 6.22 seconds, with an average duration of 2.25 seconds. Notably, the *pre-train* partition is significantly larger than the *train* partition, comprising 96,318 videos totalling 195 hours, while the *train* partition consists of 45,839 videos totalling 29 hours. For most experiments, we use the *train* partition, but we also conduct experiments using the *pre-train* partition. The *validation* and *test* partitions are designed to prevent any overlap of identities with the *pre-train* or *train* partitions, thus, preventing data leakage [Gutierrez 2014]. Figure 4.1 illustrates a few samples extracted from the LRS2 dataset.

Figure 4.1: Samples extracted from the LRS2 dataset.

With regards to the specific data requirements of the presented ADVD solution, all videos are required to have the following: (1) FPS = 25, (2) an audio sampling rate of 16000Hz, and (3) be sync-corrected. Fortunately, the LRS2 dataset [Afouras *et al.* 2018a] satisfies all three requirements in its original form. To preprocess the dataset, the S^3FD face detector [Zhang *et al.* 2017b] is applied to obtain a tight facial crop of the speaker for each frame, which is then resized to 96×96, as advised by the literature [KR *et al.* 2019; Prajwal *et al.* 2020].

4.5 Towards Utilizing a High-Resolution Dataset

While the LRS2 dataset [Afouras *et al.* 2018a] achieves the best trade-off among the previously outlined properties, it fails to satisfy two of our desired criteria i.e., the dataset only contains English videos (hence, is not multilingual) and all videos have a resolution of 160×160, which is considerably lower than real-world videos. When assessing the results achieved by existing solutions, the consequences of utilizing low-resolution datasets become apparent. These solutions are confined to low-resolution videos (typically smaller than 120×120) since photorealistic results cannot be achieved for high-resolution (real-world) videos. Despite the existence of several high-resolution datasets seemingly better suited for the visual dubbing problem, the field has not experimented much with these datasets due to the challenges synonymous with training high-resolution models, as discussed in Chapter 7. A brief overview of a few high-resolution datasets, along with the dataset used to train our high-resolution ADVD solution, is discussed next.

The VoxCeleb2 dataset [Chung *et al.* 2018] was originally designed for speaker recognition and consists of over one million utterances by 6,112 speakers extracted from YouTube videos. The dataset primarily consists of snippets from celebrity interviews at red-carpet events and outdoor stadiums among others. Despite its large scale and diverse speakers and languages, two potential concerns emerge. Firstly, the predominance of videos recorded in an interview setting may bias the vocabulary and, consequently, the phonemes and visemes included due to limited exposure to alternative recording conditions. Secondly, the dataset's creators disclose that the audio segments may contain degradation from laughter, overlapping speech, and other audio impurities. On the other hand, the recently introduced TalkingHead-1KH dataset [Wang *et al.* 2021b] contains 180,000 videos totalling 1000+ hours of footage extracted from YouTube videos and the Ryerson audio-visual dataset [Livingstone and Russo 2018]. Given its greater diversity of recording conditions and higher quality and resolution compared to the VoxCeleb2 dataset, the TalkingHead-1KH dataset serves as an ideal option for high-resolution audio-visual problems in general. Unfortunately, this dataset was not made publicly available at the time of our development period, necessitating us to explore alternative options.

The AVSpeech dataset [Ephrat *et al.* 2018] is composed of 290,000 YouTube videos, from which 2.7 million training and 180,000 testing excerpts (referred to as *sub-videos* hereon) are extracted. It includes footage of 150,000 speakers, totalling over 4,700 hours. The dataset encompasses various video types, such as interviews, public talks (TED Talks, religious discourses), how-to videos (technology, fitness, photography), and lecture videos (language, math, chemistry). This diversity covers a wide range of speakers, environments, recording conditions, and languages, as illustrated in Figure 4.2. The AVSpeech dataset outperforms other publicly available high-resolution datasets, such as VoxCeleb2 [Chung *et al.* 2018] and TalkingHead-1KH [Wang *et al.* 2021b], in satisfying the desired data properties mentioned previously. Moreover, the AVSpeech dataset features pristine audio without any interference from music, audience noise, or overlapping speech, leading to superior quality compared to the VoxCeleb2 dataset. Taking advantage of these benefits, we use the AVSpeech dataset to train our high-resolution visual dubbing solution, as detailed below.

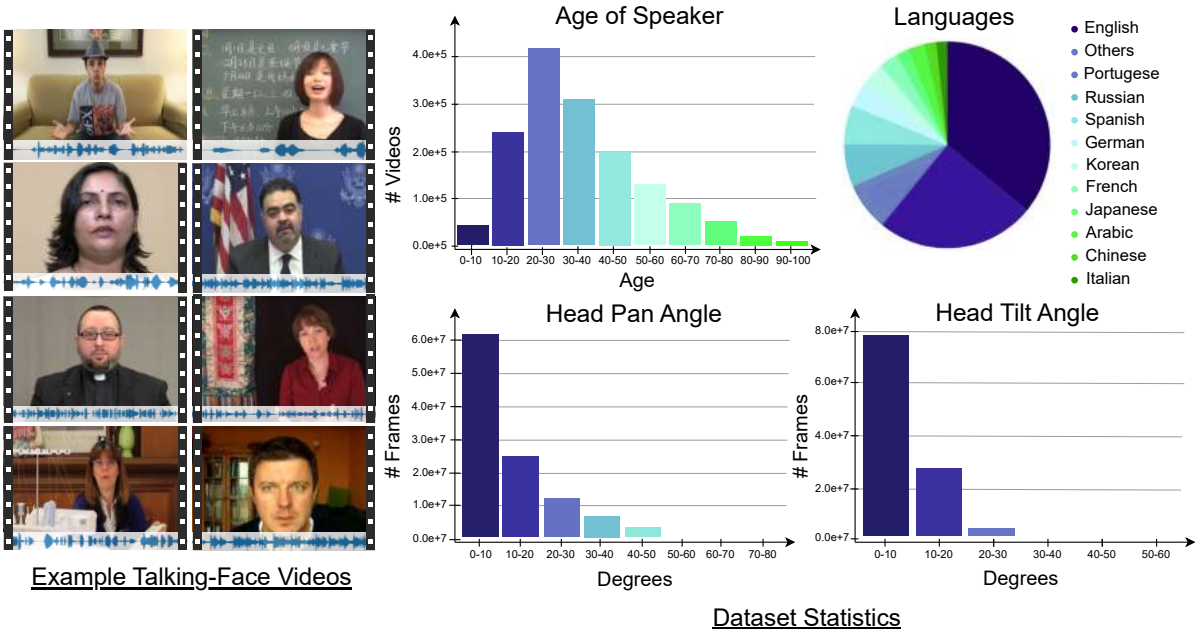


Figure 4.2: Samples extracted from the AVSpeech dataset along with a summary of statistics of the dataset.

4.6 The AVSpeech Dataset

Following the decision to use the AVSpeech dataset [Ephrat *et al.* 2018] to train the presented high-resolution implementation, it is crucial to first consider the dataset's appropriateness for the visual dubbing problem. This is achieved by examining the process through which most large-scale audio-visual datasets are compiled – automatically retrieving videos from YouTube based on specific keywords, from which several sub-videos are extracted [Nagrani *et al.* 2017; Chung *et al.* 2018; Yang *et al.* 2020]. Unfortunately, due to the noisy and unstructured nature of YouTube videos, coupled with this automated process, it is inevitable that sub-optimal videos are erroneously included in the dataset. Examples of sub-optimal sub-videos include those with poor audio quality, poor video quality, too small faces, and large audio-visual discrepancies, among others.

Based on the analysis of the AVSpeech dataset [Ephrat *et al.* 2018] provided, it is evident that the dataset in its original form is not suitable for the visual dubbing problem due to the minor imperfections of the sub-videos contained. The field has largely avoided working towards a high-resolution visual dubbing solution due to the arduous and time-consuming process of cleaning and formatting high-resolution datasets. To address this challenge, we present the first publicly available data-cleaning pipeline for high-resolution audio-visual datasets, specifically designed to clean and format the AVSpeech dataset. During the development of the data-cleaning pipeline, we adhered to three guiding principles, i.e.:

- *Little-to-no tolerance for defective data*: The premise behind the pipeline is to rectify sub-videos where possible, such as sync-correction, FPS, and audio sampling rate correction, among others, while discarding sub-videos that cannot be corrected, such as those containing too small faces or poor audio and visual quality. Given the enormous

size of the AVSpeech dataset, this allows for stringent data quality measures to be implemented. By setting high threshold values to determine the appropriateness of a video, the pipeline ensures that only data of the highest quality is retained.

- *Runtime efficiency*: Given the vast size of the dataset, it was essential to prioritize the runtime efficiency of the pipeline, ensuring the dataset could be processed within a reasonable timeframe. This requirement played a substantial role in shaping the pipeline's design, impacting everything from the selection of tools to the sequencing of operations.
- *High Configurability*: Our goal was to develop a highly configurable pipeline that can be easily adapted to meet the user's specific requirements. This flexibility enables the pipeline to be used for other audio-visual applications that previously faced challenges due to the lack of clean, large-scale, and high-resolution datasets. As a result, it accelerates progress towards high-resolution solutions for various audio-visual tasks.

Figure 4.3 illustrates an overview of our data-cleaning pipeline which is followed by a comprehensive description of each step of the pipeline which discusses the rationale behind the design choices made.

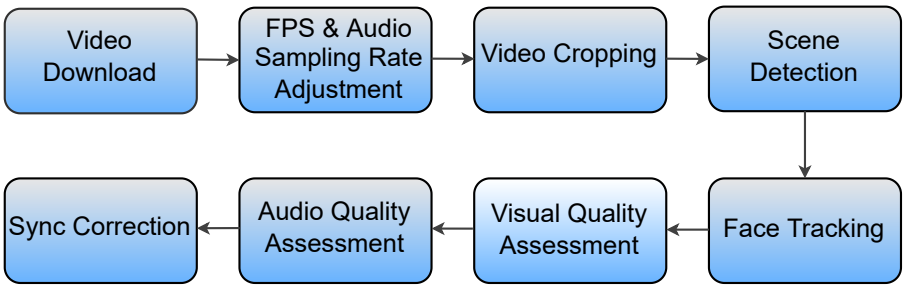


Figure 4.3: Overview of our data-cleaning pipeline.

4.7 AVSpeech Data-Cleaning Pipeline

4.7.1 Nature of the Dataset

The AVSpeech dataset is provided as two CSV files which contain the training and testing partitions respectively. Each CSV file details the following information for each sub-video within the relevant data partition:

YouTube video ID, start time, end time, X coordinate, Y coordinate

From the 290,000 videos collected from YouTube, 270,000 videos are used to construct the training partition from which 2.7M sub-videos are extracted, whereas the remaining 20,000 videos are used to compile the testing partition from which 180,000 sub-videos are extracted. The creators of the dataset partitioned the videos in a manner which eliminates any overlap of identities between the training and testing partitions, thus, preventing information leakage. To extract a sub-video from a video, the provided start and end timestamps are used to crop the relevant portion from the video. The provided $(X, Y) \in [0, 1]^2$ coordinates denote the centre point of the active speaker's face in the first frame of the sub-video. Note that the coordinates

Figure 4.4: Illustration showing how the details provided for a sub-video are used to pre-process the sub-video.

are normalized with respect to the sub-video's resolution where (0.0, 0.0) corresponds to the top left, and (1.0, 1.0) corresponds to the bottom right.

4.7.2 Video Downloads

The pipeline begins by downloading the required video from YouTube using the provided video ID and the *yt-dlp* tool[3]. For efficiency, all sub-videos extracted from the downloaded video are immediately processed, rather than processing them in the order they occur in the CSV file. The pipeline selects the highest resolution available to preserve maximal visual quality. If a video is no longer accessible due to deletion or being made private, it is discarded along with all sub-videos extracted from it, and the pipeline proceeds to the next video.

4.7.3 FPS and Audio Sampling Rate Adjustment

Akin to the LRS2 dataset [Afouras *et al.* 2018a], all videos are required to have an FPS of 25 and an audio sampling rate of 16KHz. If the original FPS is below 25, the video is discarded to avoid negatively affecting its temporal smoothness by artificially increasing the FPS. These

[3] https://github.com/yt-dlp/yt-dlp

operations are applied to the entire video, as each sub-video extracted from it must also meet these criteria, and doing so individually for each sub-video would be computationally inefficient. For all low-level operations, such as video cropping and changing the FPS and audio sampling rate, the *ffmpeg*[4] tool is used.

Thereafter, the following operations are performed on each sub-video extracted from the downloaded video.

4.7.4 Video Cropping

The sub-video is extracted from the pre-processed downloaded video using the provided start and end timestamps. If the duration of the sub-video is shorter than two seconds, it is discarded (for being too short), and this test is performed after each operation that potentially further crops the sub-video.

4.7.5 Scene Detection

Due to the possibility of the sub-video containing an abrupt scene change that may disrupt its temporal stability, scene detection is performed to identify any occurrences. This is achieved by converting all sub-video frames to the hue, saturation, and luminance (HSL) colour space and analysing the luminance channel of each frame. If the difference in luminance between two contiguous frames exceeds the user-specified threshold, a scene change is detected. To determine the optimal threshold value, we collected 100 sub-videos containing scene changes and 100 sub-videos without scene changes. Through empirical analysis, we established the optimal threshold to be 11, which most accurately classifies the collected sub-videos. The same procedure was followed whenever an optimal threshold value needed to be determined throughout the pipeline. Rather than carelessly discarding sub-videos that contain a scene change, we compute the duration from the start of the sub-video to the scene change and from the scene change to the end of the sub-video. The longer segment is then taken as the new sub-video used for all subsequent operations.

4.7.6 Face Tracking

Since the dataset is expected to contain talking-face videos, we ensure that there is at least one face present in each frame of the sub-video. This test is conducted by performing face detection on each frame and discarding the sub-video if no face is detected in any frame. After extensive experimentation with several face detectors (MTCNN [Zhang *et al.* 2016], DLib [King 2009a], and RetinaFace [Deng *et al.* 2020]), we choose the FaceBoxes face detector [Zhang *et al.* 2017a; Guo *et al.* 2018 2020] for its supreme runtime efficiency, accuracy, and robustness to pose and lighting conditions. For each detected face, an additional assessment is conducted to ensure it is sufficiently large since enlarging a too small face would considerably deteriorate the visual quality. This test verifies whether the area of the detected face accounts for at least 8% of the entire frame's area. If not, the sub-video is discarded.

[4] https://ffmpeg.org/

4.7.7 Visual Quality Assessment

Since the AVSpeech dataset [Ephrat *et al.* 2018] is curated from YouTube videos, no assumptions can be made regarding the visual quality of the contained videos. Consequently, it is crucial to assess the visual quality to eliminate sub-optimal videos, such as those that are blurry, noisy, or have low resolution. Failing to do so could harm the quality of the results achieved by the presented solution. As the primary focus is on the active speaker's face, we narrow our assessment to only the speaker's facial crop rather than the entire frame. Furthermore, since some videos may contain multiple faces, we identify the active speaker by denormalizing the coordinates provided by the dataset (which denote the centre point of the active speaker), computing the centre point of each detected facial crop, and selecting the face closest to the denormalized coordinates.

To evaluate the visual quality of the active speaker's facial crop, several no-reference image quality assessment metrics, including BRISQUE [Mittal *et al.* 2012a] and NIQE [Mittal *et al.* 2012b], were considered. Here, *no-reference* means that ground-truth data is not required to compute the metrics. However, our experiments revealed that these metrics do not consistently correlate with a human's perception of visual quality. Therefore, the variance of Laplacian (VoL), a well-known solution for blur detection [Bansal *et al.* 2016], is used as a heuristic for visual quality. We found that VoL corresponds well to a human's perception of visual quality. A frame is declared of poor quality if the VoL is below a threshold determined empirically using the protocol detailed previously. To account for a small degree of sub-optimal frames, such as those with motion blur, the sub-video is discarded only if more than 60% of the video frames are declared as poor quality.

4.7.8 Audio Quality Assessment

Despite the creators of the AVSpeech dataset [Ephrat *et al.* 2018] affirming that it contains clean audio with no interfering sounds, this does not necessarily mean that the dataset is exempt from other forms of audio degradation, such as echo and noise. Hence, we compute two no-reference audio quality assessment metrics: MOSNet [Lo *et al.* 2019] and SRMR [Falk *et al.* 2010]. MOSNet is computed using a convolutional and recurrent neural network designed to approximate human subjective ratings by predicting a Mean Opinion Score on a scale of one to five. Originally intended for assessing voice conversion speech segments, MOSNet proves useful for our purpose. On the other hand, SRMR is based on a modular spectral representation of a speech signal and has been widely used to assess the colouration, reverberation, and overall quality of speech segments (higher is better). Experiments demonstrate that both metrics correlate well with human perception of audio quality. We set the threshold for MOSNet at 2.4 and for SRMR at 3.6, as determined in the previously discussed protocol. Sub-videos failing to meet these requirements are discarded.

4.7.9 Sync Correction

As mentioned previously, ensuring that all videos are sync-corrected is essential when addressing the visual dubbing problem [Prajwal *et al.* 2020]. In the final step of the pipeline, the sub-video is passed as input to the SyncNet network [Chung and Zisserman 2016b], which produces the AV offset, distance, and confidence metrics. Subsequently, the AV offset and con-

fidence metrics are thresholded at 15 and 4.5, respectively, to determine whether the video is eligible for sync correction. If this requirement is not met, it suggests that the sub-video is either dubbed or a voiceover. Conversely, if the sub-video is eligible for sync correction, the produced AV offset measure is used to sync-correct the sub-video, where a positive AV offset indicates that the audio leads the video, whereas a negative AV offset indicates the opposite.

After pre-processing the AVSpeech dataset, approximately 1440 hours of footage remain, which accounts for 30% of the original dataset. This reduction is a result of removing sub-

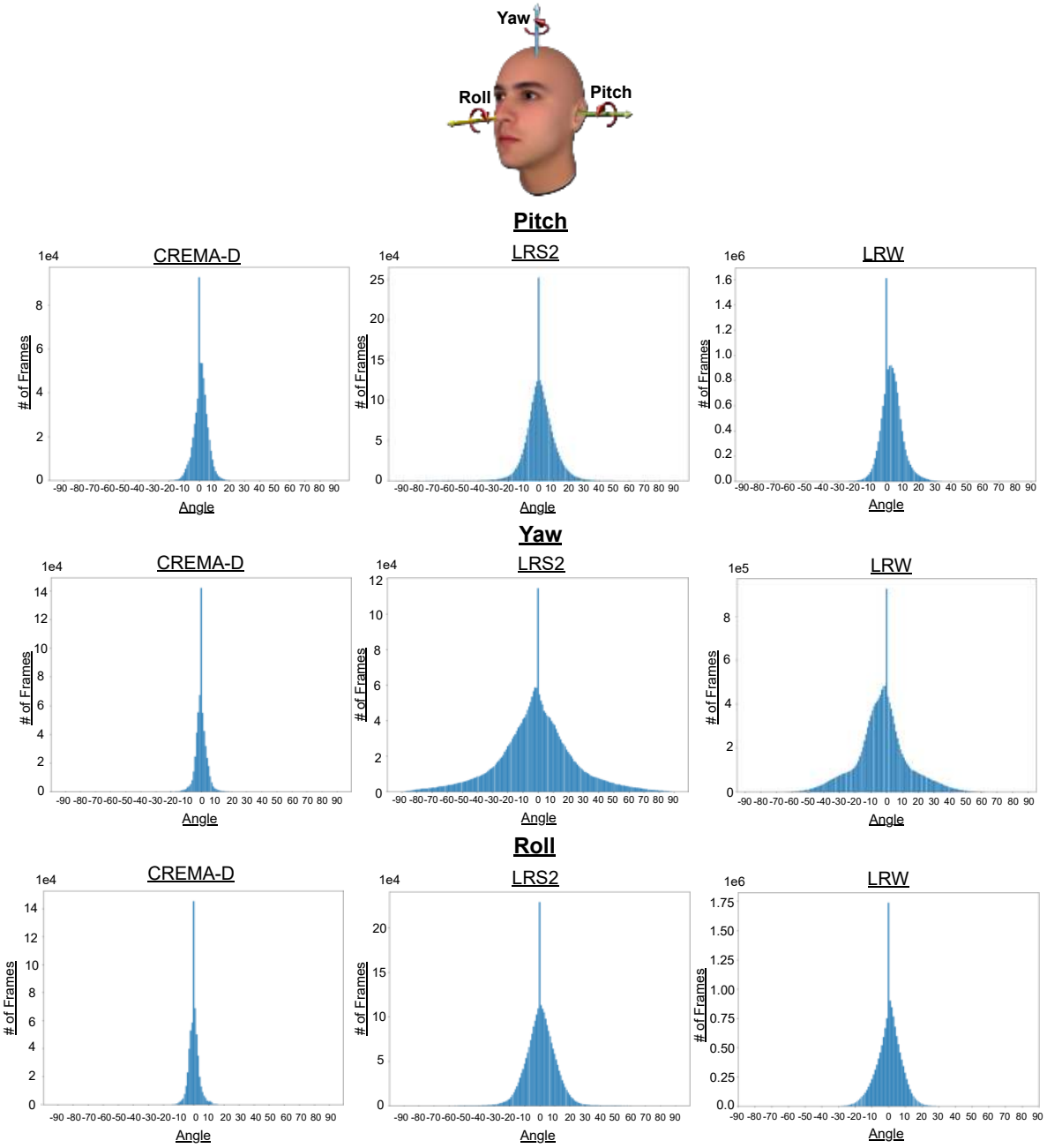


Figure 4.5: Illustration showing the head pose distribution along the pitch, yaw, and roll axes for the CREMA-D, LRS2, and LRW datasets.

optimal footage and inaccessible videos (deleted or made private on YouTube). Despite this, the pre-processed dataset still contains considerably more hours of footage compared to the majority of existing datasets in their raw (unprocessed) form, as shown in Table 4.1.

4.8 Head Pose Distribution of Existing Datasets

Exploratory data analysis of several datasets commonly used when addressing the visual dubbing problem revealed that the majority of these datasets are strongly biased toward frontal and near-frontal faces [Chung and Zisserman 2017; Cao *et al.* 2018] as shown in Figure 4.5. It is for this reason that existing solutions perform proficiently for frontal and near-frontal faces whereas the performance deteriorates as the face becomes increasingly non-frontal. This may be attributed to the lack of exposure to non-frontal faces during training. This observation, coupled with the inevitability of non-frontal faces in real-world videos, necessitates the development of the presented pose-invariant solution discussed in Chapter 8. In summary, this data bias is remediated by performing a *pose augmentation* [Cheng *et al.* 2020] which synthesizes non-frontal talking-face videos from the abundance of frontal and near-frontal footage available which is subsequently used to train the presented solution.

4.9 Conclusion

This chapter provided a comprehensive overview of the data landscape concerning the visual dubbing problem. Based on critical analyses of various datasets, we justified our selection of the datasets used to train the presented solution. This analysis also highlighted the need for developing a high-resolution, and pose-invariant, visual dubbing solution. In the following chapter, we introduce the first network of our solution – the pre-trained lip-sync discriminator, responsible for guiding the generator network on how to produce dubbed results with accurate lip-sync.

Chapter 5

Lip-Sync Discriminator

The previous chapter presented an extensive overview of various audio-visual datasets commonly used to address the visual dubbing problem. By precisely defining the desired data properties, we ensured that only datasets containing the highest quality data were selected to train the presented solution. This chapter details our pre-trained lip-sync discriminator, which provides feedback to the generator network on how to produce results with accurate lip-sync. Here, *pre-trained* refers to the fact that the lip-sync discriminator is trained from scratch and independently, meaning it is not trained adversarially with the generator network. Specifically, when training the generator network, the lip-sync discriminator remains fixed. We present an elaboration on the problem formulation, design choices, and experiments conducted.

5.1 Conceptual Overview

In order for humans to determine whether a talking-face video is in-sync or out-of-sync, i.e., whether the perceived audio coincides with the perceived mouth movements of the speaker, we leverage the audio-visual nature of videos, which simultaneously stimulates the viewer's auditory and visual systems (as shown in Figure 5.1). Specifically, viewers solely utilize the audio and visual channels of the video to classify whether it is in-sync or out-of-sync, addressing a binary classification problem known as the *Audio-Visual Temporal Synchronization* (AVTS) problem [Korbar *et al.* 2018]. Given this binary classification problem and our adoption of a GAN-based approach, we note that the discriminator network within a traditional GAN framework is a form of discriminative model that can be adapted to be a classification network [Goodfellow *et al.* 2014]. Consequently, we introduce a lip-sync discriminator, which accepts the audio and visual channels of a talking-face video as input and classifies whether it is in-sync or out-of-sync. The following discussion provides a conceptual overview of how the lip-sync discriminator is trained, along with a detailed description of its design and implementation.

Based on the problem formulation presented by Korbar *et al.* [2018], our theoretical analysis of the AVTS problem begins with a talking-face dataset $V = \{(a^{(1)},v^{(1)}), (a^{(2)}, v^{(2)}), ..., (a^{(n)}, v^{(n)})\}$ containing an equal proportion of in-sync and out-of-sync videos. Here, $a^{(n)}$ and $v^{(n)}$ denote the audio and visual streams of the n-th video, respectively. The labels $y^{(n)}$ are implicitly de-

Figure 5.1: A talking-face video simultaneously stimulates the viewer's auditory and visual systems to determine whether the video is in-sync or out-of-sync.

rived, where $y^{(n)} = 0$ represents out-of-sync videos, and $y^{(n)} = 1$ represents in-sync videos. Consequently, the final dataset representation is $V = \{(a^{(1)}, v^{(1)}, y^{(1)}), (a^{(2)}, v^{(2)}, y^{(2)}), ..., (a^{(n)}, v^{(n)}, y^{(n)})\}$. Since humans solely rely on the audio and visual streams of a video to assess synchrony, it is desirable for the lip-sync discriminator to use only this information. Fortunately, the natural synergy between the audio and visual streams of a video (providing cross-modal supervision) [Korbar *et al.* 2018] enables the model to be trained in a self-supervised manner, allowing for the abundance of talking-face videos available online to be exploited.

To account for the bi-modal nature of videos, our lip-sync discriminator is composed of two streams i.e., an audio encoder F_a and a visual encoder F_v. Each encoder accepts its respective channel from the video as input and produces a compact latent representation of all pertinent information within its modality. The objective is to learn a classification model D that minimizes the classification error, ensuring that $D(F_a(a^{(n)}), F_v(v^{(n)})) = y^{(n)}$ for as many videos as possible. This problem formulation for the AVTS problem induces a form of *cooperative learning*, since the audio and visual encoders must work together to correctly classify the input video [Korbar *et al.* 2018].

To provide context for the design of the lip-sync discriminator, we contrast our design choices to those of other notable solutions [Chung *et al.* 2017; KR *et al.* 2019; Prajwal *et al.* 2020; Yang *et al.* 2020] where necessary. The solution primarily seeks inspiration from SyncNet [Chung and Zisserman 2016b] (discussed in Section 2.3.1), which serves as the basis for numerous solutions due to its novelty and effectiveness, as well as the pre-trained lip-sync discriminator of Prajwal *et al.* [2020], which proposes numerous design enhancements to the SyncNet network. The following sections provide an overview of the input representations to the presented model, each of its encoders, and the concept that underpins the successful training of the solution.

5.2 Input Representations

5.2.1 Visual Encoder

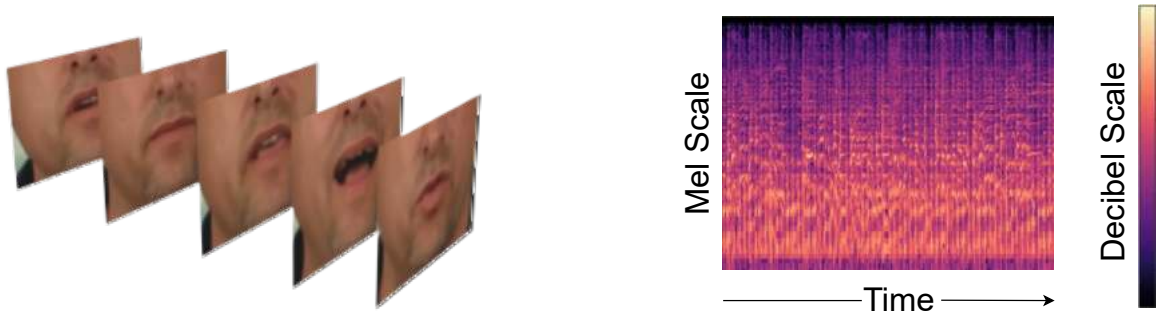
Given a talking-face video, the visual encoder accepts the facial crops extracted from the video as input. Similar to SyncNet [Chung and Zisserman 2016b] and Wav2Lip [Prajwal *et al.* 2020], only the lower half of the speaker's face is used as input since the upper half does not contain relevant information for lip-sync. This decision was empirically verified by training the lip-sync discriminator using the entire facial crop, but this did not improve the accuracy or robustness. Unlike SyncNet, which uses grayscale images, our model accepts RGB images, as the additional colour information is expected to improve lip-sync accuracy.

The poor lip-sync accuracy achieved by the lip-sync discriminator of KR *et al.* [2019] has been attributed to the sub-optimal design choice of testing for synchronicity based on a single frame and its corresponding audio segment. In contrast, using a short temporal window of frames has shown to significantly improve accuracy [Prajwal *et al.* 2020]. Doing so provides the model with the necessary temporal context to consider mouth motion. Consequently, we follow SyncNet [Chung and Zisserman 2016b] and Wav2Lip [Prajwal *et al.* 2020] and pass 0.2 seconds of footage to each encoder instead.

Since the FPS of all videos was set to 25 during the pre-processing phase, as mentioned in Chapter 4, 0.2 seconds of footage corresponds to $0.2 \times 25 = 5$ visual frames. Extensive analysis by Prajwal *et al.* [2020] showed that the optimal classification accuracy is achieved when using five visual frames as input, as this provides sufficient temporal context while keeping computational resource requirements reasonable. During training, a random facial crop C is selected from the input video, representing the start of the five-frame visual window, and the four subsequent facial crops are retrieved. The lower half of each facial crop is extracted and concatenated to form a spatiotemporal volume with dimensions $[B, 3, 5, H, W]$, where B denotes the batch size, three follows due to the RGB colour space, five denotes the temporal extent (window size), and H and W signify the spatial dimensions of the facial crops. This input representation differs from Prajwal *et al.* [2020], as they instead concatenate the five facial crops along the channel dimension to form an input representation with size $[B, 3 \times 5, H, W]$. The following section elaborates on why we believe our design is better suited for the AVTS problem. Since all facial crops are resized to 96×96 when employing the LRS2 dataset [Afouras *et al.* 2018a], the lower half of each facial crop is extracted and concatenated to form a final input representation with size $[B, 3, 5, 48, 96]$.

5.2.2 Audio Encoder

Complementary to the visual encoder, the audio encoder processes an audio representation of the speaker's speech. Since the raw waveform representation of the audio is not suitable for practical purposes, it is preferable to use well-established audio feature representations, such as mel-spectrograms or Mel-Frequency Cepstral Coefficients (MFCCs) [Bridle and Brown 1974; Mermelstein 1976]. However, there is no consensus on which of these two representations is superior and under what conditions. Given that many speech-based solutions [Shen *et al.* 2018; Skerry-Ryan *et al.* 2018] have adopted mel-spectrograms and achieved state-of-the-art results, we chose to employ mel-spectrograms for our solution.



(a) Input to visual encoder.

(b) Input to audio encoder.

Figure 5.2: Example inputs to the lip-sync discriminator.

The mel-spectrogram is computed by performing a Short-Time Fourier Transform (STFT) on the input audio segment and subsequently using a Mel filter bank to obtain the mel-spectrogram. The Mel scale is the logarithmic perceptual scale of pitches that models how humans perceive sound [Ittichaichareon *et al.* 2012]. The hyperparameters used to compute the mel-spectrograms (listed in Appendix A.1) are largely inspired by the literature [Ping *et al.* 2017; Shen *et al.* 2018]. As shown in Figure 5.2, the mel-spectrogram has time along the x-axis, the Mel scale on the y-axis, and colours representative of decibels. To extract 0.2s of audio, a time slice is selected, and the mel-spectrogram is cropped to create the final input audio representation. Specifically, since 80 mel-bands are used, and 1s corresponds to 80 audio frames (as determined by our audio hyperparameters), 0.2s of audio corresponds to $80 \times 0.2 = 16$ audio frames. Therefore, the cropped input mel-spectrogram has a shape of $[1, 80, 16]$. Due to its convenient 2D representation, mel-spectrograms serve as an ideal input representation for CNNs.

Given the binary classification problem at hand, it is necessary to have a dataset with an equal proportion of in-sync and out-of-sync videos to avoid *class imbalance* [Japkowicz and Stephen 2002; Guo *et al.* 2008]. To tackle this issue, the audio encoder's input is manipulated to randomly create out-of-sync videos with a 50% probability. This process involves selecting a video frame C at random and retrieving the four subsequent frames to form a five-frame spatiotemporal volume. This volume is then passed as input to the visual encoder, and to produce an:

- *In-sync video*: The audio from the same time slice as the visual frame C is extracted and used to crop the mel-spectrogram segment to obtain the corresponding audio segment.
- *Out-of-sync video*: The mel-spectrogram from a different time slice than the visual window is cropped to obtain the required audio segment. It is important to note that an out-of-sync video is achieved by cropping the audio from the same video as the visual window. This approach compels the model to learn spatiotemporal features, unlike extracting audio from a different video, which could lead the network to solely rely on semantic information which is unnecessary for this problem [Korbar *et al.* 2018]. Following the advice of Chung and Zisserman [2016b], the audio is extracted from a time slice that is at most 2 seconds away from the ground-truth (visual) time slice C, as the model does not gain any valuable information beyond this point.

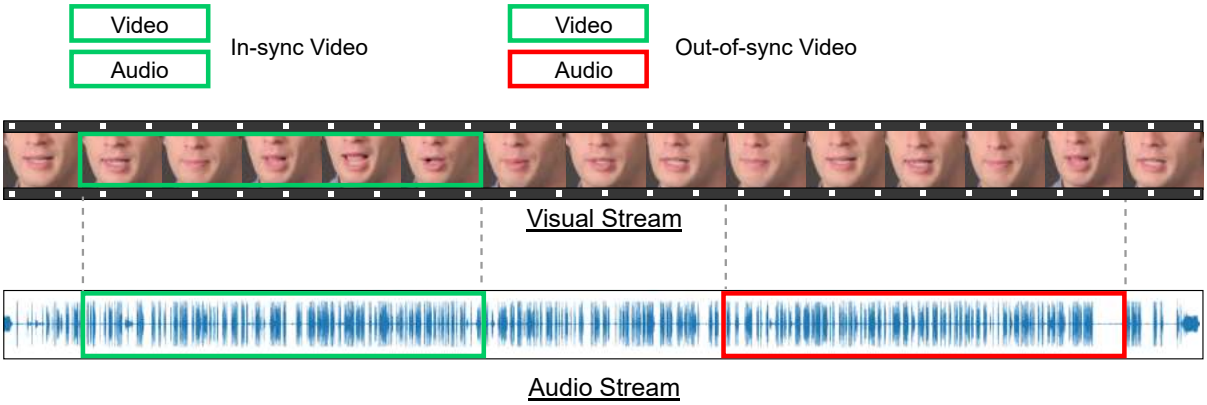


Figure 5.3: Formation of in-sync and out-of-sync videos by altering the segment cropped from the input audio.

5.3 Network Architecture

5.3.1 Visual Encoder

Achieving a temporally coherent dubbed video with high visual quality and lip-sync accuracy is crucial. Therefore, a sensible design choice is to create a visual encoder that extracts spatiotemporal features capturing both spatial and temporal aspects to model motion effectively. Chung and Zisserman [2016b] achieve this by using 3D convolutions in their visual encoder, which are proficient at learning spatiotemporal features. However, this approach is hindered by the exorbitant computation costs resulting from their large parameter count.

In contrast to Chung and Zisserman [2016b], Prajwal *et al.* [2020] use 2D convolutions in their visual encoder, which are less computationally expensive than 3D convolutions [Li *et al.* 2021]. This design choice may raise concerns due to the nature of the problem at hand. 2D convolutions lack the ability to model temporal information and motion patterns, which are crucial factors for video analysis tasks, especially for the AVTS problem. Additionally, 2D convolutions require the five visual frames to be concatenated along the channel dimension, treating all frames analogously to channels without considering the temporal ordering of the frames [Korbar *et al.* 2018].

