Learning Representations for Controllable Image Restoration

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Givi MEISHVILI

von GEORGIEN

Leiter der Arbeit:
Prof. Dr. Paolo FAVARO
Institut für Informatik

Learning Representations for Controllable Image Restoration

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Givi MEISHVILI

von GEORGIEN

Leiter der Arbeit:
Prof. Dr. Paolo FAVARO
Institut für Informatik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 31.03.2022

Der Dekan
Prof. Dr. Z. Balogh

Abstract

Learning Representations
for
Controllable Image Restoration

Givi MEISHVILI, Ph.D. in Computer Science

Universität Bern, 2022

Deep Convolutional Neural Networks have sparked a renaissance in all the subfields of computer vision. Tremendous progress has been made in the area of image restoration. The research community has pushed the boundaries of image deblurring, super-resolution, and denoising. However, given a distorted image, most existing methods typically produce a single restored output. The tasks mentioned above are inherently ill-posed, leading to an infinite number of plausible solutions. This thesis focuses on designing image restoration techniques capable of producing multiple restored results and granting users more control over the restoration process. Towards this goal, we demonstrate how one could leverage the power of unsupervised representation learning.

Image restoration is vital when applied to distorted images of human faces due to their social significance. Generative Adversarial Networks enable an unprecedented level of generated facial details combined with smooth latent space. We leverage the power of GANs towards the goal of learning controllable neural face representations. We demonstrate how to learn an inverse mapping from image space to these latent representations, tuning these representations towards a specific task, and finally manipulating latent codes in these spaces. For example, we show how GANs and their inverse mappings enable the restoration and editing of faces in the context of extreme face super-resolution and the generation of novel view sharp videos from a single motion-blurred image of a face.

This thesis also addresses more general blind super-resolution, denoising, and scratch removal problems, where blur kernels and noise levels are unknown. We resort to contrastive representation learning and first learn the latent space of degradations. We demonstrate that the learned representation allows inference of ground-truth degradation parameters and can guide the restoration process. Moreover, it enables control over the amount of deblurring and denoising in the restoration via manipulation of latent degradation features.

Dedicated to the memory of my beloved grandfathers,
***Givi Meishvili*, *Robert Begalishvili*,**
and
Roland Begalishvili

Acknowledgements

Foremost, I would like to acknowledge the role of my **Ph.D. advisor Prof. Dr. Paolo Favaro**. I appreciate the opportunities and guidance he granted me during the last couple of years. I am thankful for his dedication, commitment, and various encouraging conversations. Most importantly, I am very grateful for long discussions during which he inspired, motivated, and challenged me, pushing the boundaries of what I thought I could.

A special thanks goes to **Prof. Dr. Sabine Süsstrunk** and **Prof. Dr. Timo Kehrer** for serving as thesis examiners. I appreciate their valuable feedback.

I would like to acknowledge all the members of the Computer Vision Group (CVG) in Bern: Dragana Esser, Meiguang Jin, Qiyang Hu, Xiaochen Wang, Adrian Wälchli, Adam Bielski, Abdelhak Lemkhenter, Josué Page, Tomoki Watanabe, Riccardo Fantinel, Florence Aellen, Llukman Çerkezi, Aram Davtyan, Alp Eren Sari, Sepehr Sameni, and Viktor Shipitsin. It was a pleasure to spend time together during lunch and coffee breaks. I am pleased by the positive and friendly atmosphere that reigned in our lab.

One side of pursuing a Ph.D. was doing research and working on exciting problems. But apart from that, I want to express my appreciation to several colleagues who have turned into great friends over time. **Dr. Attila Szabó** and **Dr. Mehdi Noroozi** defended their Ph.D.-s a while ago. Nevertheless, we meet frequently. I appreciate their support and attention that went far beyond discussing scientific research. I want to express my special thanks to **Dr. Simon Jenni**, with whom I shared the office for four years. This page is not enough to tell everything. Besides brainstorming and collaborating on different projects, Simon also is a best friend that supported me in any aspect of life during the Ph.D. and still does so. He is the one who always lent his shoulder to me in difficult minutes as well as celebrated every little victory and the happy moments I had.

Thanks also to **Dr. Abdelaziz Djelouah**, **Dr. Christopher Schroers**, **Dr. Jingjing Shen**, and **Dr. Federica Bogo**, with whom I collaborated during internships at Disney Research Zurich and Microsoft Mixed Reality & AI Labs.

Nothing of this would have been imaginable without my family. Its often said that children would never pay off the effort and contribution of their parents. Indeed, I am very grateful to my mother **Inesa Begalishvili** for her immense attention, diligence, and the time she has dedicated to me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is often said that photos capture the memory of an instant in time. Parents like to capture pictures of significant events of their little ones: a birthday party, the first day at school, the first time on a bicycle, and so on. Nowadays, we can capture these moments due to the invention of imaging technology. The first known record describing a camera dates from the 4-th century BC. The Han Chinese philosopher, Mozi, documented the natural optical phenomenon known as "camera obscura." Several centuries after, in 1833, Louise Daguerre managed to figure out the world's first photographic process. The first photographic camera developed for commercial manufacture was built by Alphonse Giroux in 1839. Giroux signed a contract with Daguerre and Isidore Niépce to produce the cameras in France.

The photographic camera was considered a piece of luxury for quite a long time. Fortunately, due to significant technological advances in recent decades, imaging sensors have become so cheap and compact that almost every device currently features a built-in camera. Moreover, some devices are shipped with multiple built-in cameras. These factors led to a wide spread of imaging devices. In conjunction with the development of the internet, this allowed people to capture photos with their smartphones and share them on different social media platforms.

1.1 Image Degradations

Current cameras allow capturing an unprecedented level of detail due to the availability of high-resolution imaging sensors. However, our ever-so-special memories can still be entirely spoiled by different degradations like motion blur, the sensor's noise due to the low-light conditions, etc. Often, the details that matter the most, such as the face, are distorted.

Image blur is caused by the photographer's shaky hands and the subjects, with whom cooperation cannot be continuously established. This problem is even more evident when capturing a picture of the subject while moving with a camera phone. In this case, one might reduce the exposure time of the sensor. However, this leads to insufficient light, usually compensated by increasing the sensor's gain. Unfortunately, increased gain results in large amounts of sensors noise. Motion blur is not the only possible source of the blur. Defocus blur and spherical aberration are yet other causes of a blur. Defocus blur can be avoided by capturing the object of interest in focus. Spherical aberration can be resolved by pre-calibrating the lens of the camera. In this thesis, we focus only on motion blur.

Blur is not the only adversary during the imaging process. Sometimes, details of the beautiful scenery can be missing due to the long distance to the objects of interest in the scene. Long-distance to the object leads to a low spatial resolution of the captured object. Details might be missing even if the object is close to the

camera. In this case, a potential reason for the lack of details can be simply a limited resolution of the camera.

One can face another problem while working with old legacy content such as old photos and movies. In addition to being blurry and low-resolution, this type of content can also suffer from film scratches.

All the distortions above motivated a constant development of image restoration algorithms usually applied in a post-processing stage.

1.2 On The Ambiguities and Controllability of Restoration

The previous section briefly mentioned different image restoration problems and their possible causes. Now we will discuss some of the restoration ambiguities. We will highlight ambiguities presented in the following image restoration tasks: general deblurring, face deblurring, face super-resolution, and blind video restoration. We also briefly mention possible ways to minimize the ambiguities for specific tasks via adding additional methods of controlling and guiding the restoration process.

1.2.1 Deblurring Motion-Blurred Scene

Two Common sources of motion blur are the movement of the camera or the object itself. Photos require a finite exposure to accumulate light from the scene. Thus, objects moving during the exposure generate motion blur in an image. If we discretize the motion occurring during the exposure of the camera sensor, the blurring process can be viewed as averaging sharp instant frames over time. Therefore, the task of deblurring can be defined as the process of recovering one of the sharp instant frames. The number of output frames can be infinitely large as it depends on how fine-grained we discretize the time. The problem is ill-posed since there exist multiple plausible outputs.

Most of the algorithms in this area restore the sharp middle frame of the sequence since it corresponds to the center of mass of the local blur, which can be unambiguously extracted given the blurry input image [1], [2]. However, what if a photographer wanted to capture the state of the scene corresponding to one of the very first or last frames of the sequence? Hence, one might reformulate the deblurring problem and extract a video of all consecutive sharp frames that define the blurry input frame. Such a formulation allows more fine-grained control over the restoration process and the additional possibility of choice for the end-user. Unfortunately, this formulation of the problem leads to additional temporal ambiguities. Averaging over time destroys the temporal ordering of the instant frames. Therefore, even if our algorithm can restore N sharp frames, it's still a challenge to identify the natural order of the frames. The data-driven solution to the reformulated problem and associated ambiguities is covered in Chapter 3.

FIGURE 1.1: **Rotate and deblur motion-blurred faces.** High-level idea of our model that deblurs and rotates motion-blurred faces. Our system takes a blurry input image of the face and residual viewpoints defining how much it should rotate the restored video sequence relative to the input pose of the blurry face.

1.2.2 Deblurring Motion-Blurred Faces

In Section 1.2.1 we covered the general scene deblurring. Now we will focus on domain-specific deblurring of motion-blurred faces. We consider this problem separately due to the high social importance of faces. Following a similar line of thoughts as in Section 1.2.1, we aim to restore a sharp video sequence from a single motion-blurred image of the face.

Imagine a little daughter laughing and turning her head while her father is trying to make a lovely photoshoot. Unfortunately, the daughter's face might be blurry and partially occluded due to the rotation of the head. In the case of teleconferencing, attendees might observe a motion blur due to the face's motion. Also, the interaction is found to be more engaging when the person on the screen looks towards the receiver [3]. However, it is necessary to look directly into the camera to achieve this configuration, but this does not allow one to watch the person on the screen that one talks to. Therefore we consider a problem of recovering a sharp video rendered from an arbitrary viewpoint from a single blurry image of a face. Thus, the viewpoint will be an additional input controlling the person's gaze in a recovered sharp video. However, in addition to the inherent temporal ambiguity discussed in the previous section, our problem is even more ill-posed due to the need to recover occluded parts of the face. Luckily, we can address the ambiguity in Chapter 4 due to the domain-specific nature of the problem. We envision the system presented in Fig. 1.2. To enable the training of such a system, we have collected a synchronized, high-speed, multi-view face video dataset. The faces of 52 participants were captured in a lab setting from 8 fixed viewpoints.

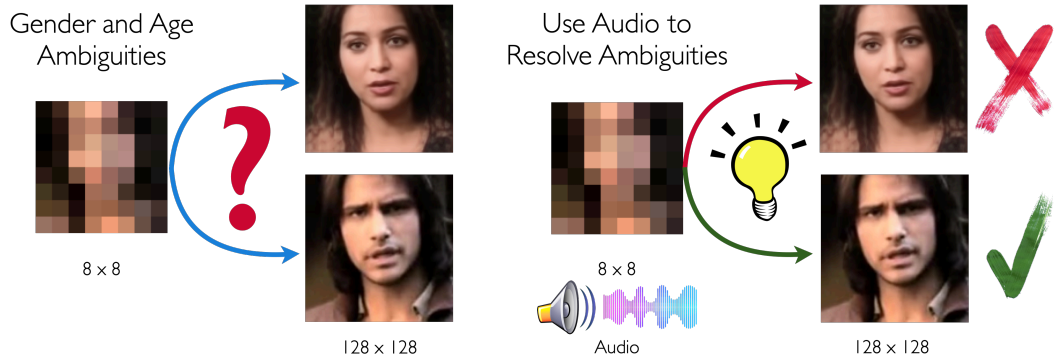


FIGURE 1.2: **Extreme face super-resolution ambiguities.** We demonstrate some of the inherent ambiguities associated with the extreme face super-resolution problem. A 8×8 low-resolution facial image can correspond to multiple identities with a different gender. On the right, we incorporate an audio speech sample and resolve such ambiguity via gender-related information presented in the audio signal.

1.2.3 Extreme Face Super-Resolution

In sections Sections 1.2.1 and 1.2.2, we discussed the motion blur; now, we will consider the challenging problem of tiny face restoration. Extreme face super-resolution refers to the task of recovering high-resolution facial images from their tiny, low-resolution counterparts. Particularly when the scaling factor is $16\times$ or above, the loss of detail can be so dire that important semantic information is lost. Let us consider a small, low-resolution face presented in Fig. 1.2. One can see that identity, gender, or age-related information is missing. The only information still available in such a low-resolution image is perhaps the viewpoint and average colors of the face and the background. Although it is possible to hallucinate numerous plausible high-resolution images from such limited information, missing attributes such as identity, gender, or age might be incorrect. For example, a low-resolution face from Fig. 1.2 can be mapped to two identities with different genders. Therefore, the problem is ill-posed since there exist multiple plausible outputs. We can incorporate alternative sources of information to address the ambiguities caused by the absence of identity, gender, and age attributes. It has been demonstrated that even a short audio speech sample of a human carries information about age and gender [4]. Therefore, we can reformulate the problem of extreme face super-resolution and use an audio signal to guide the face restoration process. In Chapter 5 we show how to train the extreme face super-resolution model that can leverage the identity, gender, and age-related information presented in the audio signal. Moreover, our new formulation allows controlling the super-resolution of a face via evaluating the model with a fixed low-resolution face and multiple different audio tracks.

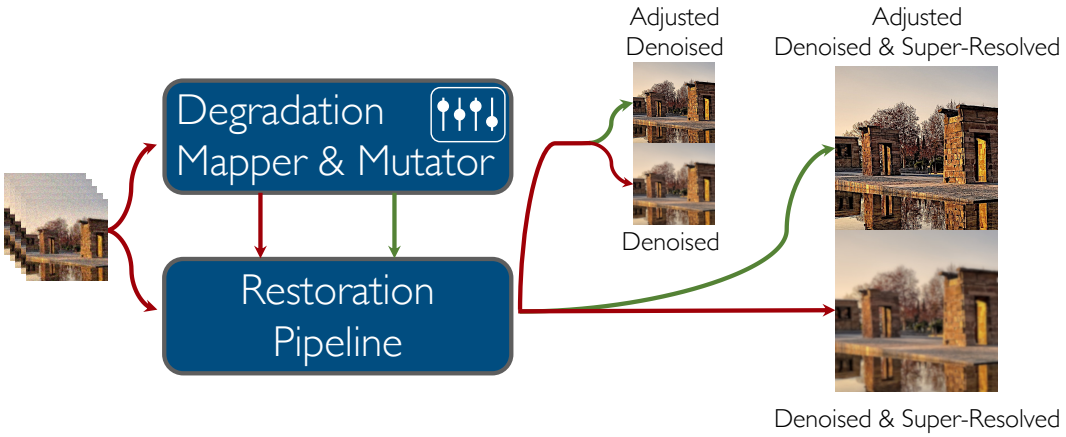


FIGURE 1.3: **Blind video super-resolution, denoising, and film scratch removal.** High-level idea of our model that performs joint video super-resolution, denoising, and film scratch removal. Degradation Mapper and Mutator blocks enable fine-grained control over the restoration process.

1.2.4 Controllable Blind Video Restoration

In Sections 1.2.1 to 1.2.3 we considered general, as well as domain-specific deblurring and super-resolution problems. We assumed that an input image is corrupted by a single source of degradation, either motion blur or extreme downsampling. However, sometimes we have a mixture of different degradations presented in an input image or a video. Input videos are sometimes available in noisy, blurry, and low-resolution format and additionally may contain scratches in the case of old legacy content. Moreover, we are frequently in a blind setting without prior knowledge of how strong each degradation is presented in the input. Therefore, in this section, in addition to blur and downsampling, we aim to simultaneously address different distortions and perform denoising and film scratch removal. We incorporate information from the temporal dimension and thus perform restoration on videos.

Addressing multiple degradations in a blind setting is a challenging task. The output of the restoration model might contain artifacts caused by exaggeration or, on the contrary, disregarding some of the initial degradations presented in the input. This undesirable behavior is caused by the model underestimating or overestimating different degradations to some extent. One particular example of this phenomenon is a combination of blur with high levels of Gaussian noise, where the model mainly focuses on denoising. Consequently, resulting in an over-smoothed solution due to the high noise levels in the input, leading to the inability to detect the initial blur presented in the input video. Thus, to alleviate these problems and control the restoration process, we first need to estimate degradations present in the input video. Once we know the strength of different degradations, we can use them to perform the restoration. However, our output might still contain certain artifacts. In this case, we can adjust the estimated degradation parameters and obtain the adjusted restoration output. In Chapter 6 we learn a latent space of different degradations allowing us to tackle a combination of various distortions with the ability to adjust the restored output if necessary.

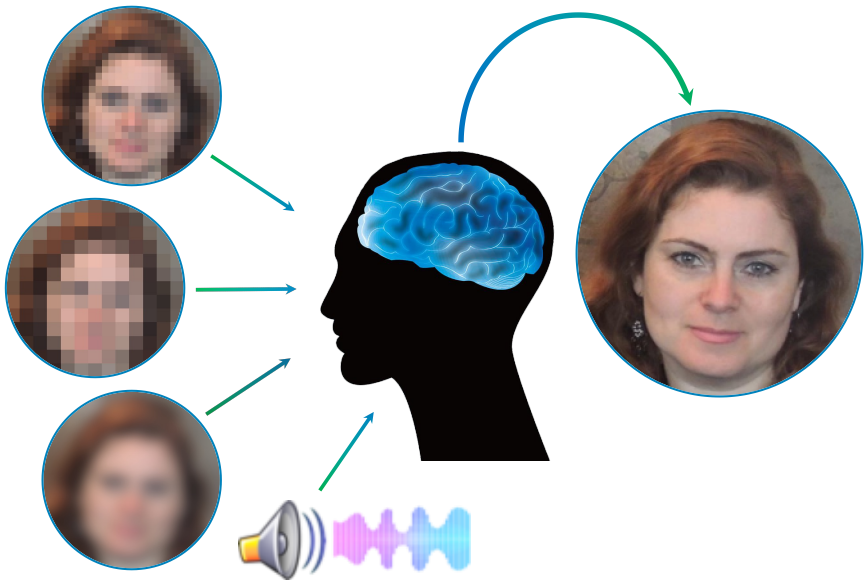


FIGURE 1.4: **Image restoration using brain.** We illustrate how the human brain observes faces with different levels of detail and associates them with the sharp face of identity. We additionally show that the human brain can associate audio speech with the corresponding identity.

1.3 Can Humans Restore?

We discussed possible image distortions arising during the image acquisition process. Now we will examine how a human visual system handles these scenarios before diving into the problem's computational and algorithmic side. Towards this goal, we will raise some questions and consequently try to address them. The main question is, "Are humans able to restore distorted images?" If yes, then how? To find the answer, let's discuss some examples.

Imagine we see a close friend standing far away from us, so far that we can not see the exact details of the face. Can we recognize our friend in this case? Well, sometimes yes, sometimes no, and sometimes we might be thinking of several possible identities. How are we able to identify our friend if he stands far enough? Well, one might say that the human brain has an excellent visual memory. However, we can not search in our memory since we don't see the details of the person's face. One explanation might be that we interacted with the friend in many different circumstances. More specifically, we might have seen him from a variety of different distances. It's fair to consider that our brain has built some latent representation where it tries to map every appearance of our friend. This might potentially explain why we can identify the people we have seen. Yet another example is the ability to hallucinate the invisible part of the human's face.

Like the previous example, let's imagine our friend running towards us. In this case, we can not focus on his face as well as before. However, the closer he is, the more confident we are in his identity. This might be possible due to the similar line of thoughts we used in the previous example.

Now let us consider yet another example. Imagine our friend is calling us and we pick up the phone. In this case, we don't see the person at all. We only hear the voice of the person. Our brain can immediately associate the person's voice with his

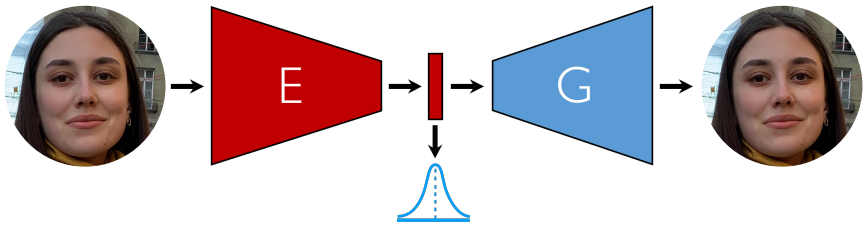


FIGURE 1.5: **Autoencoding of faces.** Facial image is first encoded to some latent space and after decoded back to image space.

face in most cases. In this case, we might think that our brain can map aural and visual modalities into some shared latent representation.

We can not make specific claims about how humans restore distorted or partial visual information. However, we can safely assume that the human brain can restore visual details. We can assume that our brain maps signals of different quality and nature (*e.g.* audio) to some latent representation. More specifically, a brain can be seen as a physical implementation of a function that encodes visual signals into a latent representation of neural activations.

Let us consider the phenomenon of dreaming during the night. It has been shown that our brain can unconsciously imagine and experience certain situations during sleep. These situations are sometimes related to our everyday experiences during the day. However, sometimes they might be random and unreal. We might consider it just as sampling and synthesis from our memory (representation).

We can conclude from these examples that our brain can perform a wide variety of image restoration tasks. Thus a biological solution exists for them. The following section will view these problems from a machine learning perspective.

1.4 Can Machines Restore?

The previous section positively answered whether the human brain could perform image restoration tasks. We hypothesized that the human brain builds latent representation where different signals are mapped. Our goal is to achieve the same computationally. We need to have a machine learning perspective on the problem towards this goal.

We will start with the analogy of the human visual cortex. Humans tend to remember things they see, especially faces, due to their high social importance. Similarly, we can incorporate a machine learning model E to map the face to some high-dimensional latent space. We can also mimic memorizing the person via associating the latent code with the person's face. More specifically, our latent feature can be processed by a machine learning model G and output a corresponding humans face. Intuitively the process of observing a person as well as memorization is shown in Fig. 1.5.

The main goal of the framework presented in Fig. 1.5 is to perform autoencoding of an input image. The natural question arising is "What are the desirable properties for models E and G?". Some of the essential requirements are: *(i)* model G should generate realistic faces; *(ii)* the shared latent space should cover a wide range of different identities; *(iii)* to allow meaningful editing capabilities the latent space should be smooth and disentangled; *(iv)* model E should be able to find the latent code corresponding to an input image. We will describe how one can train models E and G in Chapters 4 and 5.

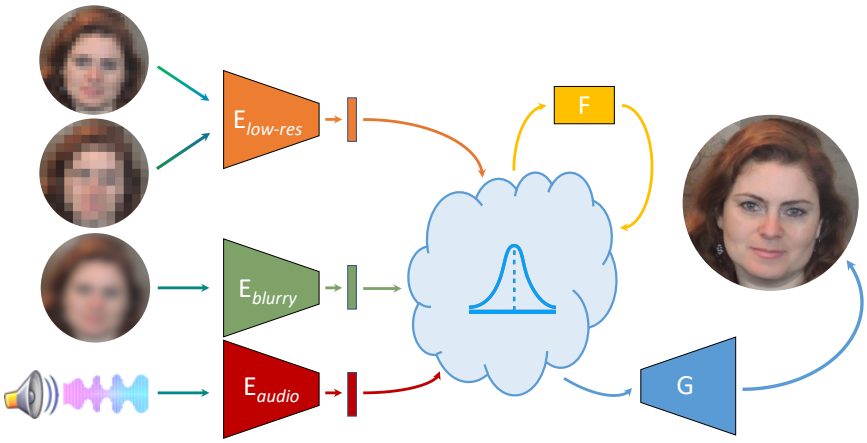


FIGURE 1.6: **Image restoration using our brain.** We illustrate how the human brain observes faces with different levels of detail and associates them with the sharp face of identity. We additionally show that the human brain can associate audio speech with the corresponding identity.

We assume that we are given model G to sample sharp natural facial images from the latent space and model E to output appropriate latent code corresponding to an input image. Now we can define some restoration task-specific components. Fig. 1.6 shows the high-level idea for different types of restoration tasks. We can now define models E_{lowres}, E_{blurry} and E_{audio} for Face Super-Resolution, Deblurring and Audio-to-Image restoration tasks respectively. Each model performs task-specific restoration of input images and maps them to the corresponding high-resolution, sharp latent code. E_{lowres} and E_{blurry} take low-resolution and blurry facial images and map them to our latent space. Similarly, encoder E_{audio} maps the human speech sample to the latent space. We also define a task-specific fusion model F that manipulates latent codes. In the context of the Chapter 5, model F will enhance the output E_{lowres} to recover some identity-related attributes based on the latent audio code provided by model E_{audio}. In the case of the Chapter 4, model F will be used to generate the novel views of deblurred output latent feature provided by model E_{blurry}. Finally, our model G takes the regressed or manipulated latent code and outputs the associated person's face.

1.5 Thesis Contributions

In this thesis, we first introduce the novel task of extracting a video sequence from a single motion-blurred image. Motion-blurred images result from an averaging process, where instant frames are accumulated over time during the exposure of the sensor. Unfortunately, reversing this process is nontrivial since averaging destroys the temporal ordering of the frames. We present a deep learning solution with novel loss functions that enable the gradual recovery of a temporal ordering by sequentially extracting pairs of frames from the middle to the end of the sequence.

This thesis studies techniques to learn image representations that enable editing, control, and guidance of different image restoration methods. Generative Adversarial models (GAN) allow an unprecedented level of generated details combined with smooth latent space. We leverage the power of GANs towards the goal of learning controllable neural face representations. We demonstrate how to learn an

inverse mapping from image space to these latent representations, tuning these representations towards a specific task, and finally manipulating latent codes in these spaces. We demonstrate the efficiency of this methodology on two novel tasks we introduced: extreme face super-resolution using audio and restoration of novel view videos from a single motion-blurred image of the face.

The thesis further addresses a more general class of blind video restoration problems. We designed a system that simultaneously addresses video deblurring, super-resolution, denoising, and scratch removal. Towards this goal, we first built the latent space of degradations via contrastive representation learning. The proposed solution restores videos by conditioning the model with latent codes from learned degradation space. This design allows the fine-grained control over the restoration process and enables modification of restored outputs via manipulating the latent codes from learned representation.

1.5.1 Chapter Outline

Chapter 2: Background. We first discuss autoencoders, GANs, and contrastive representation learning. After we provide a general overview of prior works in image deblurring, super-resolution, and denoising. More discussions of prior works specific to a given chapter are given in separate sections of the remaining chapters.
Chapter 3: Learning to Extract a Video Sequence from a Single Motion-Blurred Image. We introduce the novel task of extracting a sharp video sequence from a single motion-blurred image. Our main contribution is to introduce loss functions invariant to the temporal order. This lets a neural network choose what frame to output among the possible combinations during training. We also address the ill-posedness of deblurring by designing a network with a large receptive field implemented via resampling to achieve higher computational efficiency. Our proposed method can successfully retrieve sharp image sequences from a single motion-blurred image and generalizes well on synthetic and real datasets captured with different cameras.
Chapter 4: Learning to Deblur and Rotate Motion-Blurred Faces. We introduce the novel task of extracting a sharp video sequence of the face from an arbitrary viewpoint given a single motion-blurred image of the face. The proposed method handles the complexity of face blur by implicitly learning the geometry and motion of faces. We train a neural network to reconstruct a 3D video representation from a single image and the corresponding face gaze. We then provide a camera viewpoint relative to the estimated gaze and the blurry image as input to an encoder-decoder network to generate a video of sharp frames with a novel camera viewpoint. We demonstrate our approach on test subjects of our multi-view dataset and VIDTIMIT.
Chapter 5: Learning to Have an Ear for Face Super-Resolution. We introduce the novel task of super-resolving tiny faces using very low-resolution images and associated audio tracks. Towards this goal, we propose a model and a training procedure to extract information about a person's face from her audio track and combine it with the information extracted from her low-resolution image, which relates more to the pose and colors of the face. We demonstrate that the combination of these two inputs yields high-resolution images that better capture the correct attributes of the face. In particular, we show experimentally that audio can assist in recovering attributes such as gender, age, and identity and thus improve the correctness of the image reconstruction process. Our procedure does not make use of human annotation and thus can be easily trained with existing video datasets. Moreover, we show that our model builds a factorized representation of images and audio as it allows

one to mix low-resolution images and audio from different videos and to generate realistic faces with semantically meaningful combinations.

Chapter 6: Contrastive Learning for Controllable Blind Video Restoration. We address the task of blind image super-resolution, denoising and scratch removal simultenaously. We propose a representation learning pipeline that helps separate content from the degradation by reasoning on pairs of degraded patches, where both content and degradation vary independently and provide hard negative examples. The degradation representation is used as conditioning for a video restoration model that can denoise and upscale to arbitrary resolutions and remove film scratches. Finally, the learned representation can be mutated to fine-tune the restoration results, and both the denoising and deblurring levels can be modified. We demonstrate state-of-the-art results compared to the most recent video super-resolution and denoising methods.

Chapter 7: Conclusions and Future Work.

Chapter 2

Background

This chapter provides an overview of prior works in representation learning, deblurring, super-resolution, and denoising. The common design pattern proposed in this thesis is first to encode degraded input to some latent space, perform restoration and manipulation in the latent space and finally map the result back to the image space. Therefore, we first revisit prior work on representation learning which is the bedrock of different image restoration methods presented in this thesis. Sections 2.1 to 2.3 cover different methods of representation learning. Section 2.4 reviews algorithms addressing different types of deblurring problems. Finally, Sections 2.5 and 2.6 acknowledge prior works addressing the problems of super-resolution and denoising, respectively.

2.1 Autoencoders

The autoencoder is a model for unsupervised representation learning [5] consisting of two parts: an encoder and a decoder. During training, encoder and decoder are typically trained to minimize the mean-squared error between input and output of the model. Therefore, the model learns to reconstruct the training data. Without any additional design choices, this system can learn identity function. This is alleviated by limiting the dimensionality of the encodings denoted as a bottleneck. Bottlenecks encourage the model to focus on the structure of the data to preserve as much information as possible. The simplest autoencoder consists of a single dense layer as the encoder and a single dense layer as the decoder. However, for vision tasks, deep autoencoders with multiple convolutional layers learn better representations [6]. Two well-known extensions of the basic autoencoder model are denoising autoencoder (DAE) [7] and the variational autoencoder (VAE) [8].

In the simplest case of denoising autoencoder, the input image to the encoder is corrupted by adding random noise, and the autoencoder is trained to restore the undistorted input image. However, a significant disadvantage of DAEs is that they do not allow a random sampling of the learned data distribution. Therefore, DAE can not be used as a generative model.

In VAEs, the entries of the hidden state of the model are pushed towards a standard Normal prior via a KL-Divergence loss term resulting in multivariate Gaussian with a diagonal covariance matrix. Consequently, first VAEs are generative models that allow a random sampling of the learned data distribution via decoding samples of a standard Normal; secondly, representations learned through VAEs can capture and disentangle the factors of variation on some datasets [9]. As a result, such models allow image manipulation and editing. Numerous extensions and variations of VAEs have been proposed: a variant with discrete hidden states [10] and an adversarial training approach to enforce the prior on the hidden variables [11]. However,

many previous works on autoencoders suffered from blur presented in reconstructions.

We will often use *autoencoder* to refer to a general encoder-decoder network design pattern typical in image restoration and translation tasks [12]. The goal in these cases is not typically to learn a good representation but rather to achieve some specific image processing.

2.2 Generative Adversarial Learning

Goodfellow *et al.* [13] introduced one of the first generative adversarial network (GAN) models. The original GAN is an unsupervised generative model consisting of a generator network and a discriminator network. The generator is trained to generate samples similar to the training data from random Standard Normal noise. The discriminator is trained to judge how similar they are to the training data. Generator and discriminator are trained in an adversarial game. Ideally, the generator should faithfully model the data distribution at the game's equilibrium. Radford *et al.* [14] introduced a GAN model utilizing convolutional layers. The proposed model enabled training at higher resolution and improved the quality of the samples. The learned representation has some desirable properties. Firstly, interpolations in the latent space result in flawless, reasonable interpolations in image space. Secondly, the representation allows for semantically meaningful arithmetic operations. Therefore, directions in the latent space separate factors of variation to some extent. To leverage this representation, Donahue *et al.* [15] introduced a Bidirectional Generative Adversarial Network (BiGAN) that learns the inverse mapping of the generator. Inverse mapping allows leveraging the learned representation for image editing and meaningful semantical manipulations. Donahue *et al.* [16] showed that learning the inverse mapping also improves training and mode coverage.

Classical image restoration tasks like image super-resolution [17], deblurring [18], and image inpainting [19] have largely benefited from principles of adversarial learning. In these settings, discriminator was used as a learnable loss function quantifying the perceptual dissimilarity to some natural reference distribution of images and thus, enhancing the realism in restored images.

Part of the research community focused on addressing the inherent instability of GANs training. Salimans *et al.* [20] introduced a set of techniques and heuristics, Radford *et al.* [14] proposed improved architectural designs and hyper-parameter settings, [20] suggested using one-sided label smoothing and the injection of Gaussian noise into the layers of the discriminator, Arjovsky *et al.* [21] provided a theoretical analysis of the unstable training and the vanishing gradients phenomena. Some of the works addressed the instability issues by introducing alternative training objectives [21]–[28].

Another line of work specifically focused on the problem of very high-resolution image generation. Karras *et al.* [29] grow both the generator and discriminator progressively starting from a low resolution, they add new layers that model increasingly fine details as training progresses. This speeds the training up and significantly stabilizes it, allowing it to produce images of unprecedented quality. Karras *et al.* [30] proposed an alternative generator architecture that first processes a random noise sample using a multi-layer perceptron and passes the output to every layer of generators architecture. Authors additionally mixed two different random codes during training that led to an automatically learned, unsupervised separation of high-level

attributes (*e.g.*, pose and identity when trained on human faces) and stochastic variation in the generated images (*e.g.*., freckles, hair), and enables intuitive, scale-specific control of the synthesis. Karras *et al.* [31] further improved [30] via additional path length regularizer of the latent space resulting in a more straightforward and easier inversion of the generator.

2.3 Contrastive Representation Learning

Sections 2.1 and 2.2 covered representation learning from generative learning perspective. This section focuses on contrastive learning of representations. The main objective of these methods is to learn representations that are discriminative to certain types of data transformations while being invariant to the other types of transformations. Dosovitskiy *et al.* [32] were the very first who exploited this principle and introduced instance discrimination task. The goal is to associate each training example with its unique label while being invariant to different data augmentations defining desired invariances in the learned representation. Wang *et al.* [33] showed that properties of the contrastive loss function and the embedding space significantly influence learned representation. Wu *et al.* [34] introduced a non-parametric formulation of this task leveraging a noise-contrastive estimation, consequently enabling training on larger datasets. Chen *et al.* [35] proposed a simple framework for contrastive learning of visual representations via instance discrimination among large minibatches, requiring neither specialized architectures nor a memory bank. He *et al.* [36] proposed to avoid large minibatches by sampling negatives from a queue of past encoded samples, while [37], [38] diverted sampling negatives explicitly. Caron *et al.* [39] and Wang *et al.* [40] suggested learning a clustering of examples using the contrastive framework. Some works considered contrastive learning on videos to learn representations sensitive to temporal information [41]–[43]. Another line of works performed contrastive learning on multi-modal data: depth[44], optical flow[45], text[46], audio[47], [48].