Following the renowned success of residual learning (ResNets) [He *et al.* 2016], which has proven to facilitate the training of deep neural networks and improve their generation performance, our visual encoder is composed using seven ResNetv1 blocks. To address the shortcomings of prior solutions as mentioned above, we employ *R(2+1)D spatiotemporal blocks* [Tran *et al.* 2018b] (Section 2.2.3), serving as a drop-in replacement for traditional 3D convolutions. The claimed benefits of decomposing 3D convolutions into R(2+1)D blocks are two-fold i.e., (1) the number of non-linearities is doubled without a significant increase in parameter count, and (2) the optimization process is easier, resulting in lower training and testing errors. R(2+1)D blocks are utilized by replacing the convolutional layers in each ResNet block with an R(2+1)D block. As the final layer, a global spatiotemporal pooling operation is performed, producing a 512D embedding vector that encapsulates all pertinent visual information from the input video.

5.3.2 Audio Encoder

Due to the convenient 2D representation of the input mel-spectrograms, the audio encoder is designed as a CNN containing 2D convolutions. Since the presented visual dubbing solution is audio-driven, the audio encoder is required to extract all pertinent audio information effectively. To achieve this, the audio encoder is composed of 14 residual blocks, each containing Conv2D → BatchNorm → ReLU, inspired by Prajwal *et al.* [2020] and Isola *et al.* [2017]. The audio encoder ultimately produces a 512-dimensional embedding vector, which compactly represents all pertinent audio information.

5.4 Training

Since the audio encoder receives a mel-spectrogram representation of the input audio segment, audio features are learnt at the phoneme level rather than the word level. This language-independent nature of the network arises due to the overlap of phonemes and visemes shared among languages, similar to previous solutions [KR *et al.* 2019; Prajwal *et al.* 2020]. Similarly, the visual encoder processes visual frames of the speaker's mouth region, allowing for visual features to be learned at a viseme level. Another important property of the lip-sync discriminator is that it is trained using footage of various speakers, making it speaker-independent – meaning it is invariant to identity, age, and gender. Finally, the model is trained using the LRS2 dataset [Afouras *et al.* 2018a], which includes diverse and unconstrained environments. This makes the solution applicable to in-the-wild conditions, enhancing its overall applicability.

By using an audio and visual encoder, the lip-sync discriminator learns a joint embedding space where audio and visual embedding vectors co-exist. This enables cross-modal supervision, allowing the model to be self-supervised. We employ *contrastive learning* (Section 2.2.2) based on its success in various cross-modal tasks. Contrastive learning leverages the inherent property of embedding spaces, where distance (such as Euclidean distance) reflects semantic similarity [Chandrasekaran and Mago 2021]. In our case, a *positive pair* consists of audio and visual embeddings from an in-sync video, while a *negative pair* is composed from an out-of-sync video. The objective is to minimize the distance between audio and visual embeddings of positive pairs and maximize the distance between audio and visual embeddings of negative pairs [Wang *et al.* 2021d].

To learn a joint embedding space using contrastive learning, a notion of *similarity* between the produced audio and visual embedding vectors must be established. This is obtained by computing the *cosine similarity* between the L_2-normalized audio embedding vector a and the visual embedding vector v to determine their similarity. The use of residual blocks (which performs a ReLU activation as the final operation) ensures that the resulting embedding vectors contain non-negative entries, thus, constraining the cosine similarity to the range $[0, 1]$. A cosine similarity of zero indicates no similarity, while a value of one means the two vectors are identical. This measure is interpreted as the probability that the input video is in-sync, i.e.:

$$P_{sync}(a, v) = \frac{a \cdot v}{max(\|a\|_2 \cdot \|v\|_2, \epsilon)}. \tag{5.1}$$

During training, the cosine similarity measure and implicitly inferred class labels (i.e., $y^{(n)} = 0$ for out-of-sync videos and $y^{(n)} = 1$ for in-sync videos) are used to compute the Binary Cross-Entropy (BCE) loss, which is a natural choice for addressing binary classification problems.

Both encoders of the lip-sync discriminator are initialized with random weights and trained simultaneously. The entire network is trained from scratch, and based on empirical analyses, the model is trained with a batch size of 64 and the Adam optimizer [Kingma and Ba 2014] using a learning rate of 0.0001. For experiments using the *pre-train* and *train* partitions, the model is trained for 750K iterations on the *pre-train* partition, followed by the *train* partition until convergence. In other experiments, the *pre-train* and *train* partitions are combined to form the training dataset, and the model is trained until convergence. All experiments are implemented in the PyTorch framework [Paszke *et al.* 2019], utilizing an Nvidia GeForce RTX 3090 GPU.

The most striking characteristic of the presented lip-sync discriminator is that it is pre-trained, meaning it is not trained in tandem with the generator network. This design choice is largely inspired by the work of Prajwal *et al.* [2020] which demonstrated that training the discriminator in conjunction with the generator network significantly deteriorates the classification accuracy. They attribute this deterioration to the fact that, if the discriminator was trained using samples generated by the generator network, visual artefacts in the generated frames may be exploited as a shortcut which would deviate the model from learning fine-grained audio-visual correspondences. To quantify this, Prajwal *et al.* [2020] reported a classification accuracy of 91.6% on the LRS2 test dataset [Afouras *et al.* 2018a] using their pre-trained lip-sync discriminator, while their discriminator trained adversarially achieved an inferior accuracy of 73.5%, resulting in a 19.76% decrease in accuracy.

5.5 Experiments

5.5.1 Curriculum Learning

Curriculum learning [Bengio *et al.* 2009] is a concept that has its origins in cognitive science and psychology which attempts to mimic the ordered approach that humans (and animals) take toward learning. Humans require approximately two decades to be trained as fully functional adults in society, and this development is primarily based on a carefully formulated curriculum i.e., the education system. The core idea is to *start small* by learning basic fundamental concepts and to exploit this knowledge to ease the learning process for increasingly difficult/complex concepts. For instance, to learn university-level calculus, one begins by grasping fundamental concepts such as addition and subtraction in primary school, progressing to equations in secondary school, and exploiting this prior knowledge to ease the learning process when learning university-level calculus.

Adopting a sequential approach to learning significantly enhances the rate of learning. This is in contrast to exposing the learner to concepts in a random order of *difficulty* [Hacohen and Weinshall 2019]. Curriculum learning attempts to answer the following question: *Can improved results be achieved by ordering training samples in ascending order of difficulty as opposed to exposing the model to training samples in an arbitrary order of difficulty?* A survey of the literature reveals that curriculum learning has proven effective in improving the rate

of convergence, the quality of the local minima attained, and the generalization capabilities in numerous cases [Bengio *et al.* 2009].

To describe the curriculum learning strategy adopted, the framework of Wang *et al.* [2021c] is used, which requires a *difficulty measurer* and *training scheduler* to be defined. The difficulty measurer is used to determine the difficulty of each training sample and arrange them in ascending order of difficulty. On the other hand, the training scheduler determines when the model is exposed to new data of increased difficulty.

To avoid the need for manual annotations for the difficulty measurer, we devise an intuitive solution to control the difficulty of the out-of-sync videos exposed to the model. Initially, the visual input is extracted from time slice K, and for an in-sync video, the audio would also be extracted from time slice K. For an out-of-sync video, the audio is extracted from a time slice with a minimum distance of 25 time slices away from K. The difficulty is then incremented by extracting the audio from a time slice with a minimum distance of 24 time slices away from K. This process continues until the model may be exposed to out-of-sync videos with the audio extracted from only one time slice away from K, representing the most challenging form of out-of-sync video. Figure 5.4 illustrates this process.

Figure 5.4: Visualization of the curriculum learning strategy we adopt by controlling the segment extracted from the input audio when producing out-of-sync videos with increasing difficulty.

For the training scheduler, we aim to allocate fewer iterations to easier samples and assign the majority of training iterations to more difficult ones. This decision is based on the rationale that the model requires fewer training iterations to test for synchronicity in a video with a one second (25 time slice) offset compared to a video with a 0.04s (one time slice) offset. Following the literature [Wang *et al.* 2021c], we adopt a linear scheduler, where the number of training samples devoted to the easiest samples (i.e., when the audio is sampled with a minimum distance of 25 time slices away from K) is adjusted. Additionally, the number of

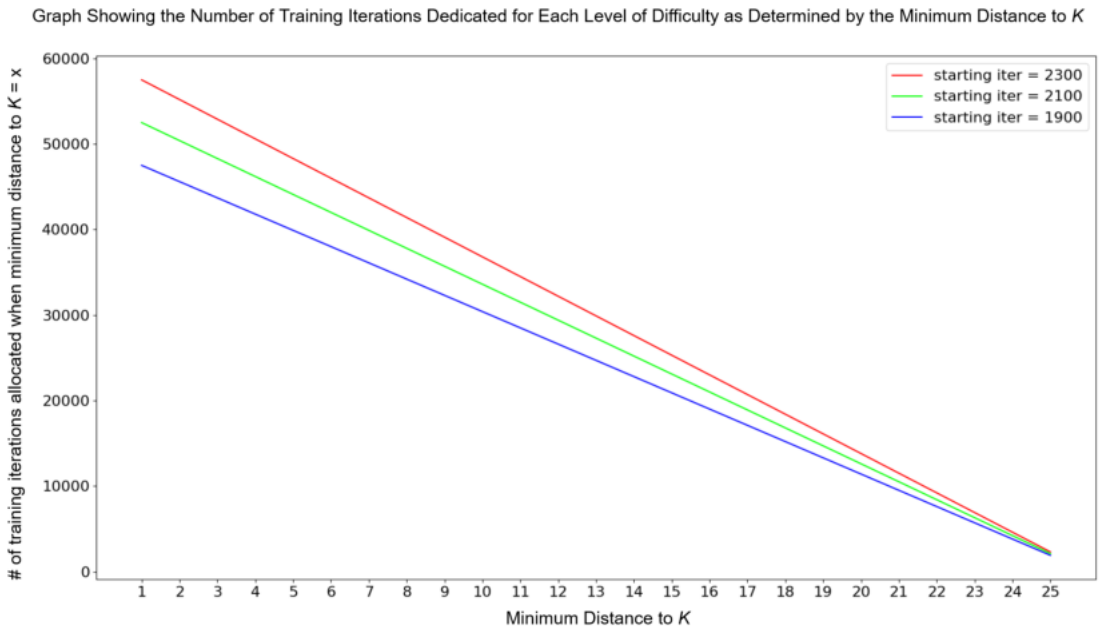


Figure 5.5: Illustration showing the three curriculum learning schedules experimented with. For the least challenging scenario (minimum distance to $K = 25$), a minimal number of training iterations are assigned. As the minimum distance to K decreases (indicating increased difficulty), we incrementally allocate more training iterations (e.g., for `start_iter` = 2100, minimum distance to $K = 25 \rightarrow 2100$ training iterations, minimum distance to $K = 24 \rightarrow 4200$ training iterations, etc.). Additionally, we observe the highest number of training iterations assigned to the most demanding samples, i.e., when the minimum distance to $K = 1$.

iterations is increased with each increment in difficulty (by decrementing the minimum distance to K). We experiment with three variants, starting with 1900, 2100, and 2300 training iterations to ensure the model is exposed to out-of-sync videos with a one time slice offset from K within a reasonable number of training iterations. In the extreme case of commencing with 2300 training iterations, the model is exposed to these videos after ≈750K training iterations. These three curriculum learning schedules are illustrated in Figure 5.5.

5.5.2 Non-Linear Projection Head

An examination of *SimCLR* [Chen *et al.* 2020a], a popular method for self-supervised representation learning, reveals the use of a *non-linear projection head*, i.e., a non-linear MLP, stacked at the top of each encoder. Instead of using the embedding vector from the original encoder, the one obtained after passing through the non-linear projection head is utilized, believed to enhance the representation quality of the original embedding vector [Chen *et al.* 2020a]. Comparing the Top-1 classification accuracy on the ImageNet dataset [Deng *et al.* 2009] without a projection head, with a linear projection head, and with a non-linear projection head, [Chen *et al.* 2020a] demonstrated that the non-linear projection head outperforms the linear one (+3%) and significantly improves performance compared to no projection head (>10%). Non-linear projection heads have been adopted to enhance the performance of various solu-

tions [Chen *et al.* 2020b; Appalaraju *et al.* 2020], including *MoCo* [Chen *et al.* 2020c], another state-of-the-art approach for self-supervised representation learning. Given its promise, we conduct experiments by adding the non-linear projection head of SimCLR [Chen *et al.* 2020a] on top of the audio and visual encoders. Each non-linear projection head is comprised of a Fully Connected (FC) layer, batch normalization layer, ReLU operator, FC layer, followed by a ReLU operator.

5.5.3 Perfect Match

Chung *et al.* [2019] propose *Perfect Match*, an extension of the SyncNet network [Chung and Zisserman 2016b], to learn more powerful cross-modal embeddings. Instead of contrasting one visual embedding vector to one audio embedding vector, they suggest contrasting one visual embedding vector to P audio embedding vectors, as illustrated in Figure 5.6. This reframes the AVTS problem as a cross-modal retrieval task through P-way feature matching. The visual embedding vector is produced in the same way as before, but the audio encoder now accepts P mel-spectrograms as input, generating P audio embedding vectors. Note that all P input audio representations are sampled from the same video as the input to the visual encoder, but only one corresponds (i.e., is in sync) with the input visual encoder.

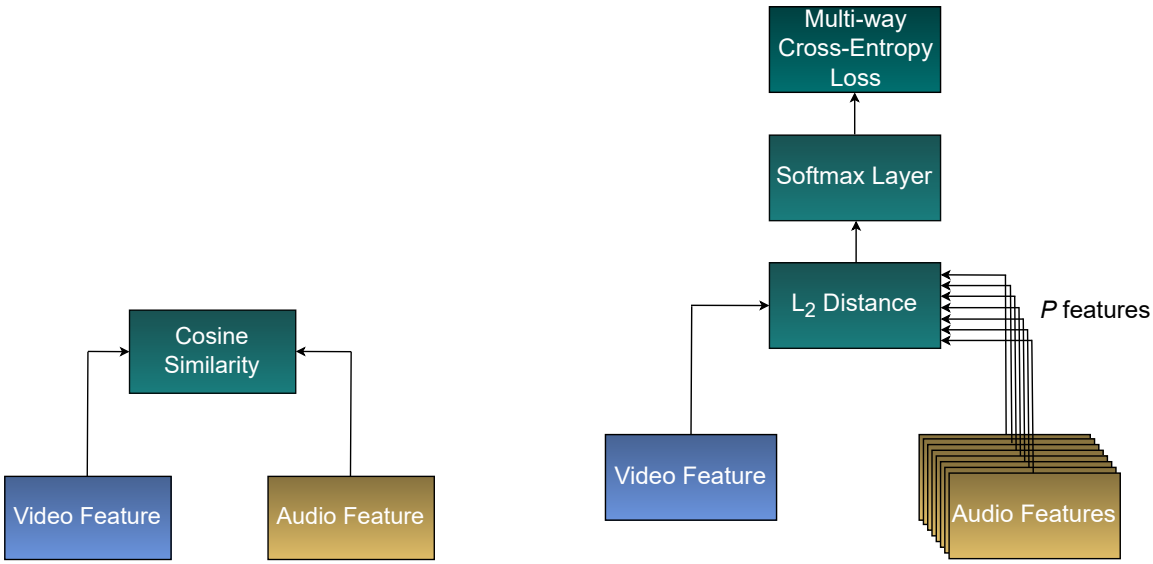
The training objective is to find the most relevant audio embedding vector, given the visual embedding vector. This is achieved by computing the Euclidean distance between the visual embedding vector and each audio embedding vector, resulting in P distance measures. The network is then trained using the cross-entropy loss with the inverse of these distance measures after passing through a softmax layer. This ensures that the similarity between the matching pair is greater than that of non-matching pairs. We integrate Perfect Match into the presented ADVD solution and vary $P \in \{8, 16, 32, 64\}$, representing the number of audio embedding vectors compared to the visual embedding vector. Note that we limit the greatest value of P in our experiments to 64 due to computational resource constraints.

5.6 Evaluation

Given the numerous experiments conducted, we analyse their impact on the classification performance of the lip-sync discriminator. Conducting a rigorous analysis of the results proved crucial since it helps to identify the best-performing discriminator based on the computed classification metrics. Subsequently, this discriminator is used to train the generator network.

5.6.1 Comparative Solutions

To contextualize the presented lip-sync discriminator's performance, a comparison is made with prior solutions to determine whether it achieves improved or reduced performance and to what extent. Firstly, the network is contrasted with the state-of-the-art Wav2Lip solution [Prajwal *et al.* 2020], the only other solution that utilizes a pre-trained lip-sync discriminator. Experiments are conducted with a baseline network consisting of 17 residual blocks (Conv3D→ BatchNorm → ReLU using traditional 3D convolutions) to assess the impact of using spatiotemporal convolutions instead of 2D convolutions in the discriminator. Additionally, an investigation is carried out to determine whether employing R(2+1)D spatiotemporal



(a) Original Solution. (b) Perfect Match.

Figure 5.6: Comparison between our original solution and Perfect Match [Chung *et al.* 2019].

blocks leads to improved results compared to traditional 3D convolutions. Finally, the solution is compared to the transformer-based solution VocaLiST [Kadandale *et al.* 2022], which is also designed to classify whether an input video is in-sync or out-of-sync. All solutions are trained using the LRS2 dataset [Afouras *et al.* 2018a], and the provided pre-trained models for each comparative solution are used during the evaluation.

5.6.2 Data

The evaluation is conducted using the LRS2 test dataset [Afouras *et al.* 2018a] which comprises of 1243 talking-face videos. The creators of the dataset ensure that there is no overlap between the training and test partitions, thus, preventing information leakage. The data is pre-processed as extensively detailed in Chapter 4 i.e., facial crops from each video are extracted and resized to 96×96 whereas the audio is resampled to a sampling rate of 16000Hz.

5.6.3 Evaluation Metrics

To compare and quantify the efficacy of the aforementioned solutions, two well-established classification metrics, namely accuracy and F1-score, are computed. The accuracy measures the number of correct predictions made out of all predictions (higher is better), while the F1-score computes the harmonic mean of the precision and recall metrics (higher is better).

5.6.4 Evaluation Protocol

To evaluate the classification performance of the aforementioned solutions using the test dataset, it is crucial to note that these solutions address a binary classification problem i.e., classifying an input talking-face video as either in-sync or out-of-sync. However, the LRS2

dataset [Afouras *et al.* 2018a] only contains in-sync videos, therefore, the data should be leveraged as before to produce out-of-sync videos to conduct the evaluation.

Similar to the training process, out-of-sync videos are produced with a 50% probability by randomly selecting a video frame. For in-sync videos, the input audio segment is extracted from the same time slice as the video frame. For out-of-sync videos, a time slice is randomly selected from which the input audio segment is extracted. However, due to the stochastic nature of this process, which involves randomly determining whether a video should be in-sync or out-of-sync and randomly selecting a time slice for out-of-sync videos, this approach is unsuitable for evaluative purposes. Using this approach for evaluation would lead to unfair comparisons among the solutions, as each test video would not be treated uniformly by each comparative solution, resulting in misleading results.

The aforementioned issue is addressed by pre-determining how each video should be used through the use of *file lists*. For each video, the following details are specified: the video name, the time slice from which the beginning of the five-frame visual input is extracted, and the time slice from which the input audio segment is extracted. The video frame is randomly selected as before, and a decision is made on whether the video should be treated as in-sync or out-of-sync with a 50% probability. In the case of in-sync videos, the placeholder *-1* is used as the time slice from which the input audio segment is extracted, indicating the use of the ground-truth time slice. On the other hand, for out-of-sync videos, a time slice is randomly selected from which the input audio is taken. To account for any inherent stochasticity that might introduce biases, ten file lists are produced using ten different seeds. The performance of each solution is then evaluated on each file list, and the averaged results are reported. As previously mentioned, the classification performance is based on the 0.2s of input audio and visual footage extracted from the input video.

5.6.5 Results

5.6.5.1 Effect of Network Architecture

The analysis begins with the Wav2Lip solution [Prajwal *et al.* 2020], which achieves an accuracy of 81%[3] and an F1-score of 0.7935. To compare the performance of 2D convolutions with traditional 3D convolutions in addressing the AVTS problem, these results are contrasted with those obtained by the 3D baseline implementation. The 3D baseline achieves an 8.49% increase in accuracy and a 10.57% increase in F1-score compared to the Wav2Lip solution. This result emphasizes the importance of considering the temporal dimension explicitly (using spatiotemporal convolutions) for the ADVD problem. Moreover, the higher F1-score achieved by the 3D baseline indicates a lower false positive and false negative rate compared to Wav2Lip.

[3]When using the official source code of Wav2Lip [Prajwal *et al.* 2020] and the evaluation protocol detailed in Section 5.6.4, we found that Wav2Lip achieves an 81% out-of-sync detection accuracy, while the authors report a 91% accuracy. Our evaluation protocol does not disadvantage the Wav2Lip solution in any way; in fact, it can be seen as an extension or generalization of the protocol adopted by Wav2Lip. Instead of evaluating using a single file list, which may lead to inaccurate or misleading results, each solution is identically assessed using ten file lists (generated using ten different seeds) and reporting the averaged performance. This approach provides a more accurate reflection of the solution's performance.

Solution	Accuracy (↑)	F1-Score (↑)
Our Solution:	–	–
Trained on the *train* partition	**0.91440**	**0.91459**
Trained on the *pre-train* & *train* partitions	0.91062	0.90812
Curriculum Learning (`start_iter` = 1900)	0.89847	0.89618
Curriculum Learning (`start_iter` = 2100)	0.89903	0.89705
Curriculum Learning (`start_iter` = 2300)	0.90008	0.89838
Non-linear Projection Head	0.88366	0.87984
Perfect Match (N=8)	0.83823	0.82017
Perfect Match (N=16)	0.82968	0.81193
Perfect Match (N=32)	0.81954	0.79599
Perfect Match (N=64)	0.78744	0.75056
3D CNN (Baseline)	0.88407	0.88052
Wav2Lip [Prajwal *et al.* 2020]	0.81488	0.79635
VocaLiST [Kadandale *et al.* 2022]	0.91287	0.91229

Table 5.1: Quantitative results showing the effect of each experiment on the accuracy and F1-score achieved in comparison to other state-of-the-art solutions.

Comparisons are also made between our solution (composed of R(2+1)D spatiotemporal blocks) and the 3D CNN baseline. The standard R(2+1)D solution trained on the *train* partition achieves superior results, with an accuracy of 0.91440 and an F1-score of 0.91459. This represents a 3.43% increase in accuracy and a 3.87% increase in F1-score compared to the 3D baseline. These results confirm that R(2+1)D spatiotemporal blocks are indeed superior to traditional 3D convolutions, as evidenced by the improved results. While the transformer-based solution of Kadandale *et al.* [2022] also achieves exceptional results similar to our standard R(2+1)D solution, we argue that our solution is significantly more effective, given that their solution has 3.6× more parameters. Our solution achieves superior results despite having only 22M parameters compared to their 80M parameters.

5.6.5.2 Effect of Training Data used

When comparing the performance achieved with and without using the *pre-train* partition of the LRS2 dataset [Afouras *et al.* 2018a], a decrease of 0.41% in accuracy and a 0.71% decrease in F1-score is observed when the *pre-train* partition is used in addition to the *train* partition. This minor decrease in performance may be attributed to the *pre-train* partition not being of the same quality as the *train* partition. This finding may also explain why most solutions that utilize the LRS2 dataset only use the *train* partition as training data. As a result of these findings, we opted to not use the *pre-train* partition for subsequent experiments, as this would prolong the training process while decreasing the model's performance.

5.6.5.3 Effect of Curriculum Learning

The experiments primarily focused on tuning the curriculum scheduler, which determines when new content of increased difficulty is exposed to the solution. This was achieved by adjusting `start_iter`, the number of training iterations compounded for each increment

in difficulty. The results achieved by the three variants are virtually indistinguishable from each other (variance of accuracy = $6.68e-7$, variance of F1-score = $1.22e-6$), suggesting that the scheduler is not strongly correlated with the results. Furthermore, a minor improvement in accuracy and F1-scores is noticed as `start_iter` increases. Although curriculum learning may have slightly reduced the performance of the superior result, the achieved results remain promising and superior to the 3D CNN baseline, Wav2Lip solution [Prajwal *et al.* 2020], and the non-linear projection head and Perfect Match experiments [Chung *et al.* 2019].

5.6.5.4 Effect of the Non-Linear Projection Head

Results show that appending a non-linear projection head to the audio and visual encoders leads to a reduction of 3.48% in accuracy and 3.95% in F1-score compared to the superior solution. This decline can be attributed to the significant increase in parameters (2×530K ≈ 1M additional parameters) caused by adding the two projection heads, making the optimization process considerably more challenging.

5.6.5.5 Effect of Adopting Perfect Match

Each of the Perfect Match [Chung *et al.* 2019] experiments resulted in a significant decrease in accuracy and F1-score compared to the superior solution. Specifically, a decrease in the range of 8.33% to 13.88% was observed. Additionally, the classification metrics are inversely proportional to the number of negative samples used (P). This observation suggests that the decrease in performance follows as a consequence of the increase in computation required as P increases. Recall, instead of contrasting a visual embedding vector to an audio embedding vector, Perfect Match contrasts the visual embedding vector to P audio embedding vectors. This is further supported by the observation that improved results are achieved when $P = 8$, which indicates a lower computational load, compared to the results achieved when $P = 64$. The decrease in performance may also be attributed to the lack of model capacity, preventing the model from achieving improved results as P (and the computational load) increases.

5.6.5.6 Comparison of Convergence Rates of CNN-based Solutions

By contrasting the loss curves of the CNN-based solutions presented in Table 5.1, the rate of convergence is examined. It is evident that the loss in the first 300K – 500K iterations of Wav2Lip [Prajwal *et al.* 2020] (when trained from scratch) and the 3D CNN baseline appears to be stagnant, and only after this period do these solutions begin to converge. However, this is not the case with R(2+1)D solutions, which start to converge from the beginning of training. This observation supports the statements made by the authors [Korbar *et al.* 2018], stating that the factorization of a 3D convolution into a 2D spatial convolution followed by a 1D temporal convolution makes the optimization process easier. Upon examining the loss curves of the Perfect Match [Chung *et al.* 2019] solutions, a small degree of divergence is observed in all four variants.

Based on our analysis, we adopted our superior lip-sync discriminator (trained on the *train* partition of the LRS2 dataset [Afouras *et al.* 2018a]) to train all of our generator networks.

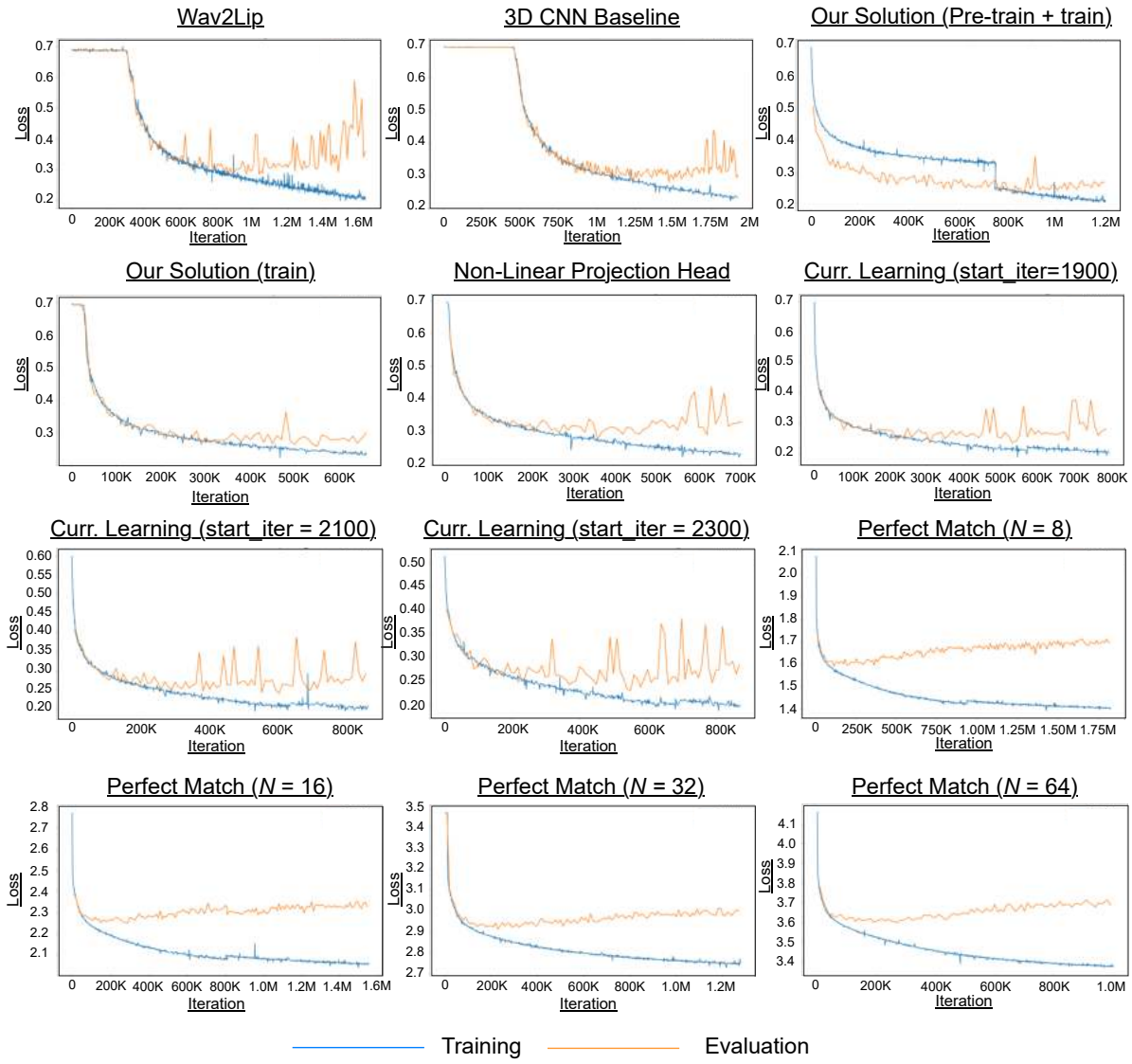


Figure 5.7: Loss curves for all conducted experiments. Notably, both Wav2Lip [Prajwal *et al.* 2020] and our 3D CNN baseline exhibit a plateau during the initial 300K - 500K iterations, while our R(2+1)D implementations display an immediate convergence, resulting in a notably accelerated rate of convergence.

5.7 Conclusion

This chapter presented an in-depth discussion on our pre-trained lip-sync discriminator, covering its input representations, network architecture, and various experiments conducted. Results indicate that the solution outperforms previous state-of-the-art solutions, including Wav2Lip and VocaLisT. The following chapter introduces the generator network which is responsible for producing the dubbed result.

Chapter 6

Generator Network

6.1 Introduction

The previous chapter provided a comprehensive description of the pre-trained lip-sync discriminator, starting with the problem formulation and progressing to the rationale behind the network's design, including inputs, network architecture, and experiments conducted. In this chapter, a similar first-principles approach is taken to analyse the complementary network within the GAN framework - the generator network.

6.2 Conceptual Overview

Given the preceding discussions, including data curation (Chapter 4) and the design of the lip-sync discriminator to ensure accurate lip-sync (Chapter 5), we now consider how all these efforts are united to enable the generator to perform ADVD. To understand how the generator network works, we consider the canonical ADVD problem formulation: Given an input talking-face video $V = \{V_1, V_2, V_3, \ldots, V_T\}$, along with an input dubbing audio segment $A = \{A_1, A_2, A_3, \ldots, A_T\}$ of equal duration, the objective is to learn a model G (parameterized by θ) that produces a new sequence of visual frames V^A. This sequence should synchronize the speaker's mouth movements with the input dubbing audio A while preserving the remaining visual elements of the scene, such as head pose, background, and lighting. This objective is expressed as:

$$V^A = \{V_1^A, V_2^A, V_3^A ..., V_T^A\} = G(V, A; \theta). \tag{6.1}$$

Upon reviewing the formulation presented in Equation (6.1), it is evident that a naïve solution would be to train the model in a supervised setting using a dataset of triples, each containing an input video, dubbing audio, and ground-truth video (the talking-face video of the speaker genuinely uttering the dubbing content) [Yang *et al.* 2020]. However, this approach would be sub-optimal since curating such a dataset would require the speaker to record two talking-face videos under identical recording conditions while uttering different content. This would inevitably introduce micro-differences in head pose and lighting. Moreover, to minimize the differences between the input and ground-truth videos (except for the speaker's mouth region), these recordings would need to be made in a controlled and constrained setting, involving

several recordings. This data acquisition process would be time-consuming and laborious, resulting in a small-scale constrained dataset, which could ultimately hinder the solution's performance.

To address the aforementioned issue, it is desirable to utilize the abundance of in-sync talking-face videos for self-supervised training. This approach involves exploiting the audio and visual channels of an input talking-face video as a form of cross-modal supervision [Nagrani *et al.* 2018; Arandjelovic and Zisserman 2018]. Therefore, the problem can be reformulated, where model G takes the audio and visual channels of a talking-face video as input and is trained to reconstruct the speaker's mouth shapes, i.e.:

$$V^A = G(V^A, A; \theta). \tag{6.2}$$

Unfortunately, this problem formulation would not suffice due to the possibility of a trivial solution where the model "cheats" by directly copying the speaker's mouth shapes from the input video to the generated result. This copying would lead to perfect visual quality and lip-sync accuracy since the copied mouth shapes would already synchronize with the input dubbing audio. However, such an approach would not provide any valuable learning for the model and would be considered a form of overfitting or generalization error known as *information leakage* [Hitaj *et al.* 2017; Yang *et al.* 2020]. To address this issue, the generator is forced to solely rely on the input dubbing audio to drive the dubbing process by masking (concealing) the speaker's mouth region using a black rectangular mask M. By doing so, the model is prevented from directly copying the mouth shapes or fine-grained details of the speaker's mouth from the input video to the generated result. The revised problem formulation is:

$$V^A = G(V^A_M, A; \theta). \tag{6.3}$$

While this measure prevents the model from copying the speaker's mouth movements from the input video using a mouth mask, all information related to the speaker's mouth region is erroneously removed as well. Consequently, the model would lack awareness of the appearance of the speaker's teeth, lips, and facial structure, making it unreasonable to expect the model to perform visual dubbing without this essential information. To address this issue, *reference frames* R are used, which are contiguous visual frames extracted from the input video from a different time slice than the input video and dubbing audio. Reference frames provide the model with the necessary appearance information it requires, and since the frames are out-of-sync, the model is discouraged from copying the mouth shapes from the reference frames. Therefore, the final problem formulation for training our ADVD solution is:

$$V^A = G(V^A_M, A, R; \theta). \tag{6.4}$$

Using the above theoretical formulation of the generator network, we provide a comprehensive description of its practical implementation. This discussion starts with a brief overview of the solution's philosophy, followed by a discussion regarding the network's inputs and architecture. Lastly, a discussion on the various experiments conducted is presented.

6.3 Solution Overview

Given the problem formulation above, we found it most appropriate to approach the ADVD problem from the perspective of *video inpainting* [Xu *et al.* 2019; Chang *et al.* 2019]. This choice is based on the observation that, for the visual channel, the mouth region of the frames to be dubbed is masked to prevent direct copying of mouth shapes from the input video. Simultaneously, reference frames are utilized to provide essential information about the appearance of the speaker's mouth region without introducing information leakage. On the other hand, the audio channel drives the dubbing process, determining the appropriate (dubbing) mouth shapes. Therefore, visual dubbing naturally becomes a matter of inpainting the speaker's masked mouth region with the appropriate mouth shape on a frame-by-frame basis. This approach eliminates the need for a series of post-processing operations that would have been required to insert the new mouth shape back into the scene [Prajwal *et al.* 2020]. In our unconstrained setting, where factors such as head pose are expected to vary significantly, the latter approach could lead to visual artefacts due to misalignment issues. We now elaborate on how we address the visual dubbing problem as one of video inpainting.

6.4 Input Representations

6.4.1 Visual Encoder

For each training video, frames containing the speaker's facial crops are extracted. Since the pre-trained lip-sync discriminator requires a five-frame input window, the generator is required to produce (dub) five frames at a time. A random facial crop f is selected to mark the start of the masked window V_M^A, and the four subsequent RGB facial crops are retrieved. All facial crops are resized to 96×96, resulting in a final masked window with shape $[B, C, T, H, W]$ = $[B, 3, 5, 96, 96]$, where B represents the batch size, C indicates the number of channels, T represents the window size, and H and W denote the spatial dimensions. The speaker's mouth region is masked by zeroing out the bottom half of all retrieved facial crops.