2.4 Deblurring

This section focuses on various categories of deblurring algorithms, which mainly differ in underlying assumptions about the blur model. Section 2.4.1 covers prior work addressing the most straightforward scenario of a uniform blur model, where blur is assumed to be the same across the image. Section 2.4.2 discusses a line of works tackling a more general non-uniform (space-varying) blur model, where blur might differ across the image. Section 2.4.3 considers video deblurring methods separately. Finally, Section 2.4.4 reviews domain-specific solutions to the face deblurring problem.

2.4.1 Uniform Deblurring

In its most general form, blur removal requires unknowns that considerably outnumber the number of measurements. Therefore, it is typically necessary to use simplifying assumptions. The most common approach assumes that the blur is uniform across the image plane. The uniform model is particularly convenient because it translates into an elegant and simple mathematical model. The problem has received enormous attention from the research community, and it is typically solved in a Bayesian framework. The uniform motion blur model describes a blurry image

as the convolution between a blur kernel and a sharp image. Since only the blurry image is given, the task of recovering both the blur and the sharp image is highly ill-posed. A classic approach to solve motion deblurring is to formulate accurate characterizations of the unknowns. These characterizations are also called priors when a Bayesian formulation is used. The problem is generally cast as an energy minimization with a term measuring how well the convolutional model matches the blurry input image and a term measuring how well the unknowns fit their priors. Image priors usually specify the distribution of gradient magnitudes. This follows from the work of Srivastava *et al.* [49] on natural image statistics. One widespread image prior choice is total variation, initially introduced by Rudin *et al.* [50]. Total variation is used to characterize sharp images by discouraging the presence of gradients and was first exploited for blind deconvolution by You *et al.* [51] and Chan *et al.* [52]. Anwar *et al.* [53] explored the potential of a class-specific image prior. Pan *et al.* [54] incorporated dark channel prior based on the observation that dark pixels from sharp images are not dark when averaged with neighboring high-intensity pixels during the blurring process. Yan *et al.* [55] presented an extremely effective image prior by combining the bright and dark channel priors of Pan *et al.* [54]. Zhou *et al.* [56] proposed a MAP-estimation framework for Blind deblurring that uses high-level edge priors. Priors for the blur have also been used to discourage blur estimates that are too close to a Dirac delta. The prior used for the blur is usually a constant [57], [58], a Gaussian prior [59] or a Laplace prior [60].

Michaeli *et al.* [61] incorporated a patch-based approach and leveraged recurrence of small image patches across different scales of a natural image. Dong *et al.* [62] addressed the influence of outliers on deblurring. Another line of works specifically addressed the case of camera shake blur [63], [64]. Gong *et al.* [65] introduced a gradient activation algorithm for blur kernel estimation. Chakrabarti [66] was the first to introduce the neural approach for uniform deblurring. However, the proposed method is limited to small blurs.

In practice, however, the uniform blur assumption is not satisfied. For example, at occlusions, motion blur may vary sharply. Also, when the camera rotates around its optical axis, the blur at the image corners is much larger than the blur at the image center. Nonetheless, the shift-invariant assumption has led to a better understanding of the general blind deconvolution problem.

2.4.2 Non-Uniform Deblurring

Recently, the general motion deblurring problem has attracted a lot of attention. In general, motion blur might be generated by an object moving relative to the camera in the scene. Its motion blur depends on several factors: its depth, its shape, and its motion trajectory. Due to the Hyun Kim *et al.* [67] proposed an energy model to estimate different motion blurs and their associated pixel-wise weights. Hyun Kim *et al.* [68] used a TV-L1 model to estimate motion flow and a latent sharp image simultaneously. Sun *et al.*[69] trained a convolutional neural network (CNN) for predicting a probability distribution of motion blurs. A sharp image is estimated by using a patch-level image prior. Pan *et al.* [70] developed an efficient algorithm to jointly estimate object segmentation and camera motion, where each layer is deblurred under the guidance of a soft-segmentation. Gong *et al.* [71] estimated a dense motion flow with a fully convolutional neural network and recovered the latent sharp image from the estimated motion flow. Bahat *et al.* [72] recover the unknown blur field by analyzing the spectral content and deblur the image from the estimated blur field with a patch recurrence prior. Pan *et al.* [73] proposed a method to learn data fitting

functions from a large set of motion-blurred images with the associated ground truth blur kernels. Nimisha *et al.* [74] used adversarial training to learn blur-invariant features, which fed to a decoder to produce a deblurred image. Recent work [1], [2], [75] generated synthetic data for dynamic scene motion blur by averaging consecutive frames captured with a high frame rate camera. This dataset could then be used to train a neural network. During training, the center frame of the averaged sharp sequence is used as the ground truth of the corresponding blurry frame. Nah *et al.* [2] trained an end-to-end model with a multi-scale convolutional neural network to restore the latent image directly.

2.4.3 Video Deblurring

Several methods consider the task of restoring a sharp sequence from a blurry video sequence. One big advantage of such methods is the possibility to leverage temporal information from distorted video frames. Zhang *et al.* [76] proposed a method that jointly estimates the motion between consecutive frames as well as blur within each frame. Sellent *et al.* [77] instead exploited a stereo video sequence. Wieschollek *et al.* [78] introduced a recurrent network architecture to deblur images by taking temporal information into account. Hyun Kim *et al.* [75] also exploits a spatio-temporal recurrent network while achieving real-time performance. Kim *et al.* [79] proposed a method for simultaneously removing general blurs and estimating optical flow from a video sequence. Ren *et al.* [80] exploited semantic segmentation of each blurry frame to understand the scene contents and used different motion models for image regions to guide the optical flow estimation. Su *et al.* [81] proposed a CNN that deblurs videos by incorporating information accumulated across frames. Pan *et al.* [82] proposed a framework to estimate the scene flow and deblur the image jointly. Park *et al.* [83] developed a method for the joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence.

2.4.4 Face Deblurring

Faces play an essential role due to their societal importance. Deblurring motion-blurred image of a face is particularly difficult due to several factors. One issue is that the motion blur generated by a moving face depends on the 3D surface of the face as well as its 3D motion. A second issue is that the moving surface causes several self-occlusions, which break the usual blur models. As an example for the last point, if a head rotates using the neck as a rotation axis, the image of the initial pose does not contain the texture of the image of the final pose and vice versa. More specifically, the area around the nose will be the combination of partial occlusion and disocclusion processes. The classic approach of [84] addressed face deblurring by leveraging facial structures from an exemplar dataset. They first collected an exemplar dataset of face images and extracted important structures from exemplars to express the structural information. Fortunately, with the rise of modern learning-based approaches, several works designed specialized neural network architectures to target face deblurring. Chrysos *et al.* [85], [86] performed face alignment to the input of the network and introduced a two-stage architecture where the first stage restores low-frequency and the second stage restores high-frequency content. Jin *et al.* [87] designed computationally efficient architecture that exploits a very large receptive field.

Some methods incorporate additional information in the form of semantic label maps [88], [89] or 3D priors from a 3DMM [90]. Lu *et al.* [91] disentangled image

content and blur and exploited cycle-consistency to learn deblurring in the unsupervised, *i.e.*, unpaired setting. Face deblurring has also been combined with super-resolution by restoring high-resolution facial images from blurry low-resolution images [92], [93].

2.5 Super-Resolution

In this section, we discuss image super-resolution. Image super-resolution refers to the problem of reconstructing high-resolution images from their low-resolution counterparts. In Section 2.5.1 we first consider methods that impose specific assumptions about degradations presented in the input. These methods assume a known blur kernel and a noise level. In Section 2.5.2 we acknowledge methods addressing the more challenging, blind super-resolution problem where blur kernel, as well as the noise level, is unknown. Finally, Section 2.5.3 reviews methods designed for more restricted, class-specific face super-resolution problem.

2.5.1 General Super-Resolution

Singe Image Super-Resolution (SISR) is a very active research area, which largely benefitted from the latest developments in deep learning (see, *e.g.*, [94]–[102]). An important part of super-resolution research works has focused on improving task-specific CNN architectures and components (see *e.g.*, [103]–[117]). A wide set of instances of this problem has been addressed, ranging from arbitrary scale factors [118], to improving the realism of the training set through accurate modeling [119], [120] or through using real zoomed in images [121], [122], to robustness against adversarial attacks [123] and generalization [124], and to modeling multiple degradations [125]–[127]. Finally, [128], [129] focus on the evaluation of the image quality in SISR. Temporal information can also be used in the context of video super-resolution [99], [130]–[137].

Advances in general super-resolution have also mainly been driven by the introduction of task-specific network architectures and components (see *e.g.*, [103]–[111], [138]–[145]). Several works incorporated adversarial training to improve the realism of super-resoled images further and alleviate over smoothed solutions introduced due to common $\mathcal{L}_1, \mathcal{L}_2$ loss functions [96], [146]–[148].

2.5.2 Blind Super-Resolution

Blind Super-Resolution methods assume that information about degradation presented in the image is unknown. Recent methods [126], [149]–[152] addressing blind image super-resolution rely on some form of *test-time* optimization to estimate the blur kernel and predict the corresponding high-resolution output. These two steps can be done separately [149], jointly [126], [150] or require a fine-tuning of the super-resolution model [151], [152]. In the case of blind video super-resolution, Pan *et al.* [153] estimate a blur kernel used in an image deconvolution step. The resulting image is then restored using a neural network and aligned adjacent frames. We can note that this strategy may not be optimal as the restoration neural network cannot directly leverage the blur kernel information. A significant step was made by Wang *et al.* [154], who avoid *test-time optimization* while still conditioning the restoration model on the estimated degradation. This provides a clear advantage. However, the proposed model is limited to images, fixed scaling factors, and the learned representation cannot be interpreted or manipulated.

2.5.3 Face Super-Resolution

The face super-resolution problem has been tackled with a wide variety of approaches. For example, Huang *et al.* [155] trained a CNN to regress wavelet coefficients of HR face, and Yu *et al.* [156] introduced a transformative discriminative autoencoder to super-resolve unaligned and noisy LR face images. More in general, recent methods addressed the problem by using additional supervision, for example, in the form of facial landmarks, heatmaps or the identity label, and multi-task learning [157]–[161]. In contrast, by using videos with corresponding audio tracks, our method does not rely on additional human annotation, and thus its training can scale more easily to large datasets. Several face super-resolution methods leverage adversarial training to improve further the realism of restored faces[157], [159], [161], [162].

2.6 Denoising

Similarly to super-resolution, a lot of progress has been made since early works based on neural networks [163]–[165]. We focus here on recent video denoising methods: Yue *et al.* [166] proposed a raw video denoising network (RViDeNet) by exploring the temporal, spatial, and channel correlations of video frames. Tassano *et al.* [167] proposed a video denoising algorithm based on a convolutional neural network model conditioned on the noise level. Maggioni *et al.* [168] introduced a multi-stage algorithm to reduce the complexity while maintaining denoising performance. These methods strongly rely on providing noise level as input. Claus *et al.* [169] addressed the blind problem using a multi-frame neural network architecture to denoise videos and considered varied noise models during training. Although more robust than specialized denoisers, results are not competitive with recent methods leveraging noise parameters at test time.

Chapter 3

Learning to Extract a Video Sequence from a Single Motion-Blurred Image

FIGURE 3.1: **Multiple frames extracted from a single motion blurred image.** On the left column we show the input image and two enlarged details with different motion blur. On the columns to the right we show the estimated 7 frames and corresponding enlargements.

It is often said that photos capture a memory, an instant in time. Technically, however, this is not strictly true. Photos require a finite exposure to accumulate light from the scene. Thus, objects moving during the exposure generate motion blur in a photo.

Motion blur is an image degradation that makes visual content less interpretable and is often seen as a nuisance. However, motion blur also combines information about both texture and motion of the objects in a single blurry image. Hence, recovering texture and motion from motion-blurred images can be used to understand the dynamics of a scene (*e.g.*, in entertainment with sports or surveillance when monitoring the traffic). The task of recovering a blur kernel and a sharp image, whose convolution gives rise to a given blurry image, is called *motion deblurring* or *blind deconvolution*. Unfortunately, this formulation of the task is accurate only for some special cases of motion blur. In particular, it holds in the instances where blur is the same across an image (the so-called shift-invariant blur [170]) or when blur can be modeled as a linear combination of a basis of shift fields (*e.g.*, in the case of camera shake [171]). However, in the case of multiple moving objects, also called *dynamic blur* [2], a blurry image is no longer some convolution of a blur pattern with a single sharp image. In this case, a blurry image is the averaging over time of instant frames, where multiple objects move independently and cause occlusions.

In this chapter, we introduce blind deconvolution with dynamic blur as the task of recovering a sequence of sharp frames from a single blurry image. As illustrated in Fig. 3.1, given a single motion-blurred image (left column), we aim at recovering a sequence of 7 frames each depicting some instantaneous motion of the objects

in the scene. To the best of our knowledge, this is the first time this problem has been posed and addressed. The two main challenges in solving this task are: 1) blur removal is an ill-posed problem, and 2) averaging over time destroys the temporal ordering of the instant frames. We use a deep learning approach and train a convolutional neural network with a large receptive field to handle the ill-posedness of deblurring. A large receptive field could be achieved by using large convolutional filters. However, such filters would have a detrimental impact on the memory requirements and the computational cost of the network. We avoid these issues by using a re-sampling layer (see Sec. 3.5). Handling the loss of the temporal ordering is instead a less well-studied problem in the literature. To make matters worse, this ordering ambiguity extends to the motion of each object in the scene, thus leading to a combinatorial explosion of valid solutions. One possible exception to this scenario is the estimation of the frame in the middle of the sequence. In most motion-blurred images, the middle frame corresponds to the center of mass of the local blur, which can be unambiguously identified given the blurry input image [1], [2]. However, as shown in the Experiments section, the other frames do not enjoy uniqueness. We find that training a neural network by defining a loss on a specific frame of the sequence, other than the middle one, yields very poor results (see Sec. 3.6). We thus analyze temporal ambiguities in Sec. 3.3 and present a novel deep learning method that sequentially extracts instant frames. Our main contribution is to train neural networks via loss functions that are invariant to the temporal ordering of the frames. These loss functions use the average of two frames and the absolute value of their difference as targets. This allows each network to choose which frames to output during training. Moreover, to make the network outputs more realistic and sharp, we use adversarial training [13]. In the Experiments section, we demonstrate that our trained networks can successfully extract videos from both synthetic and real motion-blurred images. In addition to providing accurate motion information about objects in the scene, we plan to use our method for video editing and temporal superresolution of videos. By exploiting the information embedded in motion blur, our approach can interpolate subsequent frames with high accuracy.

3.1 Background

In Section 2.4 we mentioned some of the prior works about uniform motion deblurring, non-uniform motion deblurring, and video deblurring. However, none of these approaches solves the task of extracting a video sequence from *a single* motion-blurred image. In the following sections, we first illustrate the main challenges of our problem, then we introduce our novel loss functions and show how they address these challenges. The network design is presented in Sec. 3.5 and tested on synthetic and real datasets in the Experiments section.

3.2 From Video to Image

An image $y \in \mathbf{R}^{M \times N}$ captured with exposure τ can be written as

$$y = g\left(\frac{1}{\tau}\int_0^{\tau} \tilde{x}(t)dt\right) = g\left(\frac{1}{T}\sum_{i=0}^{T-1} x[i]\right), \tag{3.1}$$

where g is the camera response function, which relates the irradiance at the image plane to the measured image intensity, and $\tilde{x}(t)$ is the instant image (irradiance) at time t. We discretize the time axis into T segments, and define a sequence of frames

$x[i]$, with $i = 1, \ldots, T$. Each frame $x[i]$ corresponds to the integral of $\tilde{x}(t)$ over a segment, *i.e.*,

$$x[i] = \frac{T}{\tau} \int_{\frac{\tau}{T}i}^{\frac{\tau}{T}(i+1)} \tilde{x}(t)dt. \tag{3.2}$$

Object motion introduces a relative shift (in pixels) of regions between subsequent instant images $\tilde{x}(t)$. Given the maximum shift Δ that we are interested in handling, and by defining negligible blur as a shift of 1 pixel, we can define the maximum number T of time segments by setting $T = \Delta$. This choice only ensures that each frame $x[i]$ will have no motion blur on average. However, motions with acceleration may cause blur larger than 1 pixel in some frames.

The motion-blur model (3.1) thus far described is quite general, as regions can shift in an unconstrained way, and subsequent instant images can introduce or remove texture (occlusions). Indeed, this model can handle the most general case of motion-blur, often called *dynamic blur*. This suggests a new formulation of motion deblurring with dynamic blur:

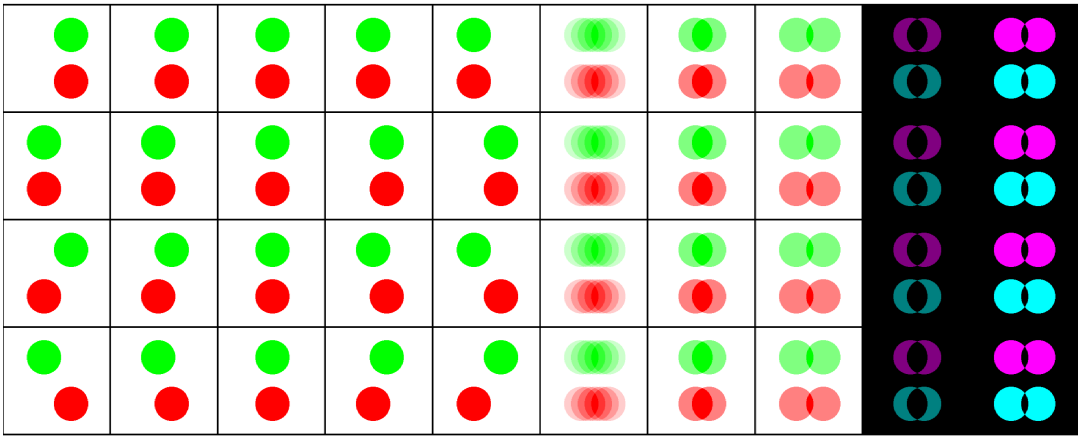
> Given a motion-blurred image y, recover the T frames $x[1], \ldots, x[T]$ satisfying model (3.1).

As mentioned in the Introduction, the task of recovering a sharp image from a blurry one is already known to be highly ill-posed. In our formulation, however, the task is made even more challenging by the loss of frame ordering in the model (3.1). It may be possible to determine the local ordering of subsequent frames by exploiting temporal smoothness. However, there exist several ambiguities that we describe and discuss in the next section. For example, given y, it is impossible to know if the ordering of the original sequence was $x[1], \ldots, x[T]$ or $x[T], \ldots, x[1]$, which corresponds to all objects moving forward or backward in time. Due to the complexity of our task, we adopt a data-driven approach. We build a dataset of blurry images with corresponding ground truth frames by exploiting high frame-rate videos as in recent methods [1], [2], and devise a novel training method with convolutional neural networks (see Sec. 3.4.3).

3.3 Unraveling Time

In our data-driven approach, we define a dataset of *input data* (a blurry image) and *target* (a sequence of frames) pairs and then train a neural network to learn this mapping. However, the averaging of frames in model (3.1) destroys the temporal ordering of the sequence $x[1], \ldots, x[T]$. This makes the recovery of the frames $x[i]$ challenging because it is impossible to define the target uniquely. We might expect that local temporal ambiguities between subsequent frames can be resolved by learning the temporal smoothness (frames are more likely to form a sequence describing smooth motions). However, several other ambiguities still remain. For example, the global motion direction is valid for forward and backward in time. This directional ambiguity applies independently to each moving object in the scene so that all motion direction combinations are valid.

We illustrate these ambiguities in Fig. 3.2 with a toy example. We consider two moving objects: a red and a green ball, both translating along the horizontal axis. The first 5 columns show all 5 frames ($T = 5$) in the averaging model. Because there are 2 objects, there are 4 possible combinations of motion directions (2^n motions with n the number of objects). These are shown in the 4 rows of the figure. Column (f)



(a) (b) (c) (d) (e) (f) (g) (h) (i) (l)

FIGURE 3.2: **Temporal ordering ambiguities.** In this toy example, we show two moving objects: a red and a green ball. Both are translating horizontally. Columns (a)-(e) show five video frames in four scenarios. Each of the four rows shows a plausible motion scenario of the two objects. Column (f) shows the blurry average of the first five columns. These averages are all identical, thus demonstrating that all four sequences are equally valid solutions. Column (g) shows the average of frame (b) and (d). Column (h) shows the average of frame (a) and (e). Column (i) shows the absolute difference of frame (b) and (d). Finally, column (h) shows the absolute difference of frame (a) and (e).

shows that the corresponding average of the frames is the same motion-blurred image in all 4 cases. Therefore, any of the target frames across the 4 rows is a valid one, and it would be unfeasible for the network to learn to predict a specific choice for just one of these 4 cases. Indeed, as we show in the Experiments section, training a network to predict a single frame results in a network that predicts a blurry output that is the average of the possible choices. There is one exception to these ambiguities. The middle frame in an odd-numbered sequence does not change across the 4 cases. This explains why prior methods [1], [2] could successfully train a neural network to predict the middle frame.

To address the temporal ordering ambiguities we introduce novel loss functions. In the next section we explore different options and show how we arrived at our proposed loss function. These cases are also discussed and evaluated in the Experiments section.

3.4 From Image to Video

Our training data has been obtained from a GoPro Hero 5 and features videos at 240 frames per second. To obtain blurry frames at standard real-time video rates (30 frames per second), we thus need to average 8 frames. However, as we have shown in our previous section, we can avoid ambiguities in the estimation of the middle frame by using an odd number of frames, and hence we use $T = 7$. However, our method can generalize to other choices of T. We denote the neural network that predicts the frame $x[i]$ with ϕ_i. Since the middle frame $x[4]$ can be predicted directly

without ambiguities, we train ϕ_4 with the following loss

$$\mathcal{L}_{\text{middle}} = |\phi_4(y) - x[4]|^2 + \mathcal{L}_{\text{perceptual}}(\phi_4(y), x[4]), \tag{3.3}$$

where $\mathcal{L}_{\text{perceptual}}$ is the perceptual loss [172]. For the perceptual loss, we use the *relu2_2* and *relu3_3* layers of vgg16 net [173]. All other losses in the sections below will focus on the other frames.

3.4.1 Globally Ordering-Invariant Loss

A first way to recover all other frames is to use a loss function based on the image formation model (3.1)

$$\mathcal{L}_{\text{model}} = \left|\sum_{i\neq 4} \hat{x}[i] - \sum_{i\neq 4} x[i]\right|_1, \tag{3.4}$$

where we have defined $\hat{x}[i] = \phi_i(y)$. This loss does not suffer from ambiguities and lets the networks decide what frames to output. In practice, however, we find that it is too weak. This loss works well only when a blurry frame is generated by averaging no more than 3 frames. We find experimentally that with more averaging frames, the network does not converge well and may not generate a meaningful sequence.

3.4.2 Pairwise Ordering-Invariant Loss

Inspired by the previous observation, we notice that any pair of symmetric frames (about the middle frame) results in the same average and absolute differences. This choice is motivated by the observations made in the previous section and illustrated in Fig. 3.2. Columns (g) and (i) in Fig. 3.2 show the average and absolute difference respectively of columns (b) and (d). These combinations yield the same target frame regardless of the object's motion direction. Thus, we propose to use a loss made of two components, one based on the sum and the other based on the absolute difference between only two frames. We find experimentally that this scheme imposes a much stronger constraint. Based on these observations, for each pair of symmetric frames (ϕ_i, ϕ_{8-i}), we propose the following loss function

$$\begin{aligned}\mathcal{L}_{\text{pair}} = \sum_{i=1}^{3} &\Big|\,|\hat{x}[i] + \hat{x}[8-i]| - |x[i] + x[8-i]|\,\Big|_1 \\ &+ \Big|\,|\hat{x}[i] - \hat{x}[8-i]| - |x[i] - x[8-i]|\,\Big|_1,\end{aligned} \tag{3.5}$$

where $\hat{x}[i] = \phi_i(y)$ and $\hat{x}[8-i] = \phi_{8-i}(\phi_i(y), y)$ for $i = 1, 2, 3$. Notice that $\phi_{8-i}(\phi_i(y), y)$ takes as inputs both the blurry image y and the output of the other network $\phi_i(y)$. The reason for this additional input is so that the network ϕ_{8-i} can learn to generate a frame different from that of $\phi_i(y)$. Therefore, it needs to "know" what frame the network $\phi_i(y)$ has chosen to generate. Compared with the loss function in eq. (3.4), this loss function is easier to optimize and converges better (see results in the Experiments section). We also find experimentally that we can further boost the performance of our networks by additionally feeding the middle frame prediction to each network. That is, we define $\hat{x}[i] = \phi_i(\phi_4(y), y)$ and $\hat{x}[8-i] = \phi_{8-i}(\phi_4(y), \phi_i(y), y)$ for $i = 1, 2, 3$.

1. Let $\hat{x}[4] = \phi_4(y)$ and minimize

$$\mathcal{L}_{\text{middle}} = |\hat{x}[4] - x[4]|^2 + \mathcal{L}_{\text{perceptual}}(\hat{x}[4], x[4]).$$

2. Let $\hat{x}[3] = \phi_3(\phi_4(y), y)$,
$\hat{x}[5] = \phi_5(\phi_3(y), \phi_4(y), y)$ and minimize

$$\begin{aligned}\mathcal{L}^{3,5}_{\text{pair}} =& \Big| |\hat{x}[3] + \hat{x}[5]| - |x[3] + x[5]| \Big|_1 \\ &+ \Big| |\hat{x}[3] - \hat{x}[5]| - |x[3] - x[5]| \Big|_1 \\ &+ \mathcal{L}^3_{\text{adv}} + \mathcal{L}^5_{\text{adv}}.\end{aligned}$$

3. Let $\hat{x}[i] = \phi_i(\phi_{i+1}(y), \phi_{i+2}(y), y)$, $\hat{x}[8-i] = \phi_{8-i}(\phi_{7-i}(y), \phi_{6-i}(y), y)$, with $i = 1, 2$ and minimize

$$\begin{aligned}\mathcal{L}^{1,2,6,7}_{\text{pair}} =& \Big| |\hat{x}[1] + \hat{x}[6]| - |x[1] + x[6]| \Big|_1 \\ &+ \Big| |\hat{x}[1] - \hat{x}[6]| - |x[1] - x[6]| \Big|_1 \\ &+ \Big| |\hat{x}[2] + \hat{x}[7]| - |x[2] + x[7]| \Big|_1 \\ &+ \Big| |\hat{x}[2] - \hat{x}[7]| - |x[2] - x[7]| \Big|_1 \\ &+ \mathcal{L}^1_{\text{adv}} + \mathcal{L}^2_{\text{adv}} + \mathcal{L}^6_{\text{adv}} + \mathcal{L}^7_{\text{adv}}.\end{aligned}$$

TABLE 3.1: Summary of networks, loss functions and training.

3.4.3 Learning a Temporal Direction

Up to this point, each pair of networks ϕ_i and ϕ_{8-i} operates independently from the other pairs. This is not ideal, as it leaves a binary ambiguity in the temporal ordering of each pair: We do not know if $\phi_i \mapsto x[i]$ and $\phi_{8-i} \mapsto x[8-i]$ or $\phi_i \mapsto x[8-i]$ and $\phi_{8-i} \mapsto x[i]$. Thus, after training, one needs to find a smooth temporal ordering of the outputs of these networks for each new input.

To avoid this additional task, we sequentially train our networks and use the outputs of the previous networks to determine the ordering for each data sample during training. This is needed only for frames away from the central core with $i = 3, 4, 5$. Moreover, once the temporal ordering chosen by the middle core networks is known to the other networks, there is no need to feed other inputs. Hence, we define the non-core networks as $\phi_i(\phi_{i+1}(y), \phi_{i+2}(y), y)$ and $\phi_{8-i}(\phi_{7-i}(y), \phi_{6-i}(y), y)$ for $i = 1, 2$. In practice, we find that ϕ_i and ϕ_{8-i} can share weights for $i = 1, 2$. This opens up the possibility of designing a recurrent network to predict all non-core frames. We also use an adversarial loss $\mathcal{L}_{\text{adv}}$ to enhance the accuracy of the output of each network ϕ. Except for the network that generates the middle frame, all other networks use the adversarial loss during training. We summarize our training losses and procedure in Table 3.1.

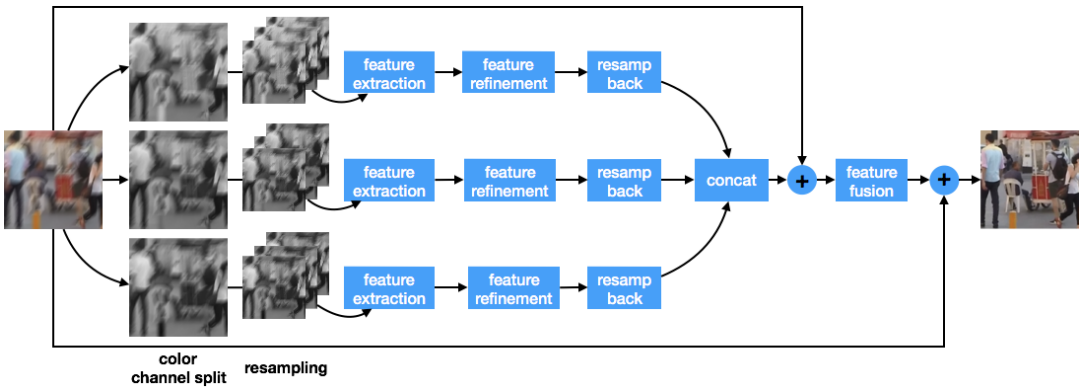


FIGURE 3.3: Middle frame prediction network architecture.

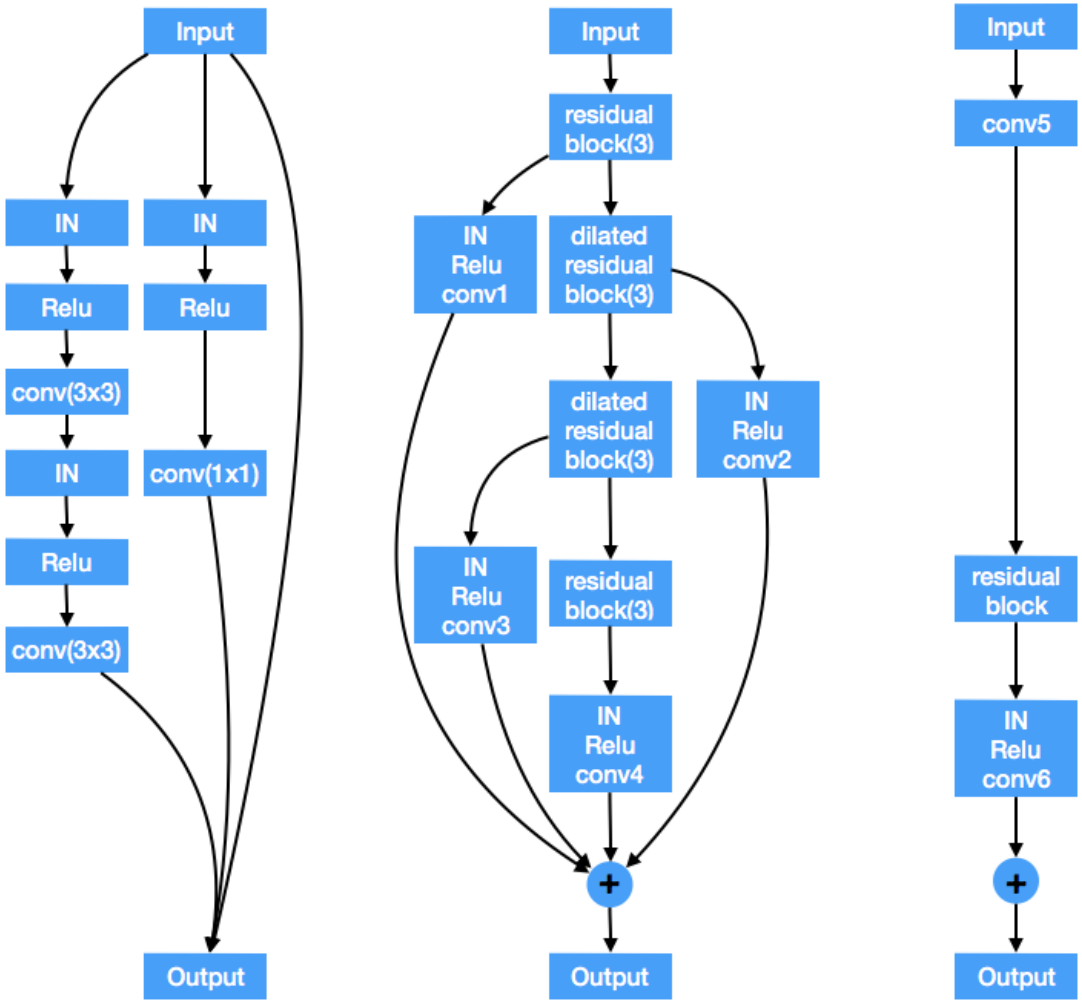


FIGURE 3.4: **Details of our architecture.** Left to right: the residual block, the feature refinement block and the feature fusion block (see Fig. 3.3).

3.5 Implementation Details

Our middle frame prediction network employs a residual learning strategy like many recent image restoration networks [2], [174]. The overall structure of the middle frame prediction network is shown in Fig. 3.3. It consists of feature extraction, feature refinement, and feature fusion. Feature extraction is a convolutional layer (conv0 in Table 3.4), where filters with size 5×5 elements are used. The architectures of the feature refinement and feature fusion blocks are shown in Fig. 3.4.

A blurry image is first split into three color channels. Resampling with factor 4 is applied to each color channel separately. Resampling creates $\texttt{factor}^2$ sub-sampled images. Each sub-sampled image is obtained by sampling the original image one pixel every `factor` pixels (along both axes). Every sub-sampled image differs by the initial sampled pixel on the original input (up to $\texttt{factor}^2$ possible initial positions). Moreover, 16 sub-sampled images are generated for each color channel. We evaluated the different resampling factors for the middle frame estimation and found that $4\times$ resampling gives a better trade-off between accuracy and execution time. Resampling can also be seen as the inverse process of the sub-pixel convolution proposed in [175].

In the feature refinement part, we use 12 residual blocks [176], where each one includes two 3×3 and one 1×1 convolution layers with a pre-activation structure [177]. The architecture of each residual block is shown on the left column of Fig. 3.4. Dilated convolutions are applied to the middle six residual blocks to increase the receptive field further. The feature extraction and refinement parts work on grayscale images, and three color-refined features are generated separately. The feature fusion part works on color images to compensate for misalignments from the three separately-generated color-refined features.

The structure of our proposed middle frame prediction network is also described in Table 3.4. For the non-middle frame prediction networks, similar architectures are also used. The differences are the feature extraction part, where features are extracted from multiple inputs separately and then concatenated, resampling factor, and the number of channels. More specifically, the number of channels (128 instead of 144), the resampling `factor` (5 instead of 4), and the feature extraction layers. For networks with two inputs, *e.g.*, $\phi_i(B, \phi_4(B))$, 64 features are extracted from B and $\phi_4(B)$ respectively, and concatenated.

Training Dataset and Implementation Details. Although there is a GoPro training set available from [2], containing 22 diverse scenes, we captured additional 20 scenes. In training, we downsample the GoPro frames to 45% of their original size (1280×720 pixels) to suppress noise. Blurry frames are generated by averaging 7 consecutive frames randomly cropped of size 320×320. For training, we use about 15K samples. Data augmentation is applied to avoid overfitting by randomly shuffling color channels, rotating images, and adding 1% white Gaussian noise. Networks are implemented using PyTorch, and training is done with 2 GTX 1080 Ti GPUs. The batch size of the middle frame prediction network and other networks are 32 and 24, respectively. Training at each stage takes one day, and the entire network training takes four days.

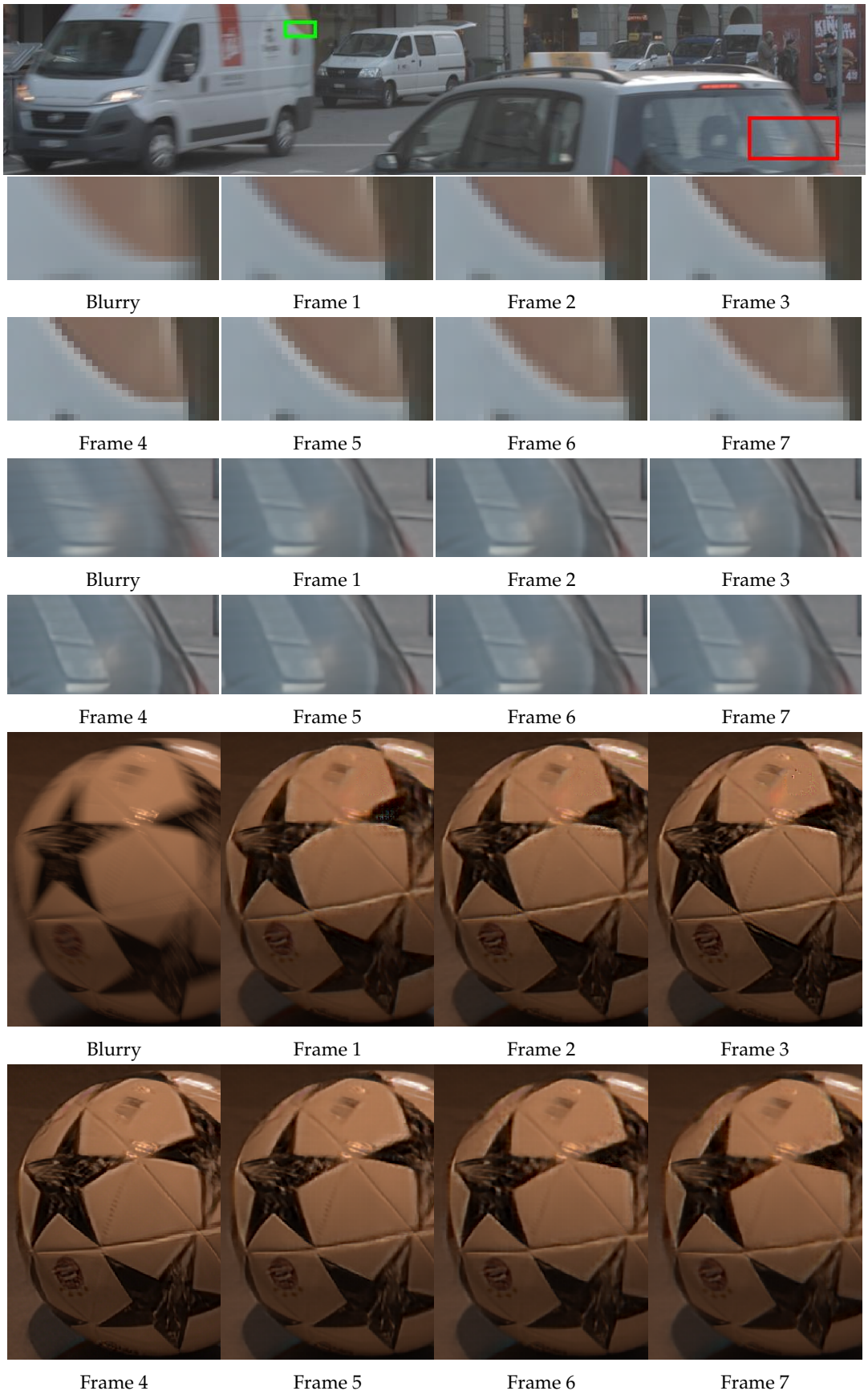


FIGURE 3.5: **Examples with real images.** Top row: an image with multiple moving objects snd static background. Rows 2 and 3: green blurry patch and reconstructed video. Rows 4 and 5: red blurry patch and reconstructed video. Notice that the reconstruction shows both cars moving left to right. This is not the true motion (it would correspond to one vehicle reversing on the street). Last two rows: a rotating ball. The network can correctly reconstruct a video with a complex motion field.

Method	45% [2] (dB)	Our testset (dB)	[2] (dB)
Nah [2]	30.52	28.19	28.48
Middle	32.20	29.02	26.98

TABLE 3.2: Comparison of the middle frame prediction networks.

3.6 Experiments

In this section, we perform a quantitative comparison of the middle frame prediction network with the state-of-the-art method [2]. For non-middle frame predictions, we carry out a qualitative evaluation as there is no existing method predicting a video sequence from a single motion-blurred input. We show some examples of video reconstructions from real motion-blurred images in Fig. 3.5. We validate our design through ablation studies of different loss functions.

3.6.1 Middle Frame Reconstruction

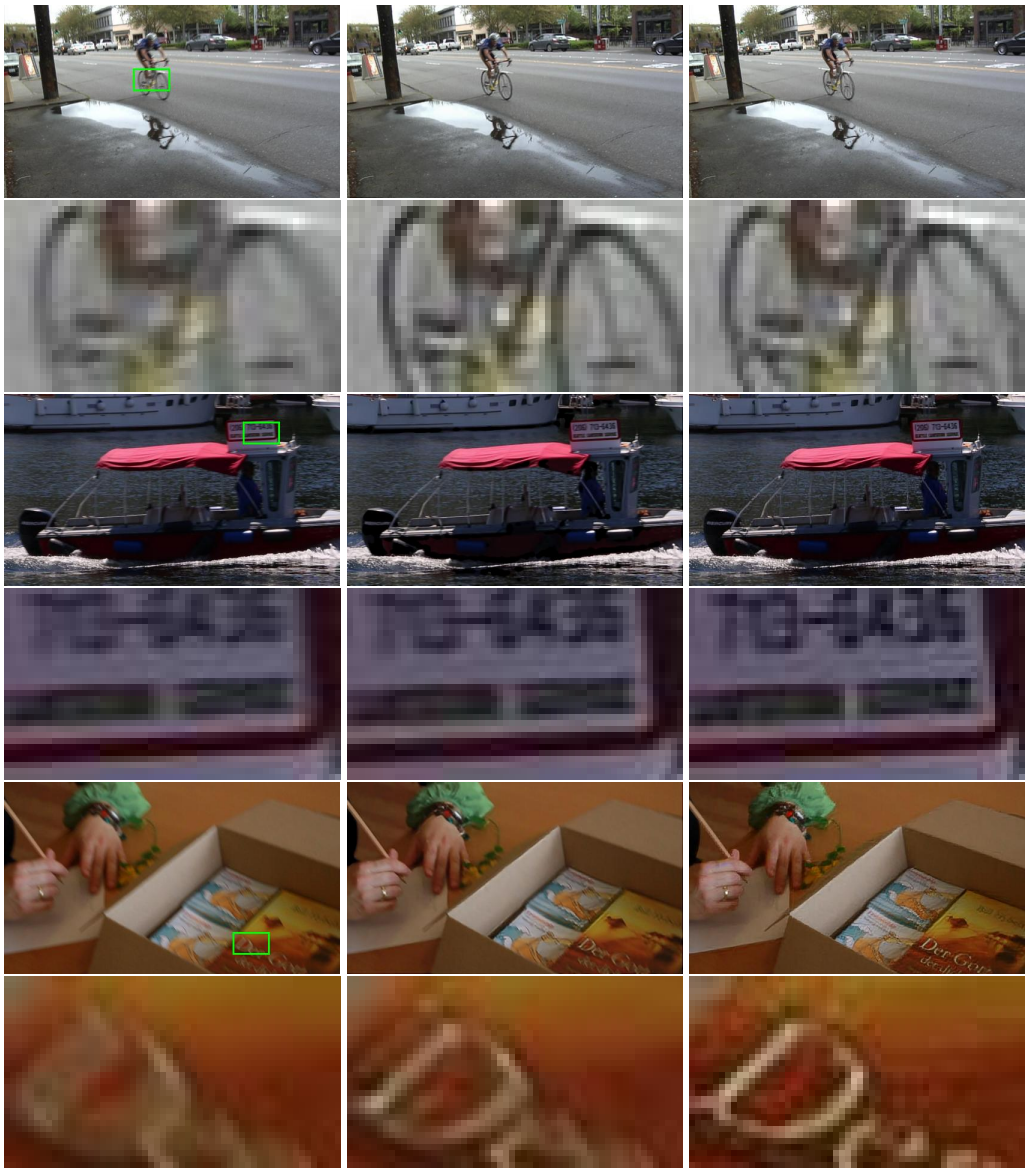
We take Nah's [2] test set, which contains 11 different sequences, and generate 1700 blurry frames by averaging seven consecutive frames. The same process is also applied to our own test set, where 450 blurry images are generated. All blurry images are downsampled to 45% as during training. Table 3.2 shows the quantitative results of Nah's network and our proposed network on two datasets. Our network is consistently performing better on the last two datasets (the first two columns in the table). This is because the motion blur in the data matches the motion blur observed by our network during training. In contrast, Nah's network was trained with much more challenging data, where motion blur could be even larger. Thus, we also evaluate our network on Nah's original 1111 test images for a fairer comparison. These images are averaged by more than 7 frames without any downsampling. In this case, Nah's network is performing better, as our network has not learned to deal with such large motion blur. However, the performance loss is not too significant.

Some visual comparisons on both synthetic and real images are shown in Figs. 3.6 and 3.7, respectively. Fig. 3.7 1 and 3 show two synthetic examples, one with an extremely large blur from Nah's original test images and the other one with a moderate blur from our test set. It can be seen that although our method does not outperform Nah's, it can give better visual results when blur is moderate. Two real examples captured with a DSLR (Nikon D7100) are shown in rows 5 and 7. In practice, we find that if a network is trained with large blurs, it may not remove moderate blur to the same extent as networks trained with small blurs. As we will show later in the Experiments section, the accuracy of the middle frame prediction has a dramatic impact on the reconstruction of the other frames.

In Table 3.3 we show the execution time for three different resolutions and the number of parameters used in Nah's and our networks. It can be seen that our middle frame prediction network is approximately ten times faster than Nah's but has half as many parameters. Additionally, in Fig. 3.6 we also compare to the state of the art video deblurring method [81]. Notice that in [81], they use five consecutive blurry frames to predict the sharp middle frame, whereas we predict the middle frame directly from only one blurry input. Three real results are shown in Fig. 3.6. It can be seen that, although our method suffers from some jpeg artifacts, it gives comparable results.

Method	320P	480P	720P	# params
Nah [2]	2.43	3.52	4.80	12M
Middle	0.24	0.30	0.45	5M
Full	0.61	0.74	1.10	17M

TABLE 3.3: Execution time comparison between the state of the art single image dynamic scene deblurring network [2] and our model on three different resolutions on a Titan X GPU.



Blurry crop Su [81] crop Proposed crop

FIGURE 3.6: **Middle frame prediction comparison.** The first column shows the blurry inputs and cropped regions. The second column corresponds to frame predictions from [81] network. Last column corresponds to frame predictions from our proposed network. Rows 1, 3, and 5 are real images from [178] and [81].

Blurry crop Nah [2] crop Proposed crop

FIGURE 3.7: **Middle frame prediction comparison.** The first column shows the blurry inputs and cropped regions. Second column corresponds to frame predictions from [2] network. Last column corresponds to frame predictions from our proposed network. The first and third rows have been generated synthetically through averaging. The fifth and seventh rows are real images.

3.6.2 Independent Frame Reconstruction

A straightforward way of estimating a sharp sequence is to replicate the training for each frame by minimizing the loss $\mathcal{L}_{\text{indep}} = \sum_{i=1}^{T} |\phi_i(y) - x[i]|^2$, where ϕ_i is a network to predict $x[i]$. In this section, we show that this scheme is not applicable beyond middle frames due to the temporal ordering ambiguity. Fig. 3.8 (a) and (g) show the results with an independent frame reconstruction scheme. The quality of the reconstructed frame worsens as the distance from the middle frame increases.

3.6.3 Global Frame Reconstruction

In this section, we show that the global ordering-invariant loss is not a good option either. Fig. 3.8 (b) and (h) show the reconstructed seven frames, where the middle frame is reconstructed independently with the loss $\mathcal{L}_{\text{middle}}$ and the other six frames are reconstructed jointly with the globally ordering-invariant loss $\mathcal{L}_{\text{model}}$. The non-middle frame prediction network does not converge well and generates artifacts.

3.6.4 Pairwise Frame Reconstruction

Fig. 3.8 (c), (d), (i) and (j) show the reconstructions with the pairwise ordering-invariant loss $\mathcal{L}_{\text{pair}}$. Rows (d) and (j) show the case where the middle frame prediction is also fed to the network, while the third row shows the case without the middle frame prediction. There are two main limitations of using $\mathcal{L}_{\text{pair}}$: 1) One has to reorder non-middle frame predictions manually; 2) Although feeding the middle frame prediction to the network gives better visual results than in the case without it, still both of these two schemes generate artifacts, especially for the frames temporally away from the middle frame.

3.6.5 Sequential Pairwise Frame Reconstruction

Fig. 3.8 (e) and (k) show the visual results with a sequential pair-wise reconstruction scheme. Notice that this scheme and the pairwise ordering scheme only differ at the 4 frame predictions $x[1], x[2], x[6]$, and $x[7]$. We can see that the sequential scheme generates fewer artifacts especially at frames $x[1]$ and $x[7]$ in Fig. 3.8 (k).

3.6.6 Teacher Forcing

We also explore the *teacher forcing* method used to train recurrent neural networks [179]. During training, we substitute the middle frame prediction $\phi_4(y)$ with the ground truth middle frame $x[4]$ in steps 2 and 3 of our full training procedure in Table 3.1. This strategy brings several benefits: 1) In practice, we observe that the teacher forcing training strategy converges faster than with a standard sequential pairwise training; 2) It also gives visually better predictions as shown in Fig. 3.8 (l), where non-middle frames are predicted by a network trained with teacher forcing; 3) The middle frame and non-middle frame prediction training can be done in parallel. We use teacher forcing training as our default network training scheme.

We use four different networks to predict all 7 frames: one for the middle frame and the others for the middle-symmetric pair-wise frames. We found that sharing the parameters of the pair-wise networks for frames 1, 2, 6, and 7 during training would not result in a loss of visual accuracy. This could make predicting more than seven frames feasible.

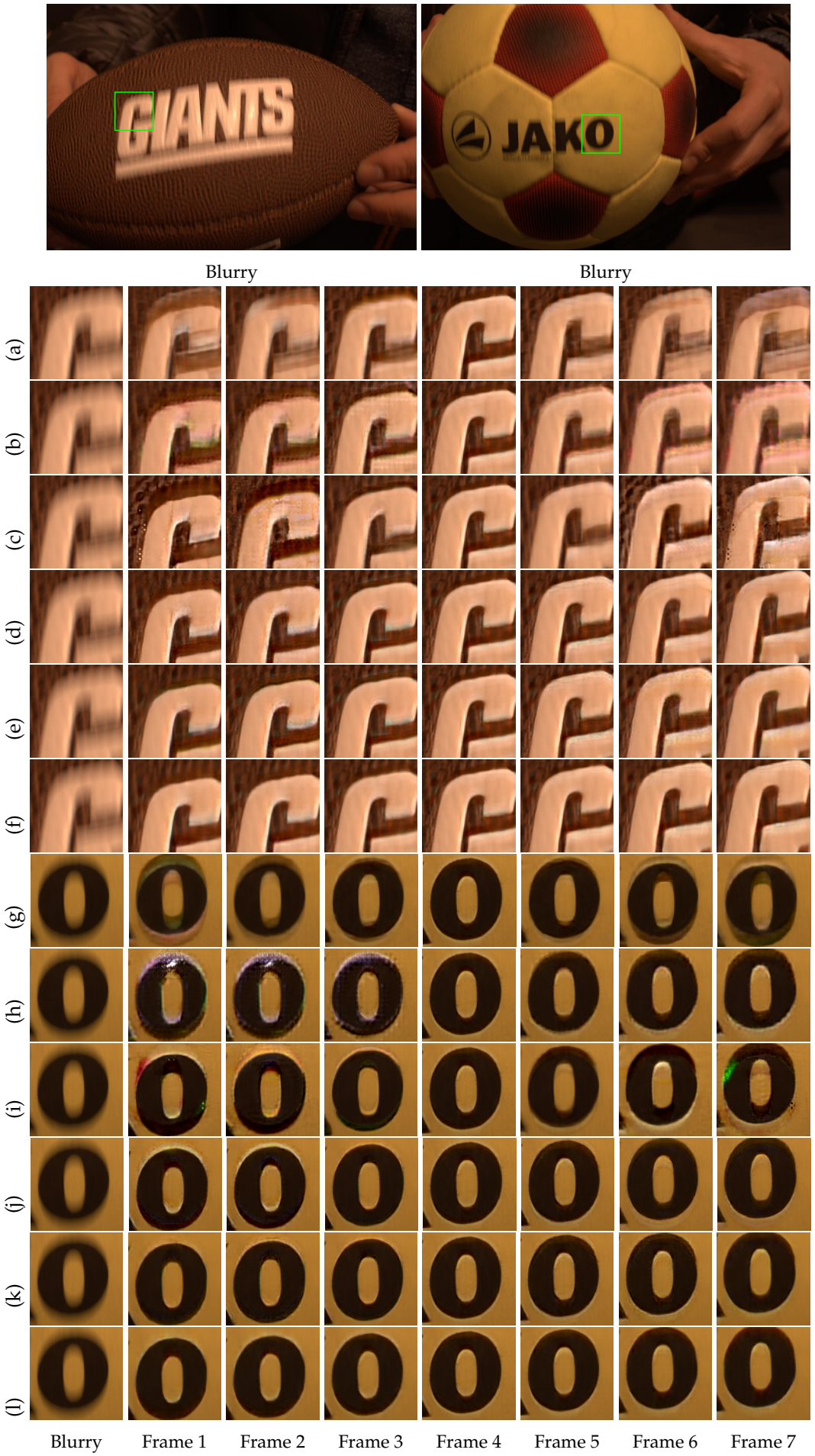


FIGURE 3.8: Ablation study on real data with different loss functions.

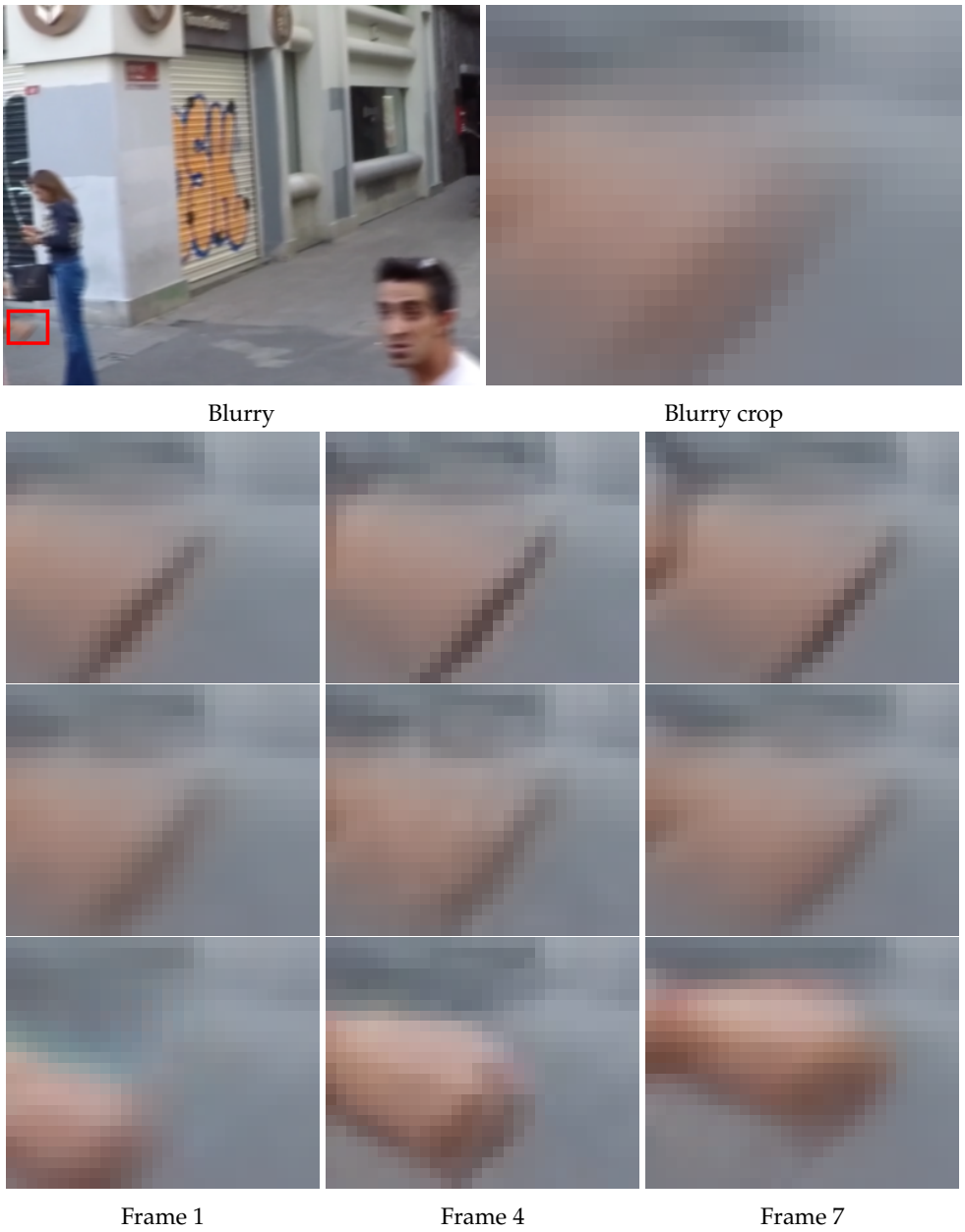


FIGURE 3.9: **A synthetic example from [2] test image.** (b)-(d) Frame 4 show our estimated middle frame, Nah's estimate and the ground truth, respectively. As can be seen, the middle frame estimates from both our method and Nah's are incorrect and affect the estimates of the other frames. Only when the ground truth middle frame is provided, the other frames can be estimated correctly.

3.6.7 Importance of the Middle Frame Estimate

We observe experimentally that a good initialization is key in making the non-middle frame prediction network work well.

Fig. 3.9 (a) shows a blurry image and an enlarged detail with significant motion blur. In Fig. 3.9 (b), we show the reconstructions of frames 1, 4 (middle), and 7 with our trained network, where we used our estimated frame 4 to recover the other frames. In Fig. 3.9 (c), we show the corresponding frames reconstructed when

feeding our network with Nah's [2] frame 4 estimate. Both cases fail to reconstruct the middle frames and the other frames. However, when we feed our networks with the ground truth middle frame (see Fig. 3.9 (d)), they can correctly reconstruct the other frames.

3.7 Discussion

In this chapter, we have presented the first method to reconstruct a video from a single motion-blurred image. We have shown that the task is more ambiguous than deblurring a single frame because the temporal ordering is lost in the motion-blurred image. We have presented a data-driven solution that allows a convolutional neural network to choose a temporal ordering at the output. We have demonstrated our model on several datasets and have shown that it generalizes on real images captured with different cameras from those used to collect the training set.

Although our system can predict seven frames from a motion-blurred image, there are two main limitations. One main limitation of our approach is that it is not robust to large blurs. Whenever our middle frame prediction network fails to remove blur, the non-middle frame prediction networks also fail.

Middle Frame Prediction Network Architecture				
Layer	Norm	Activation	Kernel	Dilation
conv0			$144 \times 16 \times 5 \times 5$	1×1
RB(1-3)	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(4)	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 3 \times 3$	2×2
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(5)	IN	ReLU	$144 \times 144 \times 3 \times 3$	2×2
	IN	ReLU	$144 \times 144 \times 3 \times 3$	4×4
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(6)	IN	ReLU	$144 \times 144 \times 3 \times 3$	4×4
	IN	ReLU	$144 \times 144 \times 3 \times 3$	8×8
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(7)	IN	ReLU	$144 \times 144 \times 3 \times 3$	8×8
	IN	ReLU	$144 \times 144 \times 3 \times 3$	4×4
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(8)	IN	ReLU	$144 \times 144 \times 3 \times 3$	4×4
	IN	ReLU	$144 \times 144 \times 3 \times 3$	2×2
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(9)	IN	ReLU	$144 \times 144 \times 3 \times 3$	2×2
	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
RB(10-12)	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 3 \times 3$	1×1
	IN	ReLU	$144 \times 144 \times 1 \times 1$	1×1
conv1-4			$16 \times 144 \times 3 \times 3$	1×1
conv5			$64 \times 3 \times 3 \times 3$	1×1
RB	IN	ReLU	$64 \times 64 \times 3 \times 3$	1×1
	IN	ReLU	$64 \times 64 \times 3 \times 3$	1×1
	IN	ReLU	$64 \times 64 \times 1 \times 1$	1×1
conv6			$3 \times 64 \times 3 \times 3$	1×1

TABLE 3.4: The layer architecture of the middle frame prediction network. RB and IN in the table indicate a residual block and instance normalization.

Chapter 4

Learning to Deblur and Rotate Motion-Blurred Faces

FIGURE 4.1: **Blurry inputs and reconstructed sharp multi-view videos on our dataset (to play the videos open the paper pdf with Adobe Acrobat Reader).** We propose a model that, given the image of a blurry face, can render a corresponding sharp video from arbitrary viewpoints.

Faces are a fundamental subject in image processing and recognition due to their role in applications such as teleconferencing, video surveillance, biometrics, video analytics, entertainment, and smart shopping, just to name a few. In particular, in the case of teleconferencing, the interaction is found to be more engaging when the person on the screen looks towards the receiver [3]. However, it is necessary to look directly into the camera to achieve this configuration. Unfortunately, this does not allow one to watch the person on the screen that one talks to. A solution to this issue is to design a system that can render the captured face from an arbitrary viewpoint. Then, it becomes possible to dynamically adapt the gaze of the face on the screen to ensure that it aims at the observer. Moreover, because of the low frame rate of web cameras, especially when used in low light, it becomes essential to solve the above task in the presence of motion blur. Since a blurry image is a result of averaging several sharp frames [2], one could pose the problem of recovering not one but a sequence of sharp frames from the single blurry input. This capability enables a smooth temporal rendering of the video. In addition, one might use this capability to deal with a limited connection bandwidth. Current software fits the available bandwidth by reducing the frame rate of the captured video. However, instead of selecting temporally distant frames, one could also transmit the average of several frames and then restore the original (high) frame rate at the destination terminal.

In this chapter, we present a method that recovers a sharp video rendered from an arbitrary viewpoint from a single blurry image of a face (see Fig. 4.1). Fig. 4.2 shows our model during the inference stage. We design a neural network and a training scheme to remove motion blur from an image and produce a video of sharp frames with a general viewpoint. Our neural network is built in two steps: First, by training a generative model that outputs face images from zero-mean Gaussian

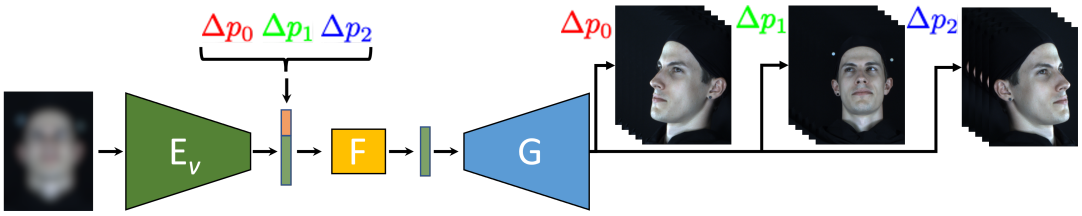


FIGURE 4.2: **Overview of our system during inference.** The encoder E_v encodes a blurry image into a sequence of latent codes that are then manipulated based on a relative viewpoint (*e.g.*, Δp_0, Δp_1, or Δp_2) via the fusion network F to produce encodings of images from a novel view. Finally, the generator G maps the novel view encodings to the image space.

noise, which we call the *latent space*, and then by training encoders to map images to the latent space. The primary motivation for using a generative model is that face rotations can be handled more easily in the latent space than in the image space. This property was recently observed for generative adversarial networks [180]. One encoder is trained so that, when concatenated with the generator, it autoencodes face images. Then, rather than using sharp images as targets in a loss, we use their encodings, the latent vectors, as targets. As a second step, we obtain a blurry image by averaging several sharp frames. Then, we train a second encoder to map the blurry image to a sequence of latent vectors that match the target latent vectors corresponding to the original sharp frames. Finally, the change of the face viewpoint requires the availability of the latent vectors corresponding to the same face instance, but rotated. To the best of our knowledge, there are no public face datasets with such data. Thus, we built a novel multiview face dataset. This dataset consists of videos captured at 112 fps of 52 individuals performing several expressions. Thanks to the high frame rate, we can simulate realistic blur through temporal averaging. Each performance is captured simultaneously from 8 different viewpoints so that it is possible to encode multiple views of the same temporal instance into target latent vectors and then train a fusion network to map the latent vector of one view and a relative viewpoint to the latent vector of another view of the same face instance. The relative viewpoint we provide as input should be the relative pose between the input and the output face poses. While we can use the viewpoint information from our calibrated camera rig during training, this information may be unknown with new data. Hence, we also train a neural network to estimate the head pose. The network learns to map an image to Basel Face Model [181] (BFM) parameters, such that, when rendered (through a differential renderer), it matches the input image.

Contributions. We make the following contributions: *(i)* We introduce BMFD, a novel high frame rate multi-view face dataset that allows more accurate modeling of natural motion blur and the incorporation of 3D constraints; *(ii)* As a novel task enabled through this data, we propose a model that, given a blurry face image, can synthesize a sharp video from arbitrary views; *(iii)* We demonstrate this capability on our multiview dataset and VIDTIMIT [182].

4.1 Background

In Section 2.4.4 we acknowledged works covering the task of face deblurring. Since our method allows the rendering of deblurred faces from novel views, we briefly discuss relevant work on novel face view synthesis. Prior works can

be divided into two main groups: full 3D face reconstruction-based works and GAN/Autoencoder-based works. In the first group of works, 3D (*e.g.* mesh and texture) is the direct output of the model. Therefore, novel view synthesis is achieved via rendering the resulting mesh and texture from any point of view. In the second group of works, 3D is not explicitly modeled or regressed; instead, it is backed into the latent representation of the model. The input image is first encoded into some latent space and decoded into some novel view using the decoder model.

4.1.1 3D Face Reconstruction

3D morphable models (3DMM) [183] provide an interpretable generative model of faces in the form of a linear combination of base shapes. In the past decades many improvements were made using more data, better scanning devices or more detailed modelling [181], [184]–[193]. 3D face reconstruction can be cast as regressing the parameters of such 3DMMs. The model parameters can be fit using multi-view images [194]–[198]. Since 3DMMs provide a strong shape prior, they also enable single-image 3D reconstruction [199]–[201]. These methods learn to estimate the model parameters by matching input images with differentiable rendering techniques [202]–[205]. We also leverage a 3DMM to learn a controllable representation of faces. In our work, these representations are used to manipulate the latent space of a StyleGAN generator.

4.1.2 Novel Face View Synthesis

Xu *et al.* [206] use an encoder-decoder architecture. The encoder extracts view independent features, which are fed to the decoder along with sampled camera parameters. Realism and pose consistency are enforced via GANs. [207] use face landmarks to guide and condition the novel face view reconstruction. A special case of novel-view synthesis on faces is face frontalization [208]–[210]. [211] design a GAN architecture for face frontalization. Their generator consists of two pathways: A global pathway processes the whole image, and a local pathway processes local patches extracted at landmarks. Tackling the opposite problem, [212] train a GAN to generate silhouette images to reduce the pose bias in existing face datasets. To the best of our knowledge, we are the first to deblur and synthesize frames from a novel view simultaneously.