We now consider the practical significance of the masked window V_M^A and how it facilitates our approach to addressing the visual dubbing problem. Since we address the problem in a self-supervised setting, it is crucial to make maximal use of all accessible information. The upper half of the speaker's face, which remains exposed, contains valuable information that does not pertain to speech, such as background and facial actions (blinking). Additionally, the upper half provides vital pose information, enabling the decoder to seamlessly inpaint the speaker's masked mouth region with the appropriate mouth shape. Due to the crucial role that masked facial crops play in our solution, we refer to them as *pose priors* [Prajwal *et al.* 2020].

In addition to the window of pose priors, a set of reference frames R is required to expose the generator to the appearance of the speaker's mouth region while preventing information leakage. Previous approaches [Prajwal *et al.* 2020] achieve this by naively selecting a facial crop at random and retrieving the four subsequent facial crops to form a five-frame reference frame window. However, we identify this approach to be prone to information leakage. The random selection of the start frame for the reference frame window could result in an overlap with the masked window, inadvertently exposing ground-truth frames to the model as ref-

erence frames during training. To address this concern, we take a precautionary measure – after randomly selecting the start frame f of the masked window, the four frames before and after f are excluded when selecting the start frame of the five-frame reference frame window, as illustrated in Figure 6.1. The reference frames are then resized to 96×96, resulting in an input shape of $[B, 3, 5, 96, 96]$. The final input representation to the visual encoder, as shown in Figure 6.2(a), is formed by concatenating the masked and reference frame windows along the channel dimension, resulting in a final input size of $[B, 6, 5, 96, 96]$.

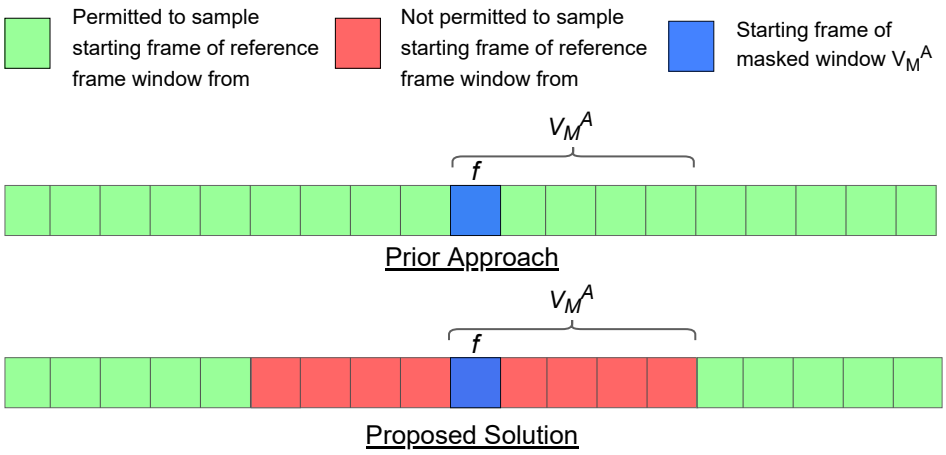


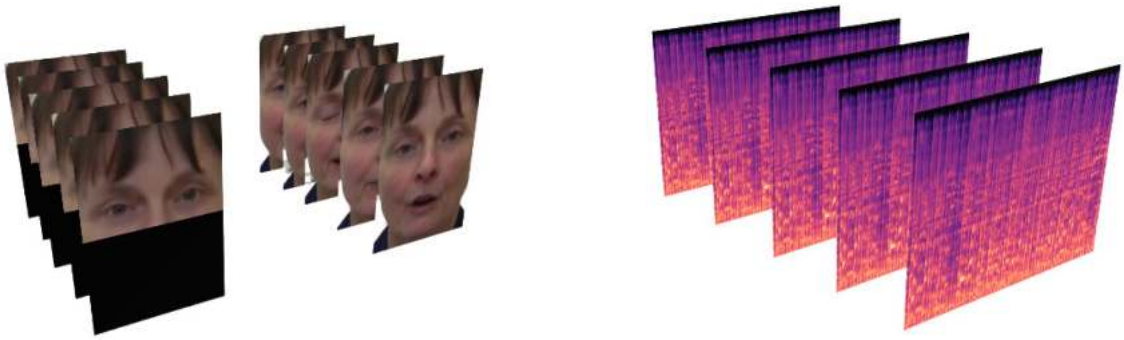
Figure 6.1: Comparison between the prior approach and the proposed solution to sampling the starting frame of the reference frame window. Notice that with the prior approach, there exists a risk of information leakage whereas with the presented approach, it is not possible for the reference frame window to overlap with the masked window V_M^A.

6.4.2 Audio Encoder

The input to the audio encoder complements the five-frame window used by the visual encoder. As discussed in Chapter 5, mel-spectrograms (instead of MFCCs) are adopted as our audio feature representation. The mel-spectrogram representation of the input dubbing audio is computed using hyperparameters inspired by state-of-the-art speaker-related solutions such as *DeepVoice3* [Ping *et al.* 2017] and *Tacotron2* [Shen *et al.* 2018] (as detailed in Appendix A.1). From the mel-spectrogram produced, five audio segments are cropped to match the five-frame window used by the visual encoder. Each mel-spectrogram excerpt represents 0.2s of audio and has a size of 80×16 as shown in Figure 6.2(b). This design ensures that each visual frame is centred on its corresponding mel-spectrogram segment, providing the model with the necessary temporal context before and after the frame being dubbed. This approach improves the temporal consistency and lip-sync accuracy achieved. The final input audio representation has a size of $[B, 5, 1, 80, 16]$.

6.5 Network Architecture

Akin to the lip-sync discriminator, the generator is composed of a visual encoder and an audio encoder due to the bi-modal nature of videos [Molholm *et al.* 2002]. Additionally, the



(a) Input to visual encoder, composed of a masked window and reference frame window.

(b) Input to audio encoder, composed of 5 stacked mel-spectrogram segments.

Figure 6.2: Example inputs to the generator network.

generator includes an image decoder responsible for producing the dubbed result on a frame-by-frame basis. After an extensive survey of recent GAN literature, we have opted for a deep residual U-Net generator [Ronneberger *et al.* 2015; Zhang *et al.* 2018], which we will elaborate on shortly. The renowned U-Net architecture is an encoder-decoder architecture with the addition of *skip connections* between intermediate encoder and decoder layers. These skip connections improve result quality by directly sharing a great deal of information between the input and output [Isola *et al.* 2017]. Our U-Net generator is specifically used within a residual learning setting [He *et al.* 2016], which has proven to ease the training process, improve generalizability, and prevent the vanishing gradient problem [He *et al.* 2020a]. We now analyse each sub-component of the generator in greater detail, discuss their role in performing visual dubbing, and explain the design choices made.

6.5.1 Visual Encoder

The visual encoder accepts the visual channel of the input video as input and encodes it to attain a compact representation of all pertinent visual information. After extensive experimentation with various architectures, we found that ResNetv1 blocks [He *et al.* 2016 2020a] achieved the best performance. Therefore, the visual encoder is composed of 13 of these blocks. Note that 2D convolutions suffice here since each pose prior is dubbed (inpainted) independently, given a set of reference frames and its corresponding mel-spectrogram segment. Additionally, all necessary information concerning temporal consistency (i.e., lip-sync) is provided by the pre-trained lip-sync discriminator. Due to the use of 2D convolutions, the input spatiotemporal volume with size $[B, 6, 5, 96, 96]$ needs to be amended before being passed as input to the network. This issue is addressed by flattening the temporal dimension into the batch dimension to attain a final input size of $[B \times 5, 6, 96, 96]$.

As the input progresses through the visual encoder, rich feature representations are learned as it is gradually downsampled (encoded) using strided convolutions. Intermediate feature representations are also passed to the image decoder through skip connections. In the end, all necessary visual information for visual dubbing is distilled into a compact 512D embedding vector. These skip connections, along with the embedding vector produced by the audio encoder, are used to generate the final dubbed result.

6.5.2 Audio Encoder

Since the presented solution is audio-driven, it was imperative to employ an audio encoder that extracts all essential audio information as effectively as possible. We use the same audio encoder as the lip-sync discriminator, consisting of 14 blocks composed of Conv2D → BatchNorm → ReLU. To accommodate the 2D convolutions, the audio window (with size $[B, 5, 1, 80, 16]$) is flattened into a final audio input size of $[B \times 5, 1, 80, 16]$. The audio encoder learns rich feature representations and encodes the input through strided convolutions, resulting in a compact 512D embedding vector that represents all essential audio information for visual dubbing. This embedding vector is then passed to the image decoder to produce the final dubbed result.

6.5.3 Image Decoder

The learned representations from the audio and visual encoders are conveyed to the image decoder, either as skip connections or embedding vectors, to generate the final dubbed result. As the input passes through the decoder, it is gradually decoded until the resulting dubbed frames, with spatial dimensions of 96×96, are produced. The image decoder is composed of 13 ResNetv1 blocks [He *et al.* 2016 2020a]. We make the notable design choice to upsample by performing a bilinear upsampling operation followed by a 2D convolution as opposed to performing a transpose convolution. This design choice follows from the observation that results produced by most GAN-based solutions that employ transpose convolutions tend to exhibit an unpleasant checkerboard effect [Odena *et al.* 2016]. After obtaining the final result, a sigmoid operation is performed to ensure that all pixel values are bounded within the range $[0, 1]$.

6.6 Training

We now discuss the loss functions employed, which may be seen as the driving force behind training the generator to produce sensible results. The sync loss, generated by the lip-sync discriminator, is the only loss used to encourage the generator to produce dubbed results with accurate lip-sync. To compute the sync loss, the lower half (i.e., the mouth region) of the five dubbed facial crops synthesized by the generator is extracted and used as input to the visual encoder of the lip-sync discriminator. Concurrently, the mel-spectrogram segment spanning this visual window (representative of 0.2s of footage) is passed as input to the audio encoder. Both the audio and visual encoders produce a 512D embedding vector, which is then used to compute the cosine similarity measure P_{sync} (Equation (5.1)) representing the probability that the audio and visual inputs are in-sync. The final sync loss $\mathcal{L}_{sync}$ conveyed to the generator is obtained by computing the BCE loss between the produced cosine similarity measure and the ground-truth label ($y = 1$), as the visual input, i.e., the synthesized result, is required to be in-sync with the input audio.

We now discuss the loss functions used to optimize for visual quality. With the plethora of loss functions available, each with its own strengths, weaknesses, and specific outcomes, conducting an extensive survey was imperative. The choice of loss function(s) is one of the most crucial design considerations, sometimes even more important than the network architecture

[Zhao *et al.* 2016]. Following a comprehensive analysis of several loss functions adopted in relevant fields such as image inpainting [Pathak *et al.* 2016; Demir and Unal 2018; Chang *et al.* 2019], super-resolution [Shi *et al.* 2016; Ledig *et al.* 2017; Wang *et al.* 2018], and image synthesis [Isola *et al.* 2017], we elaborate on each loss function that we employ and discuss its objective, strengths, weaknesses, and applicability when addressing the ADVD problem.

Upon surveying the array of loss functions that have been previously used to optimize visual quality, we discover that the most commonly used loss function is the L_2 (MSE) loss, given as:

$$\mathcal{L}_2(\hat{Y}, Y) = \frac{1}{H \times W} \sum_{x=1}^{W} \sum_{y=1}^{H} (\hat{Y}_{x,y} - Y_{x,y})^2. \tag{6.5}$$

Here, $\hat{Y}$ represents the synthesized image, while Y denotes the ground-truth image. The widespread adoption of the L_2 loss can be attributed primarily to its simplicity [Zhao *et al.* 2016]. Upon closer examination of the nature of the L_2 loss, we observe that it is a *per-pixel loss function* aimed at achieving correspondence between the generated result and ground-truth data on a pixel-by-pixel basis. Furthermore, the L_2 loss is designed to penalize large errors while being more tolerant of small errors. In contrast, the L_1 loss, presented in Equation (6.6), is another per-pixel loss function that is less sensitive to large errors [Zhao *et al.* 2016].

$$\mathcal{L}_1(\hat{Y}, Y) = \sum_{x=1}^{W} \sum_{y=1}^{H} |\hat{Y}_{x,y} - Y_{x,y}| \tag{6.6}$$

The L_2 loss is intrinsically a Peak Signal-to-Noise Ratio (PSNR) oriented loss function, aiming to maximize the PSNR measure. PSNR is a full-reference image quality assessment metric computed as:

$$PSNR(\hat{Y}, Y) = 10 \log_{10} \left(\frac{R^2}{\mathcal{L}_2(\hat{Y}, Y)} \right), \text{ where } R = 255. \tag{6.7}$$

As extensively highlighted in the literature [Ledig *et al.* 2017; Wang *et al.* 2018; Yun *et al.* 2020; Wu *et al.* 2020a; Jiang *et al.* 2021b], attempting to directly optimize the PSNR measure is fundamentally flawed because it does not correlate well with the human visual system (HVS). Kovalenko [2017] shows that a higher PSNR measure does not necessarily mean an image is more visually pleasing to a human viewer than an image with a lower PSNR measure. This counterintuitive result is because the L_2 loss maximizes the PSNR measure by averaging several plausible results. However, this inadvertently reduces the visual quality as perceived by a human, resulting in over-smoothed and blurry outputs that lack fine-grained details. Achieving high-frequency details such as skin texture, facial hair, and teeth, is highly desirable, particularly in the context of visual dubbing.

Based on the discussion presented above, we employ the L_1 loss instead of the L_2 loss for our baseline implementation. The L_1 loss has been adopted as the de-facto standard for numerous generative tasks because it produces noticeably less blurry results compared to the L_2 loss

[Zhao *et al.* 2016]. Section 6.7.1 further discusses the various intriguing loss functions that we experimented with in an attempt to improve the visual quality achieved.

To regulate the prioritization of visual quality and lip-sync accuracy achieved, which is particularly crucial when addressing the ADVD problem due to the trade-off between them [Prajwal *et al.* 2020], we assign weights to the corresponding loss functions. This task is further complicated by the lip-sync discriminator being pre-trained, making it essential to balance the influence of both the generator and the discriminator to prevent GAN collapse [Goodfellow *et al.* 2014]. Taking inspiration from Wav2Lip [Prajwal *et al.* 2020], the only other visual dubbing solution that pre-trains the lip-sync discriminator, we compute the sync loss $\mathcal{L}_{sync}$ periodically during training but delay its incorporation (i.e., backpropagation) to train the generator. This approach allows the network to first optimize for visual quality and subsequently lip-sync accuracy. However, we observed that the strategy of Prajwal *et al.* [2020] was not suitable for our purposes as our more powerful lip-sync discriminator optimized the sync loss too quickly, regardless of the delay in introducing it. To address this issue, we introduce the sync loss after 400K iterations, based on empirical observation, dedicating sufficient iterations to optimize visual quality. Additionally, we empirically set $\lambda_{sync} = 0.25$ to attenuate the feedback from the lip-sync discriminator, preventing it from becoming overpowering. The final generator loss is given as:

$$\mathcal{L}_G = \lambda_{sync} \cdot \mathcal{L}_{sync} + (1 - \lambda_{sync}) \cdot \mathcal{L}_1. \tag{6.8}$$

The generator is initialized with random weights, and based on an extensive hyperparameter search, is trained with a batch size of 32 and the Adam optimizer [Kingma and Ba 2014] with a learning rate of 0.0005 ($\beta_1 = 0.9$, $\beta_2 = 0.999$).

6.7 Experiments

6.7.1 Perceptually-motivated Loss Functions

Based on the discussion presented above, it was established that optimizing PSNR-oriented loss functions is not an ideal approach to achieving *perceptually pleasing* results for humans. Consequently, this necessitates a shift in focus from maximizing the PSNR measure to striving for results that are perceptually pleasing to humans. This involves seeking an alternate measure to optimize that is more closely related to the HVS. Upon assessing the literature on relevant image-generative tasks [Ledig *et al.* 2017; Wang *et al.* 2018; Yun *et al.* 2020; Wu *et al.* 2020a; Jiang *et al.* 2021b], *perceptually-motivated* loss functions have consistently proven to outperform PSNR-oriented solutions in terms of visual quality.

Two avenues exist for formulating a perceptually-motivated loss function. The first involves selecting an image quality assessment metric closely correlated with the human visual system (e.g., SSIM [Wang *et al.* 2004]) and transforming it into a loss function. The second, and prevailing, approach calculates the loss in the feature space, using features extracted by a pre-trained network such as VGG-16 or VGG-19 [Simonyan and Zisserman 2014; Johnson *et al.* 2016], instead of in the pixel space. Despite the significant improvement in visual quality seen in other image-related tasks using perceptually-motivated loss functions, their adoption in the field of visual dubbing is surprisingly limited. Consequently, the effect of both types of perceptually-motivated loss functions on the visual quality achieved is investigated. Below,

we elaborate on the two forms of perceptually-motivated loss functions experimented with, discuss how they work, and explain how they were configured for our experiments.

6.7.1.1 MS-SSIM + $\mathcal{L}_1$ Loss

One of the most commonly used perceptually-motivated visual quality metrics is the *Structural Similarity Index Measure* (*SSIM*) [Wang *et al.* 2004]. While providing a comprehensive overview of the formulation and rationale of the SSIM metric is beyond the scope of this work, we focus on the salient points relevant to us and direct interested readers to Wang *et al.* [2004] for a detailed explanation. In summary, the SSIM metric is based on the assumption that the HVS is highly adapted for extracting structural information from a scene. Consequently, the SSIM metric measures the degradation of structural information between the generated result and ground-truth data. In this context, structural information refers to the idea that pixels have strong dependencies amongst each other, especially spatially proximate pixels, and these dependencies contain information pertaining to the structure of objects in the scene. Furthermore, the SSIM metric is computed based on three aspects of the generated result and ground-truth data i.e., luminance, contrast, and structure.

A key property of the SSIM metric is that it is computed locally instead of globally. Rather than assessing image quality based on the entire image at once, the SSIM metric is calculated at a patch level, sliding over the entire image [Seshadrinathan *et al.* 2009]. The reasons for this are four-fold:

- Image statistical features are usually highly spatially nonstationary
- Image distortions, which may or may not be dependent on the local image statistics, may also be space-invariant
- At typical viewing distances, only one region of the image can be perceived with high resolution by a human observer at any given time (due to the foveation feature of the HVS [Geisler and Banks 1995; Wang and Bovik 2001])
- Localized quality assessment can provide a spatially varying quality map of the image which provides more information regarding the quality degradation between the generated result and the ground-truth data

When computing the SSIM metric between a patch in the generated result and ground-truth data (which we denote as x and y respectively), a Gaussian filter G_S with standard deviation S is employed. Using the image statistics computed for these patches, as determined by the support of G_S, the SSIM metric for patches x and y is computed as:

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{6.9}$$

where :

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{6.10}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{6.11}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{6.12}$$

$$\therefore SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{6.13}$$

Appendix A.2 provides a description of each symbol in Equations (6.9) – (6.13) in the interest of significantly improved readability. The SSIM metric is calculated for each patch in the generated result $\hat{Y}$ and the ground-truth image Y in a sliding-window manner, where the SSIM metric is computed centred on each pixel. The final SSIM score, which measures the image quality, is obtained by averaging all P patch-level SSIM scores, i.e.:

$$MSSIM(\hat{Y}, Y) = \frac{1}{P}\sum_{j=1}^{P} SSIM(x_j, y_j). \tag{6.14}$$

Ultimately, the SSIM loss function is given as:

$$\mathcal{L}_{SSIM}(\hat{Y}, Y) = \frac{1}{P}\sum_{j=1}^{P} 1 - SSIM(x_j, y_j). \tag{6.15}$$

Upon reviewing the literature [Wang *et al.* 2003; Zhao *et al.* 2016], it has been demonstrated that the choice of S significantly impacts the quality of results achieved when using the loss function presented above. Smaller values of S reduce the network's ability to preserve the local structure and introduce splotchy artefacts in flat regions. On the other hand, larger values of S cause the network to introduce noise in the vicinity of edges. Instead of fine-tuning S and computing the SSIM at a single scale, it has been shown that significantly improved results can be obtained by computing the SSIM at multiple scales, giving rise to *Multi-Scale SSIM* (*MS-SSIM*) [Wang *et al.* 2003]. MS-SSIM considers the scale at which local structure should be analysed, accounting for factors such as image-to-observer distance, achieved by weighting the SSIM computed at different scales based on the sensitivity of the HVS. Following common practice, we compute the SSIM at $M = 5$ scales, i.e., $\{0.5, 1.0, 2.0, 4.0, 8.0\}$, which is equivalent to iteratively downsampling the image by a factor of two and computing the SSIM. Therefore, the MS-SSIM and $\mathcal{L}_{MS-SSIM}$ are computed as:

$$MS\text{-}SSIM(x, y) = [l_M(x, y)]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j}, \tag{6.16}$$

$$\therefore \mathcal{L}_{MS-SSIM}(\hat{Y}, Y) = \frac{1}{P}\sum_{j=1}^{P} 1 - MS\text{-}SSIM(x_j, y_j). \tag{6.17}$$

By design, MS-SSIM and SSIM are not particularly sensitive to uniform biases, which may cause colour shifts or variations in brightness, often resulting in duller results [Zhao *et al.* 2016]. However, MS-SSIM has proven to preserve the contrast in high-frequency regions better than most other loss functions, such as L_1, L_2, and SSIM, among others. On the other hand, the L_1 loss preserves colour and luminance well but does not accurately capture contrast as effectively as MS-SSIM. Due to the complementary nature of these two loss functions, combining them into a single loss function, aptly named *MS-SSIM+* L_1 loss, allows the model to benefit from the best properties of both. The *MS-SSIM+* L_1 loss is given as:

$$\mathcal{L}_{MS-SSIM+L_1} = \delta \cdot \mathcal{L}_{MS-SSIM} + (1-\delta) \cdot \mathcal{L}_1. \tag{6.18}$$

δ in Equation (6.18) is a scalar used to control the contribution of each term (i.e., whether the MS-SSIM or L_1 term should be prioritized). Based on empirical analysis, δ is set to 0.5 which achieved the best trade-off between visual quality and lip-sync accuracy. When employing the $\mathcal{L}_{MS-SSIM+L_1}$ loss, the generator loss becomes:

$$\mathcal{L}_G = \lambda_{sync} \cdot \mathcal{L}_{sync} + (1-\lambda_{sync}) \cdot \mathcal{L}_{MS-SSIM+L_1} \tag{6.19}$$

6.7.1.2 Feature-based Perceptual Loss Functions

In contrast to computing the loss on a per-pixel basis, feature-based perceptual loss functions compute a distance measure (i.e., a loss) between feature representations of the produced result and ground-truth data. The notion of feature-based perceptual loss functions was first introduced by Johnson *et al.* [2016] which showed that computing the loss in the feature space, as opposed to in the pixel space, can significantly improve the perceptual quality achieved. The premise behind perceptual losses is, instead of encouraging the generated result to correspond exactly (i.e., at a pixel level) with the ground truth, the objective is to minimize the (normalized L_2) distance between their feature representations as extracted by a pre-trained network. Specifically, let $\phi_j(x)$ be the activations of the jth layer of a pre-trained network ϕ when processing image x; if j is a convolutional layer, then $\phi_j(x)$ will be a feature map with shape $C_j \times H_j \times W_j$, thus, the perceptual loss may be represented as:

$$\mathcal{L}_{percep}(\hat{Y}, Y) = \frac{1}{C_j \times H_j \times W_j} \left\| \phi_j(\hat{Y}) - \phi_j(Y) \right\|_2^2. \tag{6.20}$$

To gain an intuition of this design choice, consider a generated result that achieves high perceptual quality, but the object of interest is shifted slightly relative to the ground truth. In such a case, a per-pixel loss would incur a substantial penalty, even though the result is of high perceptual quality. On the other hand, a feature-based perceptual loss would be more tolerant, as it focuses on the overall perceptual quality. Due to their effectiveness, feature-based perceptual losses have become the de-facto standard for addressing the shortcomings of PSNR-oriented solutions [Ledig *et al.* 2017; Wang *et al.* 2018; Jiang *et al.* 2021b].

One of the most notable characteristics of feature-based perceptual losses is that they entail employing a pre-trained image classification network for feature extraction purposes. This follows since these networks have already learned to encode the perceptual and semantic

information that we would like to measure using the loss function. In addition, this allows for high-level differences, such as content and style, to be compared between the generated result and the ground truth. Common choices for pre-trained feature extraction networks are the VGG-16 and VGG-19 networks pre-trained on the ImageNet dataset [Deng *et al.* 2009; Simonyan and Zisserman 2014]. Differently, however, since the presented visual dubbing solution is an example of a facial application, a VGG-16 network pre-trained on the VGG-Face dataset [Parkhi *et al.* 2015] is used. This design choice follows from the aspiration that employing a network pre-trained on a facial dataset would learn features specific to the human face, thus, improving the perceptual quality achieved.

Another distinguishing aspect of our approach to feature-based perceptual losses is that traditionally, these losses are computed on *post-activated* features, which are features passed through an activation function. Wang *et al.* [2018] have shown that post-activated features tend to be sparse because the ReLU operator zeros out large regions of the feature maps. This sparsity provides weak supervision, resulting in sub-optimal outcomes, such as inconsistent reconstructed brightness compared to the ground truth. To address this issue, Wang *et al.* [2018] propose to employ features prior to activation, known as *pre-activated* feature maps, which offer stronger supervision, leading to sharper edges and improved visual quality.

Lastly, feature-based perceptual losses typically utilize features extracted from a single layer of the pre-trained network. However, due to the hierarchical nature of CNNs, it has been shown that early layers preserve low-level features such as colour and texture, while higher layers capture image content and spatial structure more accurately [Niu *et al.* 2018]. To strike a balanced trade-off, we adopt the approach of Wang *et al.* [2021b] and incorporate features from five layers i.e., `relu11`, `relu21`, `relu31`, `relu41`, and `relu51` of our pre-trained VGG-16 network, along with their corresponding weights, i.e., 0.03125, 0.0625, 0.125, 0.25, and 1.0, respectively. Since feature-based perceptual losses may not perform optimally in isolation, we combine them with the L_1 loss to mitigate the blurriness caused by the latter. In practice, we empirically set $\lambda_{percep} = 1e-6$ to represent the weight of the feature-based perceptual loss, resulting in the revised generator loss:

$$\mathcal{L}_G = \lambda_{sync} \cdot \mathcal{L}_{sync} + \lambda_{percep} \cdot \mathcal{L}_{percep} + (1 - \lambda_{sync} - \lambda_{percep}) \cdot \mathcal{L}_1. \tag{6.21}$$

6.7.2 Gradual Introduction of Sync Loss $\mathcal{L}_{sync}$

While the approach proposed by Wav2Lip [Prajwal *et al.* 2020] has proven to achieve satisfactory visual quality and lip-sync accuracy, the manner in which the sync loss is introduced may be deemed sub-optimal. The solution initially optimizes the visual quality and progresses towards a sense of convergence/stability; however, this is abruptly followed by the introduction of the sync loss. This sudden shock to an already (somewhat) trained model necessitates radical weight updates caused by the introduction of the sync loss. We hypothesize that the radical weight updates caused by the abrupt introduction of the sync loss in the late stages of training may degrade the rate of convergence and/or the quality of results achieved.

The phenomenon described above is comparable to the well-known PGGAN solution [Karras *et al.* 2017b], which starts by training on low-resolution data and gradually adds layers to the generator and discriminator to accommodate higher-resolution data as training advances.

Instead of introducing new layers abruptly, which could lead to radical weight updates as well, the authors employ a gradual phasing-in technique. They achieve this by weighting the new layers using a variable τ that linearly increases from zero to one. This transitional approach has demonstrated its ability to expedite training and greatly enhance stability.

To test our hypothesis, we draw inspiration from the PGGAN solution and introduce a variable τ to further weight the sync loss, which is introduced from the beginning of training. This weight τ gradually increases from zero to one as training progresses, i.e.:

$$\mathcal{L}_G = \underset{0\to1}{\tau} \cdot \lambda_{sync} \cdot \mathcal{L}_{sync} + (1 - \lambda_{sync}) \cdot \mathcal{L}_1. \tag{6.22}$$

Instead of increasing τ indefinitely during training, we specify a training iteration I at which we stop increasing τ, after which τ is fixed at one. Several experiments were conducted to determine the optimal value of I, where $I \in \{300K, 400K, 500K, 600K, 700K\}$. This training enhancement aims to improve the rate of convergence by simultaneously optimizing visual quality and lip-sync accuracy, rather than sequentially.

6.7.3 Concatenated Embedding Vectors

After conducting a thorough analysis of the state-of-the-art Wav2Lip solution [Prajwal *et al.* 2020], we observed that the visual information received by the image decoder from the visual encoder is exclusively in the form of skip connections. While valuable, the 512D embedding vector produced by the visual encoder remains unused, unlike the 512D embedding vector produced by the audio encoder, which is passed as input to the image decoder. We aim to investigate whether conveying visual information to the image decoder using the produced embedding vector, in addition to skip connections, can lead to improvements. In theory, the audio and visual embedding vectors should be similar (i.e., located nearby in the joint embedding space spanned by the two encoders) for in-sync videos and significantly different for out-of-sync videos. However, directly concatenating the 512D audio and visual embedding vectors is impractical due to the resulting 1024D embedding vector's size, which would require a large image decoder for decoding. Instead, we append an MLP (FC layer → BatchNorm → ReLU→ FC layer → ReLU) to the top of the audio and visual encoders to transform the 512D embedding vectors into 256D embedding vectors. Subsequently, the two 256D embedding vectors are concatenated to form a single 512D embedding vector, containing both audio and visual data, which is then passed as input to the image decoder.

6.7.4 Visual Quality Discriminator

Prajwal *et al.* [2020] observed that employing a powerful pre-trained lip-sync discriminator caused the generator to produce blurry mouth shapes and occasionally introduce visual artefacts, thereby compromising the visual quality achieved. To address this issue, a visual quality discriminator is used, which is trained adversarially alongside the generator. The discriminator consists of 13 blocks (Conv2D → LeakyReLU) and has approximately 14M parameters. Although this enhancement has proven to improve the visual quality, it has also shown to compromise the lip-sync accuracy.

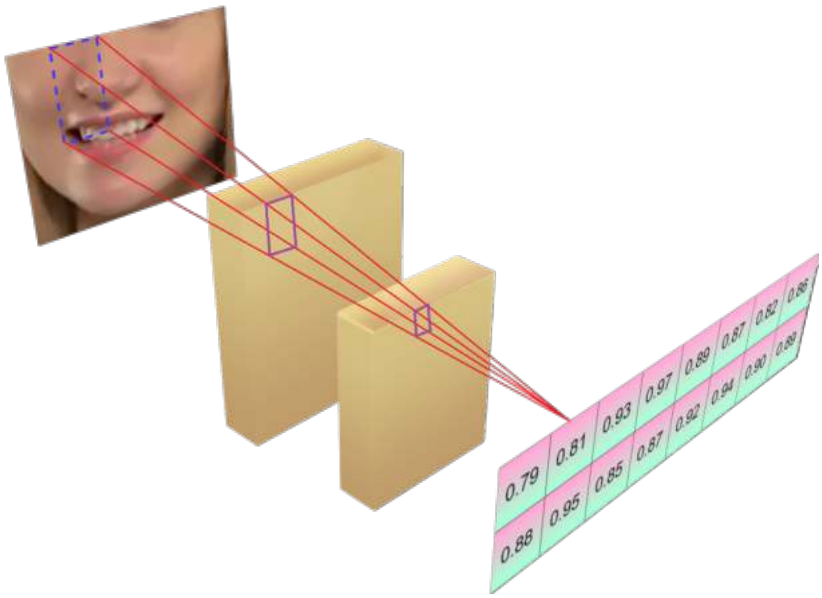


Figure 6.3: PatchGAN Discriminator. Each value of the output matrix represents the probability of whether it corresponding patch in the input image is real.

We conduct experiments using a visual quality discriminator, aiming to improve its design without relying solely on the network's size to achieve satisfactory results. Additionally, we seek to enhance the trade-off between visual quality and lip-sync accuracy by employing a PatchGAN discriminator [Isola *et al.* 2017] which has played an instrumental role in the success of various state-of-the-art GAN-based solutions [Demir and Unal 2018; Chang *et al.* 2019; Jiang *et al.* 2021c]. Unlike a regular discriminator that outputs a scalar representing the probability of an input image being real [Goodfellow *et al.* 2014], the PatchGAN discriminator penalizes at the level of local patches and generates a grid of responses. Each output element represents the probability that its corresponding patch in the input image, i.e., its receptive field, is real [Isola *et al.* 2017]. This capability allows the discriminator to provide more detailed feedback to the generator regarding how to improve the visual quality of the synthesized results.

Our visual quality discriminator is trained in a traditional GAN setting which accepts alternating batches of real and synthesized images as input. Since the generator is responsible for inpainting the appropriate mouth shape into the speaker's mouth region and borrows the upper half of the speaker's face from the ground-truth, the discriminator is solely exposed to the lower half of the speaker's face. Using the speaker's mouth region as input with size $[B \times 5, 3, 48, 96]$, the PatchGAN discriminator produces a 2×8 grid of responses for each image whereby each element represents the probability that its corresponding 24×12 patch in the input is real as shown in Figure 6.3.

In an attempt to increase the model's capacity compared to a standard PatchGAN discriminator, we duplicate each layer of the network, effectively doubling its size, resulting in 4M parameters. This makes the discriminator 3.5× smaller than the visual quality discriminator used by Wav2Lip [Prajwal *et al.* 2020]. The visual quality discriminator is trained using the Adam optimizer [Kingma and Ba 2014] with a learning rate of $1e-4$, based on empirical

analyses. Additionally, we experimented with two adversarial losses: BCE and Least Squares GAN (LSGAN) to determine which one achieves a better trade-off between visual quality and lip-sync accuracy. For the conducted experiments, we empirically set $\lambda_{adv} = 0.45$ to represent the weight of the loss received from the visual quality discriminator (denoted as $\mathcal{L}_{adv}$), and revise the generator loss to be:

$$\mathcal{L}_G = \lambda_{sync} \cdot \mathcal{L}_{sync} + \lambda_{adv} \cdot \mathcal{L}_{adv} + (1 - \lambda_{sync} - \lambda_{adv}) \cdot \mathcal{L}_1. \tag{6.23}$$

6.7.5 Relativistic Discriminator

In a standard GAN, the discriminator loss function is based on the Jensen-Shannon divergence (JSD) [Goodfellow *et al.* 2014]. The calculation of JSD involves solving the following maximum problem:

$$JSD(\mathbb{P}\|\mathbb{Q}) = \frac{1}{2}\left(\log(4) + \max_{D:X\to[0,1]} \mathbb{E}_{x_r\sim\mathbb{P}}[\log(D(x_r))] + \mathbb{E}_{x_f\sim\mathbb{Q}}[\log(1 - D(x_f))]\right). \tag{6.24}$$

The JSD is minimized i.e., $JSD(\mathbb{P}\|\mathbb{Q}) = 0$, when $D(x_r) = D(x_f) = 0.5$ for all x_r in $\mathbb{P}$ (the real data distribution) and x_f in $\mathbb{Q}$ (the learned data distribution) where x_r and x_f represent real and fake data respectively, and is maximized i.e., $JSD(\mathbb{P}\|\mathbb{Q}) = \log(2)$, when $D(x_r) = 1$ and $D(x_f) = 0$ for all x_r in $\mathbb{P}$ and x_f in $\mathbb{Q}$ [Jolicoeur-Martineau 2018]. This suggests that, if we were to directly minimize the divergence from maximum to minimum, $D(x_r)$ is expected to smoothly decrease from one to 0.5, and $D(x_f)$ to smoothly increase from zero to 0.5. When the GAN is in equilibrium, the generator (parameterized by θ) and discriminator (parameterized by w) are optimized and $D(x_r) = D(x_f) = 0.5$. Surprisingly, however, this is not the case with the standard GAN since, when the generator is trained to synthesize fake data in an attempt to make the discriminator classify the fake data as real, the discriminator is encouraged to output one i.e., $D(x_f) = 1$. Evidently, this approach deviates from the JSD minimization, which instead requires encouraging the discriminator to output 0.5 as opposed to one. In addition to minimizing the JSD, we note that whilst the standard GAN attempts to increase the probability that fake data is classified as real, no (explicit) attempt is made to decrease $D(x_r)$ to 0.5 i.e., to decrease the probability that the real data is classified as real as discussed above.