4.2 Model

Our goal is to design a model that can generate a sharp video of a face from a single motion-blurred image. Additionally, we want to synthesize novel views of these videos, *i.e.*, rotate the reconstructions. We design a modular architecture to achieve this goal (see Fig. 4.4). We give an overview of the components here and provide more details in the following subsections. The bedrock of our approach is a generative model G of sharp face images. We describe how we can leverage the generative model G by learning an inverse mapping E_s from image-space to G's latent space in section 4.2.3. The sharp image encoder E_s then acts as a teacher for a blurry image encoder E_v. In section 4.2.4 we describe how to train E_v to predict latent codes of multiple sharp frames by using encodings of E_s as targets. To perform novel view synthesis, we require to capture the 3D viewpoint of the face. To this end, we learn a viewpoint extractor E_{3D} that maps a blurry image to coefficients of a 3DMM. We

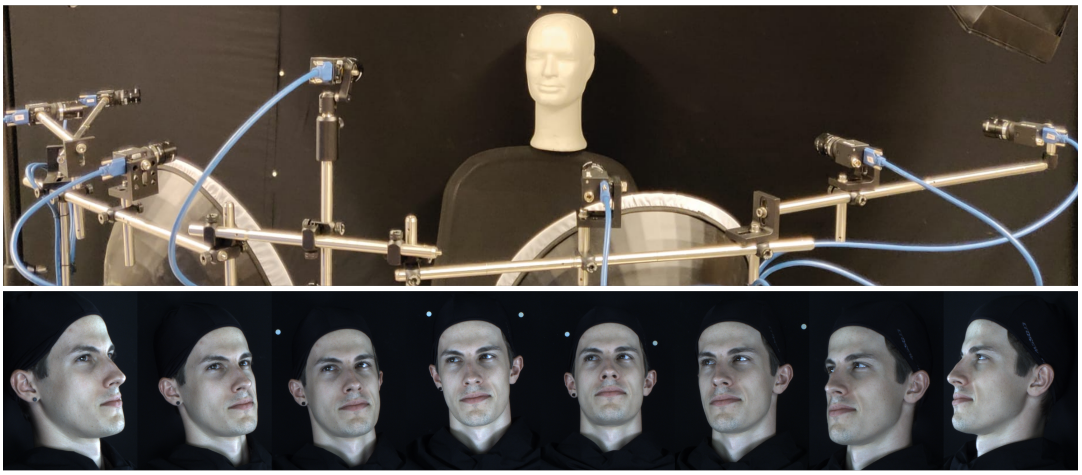


FIGURE 4.3: **Overview of our multi-view video capture setup.** We arranged eight high-speed cameras in a circular grid in our lab setting. The cameras capture synchronized videos of participants performing a wide range of facial expressions from a wide variety of viewpoints. We show an example of 8 synchronized views of one of the 52 participants in BMFD. Background and clothing are black, allowing the easier extraction of skin regions.

describe how to train E_{3D} using a differentiable renderer in section 4.2.5. The viewpoint from E_{3D} can then be used to manipulate the latent codes of a blurry image obtained through E_v. We do so by training a model F that, given relative viewpoint changes obtained through E_{3D} and latent codes from E_v, outputs updated latent codes corresponding to the desired change of viewpoint. This process is described in section 4.2.6.

4.2.1 Data

Our dataset consists of a set of sharp frames $\{y_i\}_{i=1}^{N}$. We synthesize blurry images by averaging $2m+1$ consecutive frames, *i.e.*, $x_i = \frac{1}{2m+1}\sum_{j=i-m}^{i+m} y_i$. As targets we define a sequence of 5 sharp frames $\boldsymbol{y}_i = [y_{i-m}, y_{i-m/2}, y_i, y_{i+m/2}, y_{i+m}]$. The training dataset then is given by

$$\mathcal{D} = \left\{(x_i^v, \boldsymbol{y}_i^v) \mid i = 1, \ldots, n; v = 1, \ldots, 8\right\}, \tag{4.1}$$

where the superscript v indicates the viewpoint (we omit v when it is not needed).

4.2.2 Bern Multi-View Face Dataset

Most prior face deblurring methods tackle the shift-invariant blur case, *i.e.*, blur that might arise from camera shake. Training data for such methods can be synthesized by convolving sharp face images with random blur kernels [87], [88], [91]. However, such models do not generalize well to blur caused by face motion since the resulting blurs are no longer spatially invariant. To tackle motion blur, Ren *et al.* [90] generate training data by averaging consecutive frames of the 300-VW dataset [213]. This is a valid approximation of natural motion blur when the frame rate of the videos is sufficiently high. Since the 300-VW data has a relatively low frame rate of 25-30 fps, the resulting synthetic motion blurs are not always of high quality and can exhibit ghosting artifacts. Additionally, existing face datasets exhibit a pose bias,

with most images showing faces in a frontal pose. Methods trained on such data can show poor generalization to non-frontal views.

To overcome these limitations, we introduce a dataset of high-speed, multi-view face videos. The faces of 52 participants were captured in a lab setting from 8 fixed viewpoints simultaneously. The cameras were arranged in a circular grid, ensuring that the faces are captured from all sides (see Fig. 4.3). Videos are captured at 112 frames per second at a resolution of 1440×1080. The duration of the recordings ranges between 75 and 90 seconds.

4.2.3 Inverting a Generative Face Model

In order to generate novel views of a video sequence, we rely on a generative model of face images with a latent space where manipulations that change viewpoints are feasible. Consequently, we chose to train a SyleGAN2 [214] as the generator G of sharp face images. SyleGAN2 provides state-of-the-art image quality and a smooth, disentangled latent space. To reconstruct or manipulate a given face image y_i, we require a corresponding latent code z_i, s.t. $G(z_i) = y_i$. To this end, we train a sharp image encoder E_s to invert the generator G, *i.e.*, we want that $G(E_s(y)) = y$. We adopt the inversion strategy of Meishvili *et al.* [215], where the encoder E_s is trained while the generator G is fine-tuned. The training objective is given by

$$\min_{E_s,G} \sum_{i=1}^{n} \ell_s \left(G\left(E_s(y_i)\right), y_i \right) + \lambda_g \left| G_{\text{init}} - G \right|_2^2 + \lambda_s \left| 1 - \left| E_s(y_i) \right| \right|, \tag{4.2}$$

where ℓ_s represents the following combination of different reconstruction losses:

$$\ell_s(x,y) = \lambda_{id}\mathcal{L}_{id}(x,y) + \lambda_{per}\mathcal{L}_{per}(x,y) + \lambda_{edge}\mathcal{L}_{edge}(x,y) + \left| x - y \right|,$$

$$\begin{aligned} \mathcal{L}_{id}(x,y) &= 1 - \frac{< \phi_{id}(x), \phi_{id}(y) >}{|\phi_{id}(x)| \cdot |\phi_{id}(y)|}, \\ \mathcal{L}_{per}(x,y) &= \left| \phi_{per}(x) - \phi_{per}(y) \right|_2^2, \\ \mathcal{L}_{edge}(x,y) &= \left| \mathtt{S}(x) - \mathtt{S}(y) \right|. \end{aligned}$$

$\mathcal{L}_{id}$ is a term minimizing the cosine between embeddings of a pre-trained identity classification network ϕ_{id} of Cao *et al.* [216]. $\mathcal{L}_{per}$ is a perceptual loss on features of an ImageNet pre-trained VGG16 network ϕ_{per} [217]. $\mathcal{L}_{edge}$ is a Sobel edge matching term. We used a naive Bayes classifier with Gaussian Mixture Models trained on a skin image dataset from [218] to double the contribution of the skin pixels in all the losses.

$\lambda_g = 1$ controls how much G is allowed to deviate from the initial generator parameters G_{init} (before fine-tuning), and $\lambda_s = 1$ softly enforces that the predicted latent codes lie on the unit hypersphere. During training, we gradually relax λ_g until we reach the desired reconstruction quality. Similar to [215] we regress multiple latent codes per frame, each injected at different layers of the StyleGAN2. Thus $E_s(y_i) = z_i \in \mathbb{R}^{14\times512}$. Weights controlling the contribution of each term are set as follows: $\lambda_{id} = 0.5$, $\lambda_{per} = 10^{-6}$, $\lambda_{edge} = 0.2$, $\lambda_g = 1$.

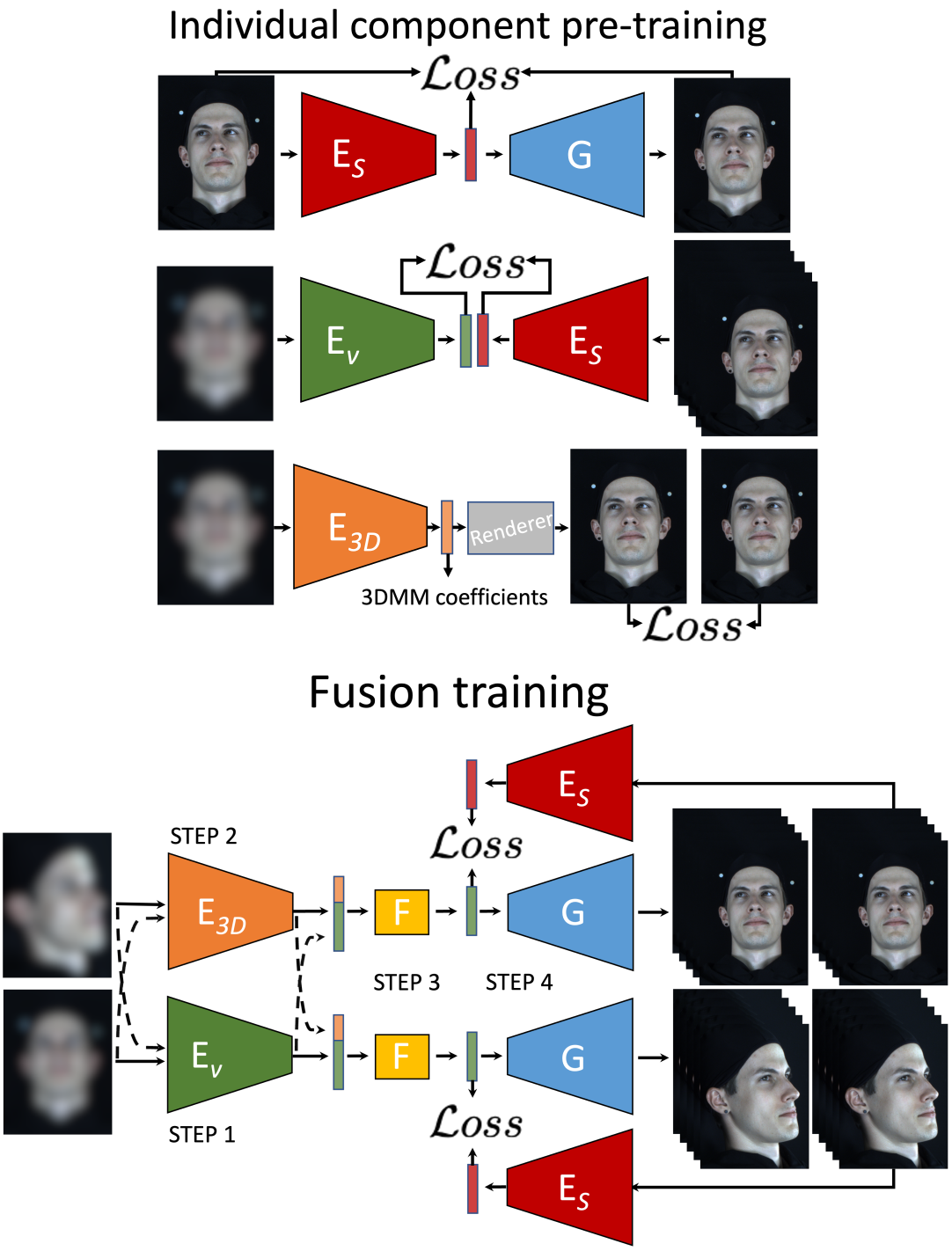


FIGURE 4.4: **Overview of the model architecture.** From top to bottom, on the right side of the figure, we show the individual pre-training stages of encoders: E_s, E_v, and E_{3D}. A sharp image generator G, is pre-trained using StyleGAN2. The training of the model F is shown on the right side of the figure. The encoder E_v encodes a blurry image into a sequence of latent codes corresponding to a sequence of sharp frames (step 1). Pose information is extracted via the viewpoint encoder E_{3D}, which is trained to regress the coefficients of a 3DMM (step 2). The predicted sharp latent codes are then manipulated based on the pose encodings via the fusion network F to produce latent codes of images from a novel view (step 3). Finally, generator G maps the novel view encodings to the image space (step 4).

4.2.4 Predicting Sharp Latent Codes from a Blurry Image

In this section we describe how to train a blurry image encoder E_v that maps a blurry image x_i to a sequence of 5 latent codes $\boldsymbol{z}_i = [z_{i-m}, z_{i-m/2}, z_i, z_{i+m/2}, z_{i+m}]$ corresponding to the target sharp frame sequence $\boldsymbol{y}_i$. We train the encoder E_v by using the the pre-trained sharp image encoder E_s as teacher. Let $\boldsymbol{z}_i = [E_s(y_{i-m}), \ldots, E_s(y_{i+m})]$ denote the sequence of target codes obtained by encoding each target sharp image in the sequence $\boldsymbol{y}_i$ with E_s.

Jin *et al.* [219] point out ambiguities when regressing a sequence of sharp frames from a blurry image. Indeed, the order of the regressed frames can be ambiguous since the output sequence is often valid whether it is played forward or backward. We handle this forward/backward ambiguity by allowing for either solution in the training objective. Let the reversed target sequence be denoted with $\bar{\boldsymbol{z}}_i = [E_s(y_{i+m}), \ldots, E_s(y_{i-m})]$. The training objective for E_v is then given by

$$\min_{E_v} \sum_{n=1}^{n} \min\left(\left|E_v(x_i) - \boldsymbol{z}_i\right|, \left|E_v(x_i) - \bar{\boldsymbol{z}}_i\right|\right), \tag{4.3}$$

where we minimize either over the forward or backward target sequence, depending on which one better matches the prediction.

4.2.5 Regressing a 3D Face Model

To perform a novel view synthesis of the reconstructed sharp frame sequence, we need to know the 3D rotation of the face. Our approach is to learn to extract the 3D viewpoint of a face by training an encoder E_{3D} to regress the coefficients of a 3DMM [181] along with camera parameters that define the rotation angles $R \in \mathbb{R}^3$, the translation $t \in \mathbb{R}^3$, and the illumination coefficients $\gamma \in \mathbb{R}^9$. The 3DMM coefficients can be grouped into components responsible for representing identity α, texture β, and facial expression δ. Given a blurry face image x_i^v from view v, we thus train a ResNet-50 [176] to regress the vector $c_i^v = (\alpha_i, \beta_i, \delta_i, \gamma_i^v, R_i^v, t_i^v) \in \mathbb{R}^{460}$ of 3D coefficients corresponding to the sharp middle frame y_i^v. The predicted 3D coefficients c_i^v are passed through a differentiable renderer ϕ [204] and the 3D encoder E_{3D} is trained by minimizing

$$\min_{E_{3D}} \sum_{i=1}^{n} \sum_{v=1}^{v_i} \ell_{im}\left(\phi\left(E_{3D}(x_i^v)\right), y_i^v\right) + \ell_{3D}\left(E_{3D}(x_i^v), y_i^v\right) + \lambda_c(|\alpha_i|^2 + |\beta_i|^2 + |\delta_i|^2), \tag{4.4}$$

where ℓ_{im} and ℓ_{3D} represents the following combination of different reconstruction losses:

$$\begin{aligned}
\ell_{im}\Big(x, y\Big) &= \lambda_{id}\mathcal{L}_{id}\Big(x, y\Big) + \lambda_{edge}\mathcal{L}_{edge}\Big(x, y\Big) + \lambda_{data}\big|x - y\big|, \\
\ell_{3D}\Big(x, y\Big) &= \lambda_{lan}\mathcal{L}_{lan}\Big(x, y\Big) + \lambda_{mview}\mathcal{L}_{mview}\Big(x\Big), \\
\mathcal{L}_{lan}\Big(x, y\Big) &= \big|\mathcal{Q}_{basel}(x) - \mathcal{Q}_{image}(y)\big|_2^2, \\
\mathcal{L}_{mview}\Big(x\Big) &= \sum_{v} |\bar{\alpha} - \alpha^v|^2 + |\bar{\beta} - \beta^v|^2 + |\bar{\delta} - \delta^v|^2.
\end{aligned}$$

where, $\lambda_{data} = 5$, $\lambda_{id} = 0.5$, $\lambda_{edge} = 30$, $\lambda_{mview} = 0.25$ and $\lambda_{lan} = 1$. We use the perspective camera model in the renderer ϕ, with an empirically selected focal length for the 3D-2D projection. The term $\mathcal{L}_{lan}$ is a MSE between 2D projections of facial

landmarks of the predicted mesh and pre-computed landmarks in sharp images. $\mathcal{Q}_{basel}$ projects the 3D landmark vertices of the reconstructed mesh onto the image (obtaining 68 facial landmarks), and $\mathcal{Q}_{image}$ extracts landmarks using the method of [220] from the ground-truth targets. $\mathcal{L}_{mview}$ ensures that the identity, texture, and expression parameters of the BFM are consistent across views for samples of our multi-view dataset.

where ℓ_{im} and ℓ_{3D} are a combination of different reconstruction losses (see supplementary for details), and $\lambda_c = 10^{-4}$ controls the amount of regularization applied to the 3DMM coefficients to prevent a degradation of face shape and texture. Note that the coefficients, α, β, γ, are shared across different views, promoting the accurate learning of facial expressions.

4.2.6 Learning to Rotate Faces in Latent Space

Given a blurry image x_i^ν from viewpoint ν and associated latent codes $\boldsymbol{z}_i^\nu = E_v(x_i^\nu)$ as well as pose information $E_{3D}(x_i^\nu)$, we aim to manipulate $\boldsymbol{z}_i^\nu$ in latent space such that the reconstruction exhibits a desired change of viewpoint. We implement this by learning a fusion network F that takes as input a pair $(z_j^\nu, \Delta p)$ consisting of a single frame encoding z_j^ν and a relative change in pose Δp. The output modified latent codes are then given by applying F to all frames in the sequence independently, *i.e.*, the modified codes are given by $\boldsymbol{z}_i^{\nu+\Delta p} = [F(z_{i-m}^\nu, \Delta p), \ldots, F(z_{i+m}^\nu, \Delta p)]$.

During training, we sample two blurry images x_i^u and x_i^v from two different viewpoints, but with the same timestamp. The change in viewpoint is then computed from $E_{3D}(x_i^u) - E_{3D}(x_i^v)$, which corresponds to $\Delta p_i^{uv} = (R_i^v - R_i^u)$, *i.e.*, the difference in the estimated 3D rotation angles between the two views. We train the fusion model F to regress the latent codes $\boldsymbol{z}_i^v$ from the pair $(\boldsymbol{z}_i^u, \Delta p_i^{uv})$ by optimizing the following objective

$$\min_F \sum_{i=1}^{n} \sum_{u \neq v} \min(|F(\boldsymbol{z}_i^u, \Delta p_i^{uv}) - \boldsymbol{z}_i^v|, |F(\boldsymbol{z}_i^u, \Delta p_i^{uv}) - \bar{\boldsymbol{z}}_i^v|), \tag{4.5}$$

where the `min` function again takes care of possible frame order ambiguities.

4.2.7 Implementation Details

We employed ResNet-50 [176] as a backbone architecture for E_s, E_v and E_{3d}. The average-pooled features are fed through fully-connected layers with 14×512 (single frame), $5 \times 14 \times 512$ (5 frames) and 460 neurons for E_s, E_v and E_{3d} respectively. The generator G is pre-trained with all hyper-parameters set to their default values on 8 NVIDIA GTX 1080Ti GPUs (see Karras *et al.* [214] for details). All other networks were trained on 3 NVIDIA GeForce RTX 3090 GPUs. The Adam optimizer [221] with a fixed learning rate of 10^{-4} was used for the training of all the networks. We used batch sizes of 72, 96, 90, 84 samples for E_s, E_v, E_{3d} and F respectively. We trained our models E_s, E_v, E_{3d} and F for 1000K, 100K, 600K, and 500K iterations each. The ratio of samples within one batch stemming from FFHQ, 300VW and BMFD is 2:1:1. All the models are trained on an image resolution of 256×256. We used random jittering of hue, brightness, saturation, and contrast for data augmentation.

Frames	Views				
	1,8	2,7	3,6	4,5	All
Middle 3	2.93	2.78	2.89	2.82	2.85
Frames 2,4	3.31	3.13	3.29	3.27	3.25
Frames 1,5	3.94	3.84	3.99	4.01	3.95
All Frames	3.50	3.35	3.51	3.49	3.46

TABLE 4.1: **Same view landmark error.** We report the landmark error (in pixels) between the ground-truth and reconstructed frame sequences without rotation.

4.3 Experiments

In this section, we perform experiments to quantify the facial pose accuracy of the reconstructed original frame sequence and novel view frame sequence. We also evaluate identity preservation under novel view synthesis. Finally, we compare our method qualitatively and quantitatively to the state-of-the-art methods of Jin *et al.* [219] and Zhou *et al.* [222].

4.3.1 Datasets

Besides our novel multi-view face dataset we also use 300VW [223], FFHQ [224] and VIDTIMIT [182] in our experiments. To synthesize motion-blurred images for training, we average *(i)* 65 consecutive frames from videos of 40 identities of our new dataset, and *(ii)* 9 consecutive frames from 65 identities of 300VW. To increase the number of identities for training and avoid overfitting, we also incorporate samples from FFHQ. Since FFHQ consists of still images, we simulate blurs by convolving images with randomly sampled 9×9 motion blur kernels. Because 300VW and FFHQ lack multiple views, we simulate them via horizontal mirroring of frames. We evaluate our method on the remaining identities of our new dataset and the VIDTIMIT dataset.

4.3.2 Pose-Regression Accuracy of E_v

We perform experiments to quantify the facial pose accuracy of the reconstructed frame sequence $G(E_v(x))$. To this end, we extract facial landmarks using the method of [220] from both the reconstructed and the ground-truth frame sequence on test subjects of our dataset. We report the MSE between them in Table 4.1 (again adjusting for the forward/backward ambiguity). We observe that the mean landmark error is slightly larger for peripheral frames (1, 2, 4, and 5) than the middle one (3). The mean landmark error is 3.46 pixels which amounts to 1.35% of the 256×256 image resolution.

4.3.3 Identity Preservation and Pose Accuracy under Novel View Synthesis

A key component of our method is the fusion model F, which performs the manipulation in the latent space that results in a change of the viewpoint. We thus

Fusion	Viewpoint Change		
	±30°	±45°	±60°
FC3	51% (86%)	27% (62%)	14% (37%)
FC3R	56% (84%)	36% (66%)	22% (45%)

TABLE 4.2: **Identity agreement between frontal and rotated sequences.** We report the Top-1 (Top-5) label agreement of a pre-trained identity classifier between frontal and rotated views. Note that the classifier has a sensitivity of 61% (87%) on average over all viewpoints.

Frames	Fusion	Views				
		1,8	2,7	3,6	4,5	All
Middle 3	FC3R	6.07	7.37	7.03	3.80	6.07
	FC3	6.67	7.51	7.02	3.99	6.30
Frames 2,4	FC3R	6.08	7.33	7.03	3.85	6.07
	FC3	6.61	7.49	6.99	4.02	6.28
Frames 1,5	FC3R	6.20	7.46	7.17	4.03	6.21
	FC3	6.63	7.63	7.14	4.20	6.40
All Frames	FC3R	6.12	7.39	7.09	3.91	6.13
	FC3	6.63	7.55	7.06	4.09	6.33

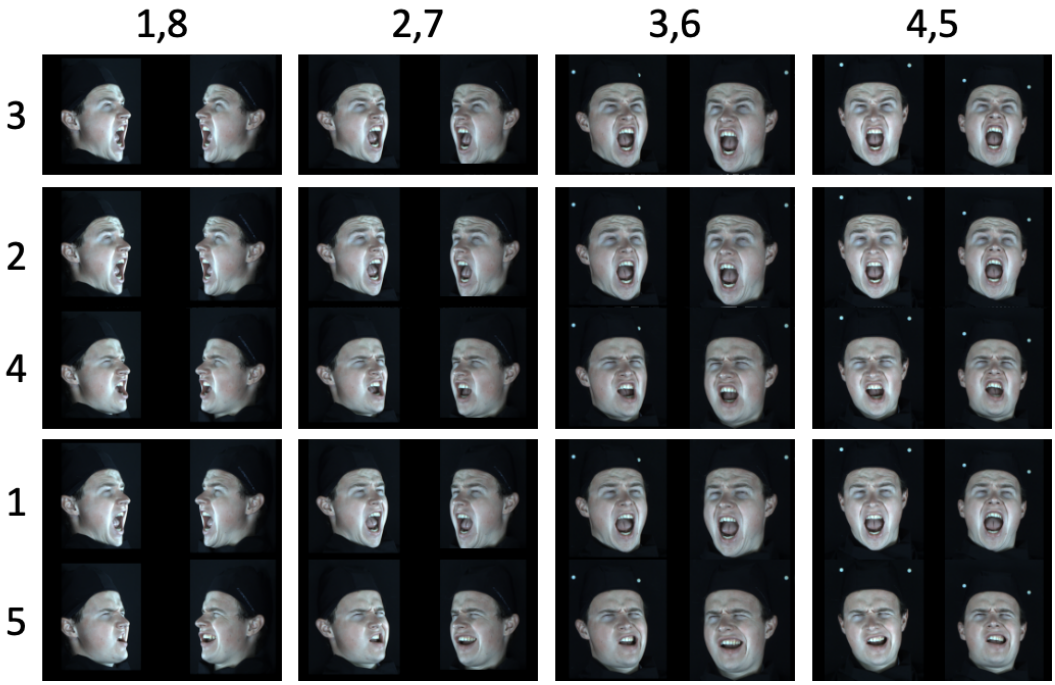


TABLE 4.3: **Face landmark accuracy for different fusion models.** In the table we report the landmark error of different frames in the reconstructed sequence (rows) and when faces are rotated to the different views in BMFD (columns). The blurry input image is taken from view 4 in all cases. An illustration of the frame and view layout is given on the right.

perform ablation experiments for different architecture designs of F, where we measure how well they reconstruct the pose in novel views and how well they preserve the identity of the face. We consider two functional designs: *(i)* $FCxR$, where F is modelled via residual computation, *i.e.*, $F(z, \Delta p) = z + MLP_x([z, \Delta p])$, and *(ii)* FCx, where F simply consists of x fully-connected layers, *i.e.*, $F(z, \Delta p) = MLP_x([z, \Delta p])$ (x indicates the number of layers in the MLP). We want F only to affect the 3D orientation of the face in our method and preserve the face identity as much as possible. To quantify the consistency of face identities under novel view synthesis, we compute the agreement of a pre-trained identity classifier [216] between a restored frontal view and reconstructions under varying amounts of rotation. We report the resulting Top-1 and Top-5 label agreements on VIDTIMIT in Table 4.2. Because the identity classifier is not perfectly robust to face rotations, we also report the estimated identity agreement of the classifier (its sensitivity) on sharp ground-truth rotations. We observe that the identity labels of rotated sequences are relatively consistent with the classifier's sensitivity on ground truth rotations up to $\pm 30^\circ$. The residual version $FCxR$ performs considerably better. To quantify the accuracy of the predicted face pose under novel view synthesis, we measure the face landmark error between the ground truth views and our reconstructions on test subjects of our multi-view dataset. Blurry frontal images (view 4) are fed through our model to reconstruct sharp frame sequences corresponding to the other seven views in our dataset. We report the mean landmark errors of different fusion models for all the views and predicted frames in Table 4.3. We observe that the average error across all views and frames varies between 6.13 and 6.33 pixels. Note that the reconstructions without rotations already show a mean landmark error of 3.46 pixels (see Table 4.1). Qualitative reconstructions of frontal and rotated frame sequences obtained with our method can be found in Figs. 4.6 and 4.7. A real-world deblurring example is presented in Fig. 4.8. Some more qualitative examples of our multi-view reconstructions on VIDTIMIT[182] can be found in Figs. 4.9 and 4.10.

4.3.4 Comparison to Prior Work

We compare to Zhou *et al.* [222] on novel face view synthesis quantitatively in Table 4.4 and qualitatively in Fig. 4.5. Since [222] is trained on non-blurry face images, we feed it with sharp frontal views from VIDTIMIT and our test set. Our method was instead evaluated on blurry input images. Despite this disadvantage, our method yields a comparable accuracy. More results are shown in the supplemental material.

We evaluated the performance of our system using conventional metrics such as PSNR and SSIM. None of the existing prior deblurring work can generate novel views from a blurry input. Therefore, we use the combination of two methods for comparison purposes. We extract the sharp video sequence from a blurry input utilizing the method of Jin *et al.* [219] and subsequently rotate the resulting frames using the method of Zhou *et al.* [222]. The mean PSNR and SSIM between ground-truth and rotated sequences are reported in Table 4.5.

FIGURE 4.5: **Qualitative novel view comparison to Zhou *et al.* [222].** We compare on VIDTIMIT (top) and BMFD (bottom). Note that [222] predicts novel views from the sharp input image on the right, whereas we predict it from the blurry image on the left.

Method	**BMFD**					**VIDTIMIT**
	1,8	2,7	3,6	4,5	All	
Zhou *et al.* [222]	7.12	6.42	5.40	5.61	6.14	3.12
Ours	6.07	7.37	7.03	3.80	6.07	3.96

TABLE 4.4: **Novel view pose error comparison.** We compare to the prior novel face view synthesis method by [222] in terms of face landmark accuracy on VIDTIMIT and BMFD.

Method	**PSNR**	**SSIM**
Jin *et al.* [219] + Zhou *et al.* [222]	16.07	0.38
Ours	19.45	0.60

TABLE 4.5: **Novel view PSNR and SSIM comparison.** We compare to the prior work in terms of PSNR and SSIM metrics on our dataset. First, the blurry input images from view 4 are fed to the method of Jin *et al.* [219], then, the resulting deblurred sequences are rotated using the method of Zhou *et al.* [222].

4.4 Discussion

In this chapter, we have presented the first method to reconstruct novel view videos from a single motion-blurred face image. Capabilities of the method were demonstrated on the VIDTIMIT dataset and a novel high frame rate, multi-view facial dataset, which we introduced. The multi-view dataset is crucial in enabling the training of our model. Moreover, our dataset is not limited to our proposed task: It can also be used to evaluate facial restoration methods for 3D reconstruction, single/video super-resolution, and temporal frame interpolation.

FIGURE 4.6: **Sample sharp video reconstructions from our model.** We show reconstructed frame sequences without viewpoint change (odd columns) and with random viewpoint changes (even columns). The first row shows the blurry input image followed by landmarks computed on the first and last frame in the reconstructed sequence. The first three examples are computed on VIDTIMIT and the last two on our test set.

FIGURE 4.7: **Sample sharp video reconstructions from our model.** We show reconstructed frame sequences without viewpoint change (odd columns) and with random viewpoint changes (even columns). The first row shows the blurry input image followed by landmarks computed on the first and last frame in the reconstructed sequence. The first three examples are computed on VIDTIMIT and the last two on our test set.

FIGURE 4.8: **Qualitative sample on real-world motion blurred face.** The first column corresponds to the blurry input image. All the other columns are output sequences rotated by a different amount. Rows from 1 to 5 correspond to the appropriate frame in the output sequence. The last column is the copy of the previous one with rectangles on top of different facial regions. Rectangles are at a fixed location with respect to the image in all frames. Note how both eyes and the nose move upwards as we go from the top to the bottom.

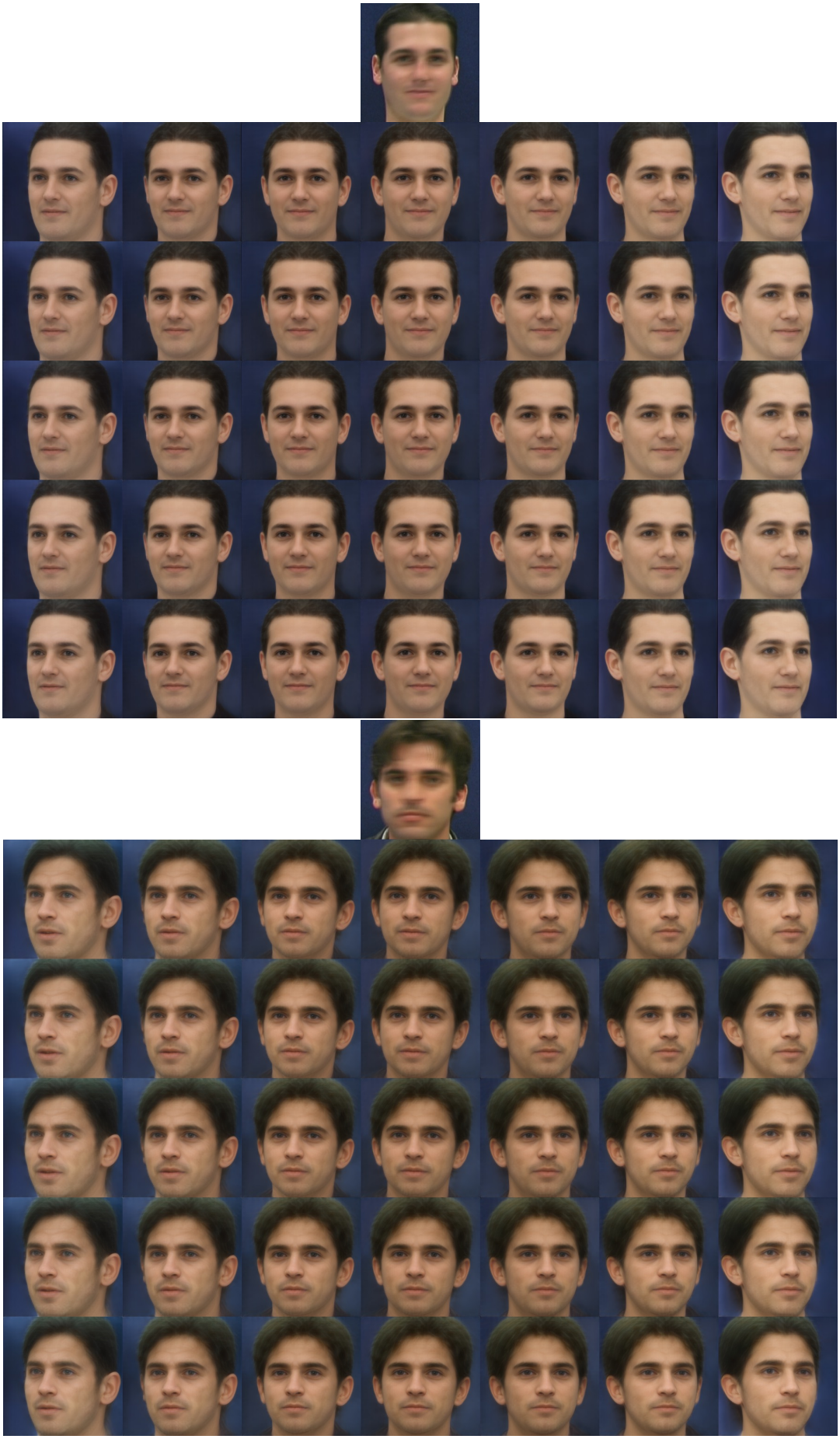


FIGURE 4.9: **Qualitative samples on VIDTIMIT.** The first column corresponds to the blurry input image. All the other columns are output sequences rotated by a different amount. Rows from 1 to 5 correspond to the appropriate frame in the output sequence.

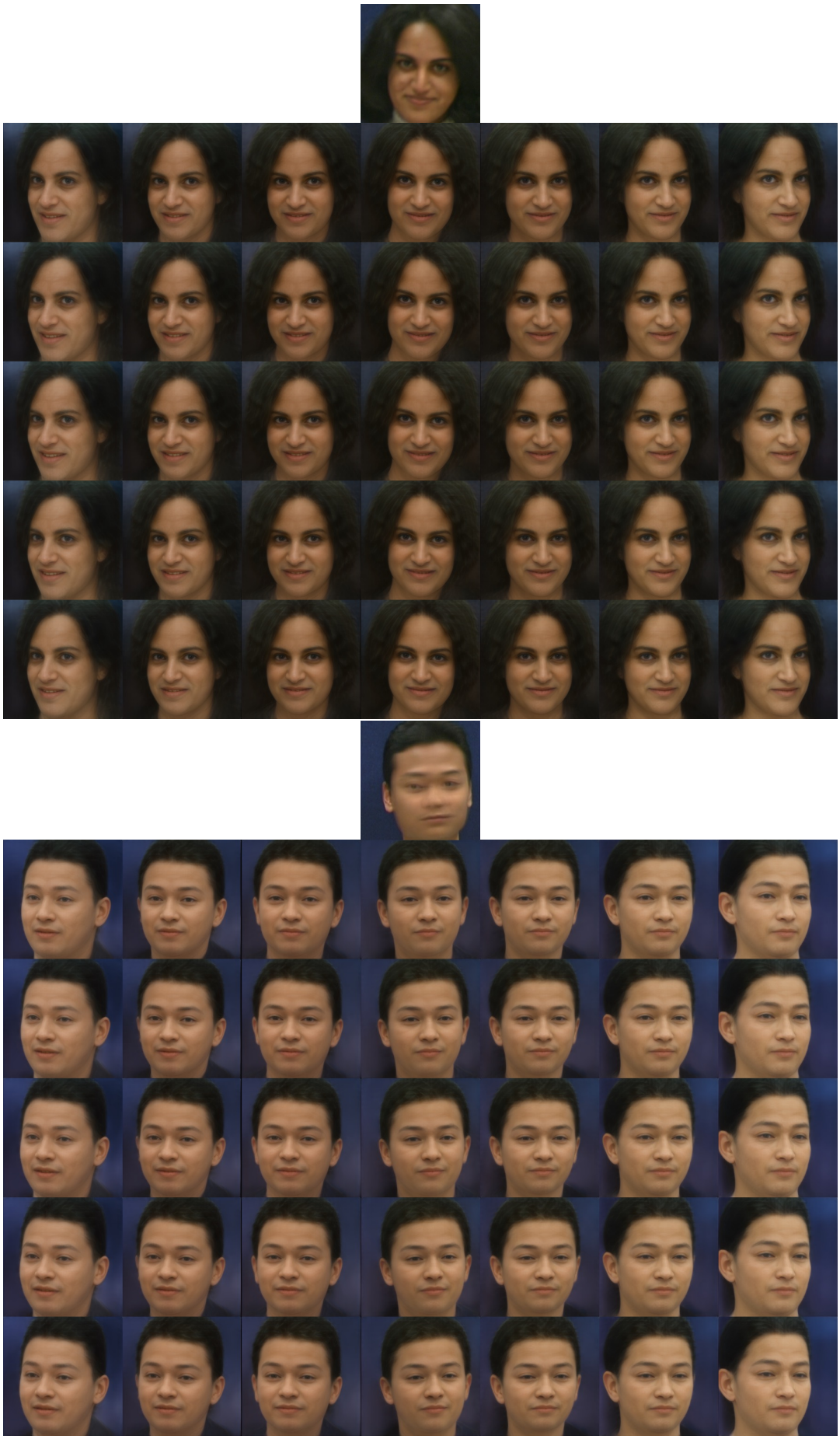


FIGURE 4.10: **Qualitative samples on VIDTIMIT.** The first column corresponds to the blurry input image. All the other columns are output sequences rotated by a different amount. Rows from 1 to 5 correspond to the appropriate frame in the output sequence.

Chapter 5

Learning to Have an Ear for Face Super-Resolution

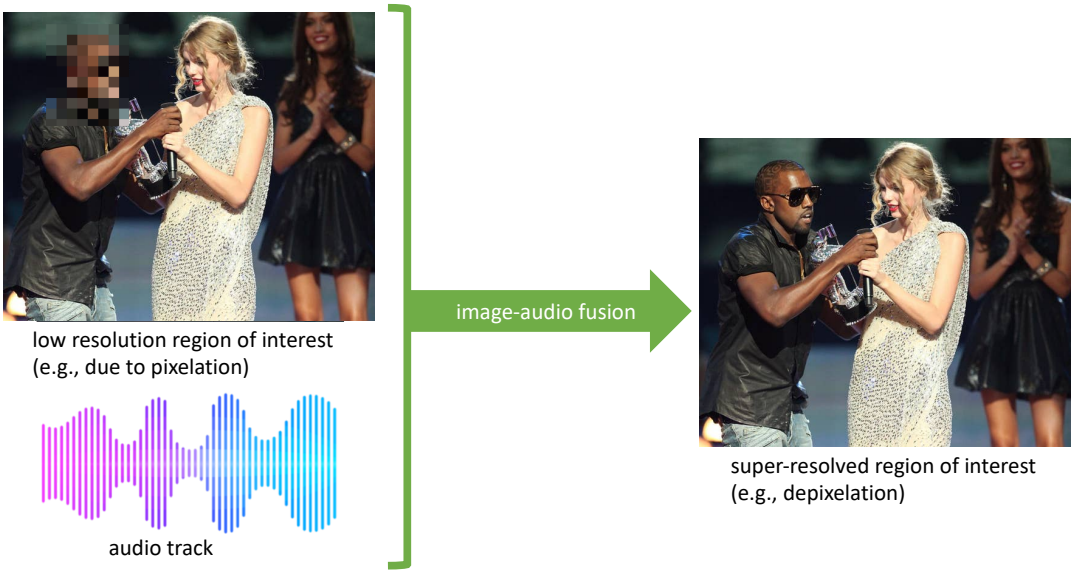


FIGURE 5.1: Pixelation is used to hide the identity of a person (left). However, audio could assist in recovering a super-resolved plausible face (right).

Image super-resolution is the task of recovering details of an image that has been captured with a limited resolution. Typically, the resolution of the input image is increased by a scaling factor of $4\times$ to $8\times$. In the more extreme case, where the scaling factor is $16\times$ or above, the loss of detail can be so considerable that important semantic information is lost. This is the case, for example, of images of faces at an 8×8 pixels resolution, where information about the original identity of the person is no longer available. The information still available in such a low-resolution image is perhaps the viewpoint and colors of the face and the background. While it is possible to hallucinate plausible high-resolution images from such limited information, useful attributes such as the identity or even just the gender or the age might be incorrect (see Fig. 5.2 (a)-(d)).

If the low-resolution image of a face is extracted from a video, we could also have access to the audio of that person. Despite the very different nature of aural and visual signals, they both capture some shared attributes of a person and, in particular, her identity. In fact, when we hear the voice of an iconic actor, we can often picture his or her face in our minds. [4] recently showed that such capability can be learned by a machine as well. The possibility to recover a full identity is typically limited to

(a) (b) (c) (d) (e) (f)

FIGURE 5.2: **Audio helps image super-resolution.** (a) and (b) are the ground-truth and 16× downsampled images respectively; (c) results of the SotA super-resolution method of Huang *et al.* [155]; (d) our super-resolution from only the low-res image; (e) audio only super-resolution; (f) fusion of both the low-res image and audio. In these cases all methods fail to restore the correct gender without audio.

a set of known people (*e.g.*, celebrities). Nonetheless, even when a person's identity is entirely new, his or her voice indicates important facial attributes such as gender, age, and ethnicity. If such information is not present in the visual data (*e.g.*, with a low-resolution image), audio could be a benefit to image processing and, in particular, image super-resolution (see Fig. 5.2 (e)-(f)). For example, in videos where the identity of a speaker is hidden via pixelation, as shown in Fig. 5.1, audio could be used to recover a more plausible face than from the lone low-resolution image.

Therefore, we propose to build a model for face super-resolution by exploiting both a low-resolution image and its audio. To the best of our knowledge, this has never been explored before. A natural way to solve this task is to build a *multimodal network* with two encoding networks, one for the low-resolution image and one for audio, and a decoding network mapping the concatenation of the encoders outputs to a high-resolution image. In theory, a multi-modal network should outperform its uni-modal counterparts. In practice, however, this does not happen with standard networks and training strategies, as shown empirically in [225]. According to [225] the performance gap is due to: 1) the difference between modalities in term of convergence and over-fitting speeds, 2) The susceptibility of multi-modal architectures to over-fitting due to their higher capacity. To address the training issues of multi-modal networks, we propose to train the low-resolution image encoder and the audio encoder separately to equalize their disentanglement accuracy. To this aim, we first train a generator G that starts from a Gaussian latent space and outputs high-resolution images (see Fig. 5.3). The generator is trained as in the recent StyleGAN of [226], which produces very high-quality samples and a latent space with a useful hierarchical structure. Then, we train a reference encoder to invert the generator by using an autoencoding constraint. The reference encoder maps a high-resolution image to the latent space of the generator, which then outputs an approximation of the input image. Then, given a matching high/low-resolution image pair, we pre-train a low-resolution image encoder E_l to map its input to the same latent representation of the reference encoder (on the high-resolution image). As a second step, we train an audio encoder E_a and a fusion network to improve the latent representation of the (fixed) low-resolution image encoder E_l. To speed up the training of the audio encoder, we also pre-train it by using as latent representation the average of

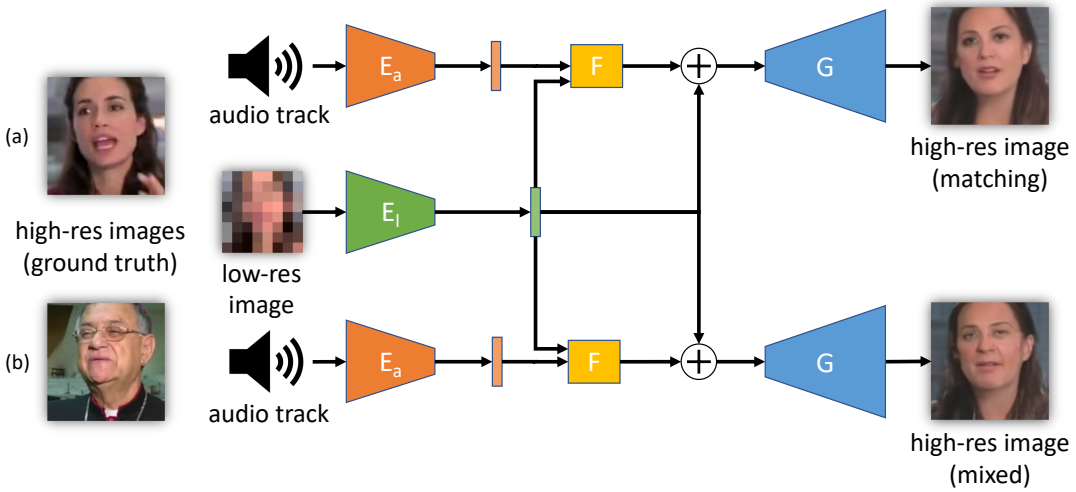


FIGURE 5.3: Simplified training and operating scheme of the proposed model. The model can be used (a) with matching inputs or (b) by mixing low-resolution images with audios from other videos. The low-resolution image (8×8 pixels) is fed to an encoder E_l to obtain an intermediate latent representation. A residual is computed by fusing in the network F the encoded audio track (through the encoder E_a) with the encoded low-resolution image. The residual is used to update the latent representation of the low-resolution image and then produce the high-resolution image through the generator G.

the outputs of the reference encoder on a high-resolution image and its horizontally mirrored version. Thanks to the hierarchical structure of the latent space learned through StyleGAN, this averaging removes information, such as the viewpoint, that audio cannot possibly carry. In Section 5.2, we describe in detail the training of each of the above models. Finally, in Section 5.3 we demonstrate experimentally that the proposed architecture and training procedure successfully fuses aural and visual data. We show that the fusion yields high-resolution images with more accurate identities, gender, and age attributes than the reconstruction based on the lone low-resolution image. We also show that the fusion is semantically meaningful by mixing low-resolution images and audio from different videos (see an example in Fig. 5.3 (b)).

Contributions: Our method builds three models for the following mappings: 1) Audio to high-resolution image; 2) Low-resolution image to high-resolution image; 3) Audio and low-resolution image to high-resolution image. The first mapping was developed concurrently to Speech2Face [4]. A notable difference is that Speech2Face is trained using a pre-trained face recognition network as additional supervision, while our method is fully unsupervised. In the second mapping, we show in our Experiments section that we achieve state-of-the-art performance at $16\times$. In the last mapping, which is the main novelty of this paper, we show that our trained model can transfer and combine facial attributes from audio and low-resolution images.

5.1 Background

In Section 2.5 we discussed super-resolution prior works including: general super-resolution, GAN based super-resolution, and face super-resolution. However, to the best of our knowledge, none of these works take as input audio signal, and

we are the first to combine audio and images in the context of a super-resolution problem. The next section covers prior works that incorporated audio signals in vision tasks.

5.1.1 Use of Audio in Vision Tasks

The use of audio in combination with video has received a lot of attention recently (see, *e.g.*, [227], [228]). Audio and video have been combined to learn to localize objects or events [229], [230], to learn how to separate audio sources [231]–[234], to learn the association between sound and object geometry and materials [235], and to predict body dynamics [236]. A significant body of work has also been devoted to the mapping of audio to visual information (see, *e.g.*, [4] and references therein).

5.2 Extreme Face Super-Resolution with Audio

Our goal is to design a model that can generate high-resolution images based on a (very) low-resolution input image and an additional audio signal. The dataset is therefore given by $\mathcal{D} = \left\{(x_i^h, x_i^l, a_i) \mid i = 1, \ldots, n\right\}$ where x_i^h is the high-resolution image, x_i^l is the low-resolution image and a_i is a corresponding audio signal. Our model consists of several components: a low-resolution encoder E_l, an audio encoder E_a, a fusion network F and a face generator G. An overview of the complete architecture is given in Fig. 5.3.

5.2.1 Combining Aural and Visual Signals

As mentioned in the introduction, a natural choice to solve our task is to train a feedforward network to match the ground truth high-resolution image given its low-resolution image and audio signal. Experimentally, we found that such a system tends to ignore the audio signal and to yield a one-to-one mapping from a low-resolution to a single high-resolution image. We believe that this problem is due to the different nature of the aural and visual signals, and the choice of the structure of the latent space. Combining both signals requires mapping their information to a common latent space through the encoders. However, we find experimentally that the audio signal requires longer processing and more network capacity to fit the latent space (this is also observed in [225]). This fitting can also be aggravated by the structure of the latent space, which might be biased more towards images than audio. Ideally, the low-resolution image should only condition the feedforward network to produce the most likely corresponding high-resolution output, and the audio signal should introduce some local variation (*i.e.*, modifying the gender or the age of the output). Therefore, for the fusion to be effective, it would be helpful if the audio could act on some fixed intermediate representation from the low-resolution image, where face attributes present in the audio are disentangled.

For these reasons, we opted to pre-train and fix the generator of a StyleGAN [226] and then train encoders to autoencode the inputs by using the generator as a decoder network. StyleGAN generators have been shown to produce realistic high-resolution images along with a good disentanglement of some meaningful factors of variation in the intermediate representations. Such models should therefore act as good priors for generating high-resolution face images, and the disentangled intermediate representations should allow better editing based on the audio signal. Formally, we learn a generative model of face images $G(z)$, where $z \sim \mathcal{N}(0, I_d)$, by optimizing the default non-saturating loss of StyleGAN (see [226] for details).

5.2.2 Inverting the Generator

Our goal is that the fusion of the information provided by the low-resolution image and audio track results in a reconstruction that is close to the corresponding high-resolution image. We pose this task as mapping an image x to its latent space target z, such that $G(z) = x$. In other words, we need to invert the pre-trained generator G. Recently, this problem has attracted the attention of the research community [237]. In this chapter, we introduce a novel GAN inversion approach, where we first pre-train the encoder E_h while the generator is fixed. Then we train the encoder E_h and the generator G (fine-tuning) through an autoencoding constraint and anchor the weights of G to its initial values through an L_2 loss. Then, the latent representation z_i corresponding to the image x_i can be generated by the encoder E_h, and used as a target by the encoders of the low-resolution images and the audio, and the fusion network.

5.2.3 Encoder Pre-Training

As a first step we train a high-resolution image encoder E_h by minimizing

$$\min_{E_h} \sum_{i=1}^{n} \left| G(z_i) - x_i^h \right|_1 + \lambda_f \ell_{\text{feat}} \left(G(z_i), x_i^h \right), \tag{5.1}$$

where $z_i = E_h(x_i^h)$, $\lambda_f > 0$ is a tuning parameter, and ℓ_{feat} is a perceptual loss based on VGG features (see Supplementary material for more details). We found that regressing a single z_i is insufficient to recover a good approximation of x_i^h. In the original style-based generator [226] each z_i is mapped to a vector w_i, which is then replicated and inserted at k different layers of the generator (each corresponding to different image scales). To improve the high-resolution reconstruction, we instead generate k different z_{ij}, $j = 1, \ldots, k$, and feed the resulting w_{ij} to the corresponding layers in the generator. The output of E_h therefore lies in $\mathbb{R}^{k \times d}$. Note that this is not too dissimilar from the training of the style-based generator, where the w-s of different images are randomly mixed at different scales.

5.2.4 Encoder and Generator Fine-Tuning

This second optimization problem can be written as

$$\min_{E_h, G} \sum_{i=1}^{n} \left| G(z_i) - x_i^h \right|_1 + \lambda_f \ell_{\text{feat}} \left(G(z_i), x_i^h \right) + \lambda_t \left| G_{\text{init}} - G \right|_2^2,$$

where $z_i = E_h(x_i^h)$, $\lambda_t > 0$ is a tuning parameter, and G_{init} denotes the weights of G after StyleGAN training. Moreover, during training, we relax the regularizer of the weights of G by reducing λ_t by a factor of 2 as soon as the overall loss is minimized (locally). The purpose of the pre-training and the regularizer decay procedure is to encourage a gradual convergence of both the encoder and the decoder without losing the structure of the latent representation of G. Examples of inversions before and after the fine-tuning are shown in Fig. 5.4. There is a visible improvement in the face's reconstruction accuracy and background. Quantitative results are shown in the Experiments section.

FIGURE 5.4: Examples of generator inversions. Top row: Autoencoding results with a fixed pre-trained generator (see eq. (5.1)). Middle row: Autoencoding results with our fine-tuned generator (see eq. (5.2)). Bottom row: Input images to the autoencoders.

5.2.5 Pre-Training Low-Res and Audio Encoders

Given the high-resolution image encoder, we now have targets z_i for the low-resolution and audio fusion. However, training a fusion model directly on these targets runs into some difficulties. As mentioned before, we find experimentally that, given enough capacity, a fusion model $F(x_i^l, a_i)$ trained to predict $z_i = E_h(x_i^h)$, ignores the audio signal a_i almost completely. To address this degenerate behavior, we train two encoders E_l and E_a separately to extract as much information from the two modalities as possible and only later fuse them. To ensure that neither of the two encoders can overfit the whole training set $\mathcal{D}$ we extract the subset $\mathcal{D}_{\text{pre}} = \left\{(x_i^h, x_i^l, a_i) \mid i = 1, \ldots, n/2\right\}$ for the encoders pre-training and use the entire $\mathcal{D}$ only for the later fusion training. The low-resolution encoder E_l is trained to regress the high-resolution encodings $z_i = E_h(x_i^h)$ from x_i^l by minimizing

$$\min_{E_l} \sum_{x_i^l, x_i^h \in \mathcal{D}_{\text{pre}}} \left| E_l\left(x_i^l\right) - z_i \right|_1 + \lambda \left| D \circ G\left(E_l\left(x_i^l\right)\right) - x_i^l \right|_1, \tag{5.2}$$

where $D \circ x$ is the $16\times$ downsampling of x and $\lambda = 40$.

In the case of the audio encoding, regressing all the information in z_i with $E_a(a_i)$ is not possible, as many of the factors of variation in z_i, *e.g.*, the pose of the face, are not present in a_i. To remove the pose from z_i we generate the targets for the audio encoder as $\bar{z}_i = \frac{1}{2}\left(E_h(x_i^h) + E_h(\hat{x}_i^h)\right)$, where $\hat{x}_i^h$ is a horizontally flipped version of the image x_i^h. As it turns out, due to the disentangled representations of G, the

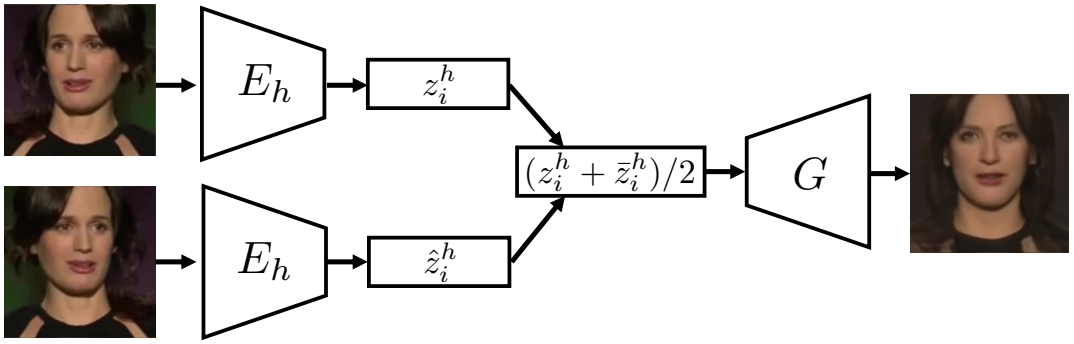


FIGURE 5.5: Illustration of how we compute the targets for the audio encoder pre-training. We feed a high-resolution training image and its horizontally flipped version through the high-resolution encoder. The resulting latent codes are then averaged and used as targets. Because of the hierarchical structure of the latent space of StyleGAN, the averaged latent code produces a face in the neutral frontal facing pose.

reconstruction $G(\bar{z}_i)$ produces a neutral frontal facing version of $G(z_i)$ (see Fig. 5.5). The audio encoder E_a is finally trained by minimizing

$$\min_{E_a} \sum_{a_i, x_i^h \in \mathcal{D}_{\text{pre}}} |E_a(a_i) - \bar{z}_i|_1 \,. \tag{5.3}$$

5.2.6 Fusing Audio and Low-Resolution Encodings

We now want to fuse the information provided by the pre-trained encoders E_l and E_a. Since the low-resolution encoder E_l already provides a good approximation to E_h, it is reasonable to use it as a starting point for the final prediction. Conceptually, we can think of E_l as providing a $z_i^l = E_l(x_i^l)$ that results in a canonical face $G(z_i^l)$ corresponding to the low-resolution image x_i^l. Ambiguities in z_i^l could then possibly be resolved via the use of audio, which would provide an estimate of the residual $\Delta z_i = z_i - z_i^l$. We therefore model the fusion mechanism as $z_i^f = E_l(x_i^l) + F\big(E_l(x_i^l), E_a(a_i)\big)$, where F is a simple fully-connected network acting on the concatenation of $E_l(x_i^l)$ and $E_a(a_i)$. Since the audio-encoding E_a might be sub-optimal for the fusion, we continue training it along with F. The limited complexity of the function F prevents the overfitting to the low-resolution encoding, but provides the necessary context for the computation of Δz_i. To summarize, we train the fusion by optimizing

$$\min_{E_a, F} \sum_{a_i, x_i^h, x_i^l \in \mathcal{D}} \left| z_i^f - z_i \right|_1 + \lambda \left| D \circ G\left(z_i^f\right) - x_i^l \right|_1 . \tag{5.4}$$

5.2.7 Implementation Details

The style-based generator G was pre-trained on the entire training set $\mathcal{D}$ with all hyper-parameters set to their default values (see [226] for details). It has seen a total of 31 million images. The high-resolution encoder E_h was trained for 715K iterations and a batch-size of 128 on the 128×128 images from $\mathcal{D}$. The low-resolution encoder E_l and the audio encoder E_a were trained on $\mathcal{D}_{\text{pre}}$. E_l was trained for 240K iterations with a batch-size of 256, and E_a was trained for 200K iterations and a batch-size of

64. The inputs x_i^l to E_l are of size 8×8 pixels and the inputs to E_a are the audio log-spectrograms of a_i of size 257×257. The fine-tuning of E_a and the training of the fusion layer F was performed for 420K iterations on $\mathcal{D}$. We used the Adam optimizer [221] with a fixed learning rate of 10^{-4} for the training of all the networks.

We provide details of the used network architectures in Tables 5.4 to 5.6. All the networks are convolutional using strided convolutions to reduce the spatial resolution. We apply instance normalization [238] to both the high-resolution encoder E_h and the low-resolution encoder E_l. Notice that we also process the audio spectrogram using a CNN architecture. However, we found that applying instance normalization to the audio-encoder E_a leads to significantly worse performance. Consequently, no normalization was applied for E_a. We use the leaky ReLU activation function in all our networks with a leak of 0.2.

To train the high-resolution encoder E_h, we used a perceptual loss on features of an ImageNet pre-trained VGG16 network. We extracted features from the outputs of the layers `conv1_1`, `conv1_2`, `conv3_2` and `conv4_2`.

The fusion network F consists of three fully-connected layers, each with a hidden dimension of 6'144. We again applied leaky ReLU activations in the hidden layers and did not use any normalization.

All networks were trained with multi-GPU training on 4 NVIDIA GTX 1080Ti GPUs.

5.3 Experiments

We demonstrate our contributions by evaluating three models with different input-output mappings: 1) Audio to high-resolution image; 2) Low-resolution image to high-resolution image; 3) Audio and low-resolution image to high-resolution image. In particular, we focus our attention on the third case as it is the main objective of this paper.

5.3.1 Dataset

We performed all our experiments on a subset of the VoxCeleb2 dataset [239]. The dataset contains over one million audio tracks extracted from 145K videos of people speaking. For the whole training set $\mathcal{D}$ we selected 104K videos with 545K audio tracks and extracted around 2M frames at 128×128 pixels such that each speaker has at least 500 associated frames. We then extracted half of this dataset to create $\mathcal{D}_{\text{pre}}$ in such a way that $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}$ contain the same speakers, but $\mathcal{D}_{\text{pre}}$ has fewer videos than $\mathcal{D}$. We selected 39K frames and 37K utterances from 25K videos not contained in the training set (again from the same speakers) for the test set. In the end, we select around 4K speakers out of the 6K speakers in the entire dataset (filtering out speakers with very few videos and audio tracks). Note that this selection is purely done to allow the evaluation via a speaker identity classifier. We call experiments **closed set** when the training and test sets share the same set of face identities; instead, we call them **open set** when the test set has identities that were not in the training set.

5.3.2 Audio-Only to High-Resolution Face

Although our main objective is to obtain super-resolved images from the fusion of low-resolution images and audio, we provide a brief comparison between our model for face reconstruction from audio ($E_a + G$) with Speech2Face [4]. Since

FIGURE 5.6: **Audio-to-Image.** To demonstrate qualitatively the capabilities of our audio-to-image model $E_a + G$ we picked several audio tracks and the corresponding generated faces by Oh *et al.* [4] from https://speech2face.github.io/supplemental/retrieve/index.html. Images in every column are generated from the same audio sources. Oh *et al.* [4] is shown on the first row, and our results on the second row.

the dataset of [4] is not public, we performed a qualitative and a quantitative comparison based on audio tracks and reconstrucitons by Oh *et al.* [4] from https://speech2face.github.io/supplemental/retrieve/index.html. In Fig. 5.6 we show the reference faces obtained by Speech2Face and our output using the same audio tracks. We can see that the gender and age match. In the second evaluation, we perform gender classification on the output of our audio-to-image model when given audio from the VoxCeleb dataset [239] as input. Given a voice of the male or female person, our $E_a + G$ model generates faces of males and females in 97% and 96% of the cases, respectively. The results match those reported by [4]. Notice that [4] uses supervision from a classifier during training while our training is completely unsupervised.

FIGURE 5.7: **Selected examples of reconstructions to some of our ablation experiments.** We show selected examples of reconstructions to some of our ablation experiments. The 8×8 pixels low-resolution inputs are shown in (a) and the corresponding 128×128 pixels ground truth images are shown in column (f). In-between, we show results for encodings from E_h in (b), E_l in (c), E_a in (d) and from our fusion model F with fine-tuned E_a in (e).

Ablation	**Acc** C_i	**Acc** C_g	**Err** C_a	**Acc** C_i	**Acc** C_g	**Err** C_a
	Closed Set			**Open Set**		
(a) E_h + fixed G	34.31%	95.60%	3.59	29.42%	92.65%	3.28
(b) E_h + tuned G	71.62%	98.20%	2.85	64.95%	95.14%	2.74
(c) E_l only	36.47%	95.51%	3.62	15.55%	91.08%	3.76
(d) E_a only	26.06%	97.07%	4.29	0.20%	**96.38**%	4.85
(e) F_1 + tuned E_a	35.91%	95.88%	3.56	15.03%	91.75%	**3.64**
(f) F + zero E_a	36.95%	95.53%	3.60	15.38%	90.89%	3.73
(g) F + fixed E_a	48.43%	97.17%	3.46	14.57%	92.86%	3.74
(h) F + tuned E_a	**51.65%**	**97.32%**	**3.31**	**15.67%**	93.11%	3.68

TABLE 5.1: Results of our ablation experiments. We report the accuracy of an identity classifier C_i and a gender classifier C_g as well as the error of an age classifier C_a on generated high-resolution images. All the models in (c)-(h) were trained using the fine-tuned generator G.

5.3.3 Identity, Gender and Age Classification Accuracy as a Performance Measure

To evaluate the capability of our model to recover gender and other identity attributes based on the low-resolution and audio inputs, we propose to use the accuracy of a pre-trained identity classifier C_i and gender classifier C_g, which achieve an accuracy of 95.25% and 99.53% respectively on the original high-resolution images. To this end, we fine-tune two VGG-Face CNNs of [240] on the training set $\mathcal{D}$ for 10 epochs on both face attributes. As one can see in Table 5.1 these classifiers perform well on the test set on both face attributes. Although we do not have the ground truth age of our dataset, we use a pre-trained age classifier C_a [241] as the reference. Then, we measure the performance of our models by checking the consistency between the classified age of the input and the output.

5.3.4 Ablations

We performed ablation experiments to understand the information retained in the encoders and justify our final model's design. The accuracy of the classifiers C_i and C_g, as well as the consistency error of C_a, are reported in Table 5.1 for the following ablation experiments:

(a)-(b) The importance of fine-tuning: In (a) we show the performance after pre-training of E_h without fine-tuning, and in (b) we show the improvement in performance with the fine-tuning of G as in eq. (5.2).

(c)-(d) Individual components: Shows the performance of the individual encoders without fusion. Results for the low-resolution encoder E_l and the audio encoder E_a should be compared to the reference high-resolution encoder E_h.

(e)-(h) Fusion strategies: The performance of different fusion strategies are reported. As a reference, we report results of a fusion model F_1 with a single fully-connected layer and fine-tuning of E_a. We compare this to a more complex fusion network F with three fully-connected layers when the audio is not used (f), the audio encoder is fixed (g), and when fine-tuning E_a (h).

We can observe that E_a can predict the correct gender more often than E_l. All the fusion approaches lead to an improvement in terms of identity prediction over E_a and E_l alone, thus showing that the information from both inputs is successfully integrated. Note that the performance of all methods in Table 5.1, including the SotA [242], is lower in the open set experiments than in the closed set ones. This is expected since all methods were trained only on identities present in the training set, and most likely, only a small amount of information is shared across identities. The open set experiments show how much the methods can identify such shared information, which is a sign of generalization. See also Fig. 5.7 for qualitative results.

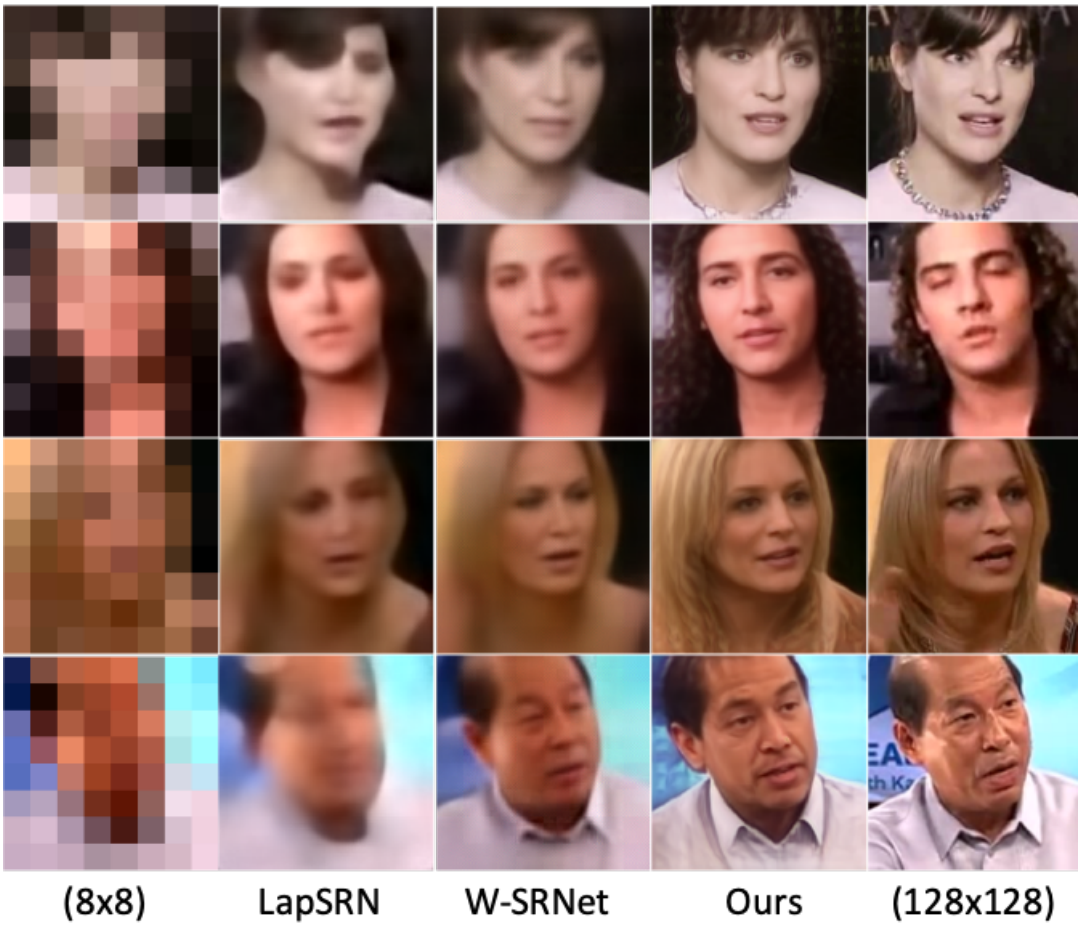


FIGURE 5.8: **Comparison to other super-resolution methods on our test set.** The first column shows the 8×8 pixels inputs; the second column shows the output of LapSRN [103]; the third column shows the output of W-SRNet [155]. Our model is shown in the fourth column. The ground-truth high-resolution image is shown in the last column.