Letting J denote the Jacobian and $C(x) \in \mathcal{F}$ (the class of functions assigned by the Integral Probability Metric (IPM) [Jolicoeur-Martineau 2018]), the gradient when employing the standard GAN can be represented as:

$$\nabla_w L_D = -\mathbb{E}_{x_r\sim\mathbb{P}}[(1 - D(x_r))\nabla_w C(x_r)] + \mathbb{E}_{x_f\sim\mathbb{Q}_\theta}[D(x_f)\nabla_w C(x_f)] \tag{6.25}$$

$$\nabla_\theta L_G = -\mathbb{E}_{z\sim\mathbb{P}_z}[(1 - D(G(z)))C(G(z))\nabla_x C(G(z))J_\theta G(z)] \tag{6.26}$$

When the discriminator is optimal, $(1 - D(x_r)) \to 0$ which indicates that the gradients for the discriminator predominantly come from fake data (i.e., the second term in Equation (6.25)).

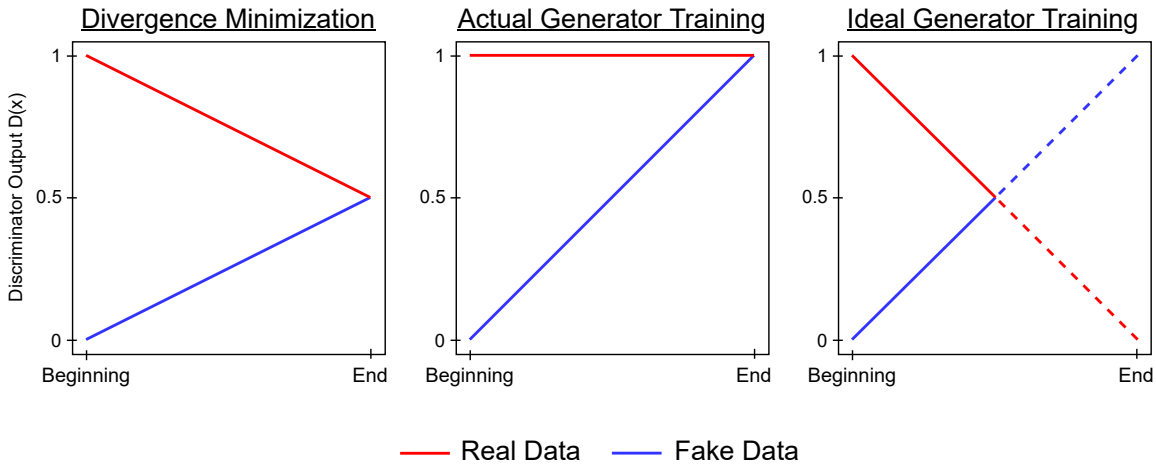


Figure 6.4: Expected discriminator output of the real and fake data for the direct minimization of the JSD, actual training of the generator when minimizing its loss function, and ideal training of the generator to minimize its loss function (lines are dotted when they cross beyond the equilibrium to signify that this may or may not be necessary) respectively. Figure adapted from Jolicoeur-Martineau [2018].

The discriminator stops learning from real data and only learns from fake data and at this point, the standard GAN no longer learns how to make the result more natural.

In an attempt to address the aforementioned issues, Jolicoeur-Martineau [2018] proposes the *relativistic discriminator* which, in contrast to a traditional discriminator which predicts the probability that the input data is real, estimates the probability that the given real data is more realistic than randomly sampled fake data (or vice versa) instead. Note, the relativistic discriminator is not an additional discriminator to our solution but instead enhances how our visual quality discriminator (and its corresponding loss function) is utilized. To make a discriminator relativistic i.e., having the output of D dependent on real and fake data, we let $C(x)$ represent the non-transformed output (i.e., $D(x) = sigmoid(C(x))$ in the case of the BCE loss), sample real and fake data $\tilde{x} = (x_r, x_f)$, and define $D(\tilde{x}) = sigmoid(C(x_r) - C(x_f))$. Using the relativistic discriminator, the corresponding (non-saturating) loss functions for the generator and discriminator (using the BCE loss) evolves from Equations (6.27) and (6.28) to Equations (6.29) and (6.30).

$$L_D = -\mathbb{E}_{x_r \sim \mathbb{P}}[\log(sigmoid(C(x_r)))] - \mathbb{E}_{x_f \sim \mathbb{Q}}[\log(1 - sigmoid(C(x_f)))] \tag{6.27}$$

$$L_G = -\mathbb{E}_{x_f \sim \mathbb{Q}}[\log(sigmoid(C(x_f)))] \tag{6.28}$$

$$L_D = -\mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})}[\log(sigmoid(C(x_r) - C(x_f)))] \tag{6.29}$$

$$L_G = -\mathbb{E}_{(x_r, x_f) \sim (\mathbb{P}, \mathbb{Q})}[\log(sigmoid(C(x_f) - C(x_r)))] \tag{6.30}$$

Jolicoeur-Martineau [2018] empirically discovered that when employing the formulation of the relativistic discriminator presented in Equations (6.29) and (6.30), the computed values exhibited high variance with large swings from different samples. As a result, they introduced a variant called the *relativistic average discriminator* (*RaGAN*), which we adopt. This variant

estimates the average probability that the given real data is more realistic than fake data. The revised formulation is given as:

$$L_D = -\mathbb{E}_{x_r \sim \mathbb{P}} \left[\log\left(\tilde{D}(x_r)\right)\right] - \mathbb{E}_{x_f \sim \mathbb{Q}}[\log(1 - \tilde{D}(x_f))] \tag{6.31}$$

$$L_G = -\mathbb{E}_{x_f \sim \mathbb{Q}} \left[\log\left(\tilde{D}(x_f)\right)\right] - \mathbb{E}_{x_r \sim \mathbb{P}}[\log(1 - \tilde{D}(x_r))] \tag{6.32}$$

$$\tilde{D}(x_r) = sigmoid(C(x_r) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f)) \tag{6.33}$$

$$\tilde{D}(x_f) = sigmoid(C(x_f) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r)) \tag{6.34}$$

In the case of our experiment in which the LSGAN loss function is employed, we also experiment with its relativistic counterpart which evolves the loss functions from Equations (6.35) and (6.36), to Equations (6.37) and (6.38).

$$L_D = \mathbb{E}_{x_r \sim \mathbb{P}}[(C(x_r) - 1)^2] + \mathbb{E}_{x_f \sim Q}[(C(x_f) - 0)^2] \tag{6.35}$$

$$L_G = \mathbb{E}_{x_f \sim \mathbb{Q}}[(C(x_f) - 1)^2] \tag{6.36}$$

$$L_D = \mathbb{E}_{x_r \sim \mathbb{P}}[(C(x_r) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f) - 1)^2] + \mathbb{E}_{x_f \sim \mathbb{Q}}[(C(x_f) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r) + 1)^2] \tag{6.37}$$

$$L_G = \mathbb{E}_{x_f \sim \mathbb{P}}[(C(x_f) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r) - 1)^2] + \mathbb{E}_{x_r \sim \mathbb{P}}[(C(x_r) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f) + 1)^2] \tag{6.38}$$

Our inspiration for experimenting with the relativistic discriminator comes from the impressive empirical findings of Jolicoeur-Martineau [2018], as well as the significantly improved results achieved by several state-of-the-art image transformation solutions [Wang *et al.* 2018; Jiang *et al.* 2021c; Du *et al.* 2021] that have adopted it. In summary, the relativistic discriminator has shown to improve the rate of convergence, significantly enhance stability, and achieve higher-quality results in GANs without any additional computational cost [Jolicoeur-Martineau 2018].

6.8 Inference

Recall that during training, the input to the audio and visual encoders was extracted from the same video, ensuring that the speaker's mouth movements were initially in-sync with the input dubbing audio. However, the speaker's mouth region was masked to prevent information leakage [Gutierrez 2014]. This self-supervised approach allowed for the abundance of talking-face videos available online to be exploited for training. On the other hand, during inference, the process involves dubbing real-world videos using audio and visual segments not present in the dataset. As a result, the input audio and video segments no longer correspond to each other, and the speaker's original mouth movements are no longer in sync with the input dubbing audio, as they were during training. We commence by discussing how the inputs to our solution are prepared during inference.

6.8.1 Input Representations

6.8.1.1 Visual Encoder

The process begins by discretizing the input talking-face video into its corresponding set of visual frames. For each visual frame, face detection [Zhang *et al.* 2017b] is performed, and if the frame contains multiple faces, the largest face (in terms of the area of the facial bounding boxes) is assumed to be the active speaker's face. Since performing face detection on a frame-by-frame basis may result in oversensitive (jittery) face detections, we smoothen the current facial bounding box by averaging over the face detections of the four subsequent frames. The resulting facial bounding box coordinates serve two key purposes i.e., to demarcate the region within the frame that is cropped to retrieve the corresponding facial crop, and secondly, they allow for the dubbed mouth shapes to be seamlessly pasted into the speaker's mouth region.

The input to the visual encoder consists of the current (smoothened) facial crop, which is resized to 96×96 to create a batched window with a size of $[B, 3, 96, 96]$. This input representation is slightly different from the one used during training. However, both representations are equivalent since each frame is dubbed independently. This modification allows for videos to be dubbed where the number of audio or visual frames no longer needs to be a multiple of five, and it also streamlines the inference process. Another copy of this window is created, in which the speaker's mouth region is masked and later inpainted with the appropriate mouth shape. The original copy of the window serves as the reference frame window, providing information to the model regarding the appearance of the speaker's mouth region. Here, it is acceptable to expose visual frames from the input video as reference frames to the model since the speaker's mouth movements are originally out-of-sync, making information leakage impossible. The two windows are then concatenated along the channel dimension to create a final input representation with a size of $[B, 6, 96, 96]$.

6.8.1.2 Audio Encoder

This process begins by enforcing the input dubbing audio to have a sampling rate of 16KHz, after which the corresponding mel-spectrogram is computed using the same audio hyperparameters used during training. For each input visual frame, its corresponding mel-spectrogram segment (i.e., audio frame) with size 80×16 is extracted, thus, forming a final batched input representation with size $[B, 1, 80, 16]$.

6.8.2 Inference Process

Using the audio and visual inputs as detailed above, the generator produces dubbed facial crops with the appropriate mouth shapes inpainted into the lower half of the facial crops. For most existing solutions [KR *et al.* 2019; Prajwal *et al.* 2020], the dubbing process hereon is simply a matter of superimposing the dubbed facial crops onto the speaker's face in the original visual frames. Despite not being evident when dubbing low-resolution videos, the consequence of adopting such as simplistic approach becomes apparent when dubbing higher-resolution videos as shown in Figures 7.1 and 7.2, and elaborated on in Chapter 7. Specifically, the results tend to blur the entire speaker's face, including the upper half. The blurring is most noticeable near the speaker's mouth region, creating an unpleasant box effect. This may happen due to the generator's imperfect visual quality and attempting to use a low-resolution solution to

(a) Illustration demonstrating the conventional approach employed by existing solutions for dubbing. The highlighted inverted area represents the region that is inpainted in by these solutions. Additionally, we draw attention to the unintentional inpainting of the background surrounding the speaker's mouth region, resulting in the undesirable box effect.

(b) Illustration of our proposed solution which is solely focused on the lower half of the speaker's face i.e., the dubbing region. Our approach avoids unnecessary inpainting of the upper half, thus preserving the clarity of the facial texture, eyes, hairline, and more. Furthermore, our solution accurately inpaints the speaker's mouth area while seamlessly integrating the background from the original frame through alpha-blending, substantially enhancing the naturalness of the output.

Figure 6.5: Comparison between the naïve inpainting approach adopted by the majority of visual dubbing solutions, and our proposed solution.

dub higher-resolution videos [Yang *et al.* 2017]. The box effect is primarily emphasized because the speaker's complexion near the jawline is typically consistent, and the blurriness of the background surrounding the mouth region and the abrupt transition between the blurry inpainted region and the rest of the photorealistic scene contribute to this effect.

In an attempt to mitigate the aforementioned visual artefacts, we deviate from the conventional approach discussed above and instead use *alpha-blending* [Smith 1995] to blend the dubbed mouth shapes (i.e., the lower half of the dubbed facial crops) into the speaker's mouth region. First, landmark detection is performed on each facial crop and the landmarks corresponding to the lower half of the speaker's face are retrieved to form a *convex hull*. The convex hull is then filled to produce a binary mask that precisely indicates the speaker's mouth shape within the facial crop. Subsequently, alpha-blending is performed by convolving the binary mask using a large Gaussian filter to ensure that the interior of the speaker's mouth region preserves maximal *activation* and is solely extracted from the dubbed facial crop while allowing for smoothing along the speaker's jawline. As shown in Figure 6.5, this approach allows for the speaker's mouth region to be precisely inpainted while borrowing the background surrounding the speaker's mouth region from the original frame, consequently improving the naturalness of the dubbed results achieved, in addition to our attempts at improving the visual quality. The entire inference process is illustrated in Figure 6.6.

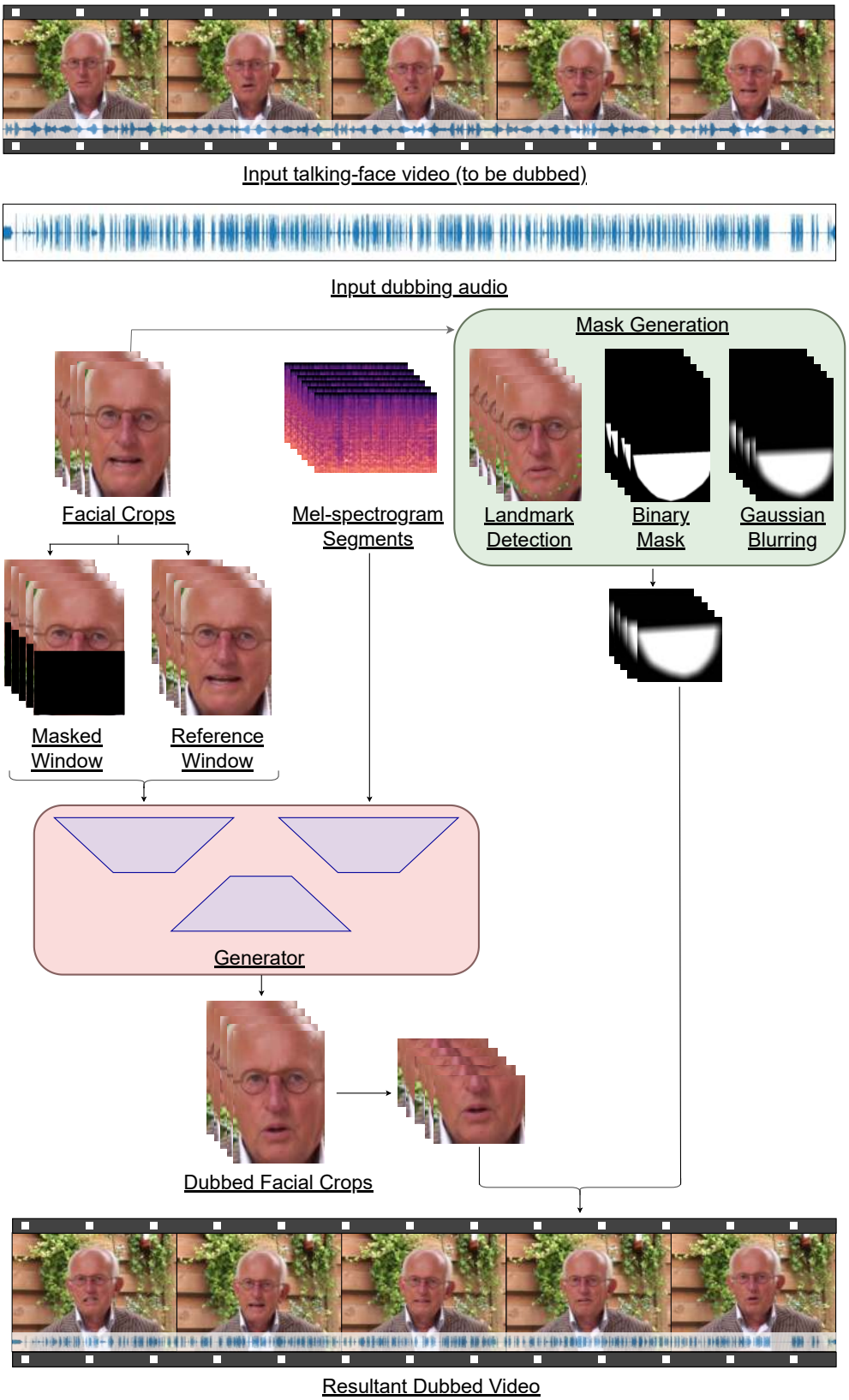


Figure 6.6: Overview of our inference pipeline.

6.8.3 Utilizing Input Dubbing Audio Synthesized by Text-To-Speech Services

Through extensive experimentation, we discovered that our solution performs well when utilizing input dubbing audio synthesized by Text-To-Speech (TTS) services to drive the dubbing process. To illustrate this, we used the Amazon TTS service[5] to synthesize the input dubbing audio, and Figure 6.7 showcases the result produced. This capability further demonstrates the robustness of the solution, improving its autonomy and taking a step toward eliminating human involvement in the dubbing process. Consequently, this enhancement also improves the efficiency and scalability of the solution.

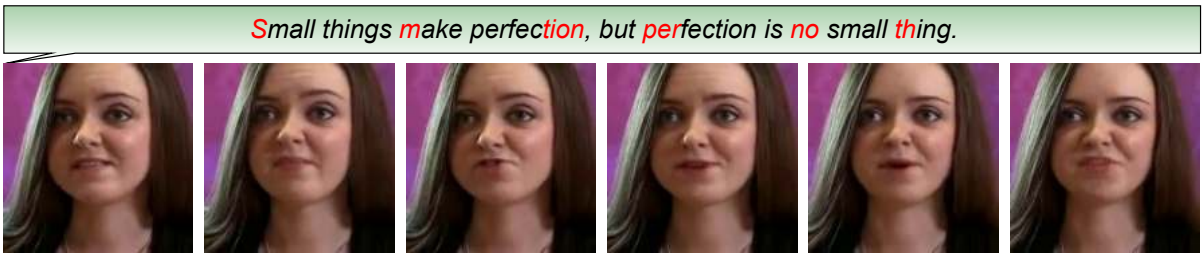


Figure 6.7: Illustration showing the result produced by our solution when using audio produced by a TTS service as the input dubbing audio. This figure also introduces the convention that we adopt when showcasing dubbed results which allows the reader to assess the visual quality and lip-sync accuracy statically. For each alphabet/phrase highlighted in red, its corresponding visual frame is presented beneath (presented from left to right).

6.9 Evaluation

Given the comprehensive discussion and numerous experiments conducted, we aim to establish the efficacy of our solution and evaluate various design choices, such as loss functions and architectures. Below, an overview is provided of the approach taken for evaluating the solution.

6.9.1 Comparative Solutions

To better understand the capabilities of the solution, a comparison is made with other state-of-the-art speaker-independent solutions. Specifically, the solution is compared with ATVG[6] [Chen *et al.* 2019], a prominent one-shot talking-face generation solution, to contrast the performance between one-shot talking-face generation solutions and visual dubbing solutions. Additionally, a comparison is made with two state-of-the-art ADVD solutions, namely Lip-

[5] https://aws.amazon.com/polly/

[6] Note that we replaced the Dlib [King 2009b] face and landmark detector with the S^3FD detector [Bulat and Tzimiropoulos 2017b] because the Dlib detector is unable to detect near-frontal or non-frontal faces. This change does not adversely affect the solution's dubbing capabilities.

GAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020], which were discussed in detail in Section 2.3.3.

6.9.2 Data

As discussed in Section 6.9.4, we do not perform *self-reenactment* to conduct the evaluation. Instead, pairs of input audio and visual segments extracted from different videos are used. The evaluation is conducted using the dubbed videos produced by the inference process detailed in Section 6.8. To ensure a fair comparison, the list of pseudo-randomly paired audio-visual samples prepared by Wav2Lip [Prajwal *et al.* 2020] is used. This list contains 14,239 audio-visual pairings extracted from different videos in the LRS2 dataset [Afouras *et al.* 2018a]. The input audio and visual segments are pre-processed identically as in our inference process, as discussed in Section 6.8.1.

6.9.3 Evaluation Metrics

The visual quality of the dubbed results is evaluated using three metrics. First, SSIM [Wang *et al.* 2004] is used, which was discussed in detail in Section 6.7.1.1. The second metric is the cumulative probability of blur detection (CPBD) metric [Narvekar and Karam 2011], a no-reference perceptual metric that assesses image sharpness without requiring ground-truth data. CPBD leverages the sensitivity of the HVS to blur detection and contrast levels by pooling localized perceived blur distortions using a probability summation model to produce a final quality score. Lastly, the Fréchet Video Distance (FVD) [Unterthiner *et al.* 2018] is computed, which is an extension of the renowned Fréchet Inception Distance (FID) metric used for evaluating GANs, tailored for videos. This metric considers visual quality, temporal coherency, and diversity of generated samples. Unlike SSIM and PSNR, FVD considers a distribution over entire videos, avoiding the drawbacks of frame-level metrics. The Fréchet Distance is computed between the feature representations of synthesized and real videos, utilizing an I3D network [Carreira and Zisserman 2017] pre-trained on the Kinetics dataset, rather than using the features produced by an Inception network [Szegedy *et al.* 2016].

To evaluate lip-sync accuracy, we adopt the common protocol of using the pre-trained lip-sync SyncNet network provided by Chung and Zisserman [2016b]. This network reportedly achieves an accuracy of 99% (averaged over a clip) when measuring lip-sync accuracy. This approach is widely adopted due to its accuracy, autonomy, and specificity to lip-sync accuracy assessment (unlike metrics such as SSIM and PSNR, which are unsuitable for this purpose). Based on the discussion presented in Section 2.3.1, which elaborates on the three measures produced by the SyncNet network, we utilize the confidence and distance measures (denoted as LS-C and LS-D, respectively) as the two lip-sync metrics when evaluating visual dubbing solutions.

6.9.4 Evaluation Protocol

To gain an intuition of the evaluation process, we first present a brief overview of how the evaluation protocol for ADVD solutions has evolved over time. This emphasizes why comparing the presented solution to speaker-specific solutions is not possible. Speaker-specific solutions [Suwajanakorn *et al.* 2017; Nakashima *et al.* 2020; Thies *et al.* 2020] require two to

five minutes of footage to construct an accurate 3DMM [Blanz and Vetter 1999] of the speaker for dubbing. Consequently, large-scale datasets such as LRS2 [Afouras *et al.* 2018a] or VoxCeleb2 [Chung *et al.* 2018] cannot be used to evaluate these solutions as they do not provide sufficient footage of each speaker. Evaluating speaker-specific solutions involves gathering footage of a handful of speakers and assessing the solution based on this confined dataset. In contrast, when evaluating a speaker-independent solution, it is desirable to use a large-scale dataset to assess how well the solution adapts to dubbing various speakers.

The second, and perhaps more significant limiting factor is that speaker-specific solutions are typically evaluated by means of *self-reenactment*, where the input audio and video are in-sync, and the solution is tasked with reconstructing the original video. This approach is useful as it allows for a comparison between the dubbed result and ground-truth data, which is the original video. However, evaluating speaker-independent solutions using this paradigm is ill-posed since the problem formulation which enables them to be speaker-independent is also the aspect which prevents them from performing self-reenactment. In other words, since the input visual frames would already be in-sync with the input dubbing audio, using a speaker-independent solution for self-reenactment would expose ground-truth frames to the model while being disguised as reference frames, thus, would result in information leakage.

To address this issue, KR *et al.* [2019] evaluated their speaker-independent solution through self-reenactment by randomly selecting the reference frames from the input video. While this approach resolves the information leakage issue, it has been shown [Prajwal *et al.* 2020] that it does not provide an accurate reflection of the solution's real-world dubbing capabilities. The random selection of five frames for the reference-frame window affects the head poses and temporal coherence of the result. This is in contrast to our training paradigm, where the reference-frame window's starting point is selected randomly, but the reference frames contained within it are consecutive, preserving temporal coherence. If we were to adopt the evaluation protocol proposed by KR *et al.* [2019], where the input reference frames are selected at random, consideration would need to be made to ensure that all solutions use the reference-frame window identically to allow for a fair comparison.

Prajwal *et al.* [2020] noticed that the previously mentioned strategy takes an unnatural approach to evaluating visual dubbing solutions. The reason is that, in reality, the reference frames are not adjusted; instead, it is the input audio that changes. Therefore, a similar evaluation approach should be adopted. By using visual frames and audio samples from different videos (as in a real-world application), normal inference can be performed. The current frame becomes the reference frame, avoiding information leakage (since mouth shapes are out-of-sync with the input audio), and the required metrics are computed from the resulting dubbed video. To ensure consistent evaluation, a test file list containing audio-visual pairs is used [Prajwal *et al.* 2020]. While this approach simplifies the evaluation process, ground-truth data is no longer available to assess lip-sync accuracy. Fortunately, the two lip-sync metrics we compute are no-reference, meaning they do not require ground-truth data. However, to calculate full-reference visual quality metrics, the input video is considered the ground-truth data and a relative comparison is made among the metrics achieved by each solution. Given the superiority of the evaluation protocol proposed by Prajwal *et al.* [2020], we adopt this approach for our evaluation.

6.9.5 Quantitative Results

Table 6.1 presents the quantitative results, followed by an analysis of each experiment's impact on visual quality and lip-sync metrics. The arrows in the table headers indicate whether lower or higher values are preferred, and the optimal value of each metric is highlighted in bold.

6.9.5.1 Effect of Employing Perceptually-Motivated Loss Functions

A review of the results achieved indicates that the use of the MS-SSIM and VGG losses significantly enhances the visual quality metrics when compared to the presented solution trained using the L_1 loss. This enhancement is apparent through the improved SSIM, increased CPBD, and reduced FVD values, all of which suggest that both variants generate outputs that are

Solution	SSIM (↑)	CPBD (↑)	FVD (↓)	LS-C (↑)	LS-D (↓)
Our Solution:					
– L1	0.89643	0.15507	20.92282	6.33912	7.79438
– L1+MS-SSIM	0.92016	0.15877	14.02665	6.35842	7.73518
– L1+VGG	0.90783	0.15692	16.78362	6.34127	7.76093
– L1+MS-SSIM, Gradual Sync	0.92001	0.15868	14.01743	6.35289	7.74489
– L1+MS-SSIM, Concat. Vectors	0.93559	0.16075	13.73770	6.32854	7.80382
– L1+MS-SSIM, Concat. Vectors, Wav2Lip VQ Disc. (BCE)	0.93864	0.16169	13.48292	6.29218	7.84491
– L1+MS-SSIM, Concat. Vectors, Wav2Lip VQ Disc. (LSGAN)	0.93851	0.16156	13.56823	6.29795	7.84723
– L1+MS-SSIM, Concat. Vectors, PatchGAN (BCE)	0.95128	0.16385	11.28192	6.27081	7.85612
– L1+MS-SSIM, Concat. Vectors, PatchGAN (LSGAN)	0.95073	0.16372	11.32918	6.27490	7.86240
– L1+MS-SSIM, Concat. Vectors, PatchGAN, Rel. Disc. (BCE)	**0.95176**	**0.16398**	**11.25203**	6.26599	7.86427
– L1+MS-SSIM, Concat. Vectors, PatchGAN, Rel. Disc. (LSGAN)	0.95095	0.16387	11.29716	6.26683	7.86604
ATVG [Chen *et al.* 2019]	0.34254	0.09649	54.60932	2.23977	9.40567
LipGAN [KR *et al.* 2019]	0.88224	0.15469	20.88000	4.34527	8.78434
Wav2Lip [Prajwal *et al.* 2020]	0.88620	0.15483	18.38556	**6.76103**	**7.16393**
Ground-Truth	–	0.19303	–	8.05287	6.48998

Table 6.1: Quantitative results.

more visually appealing, sharper, and temporally coherent. This result highlights the importance of employing perceptually motivated loss functions for generative problems, particularly the ADVD problem. Upon closer examination of the results, it is observed that our solution trained with the MS-SSIM loss achieves a 1.36% increase in SSIM, a 1.17% increase in CPBD, and a 16.42% decrease in FVD compared to our solution trained with the VGG loss. As a result, the L_1+MS-SSIM loss was adopted for all subsequent experiments.

6.9.5.2 Effect of Gradually Introducing the Sync Loss

This experiment aimed to explore whether an improved balance between visual quality and lip-sync accuracy could be achieved by simultaneously optimizing both aspects. In contrast to the sequential approach of Prajwal *et al.* [2020] which first optimizes visual quality followed by lip-sync accuracy, this experiment introduces the sync loss from the beginning of training and gradually increases the prioritization of lip-sync accuracy as training progresses. Analysing the results reveals a decrease in both visual quality and lip-sync accuracy. Specifically, SSIM decreases by 0.02%, CPBD by 0.06%, FVD by 0.06%, LS-C by 0.08%, and LS-D increases by 0.12%. This trend persists regardless of the rate at which lip-sync accuracy's priority is increased. We believe this result is due to the constant contention between visual quality and lip-sync accuracy, and by optimizing for both measures simultaneously, neither is adequately prioritized, leading to an overall decrease in performance.

6.9.5.3 Effect of Concatenating Embedding Vectors

While conducting a thorough analysis of the Wav2Lip solution [Prajwal *et al.* 2020], we discovered that the visual encoder within the generator network establishes skip connections with the image decoder to transmit visual information during result generation. However, the visual embedding vector produced is unused. This prompted us to investigate whether incorporating the visual embedding vector could lead to improved results. Results indicate that without concatenating the embedding vectors, an SSIM score of 0.92016, CPBD of 0.15877, and FVD of 14.02665 is achieved. In contrast, when the embedding vectors are concatenated, an improved SSIM score of 0.93559, CPBD of 0.16075, and FVD of 13.73770 is achieved. This enhancement in visual quality may be attributed to the additional information provided to the decoder, including the unique details within the visual embedding vector. However, despite achieving significantly improved visual quality, the concatenation of the embedding vector inadvertently dilutes the audio information received from the audio encoder, resulting in a notable decline in lip-sync metrics which is discussed in further detail in Section 6.9.5.6.

6.9.5.4 Effect of Employing a Visual Quality Discriminator

Taking inspiration from Wav2Lip [Prajwal *et al.* 2020], which demonstrated enhanced visual quality through the use of a visual quality discriminator, we aimed to explore whether further improvements could be achieved by refining the design of the employed discriminator. As a point of comparison, we also trained our generator network with the Wav2Lip discriminator and compared the results achieved by our PatchGAN discriminator [Isola *et al.* 2017]. The results indicate that, while a minor improvement is achieved when utilizing the Wav2Lip visual quality discriminator compared to not using a visual quality discriminator, this improvement is surpassed by the gains achieved by our discriminator. Training the generator with

the Wav2Lip visual quality discriminator yields a 0.32% increase in SSIM, a 0.58% increase in CPBD, and a 1.85% decrease in FVD. On the other hand, adopting our proposed discriminator for generator training leads to a more substantial improvement, specifically a 1.62% increase in SSIM, a 1.93% increase in CPBD, and a 17.87% decrease in FVD.

Perhaps what makes the outcome even more remarkable is that our discriminator achieves a significant enhancement in visual quality despite being 3.5 times smaller (in terms of parameter count) than the discriminator proposed by Wav2Lip [Prajwal *et al.* 2020]. This is primarily attributed to the nature of the PatchGAN discriminator, which penalizes at the patch level rather than the image level. As a result, it provides the generator network with more detailed feedback on improving the visual quality of the dubbed results produced. Lastly, employing the BCE loss as the adversarial loss leads to improved performance compared to when the LSGAN loss is used.

6.9.5.5 Effect of Employing a Relativistic Discriminator

Similar to the various solutions from which we drew inspiration [Wang *et al.* 2018; Jiang *et al.* 2021c; Du *et al.* 2021], utilizing a relativistic discriminator further improved the visual quality of our solution. Specifically, with the BCE loss, we achieved an SSIM score of 0.95128, CPBD of 0.16385, and FVD of 11.28192 without employing the relativistic discriminator. However, by incorporating the relativistic discriminator, we achieved an SSIM score of 0.95176, CPBD of 0.16398, and FVD of 11.25203. This results in a minor increase of 0.05% in SSIM, 0.08% in CPBD, and a minor decrease of 0.26% in FVD compared to not using a relativistic discriminator.

6.9.5.6 General Analysis

A comprehensive analysis of the results highlights that ATVG [Chen *et al.* 2019] yields the lowest visual quality and lip-sync metrics. This outcome can be attributed to the solution's inability to capture scene dynamics, which leads to an unnatural and unsatisfactory viewing experience. LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020] achieve similar levels of visual quality, likely due to their shared implementation of measures aimed at optimizing visual quality. Notably, the simplest version of the presented solution, trained with the L_1 loss, surpasses the visual quality achieved by all other comparative solutions. This is particularly noteworthy, especially considering that some solutions (such as Wav2Lip) are trained with the assistance of a visual quality discriminator. Furthermore, each enhancement designed to improve visual quality does indeed result in significant improvements.

Regarding lip-sync accuracy, ATVG [Chen *et al.* 2019] and LipGAN [KR *et al.* 2019] achieve inferior results compared to Wav2Lip [Prajwal *et al.* 2020] and the presented solution. This outcome demonstrates the effectiveness of using a pre-trained lip-sync discriminator to enforce accurate lip-sync. Despite our solution achieving impressive lip-sync metrics, we acknowledge that the results achieved are considerably lower than those of Wav2Lip, despite employing a significantly improved pre-trained lip-sync discriminator. This difference may be attributed to our design inherently prioritizing visual quality and the challenge of achieving a satisfactory balance between visual quality and lip-sync accuracy. These observations are supported by the fact that each enhancement improving visual quality inversely affects lip-sync accuracy. Figure 6.8 illustrates the relationship between achieved SSIM and LS-C scores, reflecting a similar pattern for other pairs of visual quality and lip-sync metrics. While we

acknowledge our solution's lower lip-sync measures compared to Wav2Lip, we also recognize that Wav2Lip has been noted for achieving overly optimized lip-sync metrics. Some studies [Zhou *et al.* 2021] have even shown that the lip-sync metrics of Wav2Lip can exceed those of ground-truth data in certain cases. To comprehensively assess whether our solution achieves acceptable lip-sync measures and whether significantly improved visual quality compensates for the decline in lip-sync metrics, we conduct a human subjective study (detailed in Section 6.9.7) to observe how each solution is perceived by humans, which is our primary concern.

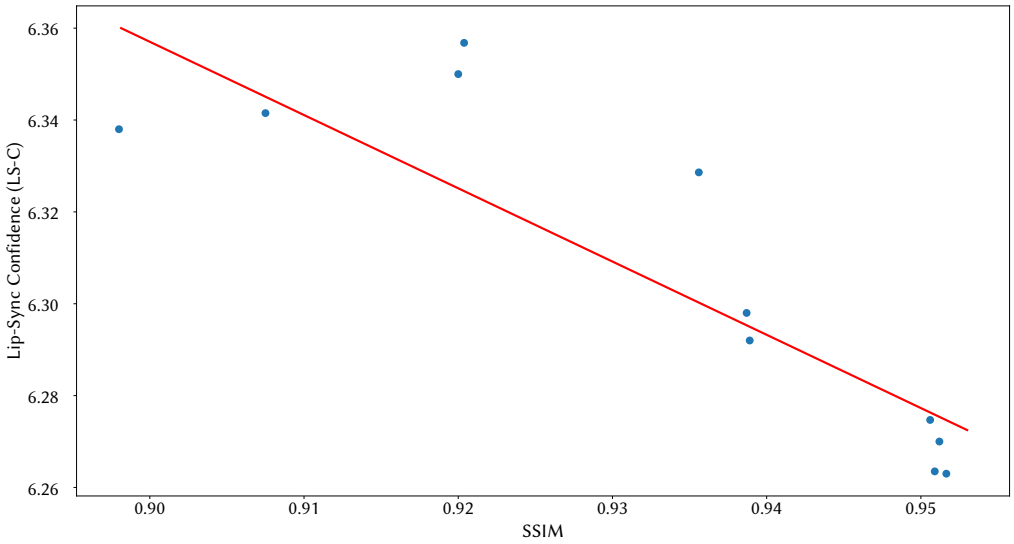


Figure 6.8: Scatter-plot showing SSIM plotted against the lip-sync confidence (LS-C) achieved for each experiment which shows that in general, each improvement in visual quality results in a deterioration in lip-sync accuracy.

6.9.6 Qualitative Results

Figures 6.9 and 6.10 present the outcomes of our solution in comparison to those achieved by comparative solutions. An examination of the results indicates that ATVG [Chen *et al.* 2019] yields visually displeasing outputs, consistent with its nature as a one-shot talking-face generation approach. Consequently, the solution fails to capture scene dynamics such as emotions, facial expressions, and head movements, resulting in unnatural outcomes. Moreover, the results exhibit blurriness, and the speaker's mouth movements appear rigid. While LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020] attain comparable levels of visual quality as depicted in Figure 6.10, LipGAN occasionally exhibits glitch-like effects stemming from uncertainty in inpainting the appropriate mouth shape, which also gives the speaker's teeth a fuzzy appearance.

The presented solution achieves the sharpest results, as evidenced by the intricate facial details captured, including wrinkles, which other solutions miss. This distinction is especially clear in Figure 6.10. This outcome is attributed to our solution's exclusive focus on inpainting the

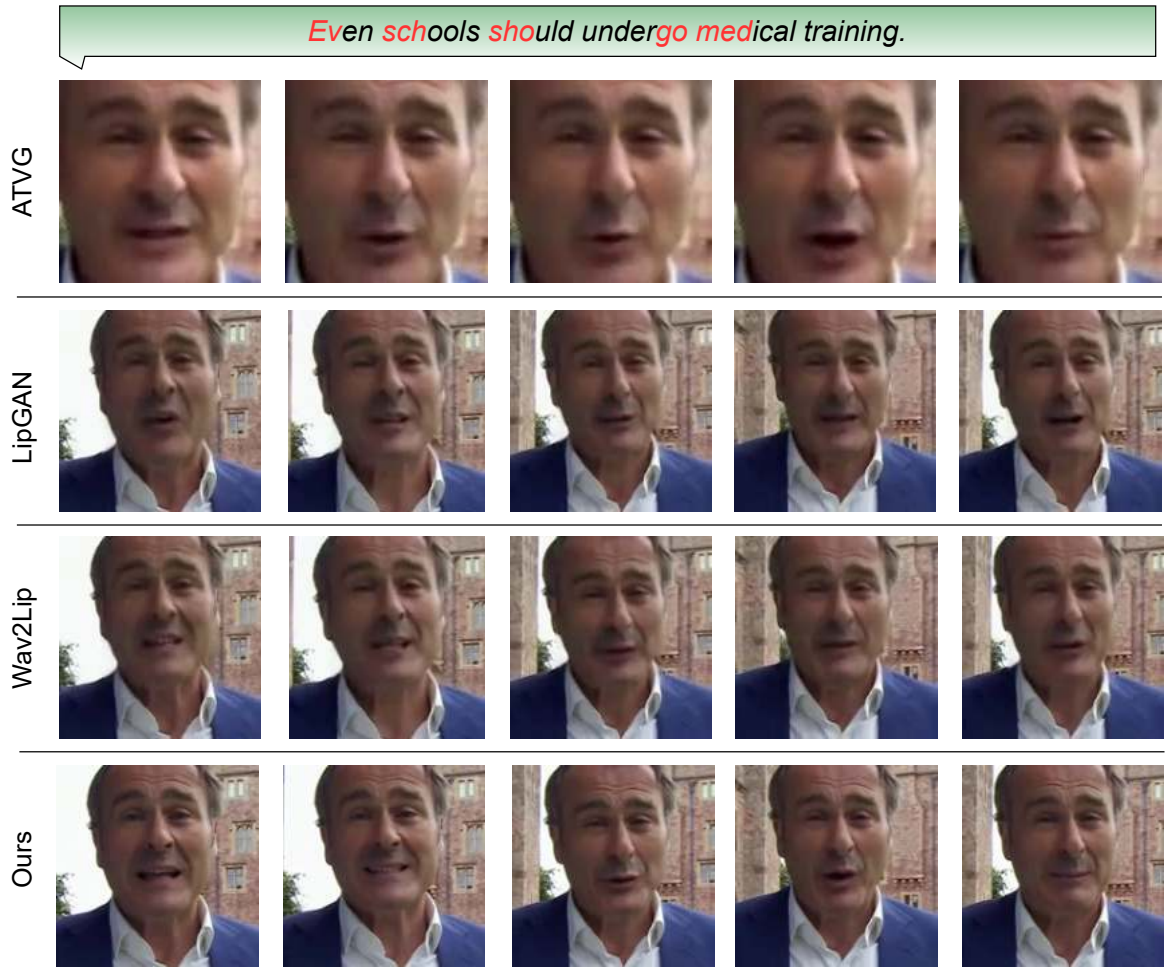


Figure 6.9: Qualitative results.

speaker's mouth region using alpha-blending, while LipGAN and Wav2Lip inpaint the entire face. This distinction also explains the more natural and vivid appearance of the speaker's hair fringe and eyes in our results compared to other solutions. Additionally, the sharpness of the speaker's teeth in our outcomes surpasses that of other solutions. Lastly, the speaker's mouth movements in our results correspond well to both the dubbing utterance and the mouth movements of Wav2Lip, which achieves superior lip-sync metrics.

6.9.7 Human Subjective Study

In addition to conducting an extensive quantitative and qualitative analysis, we deemed it appropriate to evaluate the dubbed videos from a human perspective. Since videos are dubbed for the direct consumption of humans, humans are the supreme evaluators when assessing dubbed videos [KR *et al.* 2019]. We conducted an online human subjective study through a series of pairwise comparisons, as opposed to the more commonly used Mean Opinion Score (MOS) framework. The rationale behind this decision is that it has been shown that humans may be inconsistent when evaluating images/videos on a linear scale [Kendall and Smith 1940; Sakaridis *et al.* 2018]. Through this study, we aim to establish a ranking among the three comparative solutions and our presented solution.

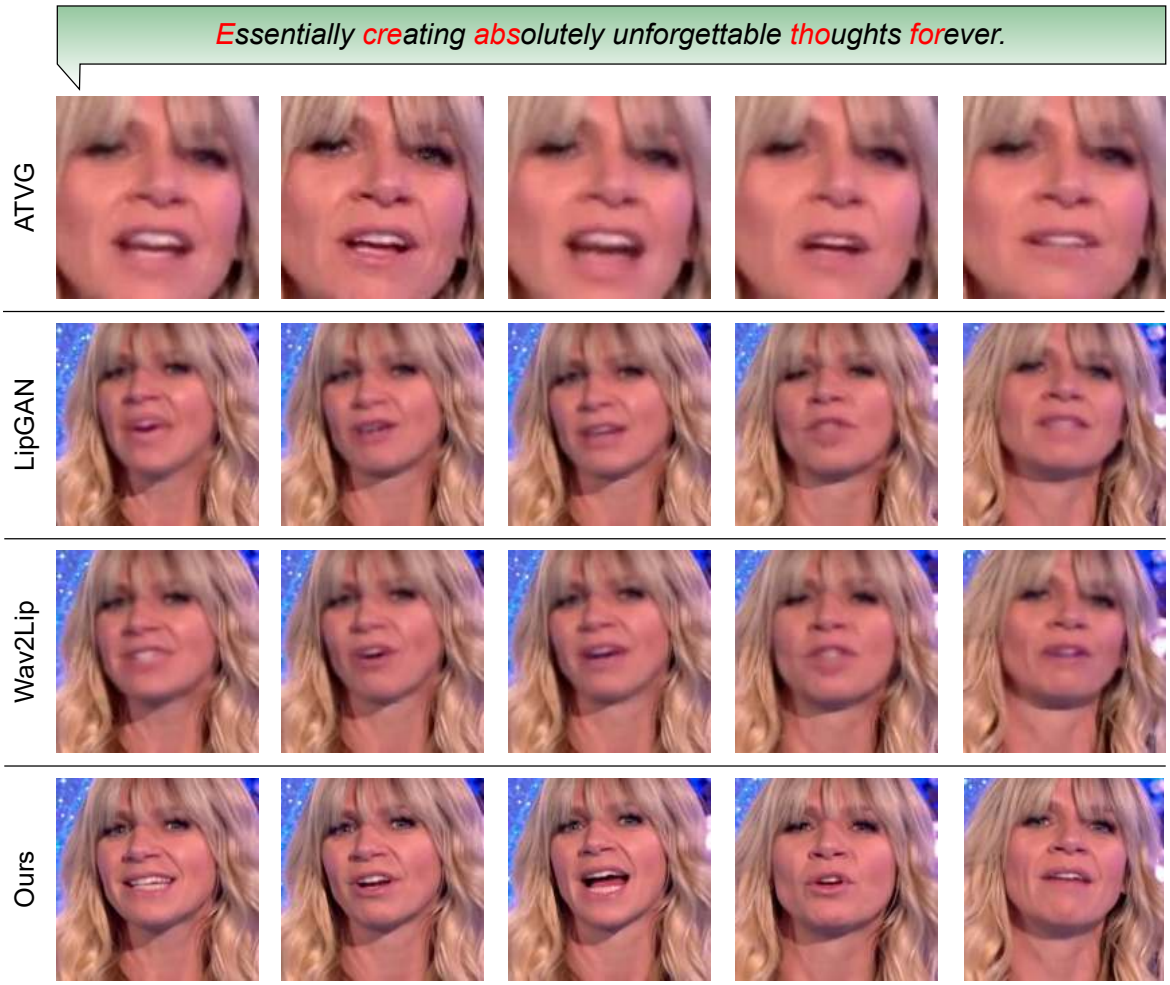


Figure 6.10: Qualitative results (continued). We present additional qualitative results in video form here.

The design of the human subjective study starts by randomly selecting an audio and visual sample from the LRS2 test dataset [Afouras *et al.* 2018a]. These samples are then used as input for each dubbing solution to generate the corresponding dubbed results. Each dubbed result is then paired and compared with every other dubbed result, resulting in six pairwise comparisons for each audio-visual input pair due to the four solutions being evaluated. For each pairwise comparison, the participant views the two dubbed videos side by side, played sequentially. The participant is then asked two questions: *Is the [visual quality/lip-sync accuracy] of video 2 much better than, better than, as good as, worse than, or much worse than the [visual quality/lip-sync accuracy] of video 1?*. With eight audio-visual input pairs, the participant conducts a total of $8 \times 6 \times 2 = 96$ pairwise comparisons. The study involved 23 anonymous participants from the general public. While testing for sync is a natural instinct for most humans, participants were also briefed on assessing visual quality (e.g., natural face texture, absence of visual artefacts like blurring, noise, and colour shifts) and lip-sync accuracy (e.g., synchronicity of mouth movements with dubbing audio), while being encouraged to use their own judgment.

6.9.8 Analysis of Results

Inspired by the literature [Jiang *et al.* 2021c], we present the results of the human subjective study by fitting a Bradley-Terry model [Bradley and Terry 1952] to establish a ranking amongst the comparative solutions. Since the study involved 23 participants assessing eight videos through six pairwise comparisons for each video, the results were processed by using the six pairwise comparisons to establish the participant's rank for the given video. When applied to all 23 participants and eight videos, this results in $23 \times 8 = 184$ visual quality rankings and 184 lip-sync rankings. Figure 6.11 presents the results of the study. To gain an intuition of how these results are interpreted, we see that the ATVG solution [Chen *et al.* 2019] was ranked 1^{st} (i.e., achieved the best visual quality) for zero videos, 2^{nd} for two videos, 3^{rd} for 119 videos, and 4^{th} for 63 videos. As a consequence of including an *as good as* option, two or more solutions may tie for the same ranking (which suggests equal efficacy).

Our analysis commences with the visual quality results achieved and we immediately observe that our solution achieves superior performance by a large margin. This is deduced by observing that the videos produced by the solution were most frequently ranked 1^{st} compared to any other solution. Furthermore, upon reviewing the number of videos produced by other solutions that were ranked 1^{st}, we note that our solution rarely tied for 1^{st} place, thus, further emphasizing the efficacy of our design. Upon contrasting the performance achieved by LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020], we see that both solutions are in close contention for 2^{nd} place, thus, suggesting that both solutions achieve a similar standard of visual quality as perceived by humans. Differently, however, we see that Wav2Lip is ranked 1^{st} more frequently than LipGAN and reciprocally, LipGAN is ranked 3^{rd} more frequently than Wav2Lip, therefore, we may deduce that Wav2Lip achieves marginally improved visual quality compared to LipGAN. Lastly, the vast majority of videos produced by ATVG were ranked 3^{rd} which we believe naturally follows as a consequence of our solution being ranked 1^{st}, and LipGAN and Wav2Lip tying for 2^{nd} for the majority of videos. Furthermore, in the case when no ties occur, ATVG is predominantly ranked 4^{th} which indicates that one-shot talking-face generation solutions are far inferior to visual dubbing solutions as perceived by humans. Evidently, these results are congruent with our quantitative and qualitative results. Through a series of paired t-tests, we establish that our results are statistically significant ($p < 10^{-4}$).

With regards to the lip-sync results achieved, we note that videos produced by Wav2Lip [Prajwal *et al.* 2020] are frequently ranked 1^{st}, aligning with our quantitative findings. In contrast to our numerical analysis, the human subjective study reveals that our solution's lip-sync accuracy is comparable to Wav2Lip's as perceived by humans. Given that videos are dubbed for human enjoyment [KR *et al.* 2019], we prioritize the results of the human study over the quantitative results. We posit that the inconsistency between the quantitative outcomes and the human study stems from humans' inherent limitations in assessing sync [Chung and Zisserman 2016b]. We suggest that once quantitative lip-sync metrics surpass this perceptual threshold, further optimization becomes imperceptible to humans, rendering excessive optimization unnecessary. As outlined in Section 10.2.1, future work could employ SyncNet's [Chung and Zisserman 2016b] AV offset measure, applying a threshold based on the human capabilities as postulated by [Chung and Zisserman 2016b] to determine videos achieving imperceptible AV offsets. This approach would provide insight into avoiding overemphasis on lip-sync accuracy at the expense of visual quality. Furthermore, we contend that by Wav2Lip [Prajwal

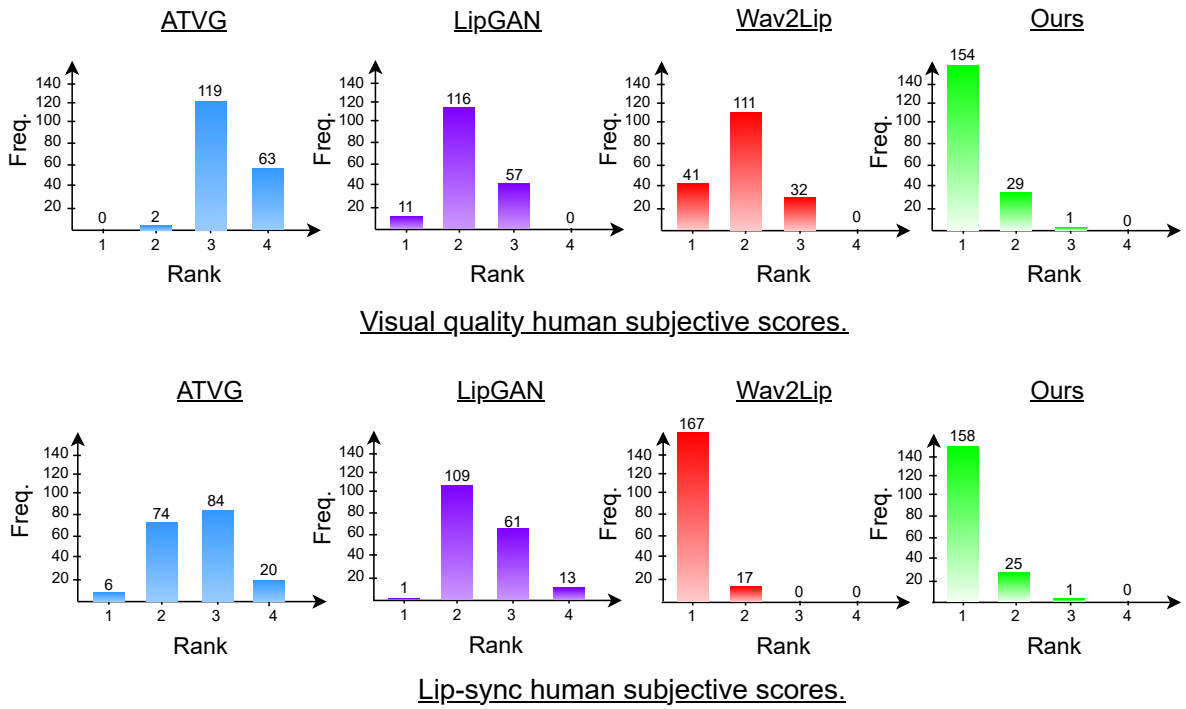


Figure 6.11: Results from our human subjective study.

et al. 2020] prioritizing the lip-sync metrics, despite failing to enhance human-perceived results, contributes to its inferior visual quality compared to our solution. This underscores the well-known trade-off between visual quality and lip-sync accuracy.

While Wav2Lip [Prajwal *et al.* 2020] and our solution frequently tie for 1^{st} place across most videos, ATVG [Chen *et al.* 2019] and LipGAN [KR *et al.* 2019] closely compete for 2^{nd} place. However, since LipGAN receives more 2^{nd} place rankings than ATVG, this indicates that LipGAN achieves better lip-sync accuracy compared to ATVG, as perceived by humans. The substantial performance gap between our solution and Wav2Lip in contrast to ATVG and LipGAN highlights the effectiveness of utilizing a pre-trained lip-sync discriminator to enforce accurate lip-sync. Through a series of paired t-tests, we established that our results are statistically significant ($p < 10^{-4}$).

6.10 Conclusion

This chapter presented a comprehensive description of our generator network, commencing with a theoretical overview of the network's functions, followed by an increasingly detailed account of the network's design, including inputs, architecture, and experiments conducted. Through a series of quantitative and qualitative analyses, the presented solution proved to achieve both superior visual quality and high levels of lip-sync accuracy compared to comparative solutions. The following chapter explains how this solution is extended to utilize high-resolution training data.

Chapter 7

High-Resolution Audio-Driven Visual Dubbing

7.1 Introduction

The preceding chapter presented our complete ADVD solution which proved to surpass the capabilities of preceding solutions. As one of the many ways in which existing ADVD solutions can be extended, we present one of the first attempts at designing a high-resolution ADVD solution. This chapter discusses the rationale behind this endeavour, as well as how the existing solution is extended to utilize high-resolution training data.

7.2 Motivation

Due to the rapid advancements made in image acquisition equipment (especially cameras) that are capable of recording high-resolution footage, this has resulted in a substantial increase in the production and consumption of high-resolution footage in recent years [Seshadrinathan *et al.* 2009; Xu *et al.* 2021; Bakhtiarnia *et al.* 2022]. Consequently, the need to utilize high-resolution data to train machine-learning models has never been greater. Doing so is also expected to improve the efficacy of solutions when applied to real-world data which is typically of high resolution. Note that the exact definition of high-resolution data varies on an application-specific basis. For most image-manipulation tasks solutions such as image inpainting [Demir and Unal 2018; Jiang *et al.* 2020], semantic segmentation [Garcia-Garcia *et al.* 2017], and image-to-image translation [Yi *et al.* 2017; Zhu *et al.* 2017a; Isola *et al.* 2017], a resolution of 1024×1024 is typically considered as high resolution, whereas for applications such as crowd counting [Gao *et al.* 2020], remote sensing (using satellite imagery) [Hu *et al.* 2015; Van Etten 2018], and medical applications [Van der Laak *et al.* 2021], data can easily reach gigapixel resolutions [Bakhtiarnia *et al.* 2022].

Defining what resolution qualifies as high-resolution within the scope of ADVD is not a straightforward task. Given that the majority of existing solutions utilize a resolution no greater than 120×120, we consider resolutions substantially surpassing this threshold to be deemed as high-resolution. As outlined in Section 7.3, training an ADVD model on high-

resolution data is inherently more complex compared to other computer-vision challenges due to the bi-modal nature of the ADVD problem. Consequently, this complexity leads to heightened demands on computational resources and time. Thus, we train our solution with a resolution of 192×192. This resolution represents a notable enhancement over previous approaches and is also the maximum resolution permitted by our computational infrastructure.

7.3 Challenges

When attempting to leverage high-resolution training data, we identify three essential requirements to be satisfied i.e., availability of an abundance of high-quality high-resolution data, access to adequate computational (hardware) resources, and time availability. Below, we elaborate on each of these aspects and consider how/why the majority of existing solutions have failed to satisfy these requirements.

7.3.1 Availability of an Abundance of High-Quality High-Resolution Data

As detailed extensively in Chapter 4, attempting to address the ADVD problem requires a dataset comprised of an abundance of high-quality talking-face videos that have been sync-corrected. This is in addition to possessing other desirable data properties such as a diverse range of identities, ethnicities, recording conditions, and head poses amongst others. During our exploration of the data landscape, we established that the LRS2 dataset [Afouras *et al.* 2018a] has been accepted as the de-facto standard, however, all videos have a resolution of 160×160 which are further cropped and resized into facial crops with size 96×96. We believe that it is for this reason that the results produced by solutions such as Wav2Lip [Prajwal *et al.* 2020] tend to exhibit an unpleasant box effect which follows as a consequence of employing a low-resolution model to perform inference on high-resolution real-world data [Yang *et al.* 2017].

The low resolution of the LRS2 dataset prompted some solutions to explore the use of higher resolution datasets, with the VoxCeleb2 dataset [Chung *et al.* 2017 2018] being the most common choice. In essence, the VoxCeleb2 dataset offers 10 times more hours of footage compared to the LRS2 dataset (as indicated in Table 4.1) and is comprised of excerpts taken from YouTube videos. Despite appearing to meet several of the aforementioned desirable data properties, this dataset holds a bias toward celebrity interviews, includes instances of corrupted audio samples, and contains videos that have not been sync-corrected. Additionally, we find it perplexing that visual dubbing solutions [Chung *et al.* 2017] that utilize this dataset opt for a resolution no larger than 120×120. Although this might seem to be a significant improvement over the LRS2 dataset, this resolution remains notably inferior when compared to the resolutions typically employed for other image manipulation tasks.

To the best of our knowledge, an ideal off-the-shelf dataset that could be used to train high-resolution visual dubbing solutions is not made publicly available. Addressing this challenge would involve either constructing a new audio-visual dataset tailored to the specific requirements of the visual dubbing problem, or thoroughly pre-processing an existing dataset. In the case of the latter, this would entail identifying the least sub-optimal dataset and subsequently

performing a series of pre-processing operations. Moreover, this process would demand a substantial level of domain expertise and technical knowledge to fully understand the sort of operations required to transform the dataset into a form suitable and optimal for the visual dubbing problem.

7.3.2 Access to Adequate Computational (Hardware) Resources

When utilizing a large-scale high-resolution dataset, the initial concern that emerges is that of storage capacity. To emphasize the importance of this issue, our pre-processed AVSpeech dataset [Ephrat *et al.* 2018] occupies 2.6TB of storage—a volume of storage that is not readily available in most circumstances.

Considering the hardware requirements from a computational perspective, we acknowledge that addressing the visual dubbing problem demands significantly higher resources compared to most other image manipulation tasks. This stems from the fact that the visual dubbing problem operates concurrently in two modalities — audio and visual. As a result, the network architecture necessitates two streams to accommodate these modalities, thereby substantially increasing the parameter count and, consequently, GPU memory usage in contrast to uni-modal tasks. Moreover, recent solutions, including the presented solution, input a window of five stacked video frames to the visual stream, further intensifying GPU memory consumption. This is in contrast with other image manipulation tasks that process individual frames independently. We posit that it is for these reasons that most other image manipulation tasks have experienced notable success in utilizing high-resolution training data, whereas the progress has been more subtle in the case of visual dubbing.

Accommodating higher-resolution input data often requires enlarging the network to achieve embedding vectors or outputs of the desired size. Naturally, this network enlargement leads to increased parameter count and GPU memory usage [Sabottke and Spieler 2020; Thompson *et al.* 2020]. To contextualize this increase in resource requirements, training our enlarged generator network (as discussed in Section 7.4.2) for high-resolution data increases the generator's parameter count and total GPU memory usage from 38M and 15GB, respectively (as in the case of our low-resolution implementation), to 50M and 39GB — representing a 31.5% increase in parameters and a 160% increase in GPU memory usage. It may be said that progress in this regard may be hindered due to the inaccessibility to sufficient computing resources and infrastructure.

7.3.3 Time Availability

Given the relatively limited research on employing high-resolution datasets for the ADVD problem, substantial experimentation was essential regarding data handling and the design of the network's architecture. This process is inherently time-consuming. To provide context for the time requirements, it took seven months to develop the data-cleaning pipeline and pre-process the AVSpeech dataset [Ephrat *et al.* 2018]. Furthermore, as previously mentioned, the utilization of higher-resolution data increases parameter count, leading to notably more challenging optimization. This complexity prolongs the training process [Sabottke and Spieler 2020]. Specifically, training the pre-trained lip-sync discriminator took 13 days, while training the generator network required 19 days.

Given the discussion presented above, we are made fully aware of the numerous challenges associated with attempting to advance the visual dubbing field to leverage high-resolution training data. Moreover, we believe that it is due to these challenges that the field has largely shied away from undertaking such an endeavour and instead has continued to employ low-resolution datasets. The solution by Yang *et al.* [2020] is the only existing ADVD solution which utilizes high-resolution training data with a resolution larger than 120×120 i.e., the authors use a resolution of 256×256. While their theoretical findings are useful, from a practical standpoint, neither the implementation nor the dataset were made available, thus, limiting the impact of their contributions.

7.4 Our Solution

To understand how the existing solution is adapted to utilize high-resolution training data, we begin by discussing the data used to train the solution. Since constructing a large-scale dataset is known to be a time-consuming and laborious process, we instead adopt the AVSpeech dataset [Ephrat *et al.* 2018] – one of the largest publicly available high-quality datasets. As detailed in Section 4.7, we design a multi-faceted data-cleaning pipeline which transforms the dataset (originally curated for speech separation) into a form that is ideal for addressing the visual dubbing problem. The resulting dataset is then used to train our high-resolution ADVD solution with a resolution of 192×192. Below, the enhancements made to the existing solution to accept high-resolution data are discussed. Since an increase in resolution only affects the visual stream, we solely concern ourselves with enhancements made to the visual encoder of the generator and lip-sync discriminator network, as well as the face decoder of the generator, since the nature of the input audio remains unchanged.

7.4.1 Lip-Sync Discriminator

To account for the higher-resolution input data, an additional R(2+1)D spatiotemporal block is added to the network, thus, increasing its model capacity. In addition, we are assured that a 512D embedding vector is produced due to the global average pooling layer at the top of the visual encoder.

7.4.2 Generator Network

To extend the generator network to ingest inputs with size $[B, 6, 192, 192]$, the visual encoder and face decoder are supplemented with an additional ResNetv1 block [He *et al.* 2016] each. In addition, the solution is trained using all experiments detailed in Section 6.7 that improved the visual quality and/or lip-sync accuracy i.e., the L_1+MS-SSIM loss, concatenated embedding vectors, PatchGAN discriminator [Isola *et al.* 2017], and relativistic discriminator [Jolicoeur-Martineau 2018] are employed.

Since the Nvidia GeForce RTX 3090 GPU that was employed previously does not possess sufficient graphics memory, all high-resolution models were instead trained using an Nvidia Quadro RTX 8000 with 48GB of graphics memory.

7.5 Evaluation

To evaluate the high-resolution ADVD solution, we adhered to the same evaluation procedure outlined for the low-resolution implementation in Section 6.9. The primary difference lies in the data used for evaluation; instead of employing pseudo-randomly audio-visual pairs sampled from the LRS2 dataset [Afouras *et al.* 2018a], pairs from AVSpeech test videos [Ephrat *et al.* 2018] were created. This approach draws inspiration from Wav2Lip [Prajwal *et al.* 2020], where two conditions are enforced: (1) the audio segment is shorter than the visual segment to avoid static frames at the end of the dubbed result, and (2) the audio and visual segments originate from different videos. This process results in 25,600 pseudo-randomly assembled audio-visual pairs for evaluation. By conducting this evaluation, we aim to explore how training the solution with high-resolution data influences the achieved performance, particularly in terms of visual quality, compared to existing state-of-the-art solutions.

7.5.1 Quantitative Results

The lip-sync discriminator, pre-trained on the high-resolution dataset, achieves an accuracy of 87.842% and an F1-score of 0.87523. Table 7.1 displays the quantitative results pertaining to the high-resolution ADVD solution, which is followed by a critical analysis of the results achieved.

Solution/Metric	**SSIM (↑)**	**CPBD** (↑)	**FVD** (↓)	**LS-Conf.** (↑)	**LS-Dist.** (↓)
ATVG [Chen *et al.* 2019]	0.46412	0.02506	84.7672	4.30108	9.49932
LipGAN [KR *et al.* 2019]	0.84621	0.03928	63.4973	4.35780	9.38862
Wav2Lip [Prajwal *et al.* 2020]	0.86470	0.05383	63.94158	**5.04172**	**8.63167**
Our Solution	**0.950191**	**0.106637**	**18.91226**	4.759881	8.89485
Ground-Truth	–	0.137942	–	6.174688	8.23574

Table 7.1: Quantitative results when evaluated on high-resolution (AVSpeech) data. For each metric, the arrow indicates whether higher or lower values are preferred, and values in bold denote the superior result for that metric. Note that in order to conduct this comparison, we used the trained solutions provided by their respective authors.

Similar to the low-resolution implementation, ATVG [Chen *et al.* 2019] achieves unsatisfactory visual quality due to its nature as a one-shot talking-face generation solution. Consequently, the solution fails to consider facial motion, expressiveness, and emotion, resulting in outcomes lacking photorealism. Contrasting these results with those of other solutions highlights the superiority of visual dubbing approaches over one-shot talking-face generation methods. In terms of lip-sync metrics, ATVG achieves comparable results to LipGAN [KR *et al.* 2019], however, both solutions perform worse than Wav2Lip [Prajwal *et al.* 2020] and the presented solution. This result emphasizes the effectiveness of utilizing a pre-trained lip-sync discriminator to ensure accurate lip-sync. When assessing the visual quality metrics

of LipGAN and Wav2Lip, it becomes evident that the difference in performance is insignificant, which follows as a consequence of both solutions employing similar design strategies for optimizing visual quality.

Throughout this document, we have hypothesized that training our ADVD solution with high-resolution data would lead to an improvement in the achieved visual quality, particularly when dubbing real-world videos. Based on the results obtained, we indeed observe that the presented solution significantly improves the visual quality metrics achieved. To illustrate the extent of this improvement, when comparing the visual quality metrics with those attained by the next best solution, Wav2Lip [Prajwal *et al.* 2020], we find that our solution achieves a 9.88% increase in SSIM, a 98% increase in CPBD, and a 70% decrease in FVD. This indicates that the dubbed results generated by our solution exhibit a higher degree of colour consistency, sharpness, and temporal coherence compared to solutions that are not trained using high-resolution data.

As anticipated, an increase in input resolution did not impact the lip-sync metrics achieved. Moreover, Wav2Lip once again outperforms our solution in terms of lip-sync metrics. As demonstrated in our low-resolution implementation, for which a human subjective study was conducted, we argued that this outcome does not necessarily signify that Wav2Lip produces results that are more natural and accurate as perceived by a human. This argument primarily stems from the inherent limitations of humans in assessing lip-sync as discussed in Section 6.9.5.6.

The significantly improved visual quality achieved by the high-resolution implementation highlights the urgency for the field to shift from low-resolution data (common in most existing solutions) to high-resolution data. We consider this research's contribution to be a catalyst for future works, simplifying a major obstacle in this transition — making an abundance of suitable high-resolution data accessible when tackling the ADVD problem.

7.5.2 Qualitative Results

A review of Figures 7.1 and 7.2 reveals that the results produced by ATVG [Chen *et al.* 2019] lack naturalness, for reasons mentioned previously. When assessing the results of LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020], it becomes clear that both solutions achieve satisfactory lip-sync levels; however, there is significant room for enhancing visual quality. Due to the poor visual quality, attributed to the blurry inpainted region, it is easy to identify the specific area within the scene that has been inpainted. Notably, both solutions apply inpainting to the entire face of the speaker. We find this design choice unconventional, given that dubbing exclusively involves morphing the speaker's mouth region. As a result of inpainting the whole face, the fine details of the face, such as wrinkles, blemishes, and teeth, are absent due to the blurriness of the inpainted section. Moreover, a distinct contrast in visual quality arises between the inpainted face and the surrounding scene, particularly noticeable along the edges of the speaker's face, such as the hairline and mouth region. The previously mentioned unpleasant box effect stems from the blurriness of the face, which unintentionally extends to inpaint the background surrounding the speaker's mouth area (this effect is particularly illustrated in the zoomed-in regions).

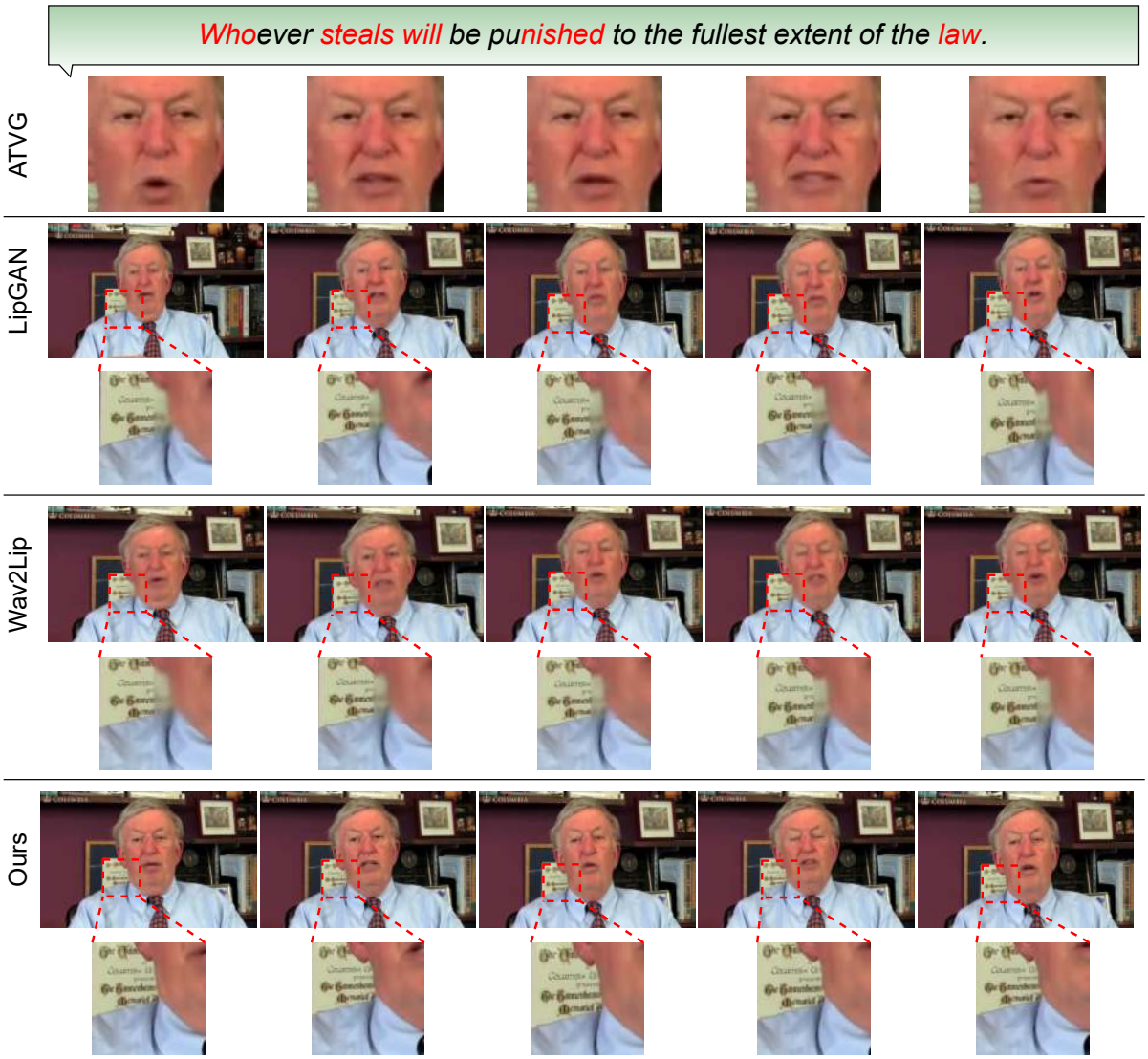


Figure 7.1: High resolution qualitative results

Upon assessing the results achieved by our solution, we notice that the unpleasant box-effect present in the results of LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020] has been successfully eliminated. This is attributed to training the solution with high-resolution data, prioritizing the visual quality achieved through the design choices made, and due to the innovative masked (alpha-blending) inpainting technique applied which precisely inpaints the speaker's mouth region. In addition, the solution solely inpaints the lower half of the speaker's face and borrows the upper half from the original frame, thus, resulting in significantly more crisp and vivid outputs compared to other solutions. Since there is no discrepancy between the upper half and lower half of the speaker's face (which has been inpainted), this proves that the solution achieves photorealistic results. This is further showcased by the photorealistic teeth that the solution achieves. Despite achieving significantly improved visual quality, the presented solution maintains a high standard of lip-sync accuracy.