		Closed Set					Open Set				
Method	**Factor**	**PSNR**	**SSIM**	**Acc C_i**	**Acc C_g**	**Err C_a**	**PSNR**	**SSIM**	**Acc C_i**	**Acc C_g**	**Err C_a**
LapSRN ([103])	4×	31.99	0.91	93.83%	99.38%	2.81	31.66	0.91	95.84%	95.37%	2.81
LapSRN ([103])	16×	22.75	0.64	5.27%	83.27%	5.16	22.39	0.62	6.80%	79.57%	5.16
W-SRNet ([155])	16×	21.55	0.67	34.91%	95.68%	4.28	19.18	0.59	13.54%	89.45%	4.57
Ours	16×	21.64	0.68	51.65%	97.32%	3.31	19.97	0.60	15.67%	93.11%	3.68

TABLE 5.2: Comparison to other general-purpose super-resolution methods at different super-resolution factors. We report PSNR and SSIM obtained on the test set. Note that the target resolution is fixed at 128×128 pixels and therefore the inputs to the 4× methods is 32×32 pixels while our model only uses 8×8 pixels input images.

5.3.5 Comparisons to Other Super-Resolution Methods

We compare to state-of-the-art super-resolution methods in Table 5.2 and Fig. 5.8. The standard metrics PSNR and SSIM, along with the accuracy of C_i and C_g, and the

FIGURE 5.9: **Low-Resolution and audio mixing.** Examples where we mix a given low-resolution image with different audio sources. The top row shows the high-resolution images from which we take the audio track. The first two columns on the left show the same high-resolution images and the corresponding low-resolution images used as input. The rest of the images in the matrix are generated by mixing the low-res from a row with the audio of a column.

Label Source	Closed Set	Open Set
Audio	10.76%	13.74%
Low-Resolution Image	89.24%	86.26%

TABLE 5.3: Agreement of C_g predictions with labels of low-resolution and audio labels on mixed reconstructions.

errors of C_a are reported for super-resolved images of our test set. Note that most methods in the literature are not trained on extreme super-resolution factors of 16×, but rather on factors of 4×. Therefore, we report the results of one method using a factor of 4× as a reference for the changes with the 16× factor. We retrained the methods of [103] and [155] on our training set before evaluating their performance. Notice that although LapSRN trained on 16× super-resolution performs better in terms of PSNR and SSIM than our method, the quality of the recovered image is clearly worse (see Fig. 5.8). This difference in the quality is instead revealed by evaluating the gender and identity classification accuracies and the age classification error of the restored images. This suggests that while PSNR and SSIM may be suitable metrics to evaluate reconstructions with small super-resolution factors, they may not be suitable to assess the reconstructions in more extreme cases, such as with a factor of 16×.

FIGURE 5.10: Examples of failure cases in our method. The 8×8 pixels low-resolution inputs are shown in (a) and the corresponding 128×128 pixels ground truth images are shown in column (e). In the middle, we show results for encodings from the high-resolution encoder E_h in (b), the low-resolution encoder E_l in (c) and from our fusion model F with the fine-tuned E_a in (d).

5.3.6 Editing by Mixing Audio Sources

Our model allows us to influence the high-resolution output by interchanging the audio tracks used in the fusion. To demonstrate this capability, we show examples where we mix a fixed low-resolution input with several different audio sources in Fig. 5.9. We provide some additional qualitative results of our mixing experiments in Fig. 5.12. These results were computed on images and audio tracks of identities that were not included in the training set (open set). Results computed on images and audio tracks from the closed test set are shown in Figs. 5.11, 5.13 and 5.14. To also quantitatively evaluate such mixing, we feed low-resolution images and audios from persons of different gender and classify the gender of the resulting high-resolution faces. In Table 5.3, we report the accuracy with respect to the ground-truth gender labels of low-resolution images and audios.

5.3.7 Failure Cases

We observe that failures may correspond more to the inherent bias presented in the training set than the training algorithm or network architecture. Failure cases sometimes happen when the gender can be easily guessed just from the low-resolution image. Some of the failure cases are reported in Fig 5.10.

5.4 Discussion

We have introduced a new paradigm for face super-resolution, where also audio contributes to the restoration of missing details in the low-resolution input image. We have described the design of a neural network and the corresponding training procedure to successfully use the audio signal despite the difficulty of extracting visual information from it. We have also shown quantitatively that audio can improve the accuracy of the restored face's identity, gender, and age. Moreover, we

FIGURE 5.11: **Mixing examples.** We show examples where we mix a given low-resolution image with different audio sources. The top row shows the high-resolution images from which we take the audio track. The column on the left show the corresponding low-resolution images used as input. The rest of the images in the matrix are generated by mixing the low-res from a row with the audio of a column.

have shown that it is possible to mix low-resolution images and audios from different videos and obtain semantically meaningful high-resolution images. A fundamental challenge in this work was the fusion of information from these very different modalities. As we have shown, valuable information is present in both. However, we observed that naive end-to-end training tends to ignore audio information. We conjecture that this problem might be a fundamental issue with current training schemes of neural networks, and its solution could provide insights on how to improve the training on tasks in general.

FIGURE 5.12: **Mixing examples.** We show examples where we mix a given low-resolution image with different audio sources. The top row shows the high-resolution images from which we take the audio track. The column on the left show the corresponding low-resolution images used as input.

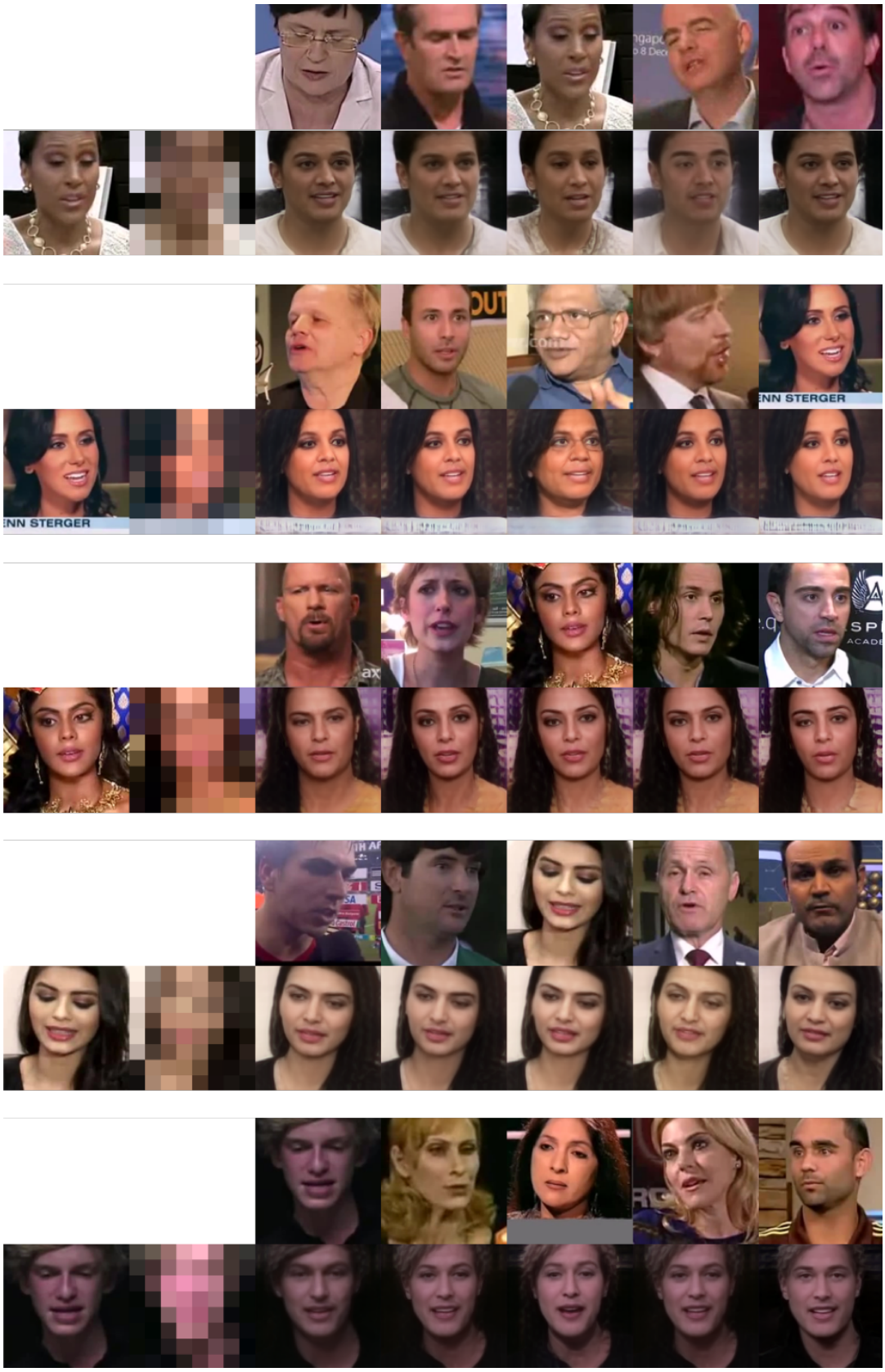


FIGURE 5.13: **Mixing examples.** We show examples where we mix a given low-resolution image with different audio sources. The top row shows the high-resolution images from which we take the audio track. The column on the left show the corresponding low-resolution images used as input.

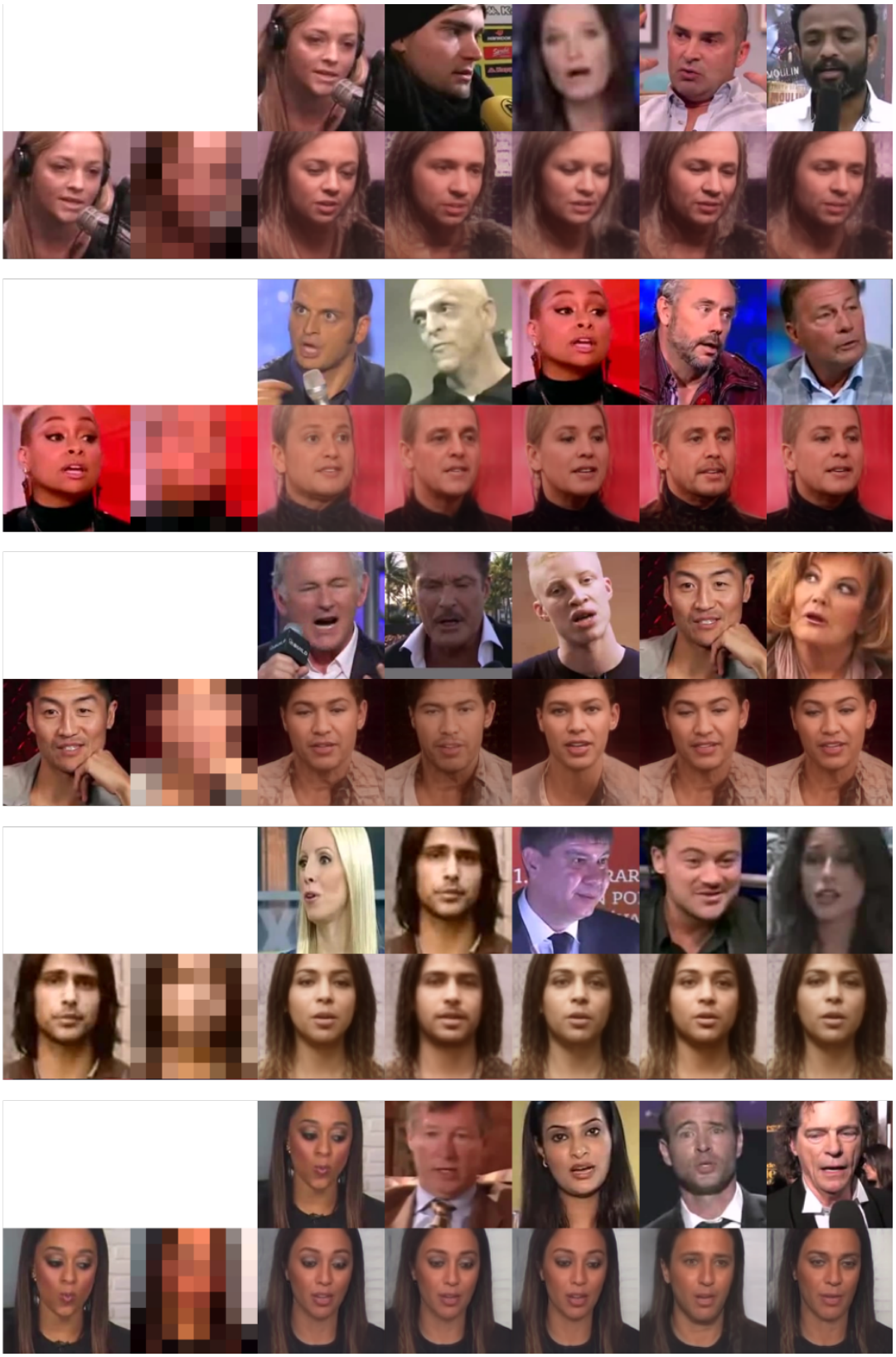


FIGURE 5.14: **Mixing examples.** We show examples where we mix a given low-resolution image with different audio sources. The top row shows the high-resolution images from which we take the audio track. The column on the left show the corresponding low-resolution images used as input.

The network architecture of the low-resolution encoder E_l					
Layer	Kernel	Stride	Norm.	Activation	# Filters
conv	3×3	1	-	`lReLU`	128
conv	3×3	2	IN	`lReLU`	128
conv	3×3	1	IN	`lReLU`	256
conv	3×3	2	IN	`lReLU`	256
conv	3×3	1	IN	`lReLU`	512
conv	3×3	2	IN	`lReLU`	512
dense	-	-	-	`lReLU`	6144
dense	-	-	-	`linear`	6144

TABLE 5.4: Images are assumed to be of size 8×8. The output size of 6144 matches the targets z_i.

The network architecture of the high-resolution encoder E_h					
Layer	Kernel	Stride	Norm.	Activation	# Filters
conv	4×4	1	-	`lReLU`	64
conv	4×4	2	IN	`lReLU`	64
conv	4×4	1	IN	`lReLU`	128
conv	4×4	2	IN	`lReLU`	128
conv	4×4	1	IN	`lReLU`	256
conv	4×4	2	IN	`lReLU`	256
conv	4×4	1	IN	`lReLU`	512
conv	4×4	2	IN	`lReLU`	512
conv	4×4	1	IN	`lReLU`	1024
conv	4×4	2	IN	`lReLU`	1024
conv	4×4	1	IN	`lReLU`	1024
conv	4×4	2	IN	`lReLU`	1024
dense	-	-	-	`linear`	6144

TABLE 5.5: Input images are of size 128×128. The output size of 6144 matches the input input of the generator which is of size 12×512.

The network architecture of the audio encoder E_a					
Layer	Kernel	Stride	Norm.	Activation	# Filters
conv	4×4	2	-	`lReLU`	64
conv	4×4	1	-	`lReLU`	64
conv	4×4	2	-	`lReLU`	64
conv	4×4	1	-	`lReLU`	128
conv	4×4	2	-	`lReLU`	128
conv	4×4	1	-	`lReLU`	256
conv	4×4	2	-	`lReLU`	256
conv	4×4	1	-	`lReLU`	512
conv	4×4	2	-	`lReLU`	512
conv	4×4	1	-	`lReLU`	1024
conv	4×4	2	-	`lReLU`	1024
conv	4×4	1	-	`lReLU`	2048
conv	4×4	2	-	`lReLU`	2048
dense	-	-	-	`lReLU`	8192
dense	-	-	-	`linear`	6144

TABLE 5.6: The input spectrograms are of size 257×257. The output size of 6144 matches the targets z_i.

Chapter 6

Contrastive Learning for Controllable Blind Video Restoration

FIGURE 6.1: **Controllable Blind Video Restoration.** Given a low resolution and degraded input video, our model can be used to denoise and/or upscale. We automatically estimate the degradation present in the image. Moreover, it is possible to manipulate the degradation representation to control the restoration result and for example increase sharpness.

With the development of video streaming services and the increased competition between the different providers in terms of catalog size, there is a regain of interest for the studios to remaster old shows and productions to make them available on their streaming platform. Our work addresses the problem of video restoration in this context of remastering legacy video content. This type of content is often available in noisy, blurry, and low-resolution format and may contain scratches. Because of this combination of degradations, the remastering process has to be carefully engineered to produce the best results and, more importantly, avoid exaggerating any of the degradations in the image.

Recent developments in deep learning have pushed the state of the art in the subproblems independently, by exploring different architectures or training settings in super-resolution [96], [97], [104], [105], [109], [136] and video denoising [166], [243].

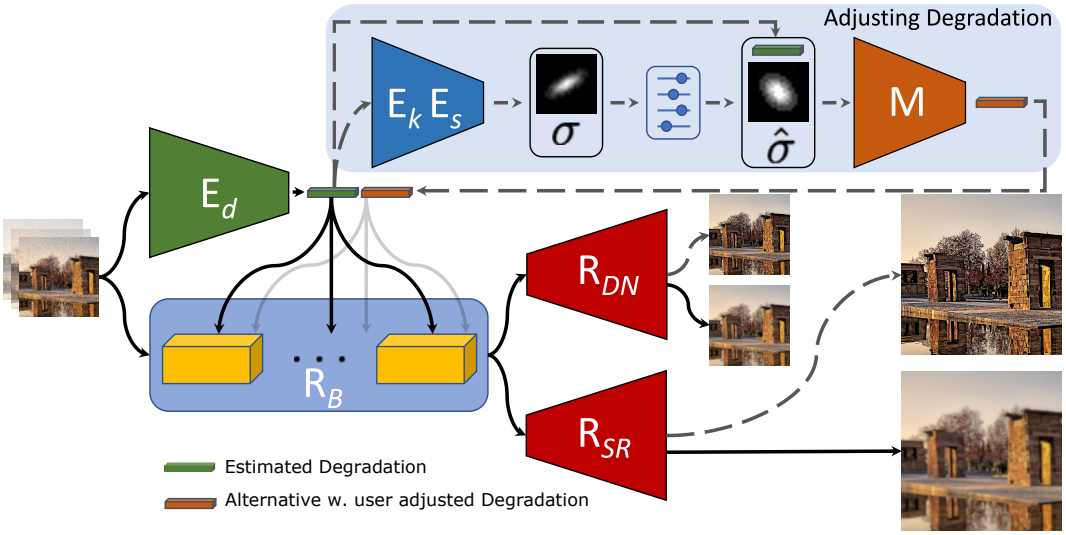


FIGURE 6.2: **Overview of our controllable restoration pipeline.** We first estimate the degradation feature by feeding the corrupted video to the encoder E_d. The degradation feature is used as conditioning for the restoration backbone R_B. The same features are used both for the Super-Resolution head SR and denoising head DN. It is possible to adjust both the denoising strength and blur kernel: E_k and E_s explicitly map back the latent representation to a blur kernel and noise levels, that can be adjusted before using the mutator M to output the corresponding degradation embedding. This mutated version of the embedding (in orange) can similarly be used as conditioning for the restoration. It corresponds to the alternative outputs indicated by the dotted arrows.

These specialized tools can be chained to denoise and upscale video content. However, sequentially applying different restoration methods may lead to sub-optimal results and increased computation costs, which has motivated jointly addressing multiple restoration problems [244]. Additionally, we can mention the blind restoration methods [126], [150] where the parameters of the degradation, such as the blur kernel, are estimated. More recently, Wang *et al.* [154] proposed an unsupervised degradation representation learning scheme for blind super-resolution without explicit degradation estimation. Although this provides a clear advantage over *test-time optimization* methods, the proposed models are limited to images, fixed scaling factors and the learned representation cannot be interpreted or manipulated. Here we propose a pipeline that is designed for video quality enhancement of older content that contains a combination of degradations. A brief overview of our method during inference is presented in Fig. 6.2. Our proposed method consists of three major steps: *(i)* extracting an interpretable and controllable representation of different degradations; *(ii)* manipulating the degradations if necessary; *(iii)* finally conditioning the restoration backbone with estimated/manipulated degradation embedding. To the best of our knowledge, there is no solution that considers the full problem of video restoration that takes into account: scratch removal, denoising and upscaling, while offering flexibility in terms of manual fine-tuning for the restoration of the signal. Our training strategy leverages contrastive learning to learn an abstract representation that distinguishes various degradations in the representation space rather than explicit estimation in the pixel space. A key difference from Wang *et al.* [154]

is the possibility to control the restoration process via manipulating the degradation features. This requires better estimates for the degradation parameters, which is possible thanks to our training strategy using pairs of degraded training samples and hard negative samples. Finally, we consider a wider range of degradations and address video restoration in a general setting where super-resolution is not limited to a discrete set of scaling factors which is necessary when processing some video formats like NTSC. Our contributions can be summarized as follows:

- A video restoration model that can jointly address the most common degradation present in legacy content.
- A new contrastive training strategy to learn an interpretable and controllable representation of different degradations.
- State of the art results in blind video restoration.

6.1 Background

In Sections 2.5 and 2.6 we covered prior works on super-resolution and denoising respectively. Unlike prior works, our pipeline addresses different video restoration tasks simultaneously and allows fine-grained control of the restoration process. The closest works that also tackle the mixed degradation restoration problem are specifically designed for old content restoration. Old content is usually available in noisy, blurry, and low-resolution format and may contain scratches. In the next section, we briefly mention some of these works.

6.1.1 Scratch Removal

Scratch removal is a classical mixed degradation problem when working with old photo/video data, and most existing methods consider it an image inpainting problem [245]–[248]. Some works consider joint restoration of images corrupted by a combination of different distortions [249], [250]. Wan *et al.* [251] proposed a novel triplet domain translation network by leveraging real photos along with massive synthetic image pairs and trained two variational autoencoders (VAEs) to respectively transform old photos and clean photos into two latent spaces. And the translation between these two latent spaces is learned with synthetic paired data.

6.2 Method

We aim to build a model that can restore videos corrupted by the most common degradations present in legacy film content: scratches, noise, and the implicit blur in the low-resolution input. We can briefly formulate the degradation model of a LR video y as follows:

$$y = S \circ \Big((x * k) \downarrow_s + n \Big) \tag{6.1}$$

where x is the HR video, $*$ is convolution operation, k is a blur kernel, $\downarrow_s$ denotes downsampling operation by factor s, n stands for noise, and S represents a film scratch as a mask that sets pixel color values to 1.

As illustrated in Fig. 6.2, we train an encoder E_d capable of extracting a latent representation for the degradation present in the input frames. For this, we leverage recent advances in contrastive learning [36], [154]. This latent representation is then

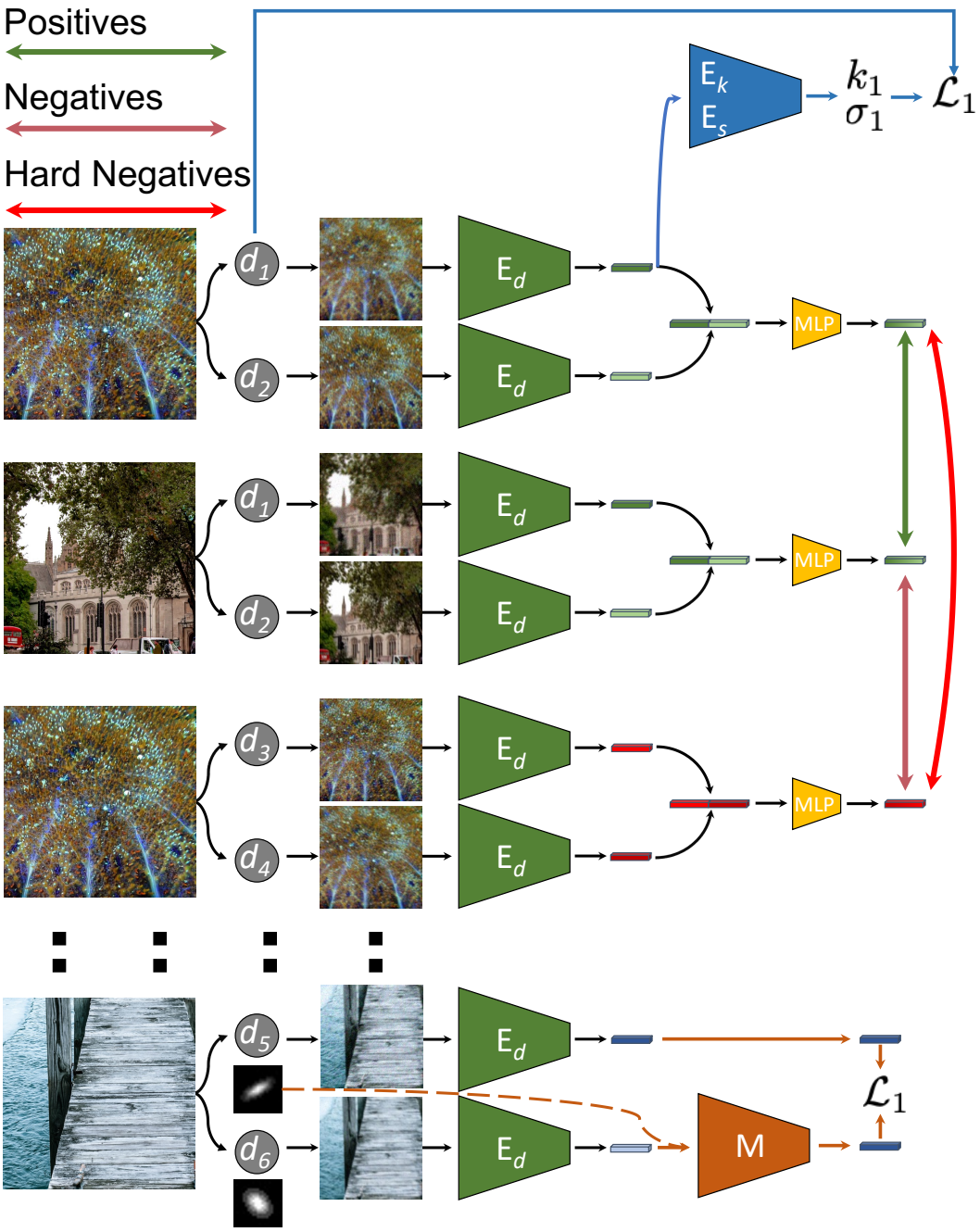


FIGURE 6.3: **Overview of our degradation learning pipeline.** We first degrade two high-resolution input images with a pair of degradations d_1, d_2. We encode low-resolution degraded image pairs using encoder E_d. Later features of the first and second rows are concatenated and passed to a two-layer MLP network. Final outputs connected with a green arrow form a positive pair for contrastive learning. A red feature from the third row creates a hard negative example for the feature from the first row since its obtained via encoding the same image corrupted with degradations d_3 and d_4. We additionally regress the blur kernel k_1 and noise level σ_1 via encoders E_k and E_s, respectively. We also learn to manipulate features using encoder M by supplying it with adjusted degradation parameters k_5, σ_5 and obtain $z_p^5 = M(z_p^6, k_5, \sigma_5)$.

used as a conditioning for the feature restoration backbone R_B, which is used both for low-resolution denoising, with R_{DN}, and the super-resolution path R_{SR}. Our model also learns to decode the degradation representation into blur kernel and noise levels. Furthermore, it is possible to modify these parameters and adjust the latent representation accordingly, thanks to the mutator model M. This flexibility is needed in the context of real-world applications where artists may want to adjust sharpness levels and denoising strength. In the following, we first present in section 6.2.1 our contrastive learning strategy, then our proposal to allow the manipulation of the learned latent representation (section 6.2.2). Finally, we take advantage of the learned representation to condition the restoration task in section 6.2.3.

6.2.1 Video Degradation Representation

The objective is to learn to extract from the input frames, a latent representation that should be discriminative towards different degradations that might be in the input. More precisely, two different videos similarly degraded should lead to two embeddings that are close to each other, while the two differently degraded versions of the same video should result in latent representations further apart. This is a more challenging objective than the one considered by Wang *et al.* [154] which is a more straightforward application of the Moco [36] representation learning framework: the loss was designed such as to push further away the embedding of patches from different images, while bringing closer patches from the same image. Such an objective doesn't encourage a clear disentanglement between the content and the degradation.

We are interested in disentangling the degradation from the content but different samples from the training set are captured with sensors of varying resolutions, exposures, and noise levels. Any high-resolution image already contains a certain amount of degradation and the application of the degradation model from Equation 6.1 will result in a mixture of two degradations: inherent from a high-resolution image and one from equation 6.1. Separating these two degradations is an ill-posed problem. Therefore, directly training the encoder E_d with a Multilayer Perceptron (MLP) that tries to optimize our contrastive learning objective is not optimal.

To address this issue, we propose to train the encoder E_d using pairs of degraded patches obtained from sampling a random high-resolution image and degrading it with two different degradations. Consequently, the MLP should focus on differences between degradations introduced during training rather than ones present in the original high-resolution video.

An overview of the training procedure is presented in Fig. 6.3. Let us denote a specific set of different degradations from equation 6.1 as $d_i \sim \mathcal{D}$ parameterized by blur kernel k_i and noise level σ_i, $y_p^i = d_i(x_p)$ as video x_p degraded with degradation d_i, and $z_p^i = E_d(y_p^i)$ as latent vector obtained by encoding y_p^i using encoder E_d. We sample pairs of degradations (d_i, d_j), (d_k, d_l), and videos x_p, x_q. We apply pairs of sampled degradations to the videos and encode them using encoder E_d:

$$x_p \rightarrow \Big(d_i(x_p), d_j(x_p)\Big) \rightarrow (y_p^i, y_p^j) \rightarrow (z_p^i, z_p^j) \tag{6.2}$$

$$x_q \rightarrow \Big(d_i(x_q), d_j(x_q)\Big) \rightarrow (y_q^i, y_q^j) \rightarrow (z_q^i, z_q^j) \tag{6.3}$$

$$x_p \rightarrow \Big(d_k(x_p), d_l(x_p)\Big) \rightarrow (y_p^k, y_p^l) \rightarrow (z_p^k, z_p^l) \tag{6.4}$$

where superscripts and subscripts denote degradations and input videos respectively. Note that embedding pairs (z_p^i, z_p^j) and (z_q^i, z_q^j) are obtained by degrading two different videos x_p and x_q, with the same pair of degradations (d_i, d_j). Therefore, they form a positive pair. Hard negative pairs (z_p^i, z_p^j) and (z_p^k, z_p^l) are obtained by degrading the same video x_p with different pairs of degradations: (d_i, d_j) and (d_k, d_l). We provide these difficult negative examples during training to force the neural representation to focus on the degradation rather than the content. Next, we define the relative degradations via concatenating the resulting embedding pairs and following the Moco framework feed them to a two-layer MLP projection head F: $\psi_p^{ij} = F\Big([z_p^i, z_p^j]\Big)$, $\psi_q^{ij} = F\Big([z_q^i, z_q^j]\Big)$, and $\psi_p^{kl} = F\Big([z_p^k, z_p^l]\Big)$. We want ψ_p^{ij} to be similar to ψ_q^{ij} since they share the same relative degradations and dissimilar to ψ_p^{kl} since degradations are different. Therefore, an InfoNCE loss is used to measure the similarity:

$$\mathcal{L}_c = \sum_{p,q}^{\mathcal{V}} \sum_{i,j}^{\mathcal{D}} -\log \frac{e^{\left(\psi_p^{ij} \cdot \psi_q^{ij} / \tau\right)}}{\sum_{t=1}^{N_Q} e^{\left(\psi_p^{ij} \cdot \psi_t / \tau\right)} + e^{\left(\psi_p^{ij} \cdot \psi_p^{kl} / \tau\right)}} \tag{6.5}$$

where N_Q is the number of samples in MoCo queue, $\mathcal{V}$ is a set of training videos, $\mathcal{D}$ is set of degradations, τ is a temperature parameter, and $\cdot$ denotes the dot product between two vectors.

In addition to optimizing for $\mathcal{L}_c$ we also estimate the parameters k_i and σ_i of applied degradation d_i from encoded feature z_p^i. We train a degradation regression that is able to recover the degradation parameters in a standardized format. This is important to allow the modification of the results and fine tuning of the outputs. Towards this goal, we train a small MLP: E_k and E_s that regress the parameters k_i and σ_i , by optimizing:

$$\mathcal{L}_k = \sum_{p}^{\mathcal{V}} \sum_{i}^{\mathcal{D}} \left| E_k \left(E_d \Big(d_i(x_p) \Big) \right) - k_i \right| \tag{6.6}$$

$$\mathcal{L}_s = \sum_{p}^{\mathcal{V}} \sum_{i}^{\mathcal{D}} \left| E_s \left(E_d \Big(d_i(x_p) \Big) \right) - \sigma_i \right| \tag{6.7}$$

where subscripts k and σ identify the specific output of the model E.

Overall training objective can be summarized as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_k \mathcal{L}_k + \lambda_s \mathcal{L}_s \tag{6.8}$$

where $\lambda_c = 1, \lambda_k = 400, \lambda_s = 1$ control the contribution of individual loss terms.

6.2.2 Learning to Manipulate Degradations

Our goal is to restore the distorted videos. However, we also want to have fine-grained control over this process. For example, one might need to correct the blur kernel, adjust the noise level and obtain the alternatively restored video. Therefore, we freeze the pre-trained encoder E_d and train the model M to perform manipulations in the latent space of degradations. Given the embedding $z_p^i = E_d\Big(d_i(x_p)\Big)$

and some new adjusted parameters k_j, σ_j, the model M enables the manipulations in the latent space and regresses the feature $z_p^j = M(z_p^i, k_j, \sigma_j)$. During training we sample video x_p, and a pair of degradations: d_i, d_j. Next, we degrade x_p obtaining $y_p^i = d_i(x_p)$ and $y_p^j = d_j(x_p)$. We compute encodings $z_p^i = E_d(y_p^i), z_p^j = E_d(y_p^j)$ using frozen encoder E_d. Finally, we train model M by minimizing the following objective:

$$\mathcal{L}_m = \sum_p^{\mathcal{V}} \sum_{i,j}^{\mathcal{D}} \left| M(z_p^i, k_j, \sigma_j) - z_p^j \right| \tag{6.9}$$

6.2.3 Learning Conditional Restoration

As illustrated in Fig. 6.2, the proposed model extracts from consecutive frames an encoding of the degradation that is present in the video. This degradation, expressed as a latent vector, is then used as conditioning for the restoration. Formally our model consists of restoration backbone R_B and two task-specific branches: R_{SR} and R_{DN} for super-resolution and denoising, respectively. The motivation for having a shared back-bone R_B is to learn features that are beneficial for different restoration tasks simultaneously. While the networks R_{SR} and R_{DN} should learn features tailored for super-resolution, denoising, and scratch removal, respectively.