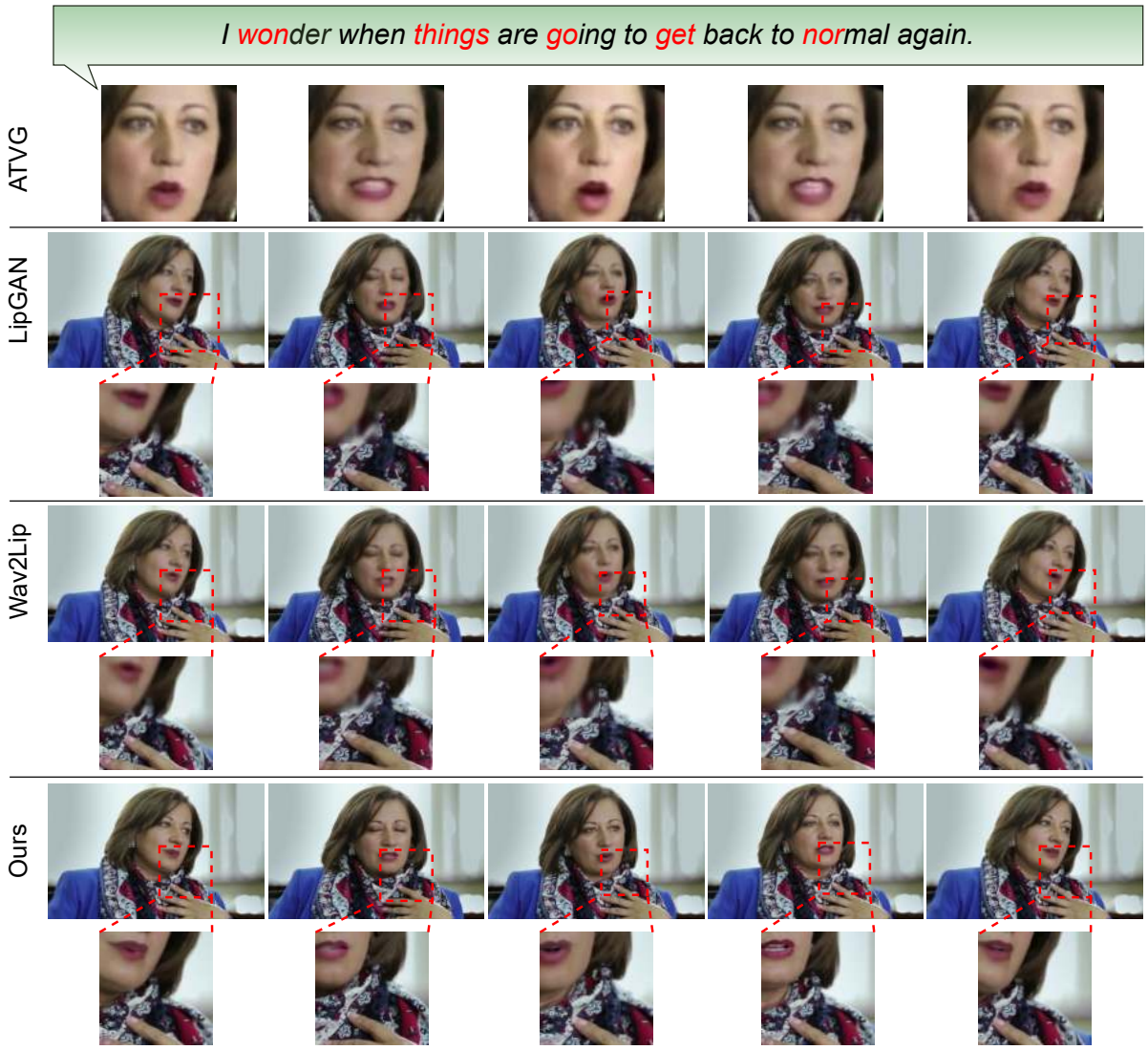


Figure 7.2: High resolution qualitative results (continued). We present additional qualitative results in video form here.

7.6 Conclusion

This chapter detailed the extension of the existing low-resolution solution to accommodate high-resolution training data. The discussion began by stating some of the challenges inherent in pursuing such an endeavour. We believe that it is due to these challenges that the field has largely shied away from actively working towards a high-resolution ADVD solution. Our findings show that the high-resolution solution, trained on the pre-processed AVSpeech dataset, produces photorealistic dubbed videos. In the following chapter, we delve into the second way in which we attempt to extend existing ADVD solutions – by improving their performance and robustness to non-frontal faces.

Chapter 8

Towards Pose-Invariant Visual Dubbing

8.1 Introduction

The primary objective of this research is to enhance the performance of ADVD solutions in terms of visual quality and lip-sync accuracy, while also expanding their capabilities to diverse settings and applications. The previous chapter discussed the first way in which we attempted to achieve this by introducing one of the first efforts toward utilizing high-resolution training data for the ADVD problem. In this chapter, we present the second approach by introducing the maiden attempt to achieve a pose-invariant ADVD solution.

8.2 Motivation

The pursuit for achieving a pose-invariant ADVD solution stems from the observation that existing solutions predominantly achieve satisfactory results when dubbing speakers with a frontal or near-frontal head pose. As the head becomes increasingly non-frontal, the performance of these solutions begins to deteriorate rapidly. Figure 8.1 showcases the results achieved by existing solutions when attempting to dub non-frontal faces. Evidently, the biggest aspect that is affected by dubbing non-frontal faces is the lip-sync accuracy which appears to be vastly impaired. Furthermore, an unpleasant shearing/tearing effect is present in the results produced by Wav2Lip [Prajwal *et al.* 2020] which gives the mouth region (specifically the teeth) a jagged appearance which reduces the naturalness of the result considerably.

We postulate that the phenomenon discussed above follows as a consequence of existing solutions being solely concerned with optimizing the visual quality and lip-sync accuracy whilst having little to no consideration for improving the capabilities and applicability of the solution. Furthermore, we believe that striving toward improving the robustness of ADVD solutions to non-frontal faces would significantly improve their applicability for real-world use cases. The reason for this is that non-frontal faces are inevitable and it would be unreasonable to expect all speakers to be in a frontal or near-frontal pose. Due to the significance of actively improving the performance of ADVD solutions for non-frontal faces, we believe that working towards addressing this problem would introduce a new research direction to the field.

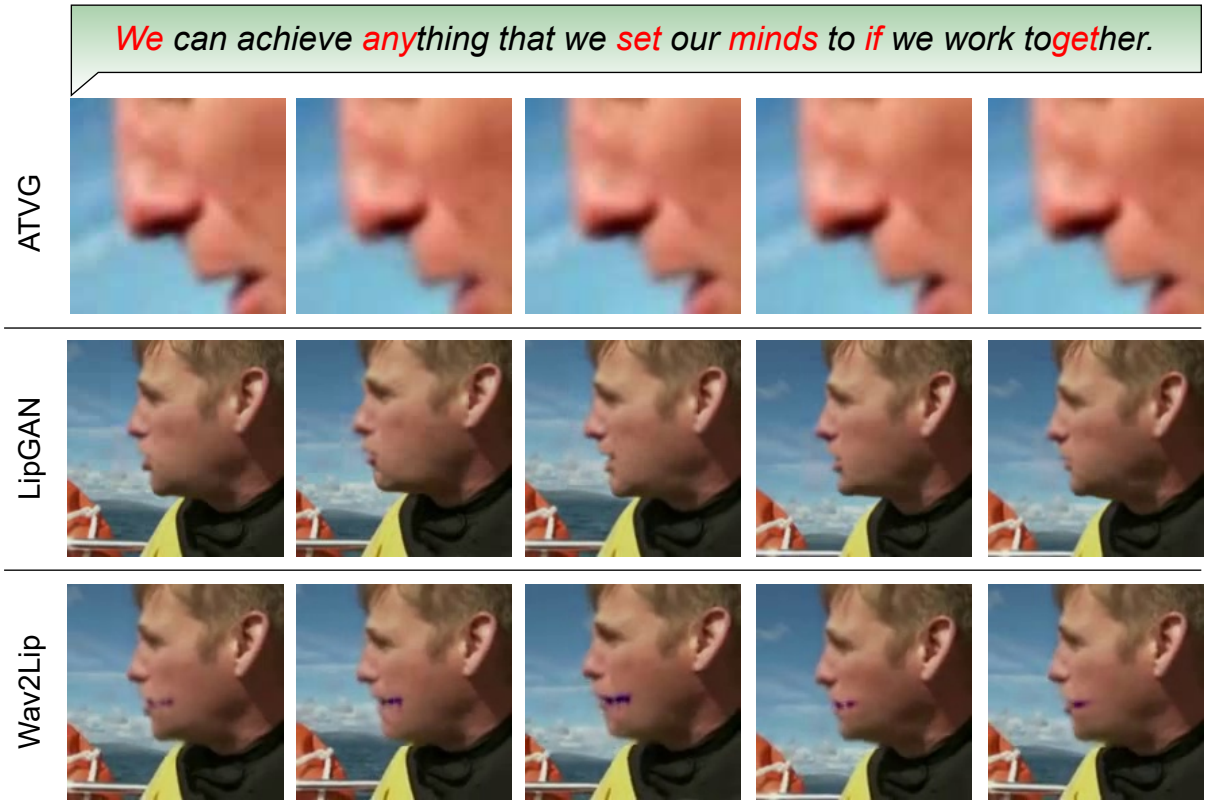


Figure 8.1: Results produced by existing solutions when attempting to dub non-frontal faces.

8.3 Challenges

To better understand the challenges posed by non-frontal faces, we start by examining how humans perceive frontal and near-frontal faces compared to non-frontal faces. When observing a human face from a frontal or near-frontal viewpoint, all facial attributes, including facial structure, eyes, and mouth region, are fully exposed and easily discernible. Consequently, training various face-related solutions on such views becomes straightforward. As the head pose becomes increasingly non-frontal along the yaw-axis, the view of the speaker's face becomes increasingly obscured, with half of the face self-occluded [Jiang *et al.* 2021a]. In this scenario, two options emerge i.e., (1) to rely solely on the partial view of the speaker's face, or (2) to adopt a human-like approach that exploits the inherent symmetry of the face to infer the occluded region [Hu *et al.* 2008; Dahmane *et al.* 2015]. While humans possess the innate ability to utilize facial symmetry to comprehend faces under varied poses, this presents a unique challenge for computer vision systems [Murphy-Chutorian and Trivedi 2008].

To contextualize the challenge posed by non-frontal faces to computer vision systems, it is important to note that this issue extends beyond the field of visual dubbing. Non-frontal faces have presented a longstanding challenge in various other face-related domains, including face recognition [Huang *et al.* 2000; Cheung *et al.* 2008; Ding and Tao 2016], emotion recognition [Hu *et al.* 2008; Tariq *et al.* 2012; Zheng *et al.* 2015], and lip-reading [Koumparoulis and Potamianos 2018; Isobe *et al.* 2021; Akhter *et al.* 2022]. To quantify the impact of non-frontal faces on these solutions, certain face recognition methods [Sengupta *et al.* 2016] have reported up

to a 10% decrease in accuracy between frontal and profile face recognition. Notably, these fields have actively addressed the challenge of non-frontal faces for several years, often employing face-frontalization methods [Hassner *et al.* 2015; Yin *et al.* 2017 2020] to *frontalize* the non-frontal faces and to subsequently perform the task to the frontalized faces which has proven to achieve promising results. This is in sharp contrast to the field of visual dubbing, where, to the best of our knowledge, no effort has been made to enhance the performance when dubbing non-frontal faces.

8.4 Problem Formulation

We argue that the aforementioned tactic may not be ideal in the case of visual dubbing as this would require performing an additional step to rotate the dubbed frontalized face back to the original non-frontal pose in a photorealistic manner. Conceptually, this procedure can easily become cumbersome due to the series of intermediate steps required to produce the dubbed result. From a practical perspective, frontalizing the face would not pose any issues since face frontalization is an active field of research and secondly, dubbing the frontalized result would not pose any issues either. Instead, the challenge arises when attempting to rotate the dubbed frontalized face back to the original non-frontal pose since the alignment with the original non-frontal face would need to be precise to achieve a photorealistic result, however, existing solutions are unable to achieve such a level of precision. Furthermore, since the sides of the dubbed frontalized face are not well exposed, rotating the face back to the original non-frontal pose exposes these regions which have missing/incorrect facial textures. This process is illustrated in Figure 8.2.

To shape our approach when addressing non-frontal faces, we began by defining the objectives that the solution should fulfil. Firstly, our approach should avoid a complete redesign of the existing solution, as such a step could potentially nullify prior contributions. Rather than straying from the current approach, we aimed to naturally extend it. Secondly, we aimed for the solution to be effective across the entire spectrum of head poses, spanning along the yaw-axis (as detailed shortly), encompassing angles within $[-90°, +90°]$. Lastly, efforts to enhance robustness to non-frontal faces should not inadvertently compromise the high levels of visual quality and lip-sync accuracy achieved by the existing solution for frontal and near-frontal poses.

While the aforementioned frontalization approach may not be directly applicable to visual dubbing, we appreciate its capacity to eliminate the need for constructing a separate non-frontal face dataset and the avoidance of a complete redesign of the existing solution to accommodate non-frontal faces. This straightforward yet intuitive solution modifies the nature of the input data to enhance the solution's robustness to non-frontal faces. Building upon this insight, in conjunction with our exploration of the factors limiting existing solutions from effectively dubbing non-frontal faces, we posit that the suboptimal outcomes achieved by current methods stem from the nature of the datasets typically employed (e.g., LRS2 [Afouras *et al.* 2018a] and VoxCeleb2 [Chung *et al.* 2018]) when addressing the visual dubbing problem. Specifically, our analysis reveals a significant bias towards frontal and near-frontal faces in these datasets, with non-frontal faces being notably underrepresented, as illustrated in Figure 4.5. Consequently, these findings may explain the satisfactory results that existing solu-

Figure 8.2: Illustration showing the process of extending the frontalization approach adopted by face recognition, emotion recognition, and lip-reading solutions, to visual dubbing. We see that frontalizing the non-frontal face and dubbing the frontalized face do not pose any issues, however, we observe that attempting to rotate the dubbed frontalized face back to the original non-frontal pose is unable to achieve photorealistic results. This follows since we observe an unpleasant stretching effect along the side of the speaker's face where the facial texture is uncertain.

tions achieve when dubbing frontal and near-frontal faces, as well as the poor results achieved when dubbing non-frontal faces.

In pursuit of addressing the abovementioned data bias, we deduced that, to the best of our knowledge, a large-scale unconstrained talking-face dataset containing a large proportion of non-frontal faces to train a pose-invariant ADVD solution does not exist. With regards to collecting our own dataset, we note that the demand for non-frontal footage is a niche, primarily being of value for certain cases of research, as well as to clinicians, VFX artists, forensics experts and police authorities [Anderson 2022]. We refrained from undertaking this endeavour due to it being a notoriously time-consuming and laborious process [Mattos and Oliveira 2018; Cheng *et al.* 2020]. Ideally, we aspired to utilize the abundance of frontal and near-frontal talking-face footage available to help improve the dubbing performance of our existing solution to non-frontal faces.

To tackle the data bias mentioned earlier, we drew inspiration from the frontalization approach mentioned earlier. However, rather than frontalizing non-frontal faces, we approached the inverse problem. This involves taking a frontal or near-frontal talking-face video and applying an arbitrary rotation to the head pose to generate a corresponding non-frontal talking-face video. Essentially, this can be seen as a *pose augmentation* that is applied to an existing dataset, synthesizing non-frontal talking-face videos from the vast collection of frontal and near-frontal talking-face videos at hand. By adopting this approach, the model is required to learn pose-invariant features to perform visual dubbing.

To contextualize the promise of this approach, Cheng *et al.* [2020] pose augment their dataset to perform pose-invariant lip-reading and achieve a 20.64% increase in accuracy compared to when the pose augmentation is not performed. Performing a pose augmentation obviates the need to collect our own non-frontal dataset and instead allows for existing large-scale

unconstrained datasets to be re-purposed. This ensures that any other capabilities of the solution are not compromised. Furthermore, performing a pose augmentation enables the head pose distribution of the dataset to be easily manipulated as required. Lastly, since this approach merely alters the nature of the data exposed to the solution, this precludes any changes to be made to the existing solution. Broadly speaking, because of this property, the pose-augmented dataset (or the pose augmentation pipeline itself) may be adopted by other existing solutions to be made pose-invariant as well.

8.5 Pose Augmentation In Practice

Our criteria for evaluating potential head rotation solutions to pose augment our dataset included the capability to (1) rotate the head to extreme angles such as profile views, (2) maintain computational efficiency, and (3) preserve the speaker's identity and original mouth movements. Leimkühler and Drettakis [2021] and Richardson *et al.* [2021] perform head rotation by projecting frontal images into the disentangled latent space of a pretrained GAN (using *GAN inversion* [Xia *et al.* 2022]), such as StyleGAN [Karras *et al.* 2019], and manipulating the latent codes corresponding to the viewpoint. However, the rotational range of this approach is limited since StyleGAN is predominantly trained on frontal faces. Other solutions utilize the input 2D RGB data to construct a 3D representation of the speaker's head which is subsequently rotated. Zhou *et al.* [2020a] proposed one of the first solutions to adopt this approach, however, the solution is unable to achieve satisfactory results for yaw angles greater than 60°. Recently, numerous solutions [Pan *et al.* 2020; Zhang *et al.* 2020] have adopted a *photo-geometric autoencoding* [Wu *et al.* 2020b] which factorizes the input image into its corresponding depth, albedo, viewpoint, and illumination. Since these solutions are online, iterative, and computationally expensive, they are unsuitable for our purposes.

Based on extensive experimentation, we established that the head rotation method proposed by Cheng *et al.* [2020] is the only solution that meets all our criteria. Therefore, this solution was adopted to pose augment our dataset. Section 2.3.2 presents a comprehensive explanation of how this head rotation solution works. In summary, head rotation is performed through 3DMM fitting [Blanz and Vetter 1999] and defining a set of key points at which depth is computed. The depth information is then triangulated using the Delaunay algorithm [Lee and Schachter 1980] to construct a 3D facial structure that is rotated to produce the non-frontal result. An additional property that makes this solution particularly suitable for our use case is that the solution is speaker independent i.e., the solution does not require an abundance of footage of the speaker to accurately reconstruct their identity. Instead, the solution acts on each frame independently and may be employed on a frame-by-frame basis to produce a resulting non-frontal video. Furthermore, the solution achieves photorealistic results whilst preserving the speaker's identity and original mouth shapes, despite allowing rotations along the pitch, yaw, and roll axes up to extreme poses. Figure 8.3 showcases the photorealistic results achieved when incrementally rotating the speaker's face along the yaw axis up to extreme (profile) views. Lastly, the solution preserves the speaker's background which significantly improves the naturalness of the result produced.

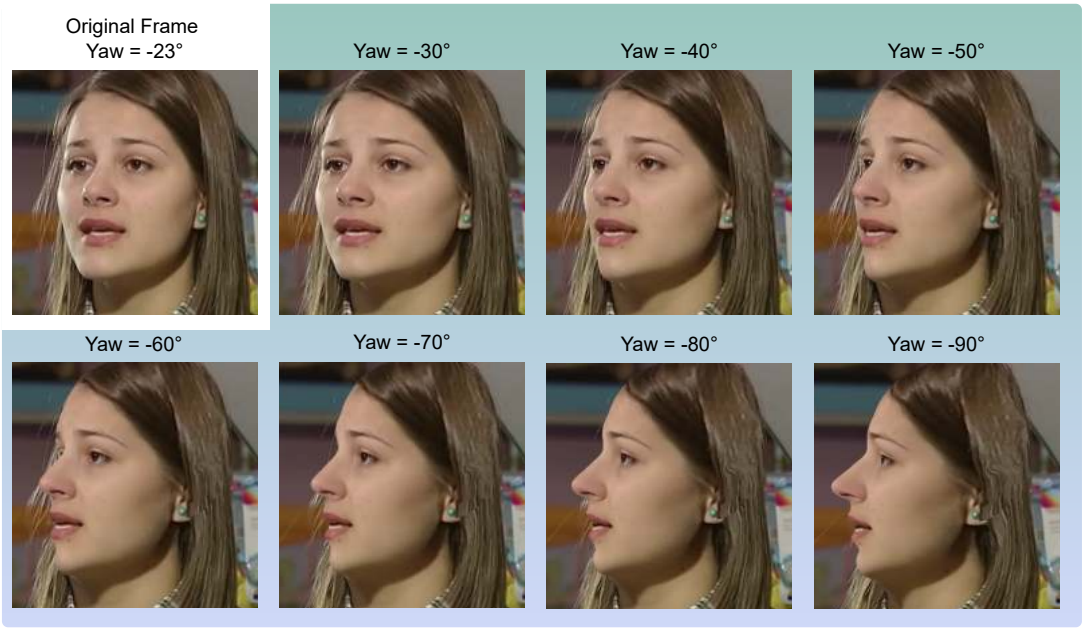


Figure 8.3: Illustration showcasing the capabilities of the head rotation solution [Cheng *et al.* 2020] that we adopt to pose augment our dataset. We see that the solution is able to rotate the head up to profile-view (±90°) whilst preserving the speaker's identity mouth shape, and background, thus, achieving photorealistic results.

8.6 Proposed Solution

Given our understanding of what makes pose augmentation the most suitable solution for our needs, along with the rationale of how the pose augmentation works, we now explain how we configure and perform pose augmentation to achieve optimal results. We begin by establishing the scope of our solution in that we only aim to improve the robustness to non-frontal rotated along the yaw axis. This follows since head rotations along the yaw axis are the most common in real-world videos, and the yaw axis is the axis along which the head pose varies the most. When surveying the literature pertaining to other facial-related applications e.g., face recognition [Huang *et al.* 2000; Cheung *et al.* 2008; Ding and Tao 2016], emotion recognition [Hu *et al.* 2008; Tariq *et al.* 2012; Zheng *et al.* 2015] and lip-reading [Koumparoulis and Potamianos 2018; Isobe *et al.* 2021; Akhter *et al.* 2022], we noticed the lack of consensus as to what should be considered as a frontal, near frontal, and non-frontal face. In an attempt to reduce the ambiguity of these subjective terms, we consider a head pose to be:

- *Frontal*: if the yaw angle resides within the range $[-5°, +5°]$
- *Near frontal*: if the yaw angle resides within the range $[-25°, -5°) \cup (+5°, +25°]$
- *Non-frontal*: If the yaw angle resides within the range $[-90°, -25°) \cup (+25°, +90°]$

When pose augmenting the dataset, it is crucial to determine which videos should undergo augmentation, as it is unnecessary to augment every single video. Additionally, a means of classifying each video as either frontal, near-frontal, or non-frontal, is required, considering that the speaker's head pose may vary rapidly within a short period of time. To achieve this,

head pose estimation [Guo *et al.* 2018] is performed where each frame is classified based on the estimated yaw angles and pose intervals defined previously. Ultimately, the video is classified based on the majority class of its frames.

We refrain from performing pose augmentation to frontal videos due to the limited visibility of the face's sides in such poses. Rotating the head to a non-frontal pose could expose areas with missing or incorrect facial textures [Cheng *et al.* 2020; Zhou *et al.* 2020a]. Therefore, only near-frontal talking-face videos were considered for pose augmentation. Additionally, pose augmentation was applied to these videos with a 50% probability to ensure that not all near-frontal videos underwent this process. When a video requires pose augmentation, the yaw angle of the first frame is estimated and denoted as γ. If $\gamma \in [-90°, 0°)$, indicating that the speaker's head is turned to the right, a target yaw angle is uniformly sampled from the range $[-90°, \gamma°]$ to which the head is rotated. On the other hand, if $\gamma \in [0°, +90°]$, indicating that the speaker's head is turned to the left, a target yaw angle within the range $[\gamma°, +90°]$ is uniformly sampled. Although the target yaw angle is uniformly sampled, it is possible to prioritize a specific range of head poses by adjusting the sampling strategy as needed.

The approach taken ensures that the head pose is rotated further only in the direction it is currently facing. Sampling a target head pose outside the specified ranges would result in a rotation in the opposite direction, thereby exposing self-occluded regions with missing facial textures. By utilizing the initial head pose γ and the target head pose, we calculate Δy, which represents the number of degrees that the head needs to be rotated. The resulting non-frontal video is generated by rotating the head pose of each frame by $\Delta y°$, thereby preserving the speaker's relative head motion and improving the naturalness of the produced result.

Similar to our initial implementation (explained in Chapter 4, Section 5.2, and Section 6.4), we employ the LRS2 dataset [Afouras *et al.* 2018a] for all experiments. This choice is motivated by our objective to demonstrate the proof of concept for achieving a pose-invariant ADVD solution through pose augmentation. Performing such experiments using high-resolution data would be impractical due to the challenges outlined in Section 7.3, which would be further compounded by pose augmentation. Consequently, we postpone this endeavour to future work, as discussed in Section 10.2.2. Essentially, the pose-invariant solution presented here is trained by modifying the nature of the training data, while keeping all other aspects unchanged, such as input representations, network architecture, and training hyperparameters. Figure 8.4 illustrates the impact of pose augmentation on the yaw head pose distribution of the LRS2 (train) dataset.

8.7 Evaluation

The only modification made to the existing solution to achieve pose invariance is the alteration of the training data's nature. Accordingly, the same evaluation procedure detailed in Section 6.9 is followed. Our objective was to assess the solution's effectiveness across a broad spectrum of head poses, necessitating careful consideration of the evaluation dataset. In addition to the desired data properties outlined in Section 4.2, we introduce an additional requirement i.e., the inclusion of talking-face footage showcasing a diverse range of head poses. Throughout our investigation, we explored several prospective datasets, including those by Anina *et al.* [2015], Wang *et al.* [2020], Zhang and Fisher [2019], and Abdrakhmanova *et*

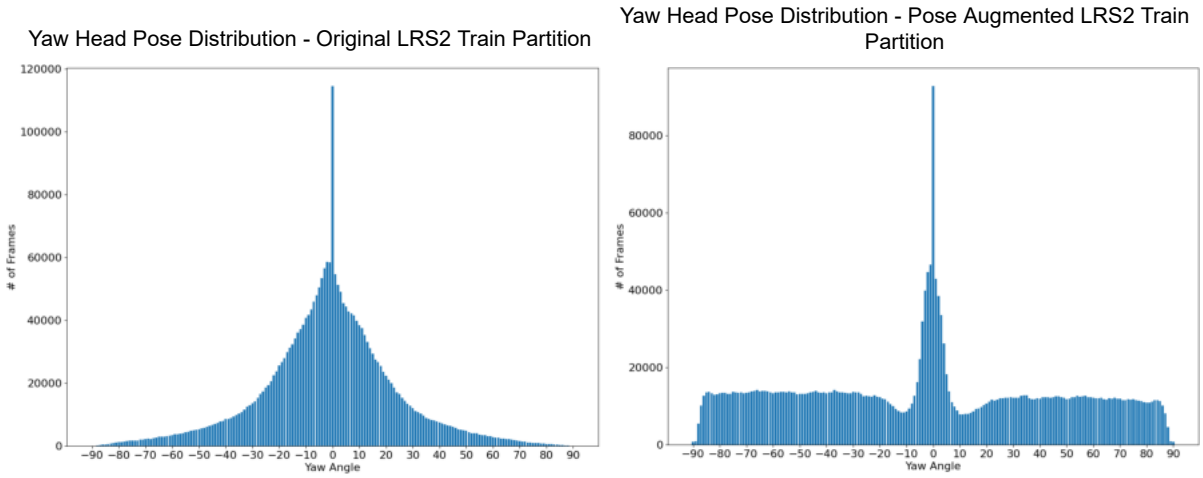


Figure 8.4: Illustration showing the impact that the pose augmentation has on the yaw head pose distribution of the LRS2 (train) dataset. Evidently, the original dataset is biased toward frontal and near-frontal faces, whereas non-frontal faces are under-represented. In contrast, the pose augmentation increases the exposure to non-frontal footage considerably. Furthermore, since the target head pose is uniformly sampled, the exposure to all non-frontal yaw angles is uniform (with approximately 15K frames for each angle).

al. [2021]. Unfortunately, none of these datasets managed to achieve a satisfactory balance among the sought-after data properties. Ultimately, our inquiry led us to the conclusion that the LRS2 test dataset [Afouras *et al.* 2018a] offers the most favourable compromise. Due to the dataset's limited representation of non-frontal head poses, we adapted our quantitative analysis approach to provide a detailed assessment of each solution's performance as the head pose becomes increasingly non-frontal.

8.7.1 Quantitative Results

For significantly improved readability, we present the quantitative results of the pose-invariant solution in Table A.2 (in Appendix A.3), dissecting the performance of each solution across the range of $[-90°, +90°]$ in $10°$ intervals. Each solution maintains its relative ranking in terms of visual quality and lip-sync accuracy, similar to the findings presented in Chapter 6. This becomes most evident when analyzing the results obtained in the simplest case, i.e., when the face is frontal. This outcome implies that, as before, the visual quality and lip-sync accuracy of ATVG [Chen *et al.* 2019] are much worse than LipGAN [KR *et al.* 2019]. Likewise, LipGAN achieves comparable measures of visual quality but exhibits reduced levels of lip-sync accuracy compared to Wav2Lip [Prajwal *et al.* 2020]. Lastly, the presented solution achieves superior measures of visual quality, whereas Wav2Lip maintains superior measures of lip-sync accuracy, as observed previously.

In addition to evaluating the overall performance of each solution, we are particularly interested in tracking the evolution of their performance as the head becomes increasingly non-frontal. Firstly, we notice that as the head transitions from a frontal to a non-frontal pose, the rate of degradation in performance is similar, regardless of whether the rotation is to the left or right.

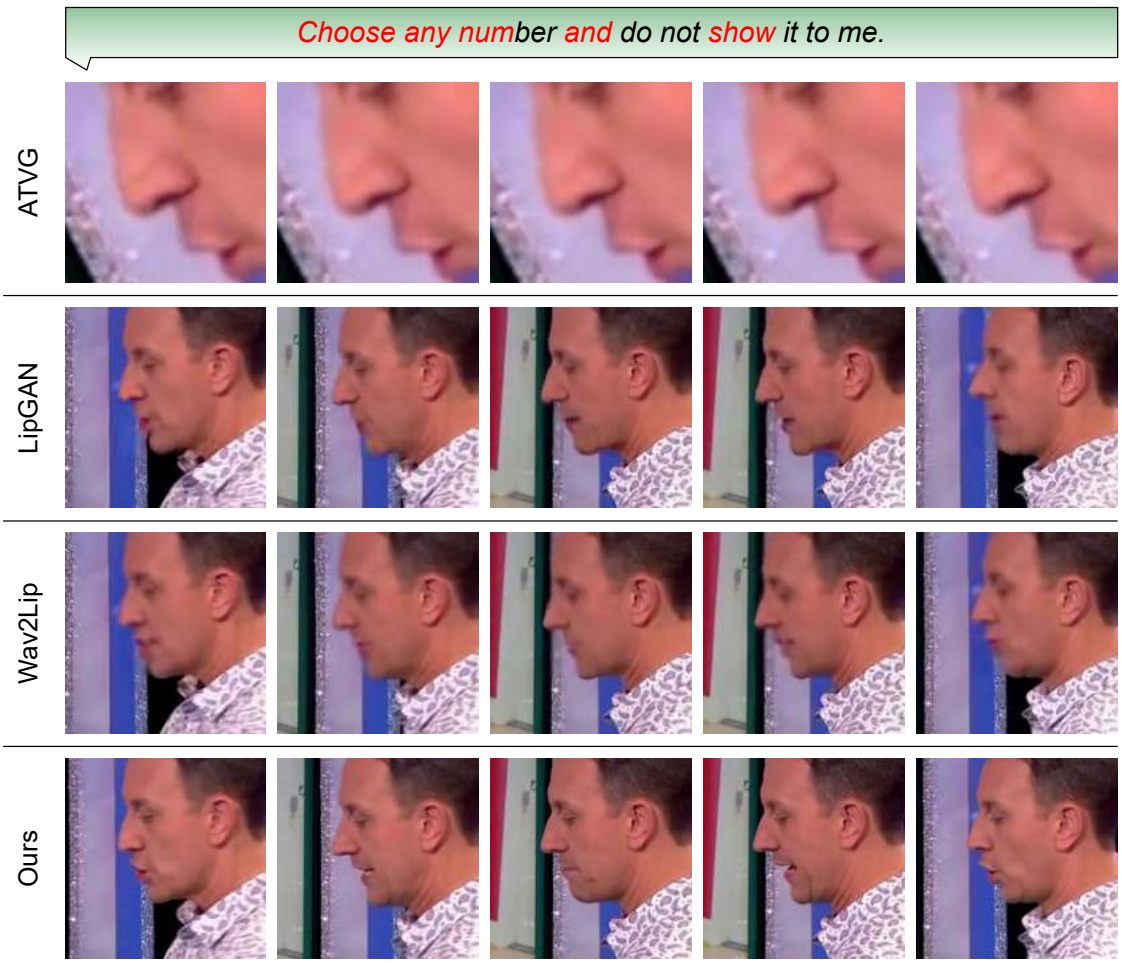


Figure 8.5: Head rotation qualitative results.

Now delving deeper, Table 8.1 presents the difference in performance (as a percentage) between the metrics achieved for frontal faces, to those achieved for non-frontal faces (specifically, faces rotated 80°+ to the right). A negative measure indicates a decrease in the metric when transitioning from a frontal to a non-frontal pose, whereas a positive value denotes an increase in the metric.

Solution/Metric	**SSIM (%)**	**CPBD** (%)	**FVD** (%)	**LS-Conf.** (%)	**LS-Dist.** (%)
ATVG [Chen *et al.* 2019]	−15.90	−22.89	+138.27	−12.02	+6.78
LipGAN [KR *et al.* 2019]	−8.65	−20.67	+151.52	−35.52	+11.33
Wav2Lip [Prajwal *et al.* 2020]	−8.21	**−15.52**	+113.59	−8.65	+4.83
Our Solution	**−4.28**	−24.19	**+33.97**	**−2.90**	**+1.36**

Table 8.1: Table showing the evolution of each metric as the head pose transitions from a frontal to a non-frontal pose.