Given a corrupted input y_p^i we first obtain the corresponding degradation embedding $E_d(y_p^i)$. We pass both y_p^i and $E_d(y_p^i)$ to the restoration backbone R_B. Consequently, the resulting final feature map from R_B is fed to R_{SR} and R_{DN} subnetworks, respectively. Therefore, we produce two outputs in this model. The first is the low-resolution denoised image and consequently the original low-resolution noise. The second is the denoised high-resolution image. Rather than outputting a fixed $4\times$ super-resolved frame, we employ Meta Upscale module [252] at the end of our R_{SR} model to enable non-integer upsampling factors and address more general scenarios. Additionally, for both super-resolution and denoising branches, our model must also remove the possible scratches presented in the video. Hence in addition to the losses mentioned in the section 6.2.1, during training models R_{SR} and R_{DN} are trained to minimize objectives $\mathcal{L}_{SR}$ and $\mathcal{L}_{DN}$ respectively.

$$\mathcal{L}_{SR} = \sum_p^{\mathcal{V}} \sum_i^{\mathcal{D}} \left| R_{SR}\left(E_d(y_p^i), y_p^i\right) - \hat{x}_p \right| \tag{6.10}$$

$$\mathcal{L}_{DN} = \sum_p^{\mathcal{V}} \sum_i^{\mathcal{D}} \left| R_{DN}\left(E_d(y_p^i), y_p^i\right) - \left(\hat{x}_p * k_i\right) \downarrow_s \right| \tag{6.11}$$

where $y_p^i = d_i(x_p)$ and $\hat{x}_p$ corresponds to the degraded video and middle sharp high-resolution ground-truth frame of the video sequence x_p. Additionally to the content losses mentioned in equations 6.10 and 6.11, we also keep fine-tunning the models from section 6.2.1. Therefore our final objective becomes:

$$\mathcal{L} = \lambda_{SR}\mathcal{L}_{SR} + \lambda_{DN}\mathcal{L}_{DN} + \lambda_c\mathcal{L}_c + \lambda_k\mathcal{L}_k + \lambda_s\mathcal{L}_s \tag{6.12}$$

where $\lambda_{SR} = 1$ and $\lambda_{DN} = 1$ are the weights of super-resolution and denoising terms respectively.

6.2.4 Implementation Details

We train our models on the Vimeo90K dataset[133]. This dataset consists of 89,800 video clips, which cover a large variety of scenes and actions. During training, we randomly sample 5 consecutive 192x192 frames from the dataset. We employed R3D-18 [253] architecture as a backbone for the encoder E_d. Before contrasting, the output features of encoder E_d are fed to two fully connected layers with 512 and 256 neurons, respectively. Contrastive *MLP* head consists of the layers presented in Table 6.5. Encoders E_k, E_s, and M are implemented using fully-connected perceptrons, presented in Tables 6.6 to 6.8, respectively. R_b, R_{SR}, and R_{DN} models consist of Degradation-aware (DA) blocks introduced by Wang *et al.* [154]. The high-level structure is borrowed from the RCAN [254] model. Model R_B starts with three 3d convolutional layers to leverage the temporal information presented in the input. Next, the aggregated feature is processed using 5 DA [154] blocks. Finally, we give the output of the last block to models R_{SR} and R_{DN} as input. We pass the input feature map from model R_B to 2 consecutive DA blocks[154]. The output of the last DA block is summed with the middle frame feature of the first convolutional layer of model R_B. Finally, we get the super-resolved output using the commonly used Pixel-Shuffle layer [255]. Alternatively, instead of Pixel-Shuffle, we employ Meta Upscale module [252] at the end of our R_{SR} model to enable non-integer upsampling factors and address more general scenarios. We pass the input feature map from model R_B to 2 consecutive DA blocks[154]. The output of the last DA block is summed with the middle frame feature of the first convolutional layer of model R_B. Finally, we get the denoised output using the last conv_2d layer.

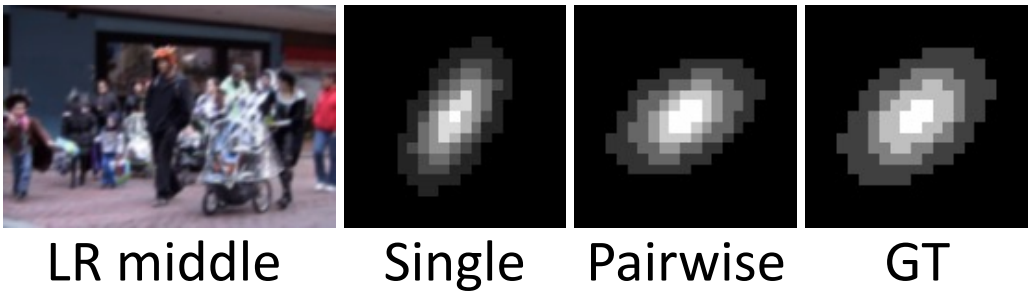
Our models were trained on 2 NVIDIA TITAN X GPUs with a mini-batch size of 32 samples. In our experiments, we used $\tau = 0.07$ and $N_Q = 8192$, respectively. Degradation encoder E_d was pre-trained for 65 epochs prior to the final fine-tuning together with models R_b, R_{SR}, and R_{DN} for additional 40 epochs. We use the Adam optimizer [221] with an initial learning rate of 10^{-4} for the training of all the networks.

6.3 Experiments

Our pipeline is simultaneously addressing multiple restoration problems. Therefore, we compare with some of the task-specific recent prior works. Specifically, we perform comparisons on video super-resolution, denoising, and scratch removal tasks.

6.3.1 Datasets & Metrics

We incorporated the Vid4 and Set8 datasets for comparison and ablation purposes. We generated multiple degraded versions of original datasets to demonstrate the capabilities of our pipeline in different settings. First, we created multiple blurry versions of each dataset using nine blur kernels presented in Table 6.2. After, we downsampled and corrupted each of the blurry datasets using AWGN of different magnitudes. And finally, we followed the method of Wan *et al.* [251] to generate scratched versions of the datasets. For quantitative comparison purposes, we employ commonly used metrics in the field: PSNR and SSIM.



Feature Contrasting	MAE↓	Kernel Similarity↑
Single	0.0008	0.9438
Pairwise	**0.0005**	**0.9821**

TABLE 6.1: Kernel estimation accuracy for single and pairwise feature contrasting strategies. The first and second rows correspond to single and pairwise feature contrasting strategies, respectively. We report Mean Absolute Error and Kernel Similarity [256].

6.3.2 Ablations

In this section, we ablate single and pairwise contrasting strategies for encoder E_d. We also evaluate the quality of mutated kernels produced by our model M.

6.3.2.1 Single vs Pairwise Contrasting

An essential component of our pipeline is the encoder E_d that learns to map the degraded videos to the latent space. As we mentioned previously, features from the latent space should reflect as much information as possible about the degradation contained in the input video. Therefore, we evaluate the latent space via the quality of the blur kernels that the encoder E_k produces given a feature from E_d as input. We use Kernel Similarity [256] and Mean Absolute Error (MAE) as evaluation metrics between ground-truth and estimated kernels. We thus perform ablation experiments for different ways to train the encoder E_d and justify the choice of the pairwise training strategy. We consider two possible design choices: *(i)* training E_d by contrasting single video embeddings, and *(ii)* training E_d by contrasting pairs of video embeddings. MAE's and Kernel Similarities are reported in Table 6.1. One can observe that the Pairwise feature contrasting strategy leads to a better quality of the estimated kernels.

6.3.2.2 Initial vs Mutated Kernels

We also evaluated how well mutator M manipulates the latent input features. Specifically, we are interested in consistency between the input kernel to the model M and kernel-related information contained in the output manipulated latent code. Towards this goal, we first feed model M with a latent code, noise level, and adjusted blur kernel; and obtain the adjusted code. After, we provide the modified latent code to the Kernel estimator E_k and estimate the adjusted kernel. Finally, we measure the MAE and Kernel Similarity between the initial adjusted blur kernel and

	σ	Method	Blur Kernels									
												All
SET8	0	Ours	27.51 / 0.82	27.17 / 0.79	24.25 / 0.67	27.76 / 0.81	26.40 / 0.77	25.52 / 0.73	27.74 / 0.81	26.44 / 0.77	25.38 / 0.72	**26.46 / 0.77**
		Tian *et al.* [131]	29.22 / 0.84	26.20 / 0.75	24.20 / 0.67	27.38 / 0.79	25.66 / 0.74	24.94 / 0.70	27.49 / 0.80	25.80 / 0.74	25.02 / 0.70	26.21 / 0.75
		Pan *et al.* [257]	26.89 / 0.79	25.76 / 0.73	23.99 / 0.66	25.82 / 0.74	24.74 / 0.70	24.66 / 0.69	26.52 / 0.77	25.02 / 0.71	24.72 / 0.69	25.35 / 0.72
		Zhang *et al.* [258]	25.80 / 0.72	24.95 / 0.70	23.95 / 0.65	25.17 / 0.71	24.09 / 0.66	24.12 / 0.66	25.38 / 0.71	24.32 / 0.67	24.22 / 0.66	24.67 / 0.68
	5	Ours	27.94 / 0.82	27.74 / 0.79	24.74 / 0.68	28.22 / 0.81	26.81 / 0.76	26.06 / 0.73	28.10 / 0.81	26.82 / 0.77	26.10 / 0.73	**26.95 / 0.77**
		Tian *et al.* [131]	29.44 / 0.83	26.51 / 0.74	24.42 / 0.66	27.71 / 0.78	25.89 / 0.73	25.18 / 0.69	27.82 / 0.79	26.06 / 0.73	25.28 / 0.69	26.48 / 0.74
		Pan *et al.* [257]	27.18 / 0.78	26.07 / 0.73	24.22 / 0.65	26.11 / 0.74	24.97 / 0.70	24.90 / 0.68	26.85 / 0.76	25.28 / 0.71	24.98 / 0.69	25.62 / 0.72
		Zhang *et al.* [258]	26.16 / 0.72	25.27 / 0.69	24.21 / 0.65	25.49 / 0.71	24.35 / 0.66	24.40 / 0.66	25.73 / 0.71	24.64 / 0.66	24.52 / 0.66	24.97 / 0.68
	10	Ours	28.51 / 0.81	28.14 / 0.78	25.26 / 0.68	28.58 / 0.80	27.13 / 0.75	26.53 / 0.73	28.52 / 0.80	27.35 / 0.76	26.72 / 0.73	**27.42 / 0.76**
		Tian *et al.* [131]	29.01 / 0.79	26.68 / 0.71	24.61 / 0.63	27.74 / 0.75	26.03 / 0.69	25.36 / 0.66	27.80 / 0.75	26.17 / 0.70	25.47 / 0.66	26.54 / 0.70
		Pan *et al.* [257]	27.32 / 0.77	26.35 / 0.71	24.45 / 0.64	26.35 / 0.73	25.21 / 0.68	25.14 / 0.67	27.06 / 0.75	25.53 / 0.70	25.23 / 0.67	25.85 / 0.70
		Zhang *et al.* [258]	26.51 / 0.72	25.61 / 0.68	24.35 / 0.63	25.86 / 0.70	24.72 / 0.65	24.69 / 0.65	26.07 / 0.70	24.99 / 0.66	24.81 / 0.65	25.29 / 0.67
	15	Ours	28.54 / 0.81	28.26 / 0.77	25.55 / 0.68	28.61 / 0.79	27.35 / 0.75	26.78 / 0.72	28.60 / 0.79	27.48 / 0.75	26.92 / 0.72	**27.57 / 0.75**
		Tian *et al.* [131]	28.17 / 0.75	26.49 / 0.67	24.58 / 0.59	27.31 / 0.70	25.87 / 0.65	25.30 / 0.62	27.35 / 0.71	25.99 / 0.66	25.40 / 0.62	26.27 / 0.66
		Pan *et al.* [257]	27.15 / 0.75	26.35 / 0.69	24.51 / 0.61	26.33 / 0.71	25.27 / 0.67	25.20 / 0.64	26.98 / 0.73	25.58 / 0.68	25.29 / 0.64	25.85 / 0.68
		Zhang *et al.* [258]	26.62 / 0.71	25.77 / 0.67	24.44 / 0.62	26.05 / 0.69	24.98 / 0.65	24.88 / 0.64	26.22 / 0.69	25.20 / 0.65	25.00 / 0.64	25.46 / 0.66
	25	Ours	27.44 / 0.79	27.52 / 0.75	25.51 / 0.67	27.63 / 0.77	26.82 / 0.73	26.48 / 0.71	27.65 / 0.78	26.88 / 0.74	26.55 / 0.71	**26.94 / 0.74**
		Tian *et al.* [131]	25.89 / 0.65	25.16 / 0.57	23.86 / 0.50	25.58 / 0.61	24.72 / 0.56	24.39 / 0.53	25.61 / 0.61	24.80 / 0.56	24.44 / 0.53	24.94 / 0.57
		Pan *et al.* [257]	25.97 / 0.70	25.44 / 0.61	24.05 / 0.54	25.52 / 0.66	24.74 / 0.62	24.61 / 0.57	25.92 / 0.66	25.00 / 0.63	24.66 / 0.57	25.10 / 0.61
		Zhang *et al.* [258]	25.96 / 0.70	25.41 / 0.65	24.30 / 0.60	25.63 / 0.67	24.85 / 0.64	24.73 / 0.62	25.72 / 0.68	24.97 / 0.64	24.79 / 0.62	25.15 / 0.65
	All	Ours	27.99 / **0.81**	**27.77 / 0.78**	**25.06 / 0.68**	**28.16 / 0.80**	**26.90 / 0.75**	**26.27 / 0.72**	**28.12 / 0.80**	**26.99 / 0.76**	**26.33 / 0.72**	**27.07 / 0.76**
		Tian *et al.* [131]	**28.35** / 0.77	26.21 / 0.69	24.33 / 0.61	27.14 / 0.73	25.63 / 0.67	25.03 / 0.64	27.21 / 0.73	25.76 / 0.68	25.12 / 0.64	26.09 / 0.68
		Pan *et al.* [257]	26.90 / 0.76	25.99 / 0.69	24.24 / 0.62	26.03 / 0.72	24.99 / 0.67	24.90 / 0.65	26.67 / 0.73	25.28 / 0.69	24.98 / 0.65	25.55 / 0.68
		Zhang *et al.* [258]	26.21 / 0.71	25.40 / 0.68	24.25 / 0.63	25.64 / 0.70	24.60 / 0.65	24.56 / 0.65	25.82 / 0.70	24.82 / 0.66	24.67 / 0.65	25.11 / 0.67
VID4	0	Ours	22.84 / 0.73	23.49 / 0.71	21.11 / 0.53	23.41 / 0.73	22.60 / 0.68	22.44 / 0.64	23.53 / 0.73	22.63 / 0.66	21.99 / 0.61	**22.67 / 0.67**
		Tian *et al.* [131]	24.62 / 0.77	22.17 / 0.62	20.79 / 0.51	23.17 / 0.69	21.86 / 0.61	21.32 / 0.55	23.09 / 0.68	21.82 / 0.60	21.32 / 0.55	22.24 / 0.62
		Pan *et al.* [257]	22.82 / 0.69	21.88 / 0.59	20.66 / 0.50	21.97 / 0.62	21.11 / 0.56	21.12 / 0.54	22.37 / 0.64	21.27 / 0.57	21.12 / 0.53	21.60 / 0.58
		Zhang *et al.* [258]	22.24 / 0.62	21.65 / 0.59	20.82 / 0.53	21.80 / 0.61	20.91 / 0.56	20.96 / 0.55	21.85 / 0.60	20.97 / 0.55	21.07 / 0.55	21.36 / 0.57
	5	Ours	23.03 / 0.74	23.59 / 0.71	21.47 / 0.55	23.60 / 0.73	22.76 / 0.67	22.55 / 0.64	23.54 / 0.73	22.65 / 0.66	22.47 / 0.63	**22.85 / 0.67**
		Tian *et al.* [131]	24.52 / 0.76	22.22 / 0.61	20.84 / 0.50	23.21 / 0.68	21.88 / 0.60	21.35 / 0.54	23.12 / 0.67	21.84 / 0.59	21.36 / 0.54	22.26 / 0.61
		Pan *et al.* [257]	22.81 / 0.69	21.94 / 0.59	20.71 / 0.49	22.03 / 0.62	21.15 / 0.56	21.16 / 0.53	22.40 / 0.64	21.31 / 0.56	21.18 / 0.53	21.63 / 0.58
		Zhang *et al.* [258]	22.29 / 0.63	21.76 / 0.59	20.96 / 0.52	21.90 / 0.61	20.99 / 0.56	21.07 / 0.55	21.91 / 0.60	21.03 / 0.55	21.16 / 0.55	21.45 / 0.57
	10	Ours	23.52 / 0.74	23.51 / 0.69	21.68 / 0.56	23.67 / 0.72	22.71 / 0.66	22.59 / 0.63	23.64 / 0.72	22.84 / 0.66	22.62 / 0.63	**22.98 / 0.67**
		Tian *et al.* [131]	24.15 / 0.72	22.21 / 0.58	20.85 / 0.47	23.09 / 0.65	21.82 / 0.57	21.39 / 0.52	22.96 / 0.64	21.82 / 0.56	21.36 / 0.51	22.18 / 0.58
		Pan *et al.* [257]	22.69 / 0.67	21.95 / 0.57	20.75 / 0.48	22.01 / 0.61	21.16 / 0.55	21.22 / 0.52	22.32 / 0.62	21.34 / 0.55	21.20 / 0.52	21.63 / 0.57
		Zhang *et al.* [258]	22.24 / 0.62	21.75 / 0.58	20.85 / 0.51	21.91 / 0.60	21.04 / 0.55	21.13 / 0.54	21.87 / 0.59	21.10 / 0.54	21.14 / 0.53	21.45 / 0.56
	15	Ours	23.36 / 0.74	23.31 / 0.68	21.69 / 0.55	23.48 / 0.72	22.73 / 0.65	22.49 / 0.62	23.56 / 0.71	22.77 / 0.65	22.49 / 0.61	**22.88 / 0.66**
		Tian *et al.* [131]	23.62 / 0.68	22.02 / 0.54	20.76 / 0.44	22.79 / 0.61	21.73 / 0.54	21.25 / 0.49	22.74 / 0.60	21.65 / 0.53	21.24 / 0.48	21.98 / 0.55
		Pan *et al.* [257]	22.51 / 0.66	21.86 / 0.55	20.72 / 0.46	21.90 / 0.60	21.18 / 0.53	21.15 / 0.50	22.26 / 0.60	21.29 / 0.54	21.15 / 0.50	21.56 / 0.55
		Zhang *et al.* [258]	22.11 / 0.61	21.62 / 0.56	20.73 / 0.49	21.82 / 0.59	21.09 / 0.54	21.04 / 0.52	21.84 / 0.58	21.07 / 0.53	21.04 / 0.52	21.37 / 0.55
	25	Ours	22.61 / 0.72	22.70 / 0.66	21.39 / 0.54	22.73 / 0.70	22.22 / 0.64	22.09 / 0.60	22.77 / 0.69	22.17 / 0.63	22.00 / 0.59	**22.30 / 0.64**
		Tian *et al.* [131]	22.33 / 0.60	21.38 / 0.47	20.35 / 0.38	21.85 / 0.54	21.09 / 0.47	20.81 / 0.42	21.78 / 0.53	21.03 / 0.46	20.73 / 0.42	21.26 / 0.48
		Pan *et al.* [257]	21.85 / 0.61	21.41 / 0.50	20.43 / 0.41	21.40 / 0.55	20.83 / 0.50	20.86 / 0.45	21.63 / 0.55	20.94 / 0.50	20.79 / 0.44	21.13 / 0.50
		Zhang *et al.* [258]	21.54 / 0.59	21.18 / 0.54	20.40 / 0.48	21.32 / 0.57	20.78 / 0.53	20.75 / 0.51	21.26 / 0.56	20.71 / 0.52	20.66 / 0.50	20.96 / 0.53
	All	Ours	23.07 / **0.73**	**23.32 / 0.69**	**21.47 / 0.55**	**23.38 / 0.72**	**22.60 / 0.66**	**22.43 / 0.63**	**23.41 / 0.72**	**22.61 / 0.65**	**22.31 / 0.61**	**22.73 / 0.66**
		Tian *et al.* [131]	**23.85** / 0.71	22.00 / 0.56	20.72 / 0.46	22.82 / 0.63	21.68 / 0.56	21.22 / 0.50	22.74 / 0.62	21.63 / 0.55	21.20 / 0.50	21.98 / 0.57
		Pan *et al.* [257]	22.54 / 0.66	21.81 / 0.56	20.65 / 0.47	21.86 / 0.60	21.09 / 0.54	21.10 / 0.51	22.20 / 0.61	21.23 / 0.54	21.09 / 0.50	21.51 / 0.56
		Zhang *et al.* [258]	22.08 / 0.61	21.59 / 0.57	20.75 / 0.51	21.75 / 0.60	20.96 / 0.55	20.99 / 0.53	21.75 / 0.59	20.98 / 0.54	21.01 / 0.53	21.32 / 0.56

TABLE 6.2: Quantitative comparison to other video super-resolution methods at 4x scaling factor. We report PSNR/SSIM values of our and competitor methods on VID4 and Set8 datasets. Different rows and columns correspond to different AWGN levels and blur kernels, respectively. Rows labeled as "All" correspond to average PSNR/SSIM values across different noise levels. Columns denoted as "All" correspond to average PSNR/SSIM values across different blur kernels.

the estimated blur kernel after manipulation. We performed the mentioned procedure on degraded videos from Set8 and obtained $MAE = 0.0004$ and $KS = 0.9837$ (Kernel Similarity).

6.3.3 Comparisons

In this section, we perform qualitative and quantitative comparisons with some of the state-of-the-art methods in video super-resolution, denoising, and scratch removal.

6.3.3.1 Video Super-Resolution

We performed a quantitative comparison with the non-blind video super-resolution approach of Tian *et al.* [131], and with the blind methods of Pan *et al.* [257], and Zhang *et al.* [258]. We report PSNR/SSIM metrics for different blur kernels and noise levels in Table 6.2. Our method achieves the best performance in all settings except for the one closest to the bicubic kernel, which is the one where

σ	Method	Dataset VID4	SET8
5	Ours	**40.85 / 0.99**	**40.22 / 0.99**
	UDVD [259]	36.66 / 0.98	38.11 / 0.97
	DVDnet [243]	37.92 / 0.99	39.30 / 0.99
	FastDVDnet [167]	40.68 / 0.99	39.80 / 0.99
10	Ours	**35.46 / 0.99**	**35.09 / 0.99**
	UDVD [259]	33.41 / 0.97	33.96 / 0.95
	DVDnet [243]	34.27 / 0.98	34.02 / 0.96
	FastDVDnet [167]	34.83 / 0.99	34.52 / 0.99
15	Ours	**32.30 / 0.99**	**31.78** / 0.98
	UDVD [259]	31.52 / 0.96	31.27 / 0.94
	DVDnet [243]	31.94 / 0.97	30.81 / 0.94
	FastDVDnet [167]	31.64 / 0.99	31.24 / **0.99**
25	Ours	27.76 / **0.99**	**27.50** / 0.97
	UDVD [259]	**28.48** / 0.95	27.48 / 0.93
	DVDnet [243]	28.15 / 0.94	26.67 / 0.89
	FastDVDnet [167]	27.11 / **0.99**	26.98 / **0.99**
All	Ours	**34.09 / 0.99**	**33.65** / 0.98
	UDVD [259]	32.52 / 0.97	32.71 / 0.95
	DVDnet [243]	33.07 / 0.97	32.7 / 0.95
	FastDVDnet [167]	33.57 / **0.99**	33.14 / **0.99**

TABLE 6.3: Quantitative comparison to the non-blind video denoising method of Tassano *et al.* [167], [243], and Sheth *et al.* [259]. We report PSNR/SSIM values on VID4 and Set8 datasets.

Method	Dataset VID4	SET8
Ours	**36.09 / 0.99**	**31.93 / 0.98**
Wan *et al.* [251]	24.54 / 0.83	26.98 / 0.86

TABLE 6.4: Quantitative comparison to the scratch removal method of Wan *et al.* [251].

naturally a specialized model [131] performs best. To understand the benefits of our multi-frame and pairwise training, we retrained the model of Wang *et al.* [154] on the Vimeo90K[133] and evaluated it on the Vid4 and Set8 test sets. Retrained model achieves 22.47 / 0.63 and, 26.80 / 0.74 for Vid4 and Set8, respectively. In contrast, our model achieves 22.73 / 0.66 on Vid4 and 27.07 / 0.76 on Set8, while addressing multiple restoration tasks simultaneously, handling non-integer scaling factors, and allowing result manipulation. Qualitative results are presented in Fig. 6.5.

6.3.3.2 Video Denoising

We performed a quantitative comparison with the video denoising methods of Tassano *et al.* [167], [243], and Sheth *et al.* [259]. We report results for different noise levels in Table 6.3. Our blind method achieves competitive performance and even slightly outperforms the model of [167], which has access to the noise level as input. Qualitative results are presented in Fig. 6.6.

FIGURE 6.4: **Degradation Manipulation.** Our pipeline takes a low-resolution corrupted video from the first column and outputs the 4x super-resolved frame. Columns 2-5 show different magnified regions of the restored frame. We show the ground-truth patch in Column 6. In column 4, we have the result obtained without adjusting the latent feature from encoder E_d. Columns 2, 3, and 5 contain results obtained using latent codes from mutator model M. One can observe how blur levels vary between different columns (2-5).

6.3.3.3 Video Scratch Removal

We performed a quantitative comparison with the method of Wan *et al.* [251]. In this experiment, we generated corrupted versions of Vid4 and Set8 by first adding AWGN with $\sigma = 5$ and applying synthetic scratches following Wan *et al.* [251]'s protocol. We report PSNR/SSIM metrics in Table 6.4. Our method outperforms the competitor's method. Note that our pipeline takes scratched videos as input while [251] takes a single scratched frame. A significant performance gap can be explained by our method leveraging information from the temporal dimension, which is not available in the case of [251]. On the other hand, [251] takes the mask of the scratched region as input, which simplifies the restoration process. Qualitative results are presented in Fig. 6.7.

6.3.3.4 Manipulating Real Videos

We demonstrate the editing capabilities of our pipeline by evaluating it on the degraded versions of 4K high-resolution real videos. The illustration is presented in Fig. 6.4. One can observe the gradual decrease of the blur level in restored frames from left to the right. Initially, we pass the feature from encoder E_d to backbone R_b and obtain the results in the 4-th column. We manipulate the blur kernel to both more and less blur. We feed the mutator M with the modified blur kernels to obtain new embeddings to condition the restoration. One can see the effect from blurry to sharper results.

6.4 Discussion

In this chapter, we proposed a discriminative learning strategy that helps separate content from the degradation by reasoning on pairs of degraded patches, where both content and degradation vary independently. The degradation representation is used as conditioning for a video restoration model that can handle denoising, super-resolution, and scratch removal. More importantly, the learned representation can be manipulated to fine-tune the results, which is crucial for real application scenarios. Our model achieves state-of-the-art results while avoiding any test time optimization contrary to many existing blind methods. An important direction for future work would be to explore a broader range of degradation, such as compression artifacts or deinterlacing. Additionally, it should still be possible to better leverage temporal information.

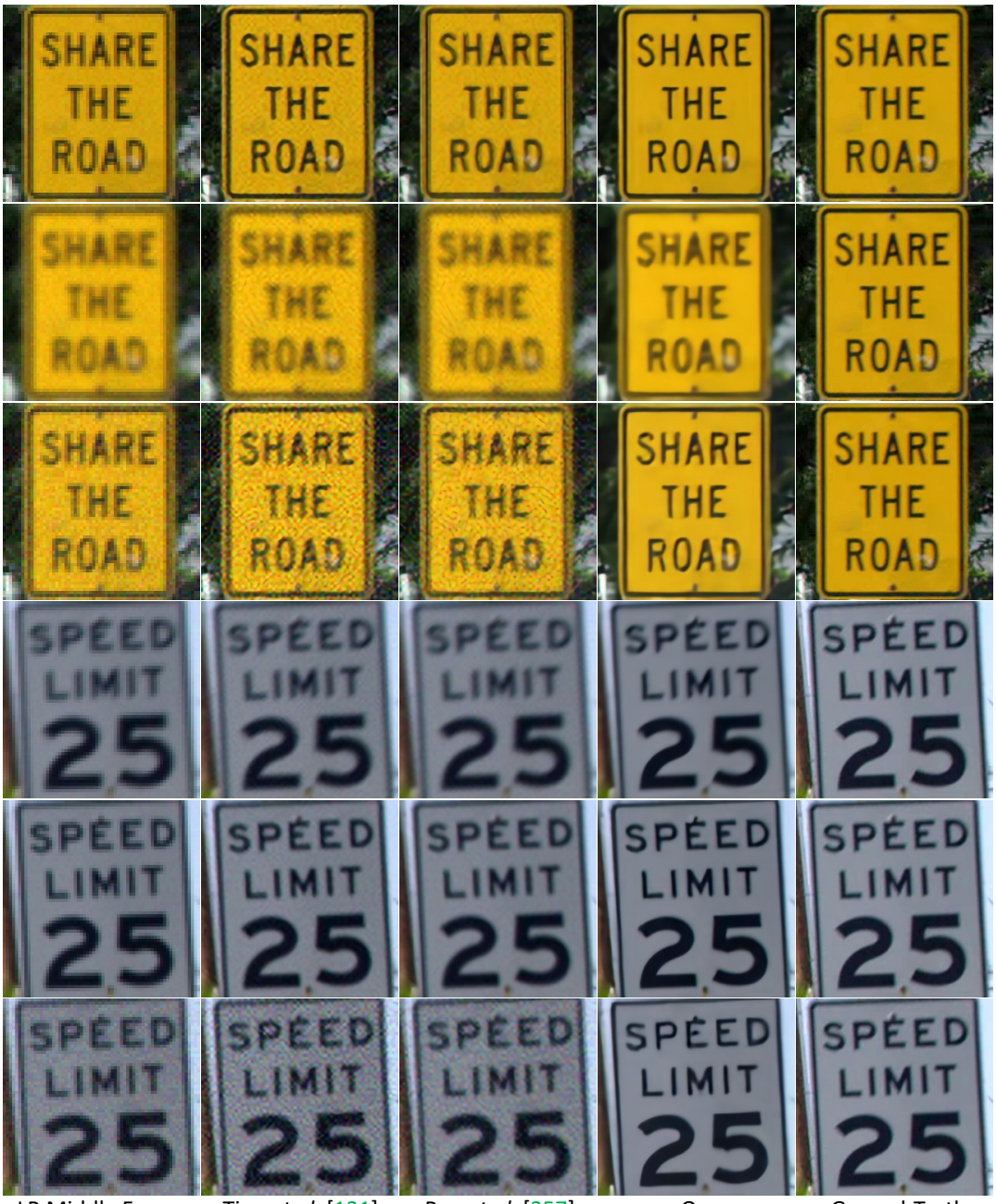


FIGURE 6.5: **Qualitative Comparison Super-Resolution.** We performed a qualitative comparison with methods of Tian *et al.* [131] and Pan *et al.* [257]. Different rows correspond to different combinations of blur kernels and a noise levels. The first column corresponds to a low-resolution input middle frame. Next, the second and third columns correspond to the restored results of Tian *et al.* [131] and Pan *et al.* [257], respectively. The fourth column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

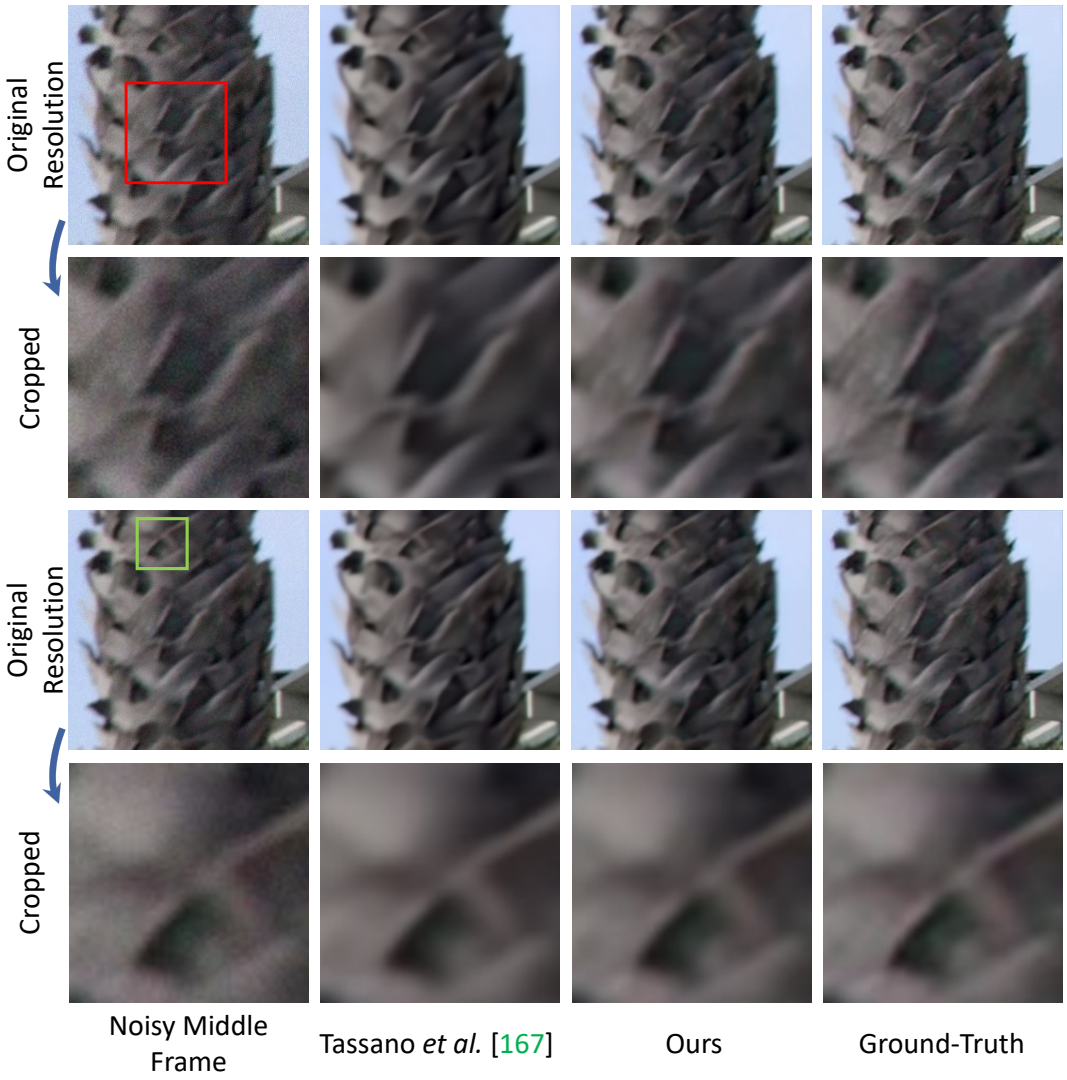


FIGURE 6.6: **Qualitative Comparison Denoising.** We performed a qualitative comparison with method of Tassano *et al.* [167]. First two rows correspond to a noise level of $\sigma = 25$. Last two rows correspond to a noise level of $\sigma = 15$. The first column corresponds to a noisy input middle frame. The second column correspond to the restored results of Tassano *et al.* [167]. The third column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

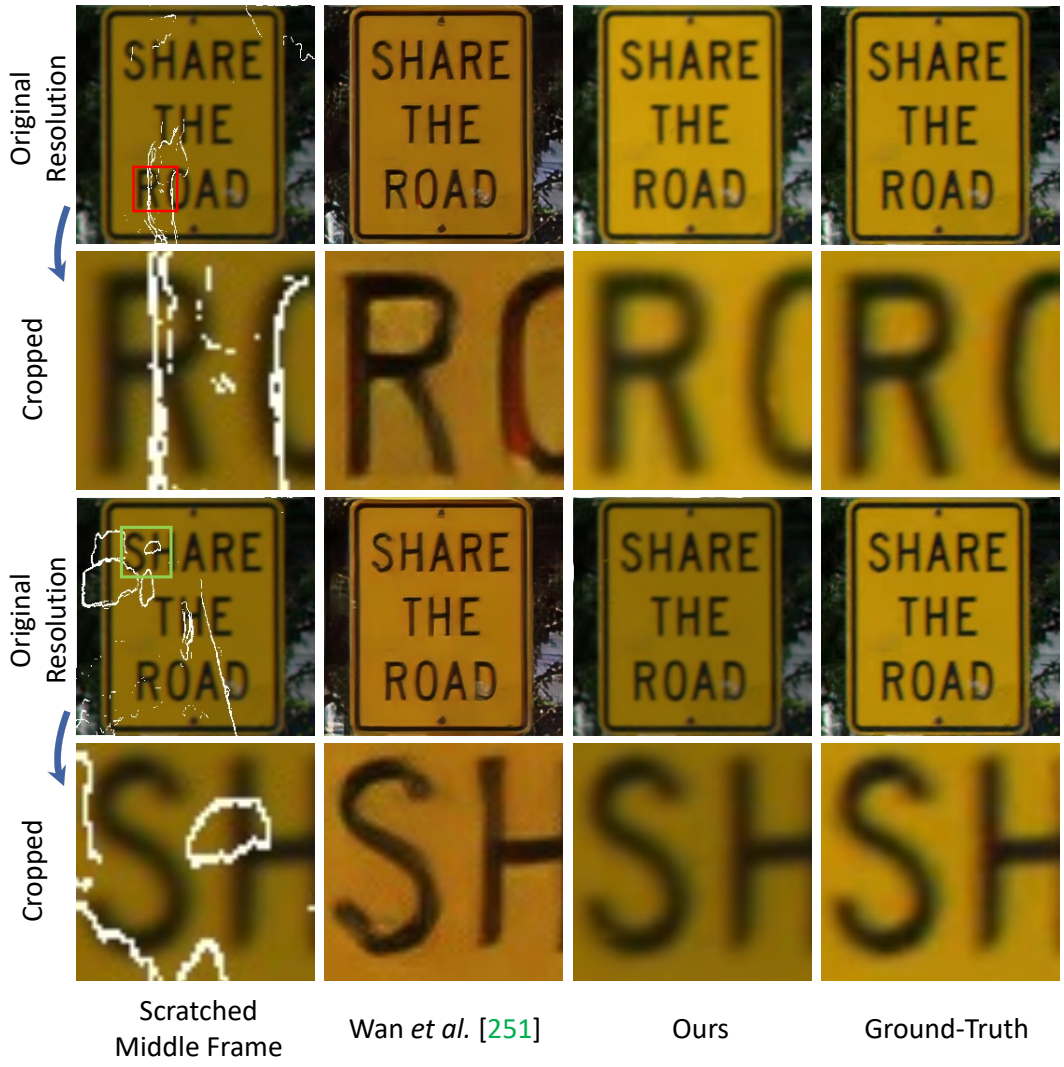


FIGURE 6.7: **Qualitative Comparison Scratch Removal.** We performed a qualitative comparison with method of Wan *et al.* [251]. The first column corresponds to a scratched input middle frame. The second column correspond to the restored results of Wan *et al.* [251]. The third column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

MLP **Head of Degradation Encoder E_d**					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	1024	-	-	`lReLU`	512
dense	512	-	-	`linear`	256

TABLE 6.5: The network architecture of contrastive *MLP* head. The input pairwise concatenated features are of size 1024. The output size is 256.