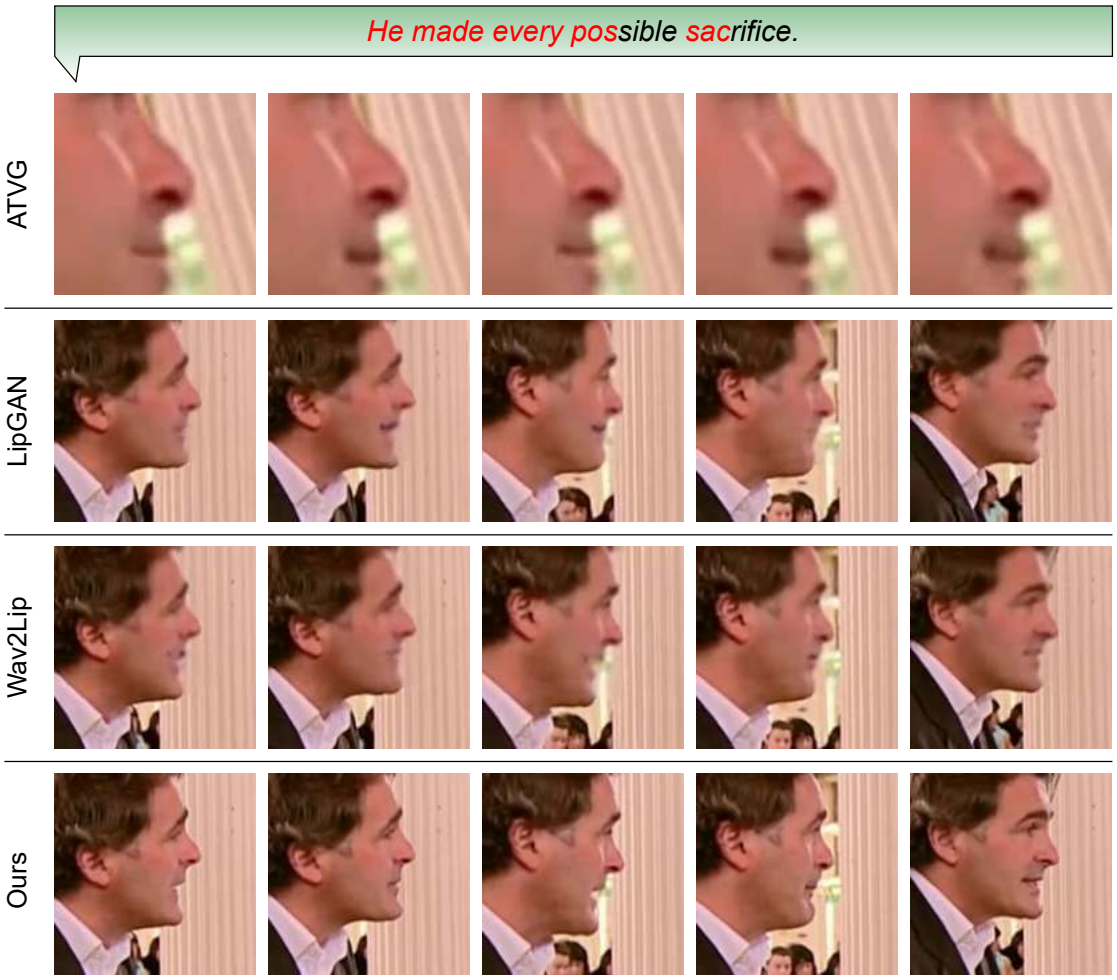


Figure 8.6: Head rotation qualitative results (continued). We present additional qualitative results in video form here.

The results presented above indicate that ATVG [Chen *et al.* 2019] experiences the most pronounced performance degradation, followed by LipGAN [KR *et al.* 2019], and then Wav2Lip [Prajwal *et al.* 2020], as the head becomes increasingly non-frontal. Furthermore, these results suggest that, unlike prior solutions that face substantial performance decline, the presented solution encounters a less severe impact. This suggests that the presented solution is notably more resilient to diverse head poses, thus, showcasing the efficacy of pose augmenting the training data to improve the solution's robustness to non-frontal faces.

8.7.2 Qualitative Results

Figures 8.5 and 8.6 showcase the results obtained from each comparative solution when dubbing extreme non-frontal (profile) faces. In addition to the extensively discussed limitations of the ATVG solution [Chen *et al.* 2019] outlined in previous chapters, a significant misalignment issue is evident in the results produced, with each output displaying an unusual tilt. We hypothesize that this anomaly stems from the solution's reliance on landmarks, a challenging task when estimating landmarks for profile faces due to self-occlusion. Regarding lip-sync accuracy, it is apparent that the solution faces considerable difficulty, as evidenced by the static

appearance of rigid mouth movements. Upon analyzing the outcomes achieved by LipGAN [KR *et al.* 2019] and Wav2Lip [Prajwal *et al.* 2020], especially in Figure 8.6, LipGAN produces unnatural mouth movements. This limitation potentially arises from uncertainties when determining the appropriate mouth shape based on the input dubbing audio. Moreover, there is an apparent increase in blur for both solutions. Due to the surface area of the speaker's mouth region being reduced for non-frontal faces, as well as the increase in blur, both solutions appear to blur the speaker's mouth with the rest of the speaker's face, especially for bilabial utterances. In contrast, our solution maintains an impressive level of visual quality, preserving intricate details (e.g., facial imperfections and teeth) that other methods fail to capture. Furthermore, a coherent synchronization between the speaker's mouth movements and the spoken phrase is observed, suggesting that the lip-sync accuracy of our solution does not deteriorate as rapidly compared to alternative methods.

8.8 Conclusion

This chapter presented our strategy for developing the first pose-invariant ADVD solution. In summary, the nature of the datasets typically used when addressing the ADVD problem was identified as the root cause of the poor results achieved by existing solutions when dubbing non-frontal faces. This issue was addressed by pose augmenting the training dataset to synthesize talking-face footage at arbitrary head poses whilst leveraging the abundance of frontal and near-frontal talking-face footage available. Results showed that doing so significantly reduces the rate of degradation of our solution's performance as the head becomes increasingly non-frontal. The following chapter discusses the ethical considerations of this research, as well as its numerous real-world applications.

Chapter 9

Applications & Ethics

9.1 Introduction

As stated throughout this document, the presented solution attempts to address the ADVD problem which pertains to employing input dubbing audio to drive (morph) the speaker's mouth movements in the input talking-face video. Furthermore, the goal is to give the viewer the impression that the speaker is genuinely uttering the dubbing content whereas this is not the case in reality. Taking the nature of this problem into account, it is crucial to consider the ethical implications of our contribution and in turn, to minimize the probability of it being exploited for malicious purposes. Therefore, this chapter is dedicated to discussing the ethical implications of the presented solution, along with an overview of its numerous real-world applications.

9.2 The Issues of Deepfakes

Upon inspecting the nature and philosophy of the presented solution, the dubbed videos produced may be considered as a form of *deepfake*. Here, we define a deepfake as:

> *A form of digital media that has been synthesized using deep-learning technology with the intention of warping a person's demeanour, such as their voice, appearance, utterances, etc., in a realistic manner.*

Since its emergence in 2017 [Somers 2020], the term deepfake has been associated with a strong negative connotation, stemming from numerous instances in which the technology has been exploited for malicious purposes [Kendja 2021; Finger 2022; Jaiman 2023]. Given that the dubbed videos produced by our solution could be considered as a form of deepfake, it is imperative to impartially evaluate the ethical considerations of both deepfakes and, by extension, our presented solution. This involves conducting a comprehensive exploration of their positive and negative applications. Through this examination, our goals are two-fold i.e., (1) to determine the extent to which concerns about their misuse are warranted, and (2) to emphasize that we are not oblivious to the potential misuse of our solution. Our assessment begins by mentioning some of the ways in which deepfakes have been used malevolently in the past.

9.2.1 The Synthesis of Non-Consensual Pornographic Content

One of the earliest reported instances of deepfakes involved the creation of non-consensual pornographic content by overlaying the faces of prominent female public figures, including celebrities, onto pornographic material [Newton and Stanfill 2020]. This act is often carried out without the victim's knowledge or consent, aiming to inflict psychological harm through defamation, humiliation, blackmail, and other means, potentially resulting in severe ethical, societal, and financial repercussions [Gosse and Burkell 2020]. Notably, a significant portion of online deepfakes, approximately 96%, manifest as non-consensual pornographic content [Zeng and Olivera-Cintrón 2019].

9.2.2 Impersonation Attacks

In the realm of audio deepfakes, voice-cloning techniques have recently been leveraged for fraudulent activities [Amezaga and Hajek 2022; Veerasamy and Pieterse 2022]. As a notable instance, Brewster [2021] shows that in 2020, attackers successfully cloned the voice of a prominent banking client which they exploited to authorize a bank transfer to their account. As a consequence of the high fidelity of the audio which led to the bank authorizing the transaction, this sophisticated attack ultimately led to the bank being defrauded for $35 million. Within the visual domain, deepfakes have been used to *spoof* biometric systems such as facial recognition and liveness detection systems [Wojewidka 2020].

9.2.3 The Spread of Misinformation

Numerous solutions akin to ours have been utilized to create deepfakes of individuals, often targeting celebrities, business leaders, or politicians, portraying them engaging in actions or uttering statements that never occurred in reality. To help put the severity of this concern into perspective, it is essential to consider the potential impact of deepfakes on significant events such as presidential elections [Diakopoulos and Johnson 2021]. This scenario could involve an individual crafting a deepfake of a candidate, falsely attributing fabricated or offensive content to them with the intent of tarnishing their image [Vaccari and Chadwick 2020]. As a consequence of the division, distrust, and confusion that this may cause the population, this may lead to mass protests, violence, and the public undermining the credibility of presidential elections and the media in general [Ireton and Posetti 2018]. Furthermore, as a consequence of the high fidelity of deepfakes that are indistinguishable from real content, this may permit an individual to falsify the validity of any (audio or video-based) evidence held against them for potential misconduct on the basis of *fake news*.

9.3 Positive Use-Cases of Deepfakes

The discussion presented above explains some of the ways in which deepfakes may be weaponized for malicious purposes and also highlights the detrimental consequences that may follow. In turn, this discussion emphasizes the importance of ensuring that deepfakes are used ethically. We believe that the media has been instrumental in educating the general public on the negative use cases of deepfakes, however, we now discuss an aspect that is underrepresented in the mainstream media i.e., the positive use cases of deepfakes. Below, we summarize a hand-

ful of positive use cases of deepfakes with a specific emphasis on the use cases of our ADVD solution.

9.3.1 Voice-Cloning for Voice *Restoration*

In cases in which an individual has lost their voice due to an illness such as cerebral palsy, Amyotrophic Lateral Sclerosis (ALS), etc., voice-cloning techniques have been employed in an attempt to "restore" the individual's voice [Meskys *et al.* 2020]. Specifically, as opposed to settling for text-to-speech (TTS) services which offer generic computer-generated voices, companies such as *CereProc*[7] instead clone the voice of a voice donor that aligns with the personality of the individual that has lost their voice. It is believed that the act of "restoring" voices plays an integral role in which the individual communicates since voice may be considered as a cornerstone of an individual's identity [Belin *et al.* 2011]. Furthermore, if the individual were to settle for a computer-generated voice, the misalignment between the voice and their personality would make for an alienating experience.

9.3.2 Positive Use-Cases of Visual Dubbing Solutions

9.3.2.1 Localization of Video Content

The primary objective of our presented solution is to improve the manner in which foreign video content such as movies, television programmes, YouTube videos, etc. are localized i.e., made comprehensible to the local audience. This follows since we find it shameful that videos have been localized using the same techniques introduced several decades ago i.e., subtitling and audio dubbing [Tveit 2009]. In contrast to these methods, visual dubbing attempts to revolutionize the localization process by using the dubbing audio to reenact the speaker's mouth movements which mitigates the shortcomings of traditional methods.

We aspire for visual dubbing solutions to be widely adopted by video-streaming services such as Netflix, Disney+, YouTube, etc. This follows since the enhanced viewing experience achieved by visual dubbing is expected to increase viewership. Furthermore, following the monumental success of foreign productions such as *Squid Game* and *Narcos* amongst many others, viewers have recently been acquainted with the immense potential of foreign content [Emmys 2022; IDMb 2022]. As a result, the significantly improved viewing experience achieved is expected to make foreign content equally as appealing as native content, consequently broadening the viewer's scope of video content consumed. Reflexively, this is expected to also broaden the audience to which video content is consumed as opposed to being confined to the audience that is familiar with the language in which the video is originally presented. To the best of our knowledge, *Flawless AI*[8] is the only commercially available visual dubbing solution, however, there exists a plethora of solutions that instead automatically produces the dubbed audio without dubbing the speaker's mouth movements.

[7] https://www.cereproc.com/en/LeeRidley
[8] https://www.flawlessai.com/

9.3.3 Multilingual Advertising Campaigns

As a notable application of deepfakes, a health charity recently partnered with David Beckham as part of the *Malaria No More* campaign [De Ruiter 2021]. In the broadcasted video[9], David Beckham appears to speak and seamlessly transition between 9 different languages including Hindi, French, and Mandarin amongst many others. To produce this video, David Beckham's voice was cloned to produce the dubbed audio segments which were subsequently used to dub his mouth movements. Taking this exceptional application of audio and video deepfake techniques into account, we identify 2 major takeaways - Firstly, the resulting video successfully captivates the viewer's attention and in turn, effectively conveys the message to viewers. Secondly, the use of deepfake techniques reduces the production costs and time required substantially in comparison to using traditional methods. Specifically, as opposed to recording the video once and being able to dub the video to any desired language, this would otherwise have required David to learn the script in 9 languages, thus, necessitating considerably more recordings to be conducted.

9.3.4 Teleconferencing

As a consequence of the COVID-19 pandemic, most social interactions were migrated to teleconferencing platforms [Wong *et al.* 2021]. Unfortunately, in regions with sparse internet connectivity, even a simple audio-visual call would pose a challenge due to the high internet requirements. Under certain circumstances, preserving the video stream may be mandatory especially since the level of engagement is significantly higher in the case of audio-visual interactions compared to audio-only interactions [Heath *et al.* 1997; Richardson *et al.* 2020]. To remediate this issue, visual dubbing may be performed as a form of *video compression* [Zhou *et al.* 2020b; Doukas *et al.* 2021] i.e., as opposed to having a real-time/live video stream, the video stream may be periodically sampled (e.g., every 5 seconds) and the speaker's mouth movements in the sampled frame can be dubbed according to the audio stream. By adopting this approach, we expect to achieve a tolerable audio-visual experience whilst reducing the internet requirements immensely.

9.3.5 Digital Avatars

Following the emergence of virtual assistants such as Siri, Cortana, Google Assistant, etc., which play an integral role in user experience, it may be argued that this experience may be enhanced if we were to augment a visual component to them [Thies *et al.* 2020]. For instance, a user may wish to assign a face to the virtual assistant's voice and as mentioned previously, the level of engagement is considerably higher for audio-visual interactions in comparison to audio-only interactions [Heath *et al.* 1997; Richardson *et al.* 2020]. Allowing the user to select a digital avatar for their virtual assistant and dubbing the mouth region with respect to the audio stream, is expected to improve the naturalness and overall experience of virtual assistants. An additional, yet futuristic, application of visual dubbing is to dub digital avatars in the metaverse to achieve a more realistic experience. This endeavour is currently being

[9] https://youtu.be/QiiSAvKJIHo

explored by Nvidia using their *Audio2Face* solution[10], however, the lip-sync accuracy achieved is not yet satisfactory.

9.4 The Ethics of Deepfakes

Based on our discussion of both views regarding the use of deepfakes, it is evident that there exist myriad positive use cases for deepfakes (and our solution). Therefore, it is irrational to deem deepfakes as *inherently morally wrong* just because they have the potential to be exploited for malicious purposes. Consequently, it is crucial that we take appropriate measures to minimize the probability of our visual dubbing solution being exploited for nefarious purposes. We are also aware that determining the ethics of deepfakes has been notoriously challenging since their inception, and we believe that their ethics should be considered on an application-specific basis. In summary, we believe that the ethics of deepfakes (and by extension, our solution) should be determined based on the following three factors:

9.4.1 Accessibility

We postulate that determining the ethics of deepfakes is primarily based on the users to whom deepfake synthesis solutions are made accessible to. Moreover, the numerous instances of deepfakes being exploited for malicious purposes may be attributed to making such powerful technology readily available to the general public. This poses a major concern since the general public is not fully aware of the potential dangers and severe repercussions that may follow when synthesizing deepfakes in an uninformed manner. This issue is further exacerbated by the fact that, in comparison to PhotoShop which requires some level of skill, deepfakes are far more straightforward to produce [Malhi 2022; Kempen 2022]. Consequently, we argue that to a great extent, the misuse of deepfakes can be reduced substantially by solely granting access to individuals that are expected to be sufficiently aware of the responsibilities when working with deepfakes, such as academics, industry professionals, etc.

9.4.2 Intention

In the case in which access to deepfake synthesis solutions is unrestricted, it is crucial also consider the user's intention for synthesizing deepfakes. As detailed previously, malicious users may intend on defaming or humiliating an individual as in the case of non-consensual pornography, or instigating mayhem during a political campaign which could result in mass protests and violence. In contrast, numerous users have also proven to use deepfakes with benign intentions, such as:

9.4.2.1 Raising Awareness

As an infamous example, BuzzFeed[11]used the idiosyncrasies of comedian Jordan Peele to synthesize a deepfake of Barack Obama making out-of-character statements. An additional ex-

[10] https://www.nvidia.com/en-us/omniverse/apps/audio2face/
[11] https://youtu.be/cQ54GDm1eL0

ample is *Deep Tom*[12] which synthesizes deepfakes of Tom Cruise for entertainment purposes whilst also raising awareness of the immense potential of deepfake technology. In such cases, the context is known, and the videos are clearly marked as fake/synthesized to prevent causing distress to the public.

9.4.2.2 Video Localization

Video localization methods attempt to address the language barrier inherent in the consumption of video content. This is achieved through devising ways of making video content originally presented in a foreign language, understandable to a local audience.

We hypothesize that the decline in negative use-cases and the promotion of positive/benign use-cases of deepfakes naturally follows as a consequence of the manner in which access to deepfake synthesis solutions is regulated as stated previously.

9.4.3 Consent

The final aspect that we consider when determining the ethics of deepfakes is that of *consent*. Since the process of synthesizing deepfakes is performed as a post-processing step to existing media, the individual(s) present in the deepfake may be unaware, thus, would not grant consent, to their content being manipulated without their knowledge. Alternatively, the individual may decline to grant consent due to the manner in which the synthesized deepfake portrays them [De Ruiter 2021].

In summary, we make the apt comparison between the use of deepfakes and medicinal drugs – when made accessible to those who genuinely require access, used by individuals with the correct intention, and with the necessary consent granted, both have an immensely positive impact on society. Conversely, when any of the 3 aforementioned requirements are not satisfied, this may lead to devastating consequences.

9.5 Preventative Measures

To establish the impact that our solution has on the field of visual dubbing solutions and how these solutions are utilized, we begin by noting that our solution does not exacerbate the misuse of deepfakes in any way. This follows due to the numerous publicly available solutions [Chung *et al.* 2017; KR *et al.* 2019; Prajwal *et al.* 2020] already in existence. In an attempt to minimize the likelihood of our solution being exploited for malicious purposes, we may consider releasing our code and models on a licence basis in which the applicant would be subject to an exhaustive screening process. In doing so, we are able to establish the role and intentions of the applicant and secondly, we restrict access to only academics, industry professionals, etc. i.e., preventing access by the general public. The rationale for wanting our solution to be accessible (albeit, to a minor degree) follows since we aspire for our solution to be a significant research contribution toward improving the visual quality, lip-sync accuracy

[12] https://www.tiktok.com/@deeptomcruise?lang=en

and generalizability of ADVD solutions. Furthermore, this also improves the transparency of our contribution which allows for the true capabilities of our solution to be easily established.

As an additional preventative measure, we may also consider embossing a translucent watermark onto all results produced by our solution. This is expected to make it easier for the untrained eye to establish that the result is synthesized whilst clearly showcasing the true capabilities of our solution. We believe that we adopt a significantly more proactive stance toward combating the misuse of deepfakes compared to the majority of existing solutions. Specifically, numerous solutions [KR *et al.* 2019; Prajwal *et al.* 2020] simply encourage that all results should be clearly and unambiguously labelled as synthesized. We do not consider this to be a sufficiently effective measure since it would be virtually impossible to enforce such a requirement in reality.

9.6 Conclusion

In this chapter, we debated the ethics of deepfakes and by extension, the ethics of our presented solution. This was achieved by taking cognizance of the pros and cons of deepfakes, after which we defined an outline based on which we believe the ethics of deepfakes (and by extension, our solution) should be determined. Furthermore, we discussed the several preventative measures that we may take to ensure that the likelihood of our solution being exploited for malicious purposes is minimized. The following chapter discusses the future work of our solution as to how it may be extended and improved on.

Chapter 10

Conclusions & Future Work

10.1 Introduction

The preceding chapter addressed the ethical aspects of the presented solution. This chapter commences by outlining the potential future directions for this research. These include validating our hypothesis that Wav2Lip achieves imperceptible levels of lip-sync accuracy to humans, thus strengthening the analysis presented in Section 6.9. Moreover, we aim to consolidate all our contributions into a high-resolution pose-invariant ADVD solution, and to explore the broader challenge of addressing the complete dubbing problem. Finally, the conclusion section in Section 10.3 provides a summary of the salient points discussed in this document.

10.2 Future Work

10.2.1 To Investigate Whether Wav2Lip [Prajwal *et al.* 2020] Achieves Levels of Lip-Sync Accuracy That Are Imperceptible to Humans

As shown in Section 6.9, the presented solution achieves superior visual quality metrics, while Wav2Lip [Prajwal *et al.* 2020] outperforms in terms of lip-sync metrics. Despite our solution achieving significantly lower measures of lip-sync accuracy compared to Wav2Lip, the human subjective study revealed that our solution achieves measures of lip-sync accuracy that is on par with Wav2Lip as perceived by humans. Recall that humans are considered the supreme evaluators when assessing dubbed videos since videos are dubbed for the viewing pleasure of humans [KR *et al.* 2019]. Furthermore, some works [Zhou *et al.* 2021] have shown that in some cases, Wav2Lip achieves higher lip-sync metrics than ground-truth data. Taking these points into account, we hypothesize that Wav2Lip tends to over-optimize the lip-sync metrics achieved beyond what can be perceived by humans. As shown by Chung and Zisserman [2016b]; ITU [1998], the threshold for detecting sync errors by an average human viewer is approximately -125ms (the audio lags the video) to +45ms (the audio leads the video).

We believe that, by over-optimizing the lip-sync beyond what can be perceived by humans, this results in an unnecessary deterioration in visual quality due to the trade-off between visual quality and lip-sync accuracy when addressing the ADVD problem [Prajwal *et al.* 2020]. To test this notion, we propose using the AV offset measure produced by SyncNet [Chung and Zisserman 2016b]. This measure could then be thresholded (employing the aforementioned human thresholds) to quantify the number of videos achieving an AV offset imperceptible to humans. In doing so, we gain an intuition as to how to prevent the lip-sync accuracy from being overoptimized at the expense of the visual quality achieved.

10.2.2 High-Resolution Pose-Invariant Visual Dubbing

As shown throughout this document, the presented solution advances the field of visual dubbing along several axes. This is achieved by improving the visual quality and lip-sync accuracy achieved (Chapter 5 and Chapter 6), presenting one of the first high-resolution ADVD solutions (Chapter 7), and presenting the first pose-invariant ADVD solution (Chapter 8). Note that each of these efforts was pursued independently, therefore, it is desirable to culminate these efforts to develop a high-resolution pose-invariant ADVD solution. As discussed in Section 7.3, undertaking such an endeavour would entail attaining an appropriate dataset, and gaining access to sufficient computing resources and as a result, training such a solution would be a time-consuming process. Specifically, this would require pose-augmenting the AVSpeech dataset [Ephrat *et al.* 2018] which would be a time-consuming process, and the immense computing resources required to train such a solution would make obtaining them even more challenging. By succeeding in training such a solution, this is expected to (1) elevate the benchmark for visual dubbing solutions, (2) introduce a new research direction to the field, and (3) significantly improve the applicability and efficacy of ADVD solutions for real-world applications.

10.2.3 Automatic Dubbing

When considering the origins of the input dubbing audio used to drive any ADVD solution, we realize that it is still produced using the traditional audio dubbing process. This process is typically performed when professionally dubbing movies, television programs, etc., which entails transcribing the video, translating the transcription to the desired language, and hiring professional dubbing artists to perform several recordings [Chaume 2020b]. Evidently, this process involves a great deal of human intervention, thus, rendering the process to be time-consuming and laborious which in turn, hinders the scalability of these solutions.

To facilitate the wider adoption of visual dubbing solutions, especially by video streaming platforms such as YouTube, Netflix, and Amazon Prime, substantial improvements in the automation of these solutions are essential. To achieve this, the reliance on input dubbing audio should be severed and instead, dubbing should be performed by solely relying on the input video (to be dubbed) and specifying the target language to which the input video should be dubbed, thus, attaining a comprehensive dubbing solution. Implementing such a solution in practice would involve transcribing the video using Automatic Speech Recognition (ASR) [Yu and Deng 2016], translating the transcription to the target language using Neural Machine Translation (NMT) [Stahlberg 2020], and utilizing Text-To-Speech (TTS) technology [Wang *et al.* 2017] to generate the dubbing audio. Some solutions [KR *et al.* 2019] go even further

in enhancing the viewing experience by replicating the speaker's voice, providing a more authentic impression that the speaker is delivering the dubbing audio.

10.3 Conclusion

Video-based content has proven to be one of the most effective ways of conveying information to a broad audience in an engaging manner, surpassing other forms of media such as images, audio, and text. This effectiveness is largely due to their bi-modal nature, which simultaneously stimulates the viewer's auditory and visual systems. In recent years, the production and consumption has grown exponentially following the advent of video-based platforms such as YouTube, Netflix, and TikTok amongst many others. Despite this surge, it is unfortunate that a video may only be fully comprehended by the audience that is familiar with the spoken language, thus, forming a language barrier. Due to the significance of addressing this issue, subtitling and audio-dubbing have emerged as the two most widely-adopted solutions for addressing this problem in recent decades, however, each is notorious for conducing an unpleasant viewing experience in its own regard.

Following an extensive survey of the literature, we identified audio-driven visual dubbing (ADVD) as the most promising path to pursue which accepts a talking-face video, along with dubbing audio, as inputs and produces a dubbed result in which the speaker's mouth movements are manipulated to appear as if the speaker is uttering the dubbing content. Despite the existence of numerous ADVD solutions, these solutions primarily focus on optimizing the visual quality and lip-sync accuracy achieved with little to no regard for expanding the capabilities and applicability of these solutions. We argue that the true potential of ADVD solutions may only be realized and subject to widespread adoption for real-world applications once these aspects are improved significantly. This naturally led to the improvement of ADVD solutions becoming the central focus of this research.

Since we believed that there was still much improvement to be made with regard to the visual quality and lip-sync accuracy achieved, we began by attempting to optimize these metrics. This was achieved by employing a deep-residual U-Net generator, along with a pre-trained lip-sync discriminator composed of R(2+1)D spatiotemporal blocks. Our results show that the presented lip-sync discriminator achieves a 91% off-sync detection accuracy on the LRS2 dataset compared to the 81% achieved by Wav2Lip. Furthermore, quantitative and qualitative analyses, as well as a human subjective study, revealed that the presented solution achieves superior visual quality whilst achieving high measures of lip-sync accuracy as well.

Upon extensively analyzing several ADVD solutions, it was discovered that the majority of solutions were trained with an input resolution no larger than 120×120, despite real-world videos typically being of much higher resolution. The consequence of this vast discrepancy in resolution becomes evident upon reviewing the results achieved by these solutions which tend to exhibit an unpleasant box effect in the speaker's mouth region due to the reduced visual quality achieved. In response, we first developed a comprehensive data-cleaning pipeline to transform the AVSpeech dataset into a form that is suitable for addressing the ADVD problem. By utilizing the pre-processed dataset and amending the solution to ingest inputs with size 192×192, we have shown to increase the SSIM by 9.88%, the CPBD by 98 and decrease the FVD by 70% compared to the next best solution i.e., Wav2Lip.

The second way in which we attempted to extend the capabilities of ADVD solutions followed from the observation that the majority of solutions achieve satisfactory results when dubbing frontal and near-frontal faces, however, the performance of these solutions deteriorates rapidly as the head becomes increasingly non-frontal along the yaw-axis. Our analysis revealed that the deterioration in performance follows as a consequence of the nature of the datasets typically used when addressing the ADVD problem which predominantly contains frontal and near-frontal faces whereas non-frontal faces are under-represented. Since a suitable dataset containing a vast diversity of head poses was not available, we resorted to performing a pose augmentation which synthesizes talking-face videos at arbitrary head poses whilst leveraging the abundance of frontal and near-frontal footage available. Our results revealed that training the solution on the pose augmented data significantly reduced the rate of deterioration in performance compared to other state-of-the-art solutions as the head becomes increasingly non-frontal.

We deem our research efforts towards improving the capabilities and applicability of ADVD solutions to be a success. This verdict is based on the significantly improved visual quality and lip-sync accuracy achieved while increasing the resolution of the training data employed, as well as improving the robustness when dubbing non-frontal faces. In addition, we strongly believe that the data-cleaning pipeline introduced will catalyze the rate of advancements made in fields previously hindered by the lack of high-quality audio-visual data. Lastly, we believe that our contributions will inspire future works to also strive toward expanding the capabilities and applicability of ADVD solutions which will ultimately lead to the widespread adoption of ADVD solutions for real-world applications.

Appendix A

Appendix

A.1 Audio Hyperparameters

We use the same audio hyperparameters for our lip-sync discriminator and generator networks. Our selection of audio hyperparameters are primarily inspired by state-of-the-art speech-based solutions *DeepVoice3* [Ping *et al.* 2017] and *Tacotron2* [Shen *et al.* 2018].

Hyperparameter	Value	Description
`n_fft`	800	Length of FFT window.
`hop_size`	200	Length of hop between STFT windows.
`num_mels`	80	Number of mel bands to generate.
`sample_rate`	16000	Sampling rate.
`max_abs_value`	4	Maximum absolute value of normalized mel-spectogram.
`preemphasis`	0.97	Pre-emphasis coefficient.
`min_level_db`	-100	Minimum dB of normalized mel-spectrogram.
`ref_level_db`	20	Reference level dB when computing the mel-spectrogram.
`fmin`	55	Minimum frequency when computing the mel-spectrogram.
`fmax`	7600	Maximum frequency when computing the mel-spectrogram.

Table A.1: Audio Hyperparameters.

A.2 Structural Similarity Index Measure (SSIM)

The SSIM metric is computed at a patch-level of an image. The measure between 2 patches x and y of common size $N \times N$ (typically 8×8) is computed as:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{A.1}$$

where :

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{A.2}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{A.3}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{A.4}$$

$$\therefore SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{A.5}$$

where:

- $l(x,y)$: luminance measure
- $c(x,y)$: contrast measure
- $s(x,y)$: structural measure
- $\alpha = \beta = \gamma = 1$, importance weightings
- μ_x: pixel sample mean of x
- μ_y: pixel sample mean of y
- σ_x^2: variance of x
- σ_y^2: variance of y
- σ_{xy}: covariance of x and y
- $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2, C_3 = C_2/2$, 3 variables to stabilize the division with weak denominator
- L: dynamic range of the pixel-values
- $k_1 = 0.01$, $k_2 = 0.03$

A.3 Pose-Invariant Quantitative Results

	Metric	[-90, -80)	[-80, -70)	[-70, -60)	[-60, -50)	[-50, -40)	[-40, -30)
AVTG	SSIM (↑)	0.24578	0.24688	0.24569	0.25618	0.26064	0.27249
	CPBD (↑)	0.02098	0.02302	0.02356	0.02379	0.02449	0.02428
	FVD (↓)	150.61944	82.44115	75.44794	69.74833	65.74285	64.49773
	LS-C (↑)	2.38074	2.42116	2.45155	2.46923	2.48012	2.50036
	LS-D (↓)	9.47125	9.41330	9.35230	9.28601	9.14849	9.02127
LipGAN	SSIM	0.82026	0.82643	0.83572	0.84258	0.84619	0.85521
	CPBD	0.11861	0.11964	0.12126	0.12282	0.12218	0.13072
	FVD	48.45671	44.75434	38.93439	34.48203	28.43525	24.98461
	LS-C	3.02189	3.46238	3.62019	3.78162	4.03983	4.28772
	LS-D	9.27089	9.09189	9.01003	8.82803	8.67343	8.50248
Wav2Lip	SSIM	0.84293	0.85019	0.85420	0.86832	0.87128	0.88792
	CPBD	0.12832	0.12983	0.13104	0.13482	0.13802	0.14025
	FVD	37.25835	20.39899	18.15866	17.79283	17.3964	17.33989
	LS-C	**6.25984**	**6.34817**	**6.46195**	**6.54225**	**6.62497**	**6.64575**
	LS-D	**7.48223**	**7.43013**	**7.35102**	**7.29593**	**7.24013**	**7.20392**
Ours	SSIM	**0.91719**	**0.92749**	**0.93572**	**0.93918**	**0.94429**	**0.94792**
	CPBD	**0.15802**	**0.15932**	**0.15988**	**0.16082**	**0.16127**	**0.16183**
	FVD	**18.48332**	**17.43912**	**16.92027**	**16.01829**	**15.6412**	**14.47639**
	LS-C	5.97914	6.01827	6.02123	6.02570	6.03582	6.05293
	LS-D	7.72311	7.71839	7.70827	7.70129	7.68191	7.65299

	Metric	[-30, -20)	[-20, -10)	[-10, 0)	[0, +10)	[+10, +20)	[+20, +30)
AVTG	SSIM (↑)	0.31380	0.31084	0.31308	0.29226	0.28104	0.27513
	CPBD (↑)	0.02521	0.02627	0.02725	0.02721	0.02903	0.02867
	FVD (↓)	63.77041	59.69720	58.82586	58.25595	63.18064	66.28374
	LS-C (↑)	2.56155	2.63642	2.70416	2.70539	2.67963	2.63293
	LS-D (↓)	8.89541	8.75109	8.63691	8.86959	9.09298	9.12740
LipGAN	SSIM	0.86813	0.88263	0.89165	0.89800	0.89248	0.88782
	CPBD	0.13593	0.14120	0.14473	0.14952	0.14774	0.14607
	FVD	22.53357	20.97927	19.61976	19.26491	20.10472	21.34802
	LS-C	4.53240	4.61759	4.68933	4.68716	4.52404	4.28716
	LS-D	8.47619	8.40764	8.31801	8.32729	8.35757	8.38854
Wav2Lip	SSIM	0.90957	0.91053	0.91540	0.91838	0.91538	0.89283
	CPBD	0.14339	0.14473	0.14297	0.15191	0.14298	0.13545
	FVD	17.30102	17.27028	17.09626	17.44381	17.99716	18.34802
	LS-C	**6.69436**	**6.81762**	**6.87549**	**6.85283**	**6.78075**	**6.73918**
	LS-D	**7.17014**	**7.15672**	**7.13995**	**7.13736**	**7.14782**	**7.19674**
Ours	SSIM	**0.94829**	**0.95182**	**0.95487**	**0.95829**	**0.95391**	**0.94891**
	CPBD	**0.16247**	**0.16392**	**0.16529**	**0.16927**	**0.16803**	**0.16743**
	FVD	**14.22432**	**14.01810**	**13.96147**	**13.7958**	**13.64102**	**13.97813**
	LS-C	6.08721	6.11027	6.13588	6.15832	6.14928	6.14197
	LS-D	7.64912	7.63875	7.62791	7.61928	7.62049	7.62971

	Metric	[+30, +40)	[+40, +50)	[+50, +60)	[+60, +70)	[+70, +80)	[+80, +90]
AVTG	SSIM (↑)	0.26643	0.26201	0.25947	0.25652	0.25390	0.24903
	CPBD (↑)	0.02742	0.02539	0.02382	0.02121	0.01809	0.01264
	FVD (↓)	73.31340	79.15455	82.10955	87.31340	98.20971	124.13221
	LS-C (↑)	2.58369	2.55303	2.53963	2.49303	2.46723	2.43293
	LS-D (↓)	9.16126	9.20215	9.22126	9.26794	9.30794	9.37959
LipGAN	SSIM	0.88432	0.88219	0.87832	0.87270	0.86528	0.84040
	CPBD	0.14448	0.14021	0.13760	0.13365	0.13002	0.12774
	FVD	24.42342	28.38273	33.19472	39.01826	42.82745	47.92741
	LS-C	3.81379	3.51693	3.12984	2.83160	2.67316	2.23866
	LS-D	8.53824	8.60230	8.78157	8.95576	9.03941	9.12320
Wav2Lip	SSIM	0.89738	0.89827	0.88697	0.88191	0.87948	0.86732
	CPBD	0.12864	0.12627	0.11802	0.11273	0.10848	0.09156
	FVD	21.22540	25.62464	28.72937	32.03463	34.62382	37.64217
	LS-C	**6.66340**	**6.54739**	**6.45901**	**6.34692**	**6.23592**	**6.16239**
	LS-D	**7.23107**	**7.29767**	**7.35925**	**7.43897**	**7.46430**	**7.54897**
Ours	SSIM	**0.94198**	**0.93799**	**0.93491**	**0.92625**	**0.92083**	**0.91928**
	CPBD	**0.16512**	**0.16301**	**0.16210**	**0.16182**	**0.16021**	**0.15918**
	FVD	**14.31843**	**4.91823**	**15.63840**	**16.92039**	**17.02252**	**18.48332**
	LS-C	6.11742	6.08196	6.06018	6.04982	6.01291	5.97914
	LS-D	7.64298	7.65283	7.77891	7.78648	7.70918	7.72311

Table A.2: Quantitative results of our pose-invariant ADVD solution (numbers in bold indicate optimal values for the corresponding head-pose interval).

References

[Abdrakhmanova *et al.* 2021] Madina Abdrakhmanova, Askat Kuzdeuov, Sheikh Jarju, Yerbolat Khassanov, Michael Lewis, and Huseyin Atakan Varol. Speakingfaces: A large-scale multimodal dataset of voice commands with visual and thermal video streams. *Sensors*, 21(10):3465, 2021.

[Afouras *et al.* 2018a] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[Afouras *et al.* 2018b] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[Akbas and Eckstein 2017] Emre Akbas and Miguel P Eckstein. Object detection through search with a foveated visual system. *PLoS computational biology*, 13(10):e1005743, 2017.

[Akhter *et al.* 2022] Naheed Akhter, Mushtaq Ali, Lal Hussain, Mohsin Shah, Toqeer Mahmood, Amjad Ali, and Ala Al-Fuqaha. Diverse pose lip-reading framework. *Applied Sciences*, 12(19):9532, 2022.

[Amezaga and Hajek 2022] Naroa Amezaga and Jeremy Hajek. Availability of voice deepfake technology and its impact for good and evil. In *Proceedings of the 23rd Annual Conference on Information Technology Education*, pages 23–28, 2022.

[Anderson 2022] Martin Anderson. *To Uncover a Deepfake Video Call, Ask the Caller to Turn Sideways*, 2022. Accessed on: 23 November 2022.

[Andriole 2006] Katherine P Andriole. Image acquisition. In *PACS*, pages 189–227. Springer, 2006.

[Anina *et al.* 2015] Iryna Anina, Ziheng Zhou, Guoying Zhao, and Matti Pietikäinen. Ouluvs2: A multi-view audiovisual database for non-rigid mouth motion analysis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–5. IEEE, 2015.

[Appalaraju *et al.* 2020] Srikar Appalaraju, Yi Zhu, Yusheng Xie, and István Fehérvári. Towards good practices in self-supervised representation learning. *arXiv preprint arXiv:2012.00868*, 2020.

[Arandjelovic and Zisserman 2018] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018.

[Arik *et al.* 2018] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. *Advances in neural information processing systems*, 31, 2018.

[Bakhtiarnia *et al.* 2022] Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis. Efficient high-resolution deep learning: A survey. *arXiv preprint arXiv:2207.13050*, 2022.

[Bansal *et al.* 2016] Raghav Bansal, Gaurav Raj, and Tanupriya Choudhury. Blur image detection using laplacian operator and open-cv. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 63–67. IEEE, 2016.

[Bau *et al.* 2019] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019.

[Bear and Harvey 2017] Helen L Bear and Richard Harvey. Phoneme-to-viseme mappings: the good, the bad, and the ugly. *Speech Communication*, 95:40–67, 2017.

[Belin *et al.* 2011] Pascal Belin, Patricia EG Bestelmeyer, Marianne Latinus, and Rebecca Watson. Understanding voice perception. *British Journal of Psychology*, 102(4):711–725, 2011.

[Bengio *et al.* 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[Blanz and Vetter 1999] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[Blanz *et al.* 2005] Volker Blanz, Patrick Grother, P Jonathon Phillips, and Thomas Vetter. Face recognition based on frontal views generated from non-frontal images. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 454–461. IEEE, 2005.

[Booth *et al.* 2018] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2):233–254, 2018.

[Bradley and Terry 1952] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[Bregler *et al.* 1997] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360, 1997.

[Brewster 2021] Thomas Brewster. Cybersecurity editors' pick fraudsters cloned company director's voice in $35 million bank heist, police find. 2021. Accessed on: 07 November 2022.

[Bridle and Brown 1974] John S Bridle and Michael D Brown. An experimental automatic word recognition system. *JSRU report*, 1003(5):33, 1974.

[Brock *et al.* 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[Bulat and Tzimiropoulos 2017a] Adrian Bulat and Georgios Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3706–3714, 2017.

[Bulat and Tzimiropoulos 2017b] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[Cakir 2006] Ismail Cakir. The use of video as an audio-visual material in foreign language teaching classroom. *Turkish Online Journal of Educational Technology-TOJET*, 5(4):67–72, 2006.

[Cao *et al.* 2013] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013.

[Cao *et al.* 2014] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

[Cao *et al.* 2018] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018.

[Carreira and Zisserman 2017] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[Chakravarty and Tuytelaars 2016] Punarjay Chakravarty and Tinne Tuytelaars. Cross-modal supervision for learning active speaker detection in video. In *European Conference on Computer Vision*, pages 285–301. Springer, 2016.

[Chandrasekaran and Mago 2021] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021.

[Chang *et al.* 2019] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9066–9075, 2019.

[Chatfield *et al.* 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[Chaudhari *et al.* 2020] Poonam Chaudhari, Himanshu Agrawal, and Ketan Kotecha. Data augmentation using mg-gan for improved cancer classification on gene expression data. *Soft Computing*, 24(15):11381–11391, 2020.

[Chaume 2013] Frederic Chaume. The turn of audiovisual translation: New audiences and new technologies. *Translation spaces*, 2(1):105–123, 2013.

[Chaume 2018] Frederic Chaume. An overview of audiovisual translation: Four methodological turns in a mature discipline. *Journal of Audiovisual Translation*, 1(1):40–63, 2018.

[Chaume 2020a] Frederic Chaume. *Audiovisual translation: dubbing*. Routledge, 2020.

[Chaume 2020b] Frederic Chaume. Dubbing. In *The Palgrave handbook of audiovisual translation and media accessibility*, pages 103–132. Springer, 2020.

[Chen *et al.* 2019] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7832–7841, 2019.

[Chen *et al.* 2020a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.* 2020b] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[Chen *et al.* 2020c] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[Cheng *et al.* 2020] Shiyang Cheng, Pingchuan Ma, Georgios Tzimiropoulos, Stavros Petridis, Adrian Bulat, Jie Shen, and Maja Pantic. Towards pose-invariant lip-reading. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4357–4361. IEEE, 2020.

[Cheung *et al.* 2008] Kin-Wang Cheung, Jiansheng Chen, and Yiu-Sang Moon. Pose-tolerant non-frontal face recognition using ebgm. In *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6. IEEE, 2008.

[Chiaro 2009] Delia Chiaro. Issues in audiovisual translation. In *The Routledge companion to translation studies*, pages 155–179. Routledge, 2009.

[Chuang *et al.* 2020] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[Chung and Zisserman 2016a] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian conference on computer vision*, pages 87–103. Springer, 2016.

[Chung and Zisserman 2016b] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian conference on computer vision*, pages 251–263. Springer, 2016.

[Chung and Zisserman 2017] Joon Son Chung and AP Zisserman. Lip reading in profile. 2017.

[Chung *et al.* 2017] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017.

[Chung *et al.* 2018] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[Chung *et al.* 2019] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3965–3969. IEEE, 2019.

[Cintas and Remael 2014] Jorge Díaz Cintas and Aline Remael. *Audiovisual translation: subtitling*. Routledge, 2014.

[Clark *et al.* 2019] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[Cooke *et al.* 2006] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu 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.

[Curry 2022] David Curry. *Video Streaming App Revenue and Usage Statistics (2022)*, 2022. Accessed on: 1 December 2022.

[Dahmane *et al.* 2015] Afifa Dahmane, Slimane Larabi, Ioan Marius Bilasco, and Chabane Djeraba. Head pose estimation based on face symmetry analysis. *Signal, Image and Video Processing*, 9(8):1871–1880, 2015.

[De Ruiter 2021] Adrienne De Ruiter. The distinct wrong of deepfakes. *Philosophy & Technology*, 34(4):1311–1332, 2021.

[Demir and Unal 2018] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018.

[Deng *et al.* 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Deng *et al.* 2018] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7093–7102, 2018.

[Deng *et al.* 2020] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5203–5212, 2020.

[Derda 2021] Izabela Derda. "did you know that david beckham speaks nine languages?": Ai-supported production process for enhanced personalization of audio-visual content. *Creative Industries Journal*, pages 1–16, 2021.

[Di Giovanni 2018] Elena Di Giovanni. Dubbing, perception and reception. *Reception studies and audiovisual translation*, pages 159–177, 2018.

[Diakopoulos and Johnson 2021] Nicholas Diakopoulos and Deborah Johnson. Anticipating and addressing the ethical implications of deepfakes in the context of elections. *New Media & Society*, 23(7):2072–2098, 2021.

[Díaz-Cintas 2013] Jorge Díaz-Cintas. Subtitling: Theory, practice and research. In *The Routledge handbook of translation studies*, pages 291–305. Routledge, 2013.

[Ding and Tao 2016] Changxing Ding and Dacheng Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on intelligent systems and technology (TIST)*, 7(3):1–42, 2016.

[Donahue *et al.* 2016] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[Doukas *et al.* 2021] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021.

[Du *et al.* 2021] Wenchao Du, Hu Chen, Hongyu Yang, and Yi Zhang. Disentangled generative adversarial network for low-dose ct. *EURASIP Journal on Advances in Signal Processing*, 2021(1):1–16, 2021.

[Durugkar *et al.* 2016] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[Emmys 2022] Emmys. *Awards & Nominations - Squid Game*, 2022. Accessed on: 1 December 2022.

[Ephrat *et al.* 2018] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[Estellers and Thiran 2012] Virginia Estellers and Jean-Philippe Thiran. Multi-pose lipreading and audio-visual speech recognition. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–23, 2012.

[Falk *et al.* 2010] Tiago H Falk, Chenxi Zheng, and Wai-Yip Chan. A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1766–1774, 2010.

[Fard *et al.* 2021] Azin Shokraei Fard, David C Reutens, and Viktor Vegh. Cnns and gans in mri-based cross-modality medical image estimation. *arXiv preprint arXiv:2106.02198*, 2021.

[Fernandez-Lopez and Sukno 2018] Adriana Fernandez-Lopez and Federico M Sukno. Survey on automatic lip-reading in the era of deep learning. *Image and Vision Computing*, 78:53–72, 2018.

[Finger 2022] Lutz Finger. *Deepfakes - The Danger Of Artificial Intelligence That We Will Learn To Manage Better*, 2022. Accessed on: 24 November 2022.

[Frid-Adar *et al.* 2018] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[Gao *et al.* 2020] Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang, and Yunhong Wang. Cnn-based density estimation and crowd counting: A survey. *arXiv preprint arXiv:2003.12783*, 2020.

[Garcia-Garcia *et al.* 2017] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[Garrido *et al.* 2015] Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Perez, and Christian Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer graphics forum*, volume 34, pages 193–204. Wiley Online Library, 2015.

[Ge *et al.* 2018] Hao Ge, Yin Xia, Xu Chen, Randall Berry, and Ying Wu. Fictitious gan: Training gans with historical models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[Geisler and Banks 1995] Wilson S Geisler and Martin S Banks. Visual performance. *Handbook of optics*, 1:2, 1995.

[Goodfellow *et al.* 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[Goodrow 2017] Cristos Goodrow. *You know what's cool? A billion hours*, 2017. Accessed on: 18 October 2022.

[Gosse and Burkell 2020] Chandell Gosse and Jacquelyn Burkell. Politics and porn: how news media characterizes problems presented by deepfakes. *Critical Studies in Media Communication*, 37(5):497–511, 2020.

[Gross *et al.* 2006] Ralph Gross, Iain Matthews, and Simon Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006.

[Gulrajani *et al.* 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[Guo *et al.* 2008] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE, 2008.

[Guo *et al.* 2018] Jianzhu Guo, Xiangyu Zhu, and Zhen Lei. *3DDFA*. https://github.com/cleardusk/3DDFA, 2018.

[Guo *et al.* 2020] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Gutierrez 2014] Daniel Gutierrez. *Ask a Data Scientist: Data Leakage*, 2014. Accessed On: 23 November 2022.

[Hacohen and Weinshall 2019] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR, 2019.

[Hadsell *et al.* 2006] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

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

[Hassner *et al.* 2015] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4295–4304, 2015.

[He *et al.* 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.* 2020a] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12):5349–5362, 2020.

[He *et al.* 2020b] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[Heath *et al.* 1997] Christian Heath, Paul Luff, and Abigail Sellen. Reconfiguring media space: Supporting collaborative work. *Video-mediated communication*, pages 323–347, 1997.

[Heiss 2004] Christine Heiss. Dubbing multilingual films: A new challenge? *Meta: journal des traducteurs/Meta: Translators' Journal*, 49(1):208–220, 2004.

[Hitaj *et al.* 2017] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[Hoang *et al.* 2018] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. In *International conference on learning representations*, 2018.

[Hochreiter and Schmidhuber 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hu *et al.* 2008] Yuxiao Hu, Zhihong Zeng, Lijun Yin, Xiaozhou Wei, Jilin Tu, and Thomas S Huang. A study of non-frontal-view facial expressions recognition. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[Hu *et al.* 2015] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, 2015.

[Huang *et al.* 2000] Fu Jie Huang, Zhihua Zhou, Hong-Jiang Zhang, and Tsuhan Chen. Pose invariant face recognition. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 245–250. IEEE, 2000.

[Huang *et al.* 2018] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018.

[IDMb 2022] IDMb. *Awards & Nominations - Narcos*, 2022. Accessed on: 1 December 2022.

[Interworldstats 2022] Interworldstats. *INTERNET WORLD USERS BY LANGUAGE*, 2022. Accessed on: 1 December 2022.

[Ioffe and Szegedy 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[Ireton and Posetti 2018] Cherilyn Ireton and Julie Posetti. *Journalism, fake news & disinformation: handbook for journalism education and training*. Unesco Publishing, 2018.

[Isobe *et al.* 2021] Shinnosuke Isobe, Satoshi Tamura, Satoru Hayamizu, Yuuto Gotoh, and Masaki Nose. Multi-angle lipreading using angle classification and angle-specific feature integration. In *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–5. IEEE, 2021.

[Isola *et al.* 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[Ittichaichareon *et al.* 2012] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. Speech recognition using mfcc. In *International conference on computer graphics, simulation and modeling*, volume 9, 2012.

[ITU 1998] Radiocommunication ITU. *Relative timing of sound and vision for broadcasting*, 1998.

[Jackson *et al.* 2017] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Proceedings of the IEEE international conference on computer vision*, pages 1031–1039, 2017.

[Jaiman 2023] Ashish Jaiman. *The danger of deepfakes*, 2023. Accessed on: 03 January 2023.

[Japkowicz and Stephen 2002] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[Jiang *et al.* 2020] Yi Jiang, Jiajie Xu, Baoqing Yang, Jing Xu, and Junwu Zhu. Image inpainting based on generative adversarial networks. *IEEE Access*, 8:22884–22892, 2020.

[Jiang *et al.* 2021a] Bin Jiang, Qiuwen Zhang, Zuhe Li, Qinggang Wu, and Huanlong Zhang. Non-frontal facial expression recognition based on salient facial patches. *EURASIP Journal on Image and Video Processing*, 2021(1):1–19, 2021.

[Jiang *et al.* 2021b] Junjun Jiang, Chenyang Wang, Xianming Liu, and Jiayi Ma. Deep learning-based face super-resolution: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[Jiang *et al.* 2021c] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021.

[Johnson *et al.* 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[Jolicoeur-Martineau 2018] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[Ju *et al.* 2022] Yeong-Joon Ju, Gun-Hee Lee, Jung-Ho Hong, and Seong-Whan Lee. Complete face recovery gan: Unsupervised joint face rotation and de-occlusion from a single-view image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3711–3721, 2022.

[Kadandale *et al.* 2022] Venkatesh S Kadandale, Juan F Montesinos, and Gloria Haro. Vocalist: An audio-visual synchronisation model for lips and voices. *arXiv preprint arXiv:2204.02090*, 2022.

[Karras *et al.* 2017a] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[Karras *et al.* 2017b] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[Karras *et al.* 2019] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[Kempen 2022] Annalise Kempen. Community safety tips "deepfake": Tips on how to separate fact from fiction. *Servamus Community-based Safety and Security Magazine*, 115(10):52–53, 2022.

[Kendall and Smith 1940] Maurice G Kendall and B Babington Smith. On the method of paired comparisons. *Biometrika*, 31(3/4):324–345, 1940.

[Kendja 2021] Avondale Kendja. The dangers of deepfakes. 2021. Accessed on: 06 November 2022.

[Kim *et al.* 2018] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[Kim *et al.* 2019a] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[Kim *et al.* 2019b] Hyeongwoo Kim, Mohamed Elgharib, Michael Zollhöfer, Hans-Peter Seidel, Thabo Beeler, Christian Richardt, and Christian Theobalt. Neural style-preserving visual dubbing. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[Kim *et al.* 2021] You Jin Kim, Hee-Soo Heo, Soyeon Choe, Soo-Whan Chung, Yoohwan Kwon, Bong-Jin Lee, Youngki Kwon, and Joon Son Chung. Look who's talking: Active speaker detection in the wild. *arXiv preprint arXiv:2108.07640*, 2021.

[King 2009a] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[King 2009b] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[Kingma and Ba 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Koolstra *et al.* 2002] Cees M Koolstra, Allerd L Peeters, and Herman Spinhof. The pros and cons of dubbing and subtitling. *European Journal of Communication*, 17(3):325–354, 2002.

[Korbar *et al.* 2018] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 31, 2018.

[Kotar *et al.* 2021] Klemen Kotar, Gabriel Ilharco, Ludwig Schmidt, Kiana Ehsani, and Roozbeh Mottaghi. Contrasting contrastive self-supervised representation learning pipelines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9949–9959, 2021.

[Kottahachchi and Abeysinghe 2022] Buddhika Kottahachchi and Sasakthi Abeysinghe. *Overcoming the language barrier in videos with Aloud*, 2022. Accessed onL 1 December 2022.

[Koumparoulis and Potamianos 2018] Alexandros Koumparoulis and Gerasimos Potamianos. Deep view2view mapping for view-invariant lipreading. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 588–594. IEEE, 2018.

[Kovalenko 2017] Boris Kovalenko. Super resolution with generative adversarial networks. *cs231n. stanford. edu/reports*, 2017.

[KR *et al.* 2019] Prajwal KR, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and CV Jawahar. Towards automatic face-to-face translation. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1428–1436, 2019.

[Krämer and Böhrs 2017] Andreas Krämer and Sandra Böhrs. How do consumers evaluate explainer videos? an empirical study on the effectiveness and efficiency of different explainer video formats. *Journal of Education and Learning*, 6(1):254–266, 2017.

[Kreps 1989] David M Kreps. Nash equilibrium. In *Game Theory*, pages 167–177. Springer, 1989.

[Lapedriza *et al.* 2005] Agata Lapedriza, David Masip, and Jordi Vitria. Are external face features useful for automatic face classification? In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 151–151. IEEE, 2005.

[Ledig *et al.* 2017] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[Lee and Schachter 1980] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.

[Lee *et al.* 2012] Youn Joo Lee, Sung Joo Lee, Kang Ryoung Park, Jaeik Jo, and Jaihie Kim. Single view-based 3d face reconstruction robust to self-occlusion. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–20, 2012.

[Leimkühler and Drettakis 2021] Thomas Leimkühler and George Drettakis. Freestylegan: Free-view editable portrait rendering with the camera manifold. *arXiv preprint arXiv:2109.09378*, 2021.

[Li *et al.* 2018] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. On the limitations of first-order approximation in gan dynamics. In *International Conference on Machine Learning*, pages 3005–3013. PMLR, 2018.

[Li *et al.* 2021] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. 2d or not 2d? adaptive 3d convolution selection for efficient video recognition. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6155–6164, 2021.

[Limov 2020] Brad Limov. Click it, binge it, get hooked: Netflix and the growing us audience for foreign content. *International Journal of Communication*, 14:20, 2020.

[Lin *et al.* 2021] Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. *Advances in neural information processing systems*, 34:9625–9638, 2021.

[Livingstone and Russo 2018] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[Lo *et al.* 2019] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. Mosnet: Deep learning based objective assessment for voice conversion. *arXiv preprint arXiv:1904.08352*, 2019.

[Mailhac 2000] Jean-Pierre Mailhac. Subtitling and dubbing, for better or worse? the english video versions of gazon maudit. In *On translating french literature and film II*, pages 129–154. Brill, 2000.

[Malhi 2022] Mehhma Malhi. *To see no longer means to believe: The harms and benefits of deepfake*, 2022. Accessed on: 12 November 2022.

[Mansourifar *et al.* 2019] Hadi Mansourifar, Lin Chen, and Weidong Shi. Virtual big data for gan based data augmentation. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1478–1487. IEEE, 2019.

[Mao *et al.* 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[Martínez 2004] Xènia Martínez. Film dubbing. *Topics in audiovisual translation*, 56:3, 2004.

[Masi *et al.* 2016] Iacopo Masi, Anh Tuan Tran, Tal Hassner, Jatuporn Toy Leksut, and Gérard Medioni. Do we really need to collect millions of faces for effective face recognition? In *European conference on computer vision*, pages 579–596. Springer, 2016.

[Mattos and Oliveira 2018] Andrea Britto Mattos and Dario Augusto Borges Oliveira. Multi-view mouth renderization for assisting lip-reading. In *Proceedings of the 15th International Web for All Conference*, pages 1–10, 2018.

[Mejías-Climent 2021] Laura Mejías-Climent. *Enhancing Video Game Localization Through Dubbing*. Springer, 2021.

[Menéndez *et al.* 1997] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.

[Mermelstein 1976] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

[Meskys *et al.* 2020] Edvinas Meskys, Julija Kalpokiene, Paulius Jurcys, and Aidas Liaudanskas. Regulating deep fakes: legal and ethical considerations. *Journal of Intellectual Property Law & Practice*, 15(1):24–31, 2020.

[Mittal *et al.* 2012a] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

[Mittal *et al.* 2012b] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[Mogren 2016] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[Molholm *et al.* 2002] Sophie Molholm, Walter Ritter, Micah M Murray, Daniel C Javitt, Charles E Schroeder, and John J Foxe. Multisensory auditory–visual interactions during early sensory processing in humans: a high-density electrical mapping study. *Cognitive brain research*, 14(1):115–128, 2002.

[Mori *et al.* 2012] Masahiro Mori, Karl F MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics & automation magazine*, 19(2):98–100, 2012.

[Murphy-Chutorian and Trivedi 2008] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626, 2008.

[Nagrani *et al.* 2017] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[Nagrani *et al.* 2018] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Learnable pins: Cross-modal embeddings for person identity. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–88, 2018.

[Nakashima *et al.* 2020] Yuta Nakashima, Takaaki Yasui, Leon Nguyen, and Noboru Babaguchi. Speech-driven face reenactment for a video sequence. *ITE Transactions on Media Technology and Applications*, 8(1):60–68, 2020.

[Narvekar and Karam 2011] Niranjan D Narvekar and Lina J Karam. A no-reference image blur metric based on the cumulative probability of blur detection (cpbd). *IEEE Transactions on Image Processing*, 20(9):2678–2683, 2011.

[Nayak *et al.* 2022] Shravan Nayak, Christian Schuler, Debjoy Saha, and Timo Baumann. A deep dive into neural synchrony evaluation for audio-visual translation. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION*, pages 642–647, 2022.

[Neekhara *et al.* 2021] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Expressive neural voice cloning. In *Asian Conference on Machine Learning*, pages 252–267. PMLR, 2021.

[Newell *et al.* 2016] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[Newton and Stanfill 2020] Olivia B Newton and Mel Stanfill. My nsfw video has partial occlusion: deepfakes and the technological production of non-consensual pornography. *Porn Studies*, 7(4):398–414, 2020.

[Ninomiya *et al.* 2015] Hiroshi Ninomiya, Norihide Kitaoka, Satoshi Tamura, Yurie Iribe, and Kazuya Takeda. Integration of deep bottleneck features for audio-visual speech recognition. In *Sixteenth annual conference of the international speech communication association*, 2015.

[Niu *et al.* 2018] Zhong-Han Niu, Lu-Fei Liu, Kai-Jun Zhang, Jian-Feng Dong, Yu-Bin Yang, and Xiao-Jiao Mao. Single image super-resolution via perceptual loss guided by denoising auto-encoder. In *Pacific Rim International Conference on Artificial Intelligence*, pages 126–136. Springer, 2018.

[Odena *et al.* 2016] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[Oord *et al.* 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Pan *et al.* 2020] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020.

[Park *et al.* 2017] Minsu Park, Jaram Park, Young Min Baek, and Michael Macy. Cultural values and cross-cultural video consumption on youtube. *PLoS one*, 12(5):e0177865, 2017.

[Parkhi *et al.* 2015] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[Paszke *et al.* 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Pathak *et al.* 2016] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[Paysan *et al.* 2009] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009.

[Petridis and Pantic 2016] Stavros Petridis and Maja Pantic. Deep complementary bottleneck features for visual speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2304–2308. IEEE, 2016.

[Ping *et al.* 2017] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.

[Prajwal *et al.* 2020] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 484–492, 2020.

[Radford *et al.* 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Reinhard *et al.* 2010] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[Remael 2010] Aline Remael. Audiovisual translation. *Handbook of translation studies*, 1:12–17, 2010.

[Richardson *et al.* 2020] Daniel C Richardson, Nicole K Griffin, Lara Zaki, Auburn Stephenson, Jiachen Yan, Thomas Curry, Richard Noble, John Hogan, Jeremy I Skipper, and Joseph T Devlin. Engagement in video and audio narratives: contrasting self-report and physiological measures. *Scientific Reports*, 10(1):1–8, 2020.

[Richardson *et al.* 2021] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[Riggio 2010] Francesca Riggio. Dubbing vs. subtitling. *Multilingual computing & technology*, 21(7):31, 2010.

[Ronneberger *et al.* 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Saber and Tekalp 1998] Eli Saber and A Murat Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 19(8):669–680, 1998.

[Sabottke and Spieler 2020] Carl F Sabottke and Bradley M Spieler. The effect of image resolution on deep learning in radiography. *Radiology. Artificial intelligence*, 2(1), 2020.

[Sakaridis *et al.* 2018] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018.

[Salimans *et al.* 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[Sayed *et al.* 2018] Nawid Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*, pages 228–243. Springer, 2018.

[Schneiders 2020] Pascal Schneiders. What remains in mind? effectiveness and efficiency of explainers at conveying information. *Media and Communication*, 8(1):218–231, 2020.

[Sengupta *et al.* 2016] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016.

[Sermanet *et al.* 2018] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[Seshadrinathan *et al.* 2009] Kalpana Seshadrinathan, Thrasyvoulos N Pappas, Robert J Safranek, Junqing Chen, Zhou Wang, Hamid R Sheikh, and Alan C Bovik. Image quality assessment. In *The essential guide to image processing*, pages 553–595. Elsevier, 2009.

[Shen *et al.* 2018] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[Shi *et al.* 2016] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[Simonyan and Zisserman 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Skerry-Ryan *et al.* 2018] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018.

[Smith 1995] Alvy Ray Smith. Alpha and the history of digital compositing. *URL: http://www. alvyray. com/Memos/7_alpha. pdf, zuletzt abgerufen am*, 24:2010, 1995.

[Somers 2020] Meredith Somers. Deepfakes, explained. 2020. Accessed on: 21 November 2022.

[Song *et al.* 2022] Linsen Song, Wayne Wu, Chen Qian, Ran He, and Chen Change Loy. Everybody's talkin': Let me talk as you want. *IEEE Transactions on Information Forensics and Security*, 17:585–598, 2022.

[Stahlberg 2020] Felix Stahlberg. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418, 2020.

[Stains 2022] Howard James Stains. *Feline - Mammal Family*, 2022. Accessed on: 1 December 2022.

[Surís *et al.* 2018] Didac Surís, Amanda Duarte, Amaia Salvador, Jordi Torres, and Xavier Giró-i Nieto. Cross-modal embeddings for video and audio retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[Suwajanakorn *et al.* 2017] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[Szegedy *et al.* 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[Tariq *et al.* 2012] Usman Tariq, Jianchao Yang, and Thomas S Huang. Multi-view facial expression recognition analysis with generic sparse coding feature. In *European Conference on Computer Vision*, pages 578–588. Springer, 2012.

[Thies *et al.* 2016] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[Thies *et al.* 2020] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *European conference on computer vision*, pages 716–731. Springer, 2020.

[Thompson *et al.* 2020] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

[Tran *et al.* 2018a] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[Tran *et al.* 2018b] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[Tran *et al.* 2018c] Luan Tran, Xi Yin, and Xiaoming Liu. Representation learning by rotating your faces. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3007–3021, 2018.

[Tran *et al.* 2018d] Ngoc-Trung Tran, Tuan-Anh Bui, and Ngai-Man Cheung. Dist-gan: An improved gan using distance constraints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–385, 2018.

[Tsao and Livingstone 2008] Doris Y Tsao and Margaret S Livingstone. Mechanisms of face perception. *Annual review of neuroscience*, 31:411, 2008.

[Tveit 2009] Jan-Emil Tveit. Dubbing versus subtitling: Old battleground revisited. In *Audio-visual translation*, pages 85–96. Springer, 2009.

[Twaddell 1935] W Freeman Twaddell. On defining the phoneme. *Language*, 11(1):5–62, 1935.

[Unterthiner *et al.* 2018] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[Vaccari and Chadwick 2020] Cristian Vaccari and Andrew Chadwick. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media+ Society*, 6(1):2056305120903408, 2020.

[Van der Laak *et al.* 2021] Jeroen Van der Laak, Geert Litjens, and Francesco Ciompi. Deep learning in histopathology: the path to the clinic. *Nature medicine*, 27(5):775–784, 2021.

[Van Etten 2018] Adam Van Etten. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv preprint arXiv:1805.09512*, 2018.

[Veerasamy and Pieterse 2022] Namosha Veerasamy and Heloise Pieterse. Rising above misinformation and deepfakes. In *International Conference on Cyber Warfare and Security*, volume 17, pages 340–348, 2022.

[W3Techs 2022] W3Techs. *Usage statistics of content languages for websites*, 2022. Accessed on: 2 December 2022.

[Wang and Bovik 2001] Zhou Wang and Alan C Bovik. Embedded foveation image coding. *IEEE Transactions on image processing*, 10(10):1397–1410, 2001.

[Wang and Sung 2007] Jian-Gang Wang and Eric Sung. Em enhancement of 3d head pose estimated by point at infinity. *Image and Vision Computing*, 25(12):1864–1874, 2007.

[Wang *et al.* 2003] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[Wang *et al.* 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Wang *et al.* 2016] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*, 2016.

[Wang *et al.* 2017] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*, 164, 2017.

[Wang *et al.* 2018] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[Wang *et al.* 2019] Chaoyue Wang, Chang Xu, Xin Yao, and Dacheng Tao. Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*, 23(6):921–934, 2019.

[Wang *et al.* 2020] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *ECCV*, 2020.

[Wang *et al.* 2021a] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. *arXiv preprint arXiv:2107.09293*, 2021.

[Wang *et al.* 2021b] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021.

[Wang *et al.* 2021c] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[Wang *et al.* 2021d] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12586–12595, 2021.

[Wang *et al.* 2022] Ganglai Wang, Peng Zhang, Lei Xie, Wei Huang, and Yufei Zha. Attention-based lip audio-visual synthesis for talking face generation in the wild. *arXiv preprint arXiv:2203.03984*, 2022.

[Wen *et al.* 2020] Xin Wen, Miao Wang, Christian Richardt, Ze-Yin Chen, and Shi-Min Hu. Photorealistic audio-driven video portraits. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3457–3466, 2020.

[Weng 2019] Lilian Weng. From gan to wgan. *arXiv preprint arXiv:1904.08994*, 2019.

[Wickstrøm *et al.* 2022] Kristoffer Wickstrøm, Michael Kampffmeyer, Karl Øyvind Mikalsen, and Robert Jenssen. Mixing up contrastive learning: Self-supervised representation learning for time series. *Pattern Recognition Letters*, 155:54–61, 2022.

[WikiPedia 2023] WikiPedia. *Structural similarity*, 2023. Accessed on: 12 January 2023.

[Wilson *et al.* 2010] Elizabeth AH Wilson, Denise C Park, Laura M Curtis, Kenzie A Cameron, Marla L Clayman, Gregory Makoul, Keith Vom Eigen, and Michael S Wolf. Media and memory: the efficacy of video and print materials for promoting patient education about asthma. *Patient education and counseling*, 80(3):393–398, 2010.

[Wojewidka 2020] John Wojewidka. The deepfake threat to face biometrics. *Biometric Technology Today*, 2020(2):5–7, 2020.

[Wong *et al.* 2021] Adrian Wong, Serene Ho, Olusegun Olusanya, Marta Velia Antonini, and David Lyness. The use of social media and online communications in times of pandemic covid-19. *Journal of the Intensive Care Society*, 22(3):255–260, 2021.

[Wu *et al.* 2020a] Qiong Wu, Chunxiao Fan, Yong Li, Yang Li, and Jiahao Hu. A novel perceptual loss function for single image super-resolution. *Multimedia Tools and Applications*, 79(29):21265–21278, 2020.

[Wu *et al.* 2020b] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020.

[Xia *et al.* 2022] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Xie *et al.* 2017] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017.

[Xu *et al.* 2018] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[Xu *et al.* 2019] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2019.

[Xu *et al.* 2021] Haoran Xu, Xinya Li, Kaiyi Zhang, Yanbai He, Haoran Fan, Sijiang Liu, Chuanyan Hao, and Bo Jiang. Sr-inpaint: A general deep learning framework for high resolution image inpainting. *Algorithms*, 14(8):236, 2021.

[Yang *et al.* 2017] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017.

[Yang *et al.* 2020] Yi Yang, Brendan Shillingford, Yannis Assael, Miaosen Wang, Wendi Liu, Yutian Chen, Yu Zhang, Eren Sezener, Luis C Cobo, Misha Denil, et al. Large-scale multilingual audio visual dubbing. *arXiv preprint arXiv:2011.03530*, 2020.

[Yi *et al.* 2017] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[Yin *et al.* 2017] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3990–3999, 2017.

[Yin *et al.* 2020] Yu Yin, Songyao Jiang, Joseph P Robinson, and Yun Fu. Dual-attention gan for large-pose face frontalization. In *2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020)*, pages 249–256. IEEE, 2020.

[Yu and Deng 2016] Dong Yu and Li Deng. *Automatic speech recognition*, volume 1. Springer, 2016.

[Yu *et al.* 2018] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[Yun *et al.* 2020] Jung Un Yun, Byungho Jo, and In Kyu Park. Joint face super-resolution and deblurring using generative adversarial network. *IEEE Access*, 8:159661–159671, 2020.

[Zeng and Olivera-Cintrón 2019] Catherine Zeng and Rafael Olivera-Cintrón. Preparing for the world of a perfect deepfake. *Dostopno na: https://czeng. org/classes/6805/Final. pdf (18. 6. 2020)*, 2019.

[Zhang and Fisher 2019] Jie Zhang and Robert B Fisher. 3d visual passcode: Speech-driven 3d facial dynamics for behaviometrics. *Signal processing*, 160:164–177, 2019.

[Zhang *et al.* 2016] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

[Zhang *et al.* 2017a] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. Faceboxes: A cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2017.

[Zhang *et al.* 2017b] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017.

[Zhang *et al.* 2018] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[Zhang *et al.* 2019] Yuan-Hang Zhang, Jingyun Xiao, Shuang Yang, and Shiguang Shan. Multi-task learning for audio-visual active speaker detection. *The ActivityNet Large-Scale Activity Recognition Challenge*, pages 1–4, 2019.

[Zhang *et al.* 2020] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.

[Zhao *et al.* 2016] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[Zhao *et al.* 2017] Jian Zhao, Lin Xiong, Panasonic Karlekar Jayashree, Jianshu Li, Fang Zhao, Zhecan Wang, Panasonic Sugiri Pranata, Panasonic Shengmei Shen, Shuicheng Yan, and Jiashi Feng. Dual-agent gans for photorealistic and identity preserving profile face synthesis. *Advances in neural information processing systems*, 30, 2017.

[Zheng *et al.* 2010] Wenming Zheng, Hao Tang, Zhouchen Lin, and Thomas S Huang. Emotion recognition from arbitrary view facial images. In *European Conference on Computer Vision*, pages 490–503. Springer, 2010.

[Zheng *et al.* 2015] Wenming Zheng, Hao Tang, and Thomas S Huang. Emotion recognition from non-frontal facial images. *Emotion Recognition: A Pattern Analysis Approach*, pages 183–213, 2015.

[Zhou *et al.* 2020a] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5911–5920, 2020.

[Zhou *et al.* 2020b] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.

[Zhou *et al.* 2021] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4176–4186, 2021.

[Zhu *et al.* 2015] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 787–796, 2015.

[Zhu *et al.* 2016] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

[Zhu *et al.* 2017a] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[Zhu *et al.* 2017b] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017.

[Zhu *et al.* 2019] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019.

[Zolfaghari *et al.* 2021] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1450–1459, 2021.