Encoder E_k					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	512	-	-	`lReLU`	441
dense	441	-	-	`lReLU`	441
dense	441	-	-	`Softmax`	441

TABLE 6.6: The input embedding is 512 dimensional vector. The output size is 441 which gives 21×21 blur kernel after reshaping.

Encoder E_s					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	512	-	-	`lReLU`	128
dense	128	-	-	`linear`	1

TABLE 6.7: The input degradation embedding is 512 dimensional vector. The output size is 1.

Degradation Mutator M					
Layer	Kernel	Stride	Norm.	Activation	# Filters
dense	954	-	-	`lReLU`	954
dense	954	-	-	`lReLU`	954
dense	954	-	-	`linear`	512

TABLE 6.8: The input embedding is 954 dimensional vector. The output size is 512 dimensional vector.

Restoration Backbone R_B					
Layer	Kernel	Stride	Norm.	Activation	# Filters
conv_3d	$1 \times 5 \times 5$	1	-	`linear`	128
conv_3d	$3 \times 7 \times 7$	1	-	`lReLU`	128
conv_3d	$3 \times 7 \times 7$	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128

Super-Resolution Branch R_{SR}					
Layer	Kernel	Stride	Norm.	Activation	# Filters
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
Pixel-Shuffle	-	1	-	`linear`	3

Denoising Branch R_{DN}					
Layer	Kernel	Stride	Norm.	Activation	# Filters
DA	-	1	-	`lReLU`	128
DA	-	1	-	`lReLU`	128
conv_2d	3×3	1	-	`linear`	3

Chapter 7

Conclusions

In this thesis, we studied challenging and ill-posed image and video restoration tasks. Throughout Chapters 3 to 5, we introduced three novel computational photography tasks. Common to all problems that we covered are inherent ambiguities due to the limited or missing amount of information presented in the input. Chapter 3 introduced the novel problem of extracting a video sequence from a single motion-blurred image. We investigated the associated temporal ambiguities and proposed novel frame-ordering invariant loss functions. We showed that our loss functions and a specific system design enable the extraction of temporally coherent sharp video sequences from single motion-blurred images. This opened up two possibilities for the end-user of the framework. Firstly, the capability to obtain a sharp video sequence corresponding to the blurry input. Secondly, the ability to choose a specific sharp frame from the sequence rather than a single central deblurred output. In Chapters 4 to 6, we tackled various degradations by first learning the proper latent representation of the data via leveraging the advances in generative adversarial and contrastive learning. Furthermore, we introduced an effective framework for inverting the learned representations. First, we learn the inverse of the frozen representation and then further finetune the representation and the inverse jointly. We demonstrated how one could leverage the representation together with its inverse mapping to enable image restoration, editing, and generation of multiple differently restored versions of the degraded input.

In Chapter 4, we proposed the novel task of deblurring and rotating motion-blurred faces. Towards this goal, we collected a new Bern Multi-View Face Dataset. Our unique dataset enabled simulating realistic motion blur through averaging sharp ground-truth frames. Moreover, it allowed the enforcement of multi-view constraints, which are crucial to synthesizing sharp videos from a new camera view. Given a generative model of faces, its inverse together with our dataset, we proposed a method that generates a sharp video sequence from a blurry face. Moreover, our solution allows controlling the viewpoint of the output face, which is an additional feature for the system's end-user.

In Chapter 5, we pushed the boundaries of facial degradations to the edge and tackled the problem of extreme face super-resolution. Specifically, we considered the upsampling of 8×8 facial images by a factor of $16\times$. The task is inherently ambiguous due to the limited information presented in the input. We compensated for the lack of information by additionally leveraging the information presented in the short audio speech sample corresponding to the distorted 8×8 image of the person. We were the first to introduce the task of extreme face super-resolution using audio. Our solution first maps two different modalities (low-resolution image and audio sample) to a shared representation, fuses the resulting latent codes in the representation, and finally maps the fused result to the high-resolution image space. A byproduct of our system is the ability to obtain multiple reconstructions from a

single low-resolution image via supplying the method with a fixed low-resolution image and multiple audio tracks of the distorted identity. This allows additional flexibility to the final consumers of the model.

In Chapter 6, we addressed the problem of blind video super-resolution. We focused on reversing the degradation process where a sharp image is sequentially convolved with an unknown anisotropic gaussian blur kernel, sub-sampled by a factor of $4\times$, summed with unknown gaussian noise, and finally scratched. Similar to Chapters 4 and 5, here, we also first learned a representation; however, we learned the representation of degradations this time. Next, we showed how the learned representation guides the restoration process. Furthermore, we demonstrated that our latent space allows: *(i)* regressing the original degradation parameters such as blur kernel and a noise level; *(ii)* encoding degradation parameters back to the latent space. More importantly, we showed that our system allows us to adjust the resulting output by manipulating degradation parameters in the latent space and passing the feature corresponding to the revised parameters to the restoration branch. This enables more flexibility and fine-grained control over reconstructing missing high-frequency details in the input video. For example, an end-user can adjust the sharpness and noise present in the final output.

Methods presented throughout Chapters 3 to 6 addressed different image or video restoration tasks. However, besides restoring degraded inputs, our models also empower users to have more control over the output and freedom of choice between multiple plausible results. The method presented in Chapter 3 allows choosing a frame from the resulting sharp video sequence. In Chapter 4, users can additionally obtain a novel view of a restored video sequence of the face. Given a very low-resolution image of a face and corresponding audio, the method presented in Chapter 5 enables obtaining multiple high-resolution reconstructions. Finally, one can control the resulting sharpness and noise in the context of the blind video restoration method introduced in Chapter 6. These editing capabilities were achieved by leveraging invertible image representations.

We presented different modular image restoration techniques that heavily rely on learning invertible image representations. We demonstrated that these latent spaces opened up the doors for controllable image restoration. Therefore, advances in learning better representations will further boost the performance and abilities of the presented methods. The research community has seen tremendous progress in generative adversarial training. However, there remain some challenges while training these systems. We outline below four possible directions that continue this work towards this goal.

Pushing the Limits of Invertible Generative Modeling. Some methods presented in this thesis heavily rely on generative adversarial learning. Therefore, any future improvements in this area will be substantially beneficial. While generative adversarial networks can generate human faces with unprecedented resolution and realism, it remains challenging to train their inverse models. More specifically, the final system occasionally cannot precisely reconstruct some facial details. These details are essential for such a sensitive subject as a human face. Therefore, exploring generative models and their inverses with richer representation capabilities is crucial.

Extra Control via Further Disentanglement of the Latent Space. Recent advances in generative modeling already achieve a certain degree of disentanglement of different factors of variation in learned representations. This allowed us to rotate faces

in the latent space in Chapter 4 and incorporate audio in the face super-resolution method presented in Chapter 5. However, one might envision a system that supports a richer set of editing capabilities. For example, one might want to manipulate attributes like shape, brightness, and expressions. Towards this goal, we believe that achieving a higher degree of disentanglement is one of the promising directions.

3D Aware Representations. Current advances in novel view synthesis literature are largely based on breakthroughs in Neural Rendering [260]–[265]. Nerfs achieved an unprecedented quality of reconstructions. Unfortunately, these methods are computationally demanding. Therefore, the research community is currently focusing on decreasing training and inference time. Recently, Chan *et al.* [260] proposed geometry-aware 3D generative adversarial networks based on StyleGAN and Nerfs. We believe that future progress in this direction and incorporation of these ideas in proposed solutions can improve the 3D consistency of our restoration pipelines.

Incorporating Other Modalities. In Chapter 5 we demonstrated how leveraging information presented in an audio speech sample can assist in recovering a high-resolution image of the face. What about other modalities? Radford *et al.* [266] successfully connected language and image domains via Contrastive Language-Image Pre-training (CLIP). Patashnik *et al.* [267] proposed interactive text-driven image manipulation by capitalizing on CLIP and StyleGAN. Therefore, it might be promising to incorporate the text domain in the context of image restoration. One can reduce the ambiguities during the restoration process of severely degraded input images by incorporating information presented in the text.

Bibliography

[1] M. Noroozi *et al.*, "Motion deblurring in the wild", in *GCPR*, 2017.

[2] S. Nah *et al.*, "Deep multi-scale convolutional neural network for dynamic scene deblurring", in *CVPR*, 2017.

[3] M. Tomasello *et al.*, "Reliance on head versus eyes in the gaze following of great apes and human infants: The cooperative eye hypothesis.", eng, *J Hum Evol*, vol. 52, no. 3, 2007, ISSN: 0047-2484 (Print); 0047-2484 (Linking).

[4] T. H. Oh *et al.*, "Speech2face: Learning the face behind a voice", in *CVPR*, 2019.

[5] G. E. Hinton *et al.*, "Autoencoders, minimum description length, and helmholtz free energy", *Advances in neural information processing systems*, pp. 3–3, 1994.

[6] J. Masci *et al.*, "Stacked convolutional auto-encoders for hierarchical feature extraction", in *International conference on artificial neural networks*, Springer, 2011, pp. 52–59.

[7] P. Vincent *et al.*, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion", *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[8] D. P. Kingma *et al.*, "Auto-encoding variational bayes", in *ICLR*, 2014.

[9] I. Higgins *et al.*, "Beta-vae: Learning basic visual concepts with a constrained variational framework", in *ICLR*, 2016.

[10] A. v. d. Oord *et al.*, "Neural discrete representation learning", *arXiv preprint arXiv:1711.00937*, 2017.

[11] A. Makhzani *et al.*, "Adversarial autoencoders", *arXiv preprint arXiv:1511.05644*, 2015.

[12] P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks", in *CVPR*, 2016.

[13] I. Goodfellow *et al.*, "Generative adversarial nets", in *NeurIPS*, 2014.

[14] A. Radford *et al.*, "Unsupervised representation learning with deep convolutional generative adversarial networks", *arXiv:1511.06434*, 2015.

[15] J. Donahue *et al.*, "Adversarial feature learning", *International Conference on Learning Representations*, 2017.

[16] J. Donahue *et al.*, "Large scale adversarial representation learning", *arXiv preprint arXiv:1907.02544*, 2019.

[17] C. Ledig *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*, 2017, pp. 4681–4690.

[18] O. Kupyn *et al.*, "Deblurgan: Blind motion deblurring using conditional adversarial networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.

[19] D. Pathak *et al.*, "Context encoders: Feature learning by inpainting", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[20] T. Salimans *et al.*, "Improved techniques for training gans", in *Advances in Neural Information Processing Systems*, 2016, pp. 2226–2234.

[21] M. Arjovsky *et al.*, "Towards principled methods for training generative adversarial networks", *arXiv preprint arXiv:1701.04862*, 2017.

[22] M. Arjovsky *et al.*, "Wasserstein gan", *arXiv:1701.07875*, 2017.

[23] I. Gulrajani *et al.*, "Improved training of wasserstein gans", in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[24] X. Mao *et al.*, "Least squares generative adversarial networks", in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2017, pp. 2813–2821.

[25] J. Zhao *et al.*, "Energy-based generative adversarial network", *arXiv preprint arXiv:1609.03126*, 2016.

[26] D. Berthelot *et al.*, "Began: Boundary equilibrium generative adversarial networks", *arXiv preprint arXiv:1703.10717*, 2017.

[27] N. Kodali *et al.*, "How to train your dragan", *arXiv preprint arXiv:1705.07215*, 2017.

[28] K. Roth *et al.*, "Stabilizing training of generative adversarial networks through regularization", in *Advances in Neural Information Processing Systems*, 2017, pp. 2015–2025.

[29] T. Karras *et al.*, "Progressive growing of gans for improved quality, stability, and variation", in *ICLR*, 2018.

[30] T. Karras *et al.*, "A style-based generator architecture for generative adversarial networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[31] T. Karras *et al.*, "Analyzing and improving the image quality of stylegan", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[32] A. Dosovitskiy *et al.*, "Discriminative unsupervised feature learning with convolutional neural networks", in *Advances in Neural Information Processing Systems*, 2014, pp. 766–774.

[33] T. Wang *et al.*, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere", in *International Conference on Machine Learning*, PMLR, 2020, pp. 9929–9939.

[34] Z. Wu *et al.*, "Unsupervised feature learning via non-parametric instance discrimination", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[35] T. Chen *et al.*, "A simple framework for contrastive learning of visual representations", *arXiv preprint arXiv:2002.05709*, 2020.

[36] K. He *et al.*, "Momentum contrast for unsupervised visual representation learning", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] J.-B. Grill *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning", *arXiv preprint arXiv:2006.07733*, 2020.

[38] X. Chen *et al.*, "Exploring simple siamese representation learning", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 750–15 758.

[39] M. Caron *et al.*, "Unsupervised learning of visual features by contrasting cluster assignments", *arXiv preprint arXiv:2006.09882*, 2020.

[40] X. Wang *et al.*, "Unsupervised feature learning by cross-level instance-group discrimination", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 586–12 595.

[41] R. Qian *et al.*, "Spatiotemporal contrastive video representation learning", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6964–6974.

[42] I. Dave *et al.*, "Tclr: Temporal contrastive learning for video representation", *arXiv preprint arXiv:2101.07974*, 2021.

[43] M. Patrick *et al.*, *Multi-modal self-supervision from generalized data transformations*, 2021. [Online]. Available: https://openreview.net/forum?id=mgVbI13p96.

[44] Y. Tian *et al.*, "Contrastive multiview coding", *arXiv preprint arXiv:1906.05849*, 2019.

[45] T. Han *et al.*, "Self-supervised co-training for video representation learning", in *Neurips*, 2020.

[46] A. Miech *et al.*, "End-to-End Learning of Visual Representations from Uncurated Instructional Videos", in *CVPR*, 2020.

[47] T. Afouras *et al.*, "Self-supervised learning of audio-visual objects from video", in *European Conference on Computer Vision*, 2020.

[48] H. Alwassel *et al.*, "Self-supervised learning by cross-modal audio-video clustering", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[49] A. Srivastava *et al.*, "On advances in statistical modeling of natural images", *Journal of Mathematical Imaging and Vision*, vol. 18, no. 1, pp. 17–33, 2003. DOI: 10.1023/A:1021889010444. [Online]. Available: https://doi.org/10.1023/A:1021889010444.

[50] L. I. Rudin *et al.*, *Nonlinear total variation based noise removal algorithms*, 1992.

[51] Y. You *et al.*, "A regularization approach to joint blur identification and image restoration", English (US), *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 416–428, 1996, ISSN: 1057-7149. DOI: 10.1109/83.491316.

[52] T. Chan *et al.*, "Total variation blind deconvolution", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998. DOI: 10.1109/83.661187.

[53] S. Anwar *et al.*, "Class-specific image deblurring", in *ICCV*, 2015.

[54] J. Pan *et al.*, "Blind image deblurring using dark channel prior", in *CVPR*, 2016.

[55] Y. Yan *et al.*, "Image deblurring via extreme channels prior", in *CVPR*, 2017.

[56] Y. Zhou *et al.*, "A map-estimation framework for blind deblurring using high-level edge priors", in *ECCV*, 2014.

[57] D. Perrone *et al.*, "Blind deconvolution via lower-bounded logarithmic image priors", in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2015, pp. 112–125.

[58] D. Perrone *et al.*, "Total variation blind deconvolution: The devil is in the details", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2909–2916.

[59] S. Cho *et al.*, "Fast motion deblurring", *ACM Trans. Graph.*, vol. 28, no. 5, 145:1–145:8, Dec. 2009, ISSN: 0730-0301. DOI: 10.1145/1618452.1618491. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618491.

[60] Q. Shan *et al.*, "High-quality motion deblurring from a single image", *ACM Transactions on Graphics (SIGGRAPH)*, 2008.

[61] T. Michaeli *et al.*, "Blind deblurring using internal patch recurrence", in *ECCV*, 2014.

[62] J. Dong *et al.*, "Blind image deblurring with outlier handling", in *ICCV*, 2017.

[63] M. Hirsch *et al.*, "Fast removal of non-uniform camera shake", in *ICCV*, Piscataway, NJ, USA: IEEE, Nov. 2011.

[64] H. Zhang *et al.*, "Non-uniform camera shake removal using a spatially-adaptive sparse penalty", in *NeurIPS*, 2013.

[65] D. Gong *et al.*, "Blind image deconvolution by automatic gradient activation", in *CVPR*, 2016.

[66] A. Chakrabarti, "A neural approach to blind motion deblurring", in *ECCV*, 2016.

[67] T. Hyun Kim *et al.*, "Dynamic scene deblurring", in *ICCV*, 2013.

[68] T. Hyun Kim *et al.*, "Segmentation-free dynamic scene deblurring", in *CVPR*, 2014.

[69] J. Sun *et al.*, "Learning a convolutional neural network for non-uniform motion blur removal", in *CVPR*, 2015.

[70] J. Pan *et al.*, "Soft-segmentation guided object motion deblurring", in *CVPR*, 2016.

[71] D. Gong *et al.*, "From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur", in *CVPR*, 2017.

[72] Y. Bahat *et al.*, "Non-uniform blind deblurring by reblurring", in *ICCV*, 2017.

[73] J. Pan *et al.*, "Learning discriminative data fitting functions for blind image deblurring", in *ICCV*, 2017.

[74] T. M. Nimisha *et al.*, "Blur-invariant deep learning for blind-deblurring", in *ICCV*, 2017.

[75] T. Hyun Kim *et al.*, "Online video deblurring via dynamic temporal blending network", in *ICCV*, 2017.

[76] H. Zhang *et al.*, "Intra-frame deblurring by leveraging inter-frame camera motion", in *CVPR*, 2015.

[77] A. Sellent *et al.*, "Stereo video deblurring", in *ECCV*, 2016.

[78] P. Wieschollek *et al.*, "Learning blind motion deblurring", in *ICCV*, 2017.

[79] T. H. Kim *et al.*, "Dynamic scene deblurring using a locally adaptive linear blur model", *CoRR*, 2016.

[80] W. Ren *et al.*, "Video deblurring via semantic segmentation and pixel-wise non-linear kernel", in *ICCV*, 2017.

[81] S. Su *et al.*, "Deep video deblurring for hand-held cameras", in *CVPR*, 2017.

[82] L. Pan *et al.*, "Simultaneous stereo video deblurring and scene flow estimation", in *CVPR*, 2017.

[83] H. Park *et al.*, "Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence", in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[84] J. Pan *et al.*, "Deblurring face images with exemplars", in *ECCV*, 2014.

[85] G. G. Chrysos *et al.*, "Deep face deblurring", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[86] G. G. Chrysos *et al.*, "Motion deblurring of faces", *International Journal of Computer Vision*, vol. 127, no. 6, 2019.

[87] M. Jin *et al.*, "Learning face deblurring fast and wide", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[88] Z. Shen *et al.*, "Deep semantic face deblurring", in *CVPR*, 2018.

[89] R. Yasarla *et al.*, "Deblurring face images using uncertainty guided multi-stream semantic networks", *IEEE Transactions on Image Processing*, vol. 29, 2020.

[90] W. Ren *et al.*, "Face video deblurring using 3d facial priors", in *ICCV*, 2019.

[91] B. Lu *et al.*, "Unsupervised domain-specific deblurring via disentangled representations", in *CVPR*, 2019.

[92] X. Xu *et al.*, "Learning to super-resolve blurry face and text images", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[93] Y. Song *et al.*, "Joint face hallucination and deblurring via structure generation and detail enhancement", *International Journal of Computer Vision*, 2019.

[94] C. Dong *et al.*, "Learning a deep convolutional network for image super-resolution", in *European conference on computer vision*, Springer, 2014, pp. 184–199.

[95] J. Kim *et al.*, "Accurate image super-resolution using very deep convolutional networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[96] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[97] Y. Wang *et al.*, "A fully progressive approach to single-image super-resolution", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[98] X. He *et al.*, "Ode-inspired network design for single image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[99] M. Haris *et al.*, "Recurrent back-projection network for video super-resolution", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[100] Z. Zhang *et al.*, "Image super-resolution by neural texture transfer", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[101] X. Deng *et al.*, "Wavelet domain style transfer for an effective perception-distortion tradeoff in single image super-resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[102] S. Y. Kim *et al.*, "Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[103] W.-S. Lai *et al.*, "Deep laplacian pyramid networks for fast and accurate super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[104] Y. Zhang *et al.*, "Residual dense network for image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[105] J. Li *et al.*, "Multi-scale residual network for image super-resolution", in *The European Conference on Computer Vision (ECCV)*, 2018.

[106] N. Ahn *et al.*, "Fast, accurate, and lightweight super-resolution with cascading residual network", in *The European Conference on Computer Vision (ECCV)*, 2018.

[107] Y. Tai *et al.*, "Image super-resolution via deep recursive residual network", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[108] Y. Zhang *et al.*, "Image super-resolution using very deep residual channel attention networks", in *The European Conference on Computer Vision (ECCV)*, 2018.

[109] X. Wang *et al.*, "Recovering realistic texture in image super-resolution by deep spatial feature transform", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[110] Y. Qiu *et al.*, "Embedded block residual network: A recursive restoration model for single-image super-resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[111] Z. Li *et al.*, "Feedback network for image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[112] K. Zhang *et al.*, "Deep unfolding network for image super-resolution", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[113] Y.-S. Xu *et al.*, "Unified dynamic convolutional network for super-resolution with variational degradations", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[114] Y. Mei *et al.*, "Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[115] F. Yang *et al.*, "Learning texture transformer network for image super-resolution", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[116] X. Deng *et al.*, "Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9189–9198.

[117] Y. Zhang *et al.*, "Mr image super-resolution with squeeze and excitation reasoning attention network", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 425–13 434.

[118] X. Hu *et al.*, "Meta-sr: A magnification-arbitrary network for super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[119] X. Xu *et al.*, "Towards real scene super-resolution with raw images", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[120] J. Cai *et al.*, "Toward real-world single image super-resolution: A new benchmark and a new model", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[121] C. Chen *et al.*, "Camera lens super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[122] X. Zhang *et al.*, "Zoom to learn, learn to zoom", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[123] J.-H. Choi *et al.*, "Evaluating robustness of deep image super-resolution against adversarial attacks", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[124] R. Zhou *et al.*, "Kernel modeling super-resolution on real low-resolution images", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[125] K. Zhang *et al.*, "Learning a single convolutional super-resolution network for multiple degradations", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[126] J. Gu *et al.*, "Blind super-resolution with iterative kernel correction", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[127] H. Zhang *et al.*, "Deep stacked hierarchical multi-patch network for image deblurring", in *CVPR*, 2019.

[128] M. S. Rad *et al.*, "Srobb: Targeted perceptual loss for single image super-resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[129] J. W. Soh *et al.*, "Natural and realistic single image super-resolution with explicit natural manifold discrimination", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[130] X. Wang *et al.*, "Edvr: Video restoration with enhanced deformable convolutional networks", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[131] Y. Tian *et al.*, "Tdan: Temporally-deformable alignment network for video super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[132] Y. Jo *et al.*, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[133] T. Xue *et al.*, "Video enhancement with task-oriented flow", *International Journal of Computer Vision (IJCV)*, vol. 127, no. 8, pp. 1106–1125, 2019.

[134] X. Tao *et al.*, "Detail-revealing deep video super-resolution", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[135] D. Liu *et al.*, "Robust video super-resolution with learned temporal dynamics", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[136] J. Caballero *et al.*, "Real-time video super-resolution with spatio-temporal networks and motion compensation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[137] A. Kappeler *et al.*, "Video super-resolution with convolutional neural networks", *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, 2016. DOI: 10.1109/TCI.2016.2532323.

[138] Z. Hui *et al.*, "Fast and accurate single image super-resolution via information distillation network", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[139] M. Haris *et al.*, "Deep back-projection networks for super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[140] W. Han *et al.*, "Image super-resolution via dual-state recurrent networks", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[141] S. Li *et al.*, "Fast spatio-temporal residual network for video super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[142] L. Wang *et al.*, "Learning parallax attention for stereo image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[143] T. Dai *et al.*, "Second-order attention network for single image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[144] P. Yi *et al.*, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[145] H. Zhang *et al.*, "Two-stream action recognition-oriented video super-resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[146] S.-J. Park *et al.*, "Srfeat: Single image super-resolution with feature discrimination", in *The European Conference on Computer Vision (ECCV)*, 2018.

[147] A. Bulat *et al.*, "To learn image super-resolution, use a gan to learn how to do image degradation first", in *The European Conference on Computer Vision (ECCV)*, 2018.

[148] W. Zhang *et al.*, "Ranksrgan: Generative adversarial networks with ranker for image super-resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[149] T. Michaeli *et al.*, "Nonparametric blind super-resolution", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 945–952.

[150] V. Cornillere *et al.*, "Blind image super-resolution with spatially variant degradations", *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.

[151] A. Shocher *et al.*, ""zero-shot" super-resolution using deep internal learning", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3118–3126.

[152] S. Bell-Kligler *et al.*, "Blind super-resolution kernel estimation using an internal-gan", *arXiv preprint arXiv:1909.06581*, 2019.

[153] J. Pan *et al.*, "Deep blind video super-resolution", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4811–4820.

[154] L. Wang *et al.*, "Unsupervised degradation representation learning for blind super-resolution", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 581–10 590.

[155] H. Huang *et al.*, "Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution", in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[156] X. Yu *et al.*, "Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[157] A. Bulat *et al.*, "Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[158] X. Yu *et al.*, "Face super-resolution guided by facial component heatmaps", in *The European Conference on Computer Vision (ECCV)*, 2018.

[159] Y. Chen *et al.*, "Fsrnet: End-to-end learning face super-resolution with facial priors", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[160] K. Zhang *et al.*, "Super-identity convolutional neural network for face hallucination", in *The European Conference on Computer Vision (ECCV)*, 2018.

[161] X. Yu *et al.*, "Super-resolving very low-resolution face images with supplementary attributes", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[162] X. Xu *et al.*, "Learning to super-resolve blurry face and text images", in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[163] V. Jain *et al.*, "Natural image denoising with convolutional networks", *Advances in neural information processing systems*, vol. 21, 2008.

[164] H. C. Burger *et al.*, "Image denoising: Can plain neural networks compete with bm3d?", in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 2392–2399.

[165] J. Xie *et al.*, "Image denoising and inpainting with deep neural networks", in *Advances in neural information processing systems*, 2012, pp. 341–349.

[166] H. Yue *et al.*, "Supervised raw video denoising with a benchmark dataset on dynamic scenes", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[167] M. Tassano *et al.*, "Fastdvdnet: Towards real-time deep video denoising without flow estimation", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[168] M. Maggioni *et al.*, "Efficient multi-stage video denoising with recurrent spatio-temporal fusion", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3466–3475.

[169] M. Claus *et al.*, "Videnn: Deep blind video denoising", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[170] W.-S. Lai *et al.*, "A comparative study for single image blind deblurring", in *CVPR*, 2016.

[171] R. Fergus *et al.*, "Removing camera shake from a single photograph", in *SIGGRAPH*, 2006.

[172] J. Johnson *et al.*, "Perceptual losses for real-time style transfer and super-resolution", in *ECCV*, 2016.

[173] K. Simonyan *et al.*, "Very deep convolutional networks for large-scale image recognition", in *ICLR*, 2015.

[174] J. Kim *et al.*, "Deeply-recursive convolutional network for image super-resolution", in *CVPR*, 2016.

[175] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network", in *CVPR*, 2016.

[176] K. He *et al.*, "Deep residual learning for image recognition", in *CVPR*, 2016.

[177] ——, "Identity mappings in deep residual networks", in *ECCV*, 2016.

[178] S. Cho *et al.*, "Video deblurring for hand-held cameras using patch-based synthesis", *ACM Trans. Graph.*, 2012.

[179] I. Goodfellow *et al.*, *Deep Learning*. MIT Press, 2016.

[180] A. Voynov *et al.*, *Unsupervised discovery of interpretable directions in the gan latent space*, 2020. arXiv: 2002.03754 [cs.LG].

[181] P. Paysan *et al.*, "A 3d face model for pose and illumination invariant face recognition", in *AVSS*, 2009.

[182] C. Sanderson *et al.*, "Multi-region probabilistic histograms for robust and scalable identity inference", in *Advances in Biometrics*, M. Tistarelli *et al.*, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[183] V. Blanz *et al.*, "A morphable model for the synthesis of 3d faces", in *SIGGRAPH*, 1999.

[184] J. Booth *et al.*, "A 3d morphable model learnt from 10,000 faces", in *CVPR*, 2016.

[185] T. Bolkart *et al.*, "A robust multilinear model learning framework for 3d faces", in *CVPR*, 2016.

[186] W. Peng *et al.*, "Parametric t-spline face morphable model for detailed fitting in shape subspace", in *CVPR*, 2017.

[187] L. Tran *et al.*, "Nonlinear 3d face morphable model", in *CVPR*, 2018.

[188] A. Ranjan *et al.*, "Generating 3d faces using convolutional mesh autoencoders", in *ECCV*, 2018.

[189] F. Liu *et al.*, "3d face modeling from diverse raw scan data", in *ICCV*, 2019.

[190] S. Ploumpis *et al.*, "Combining 3d morphable models: A large scale face-and-head model", in *CVPR*, 2019.

[191] L. Tran *et al.*, "Towards high-fidelity nonlinear 3d face morphable model", in *CVPR*, 2019.

[192] B. Egger *et al.*, "3d morphable face models - past, present and future", *ACM Transactions on Graphics*, vol. 39, no. 5, Aug. 2020.

[193] H. Yang *et al.*, "Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction", in *CVPR*, 2020.

[194] M. Piotraschke *et al.*, "Automated 3d face reconstruction from multiple images using quality measures", in *CVPR*, 2016.

[195] F. Wu *et al.*, "Mvf-net: Multi-view 3d face morphable model regression", in *CVPR*, 2019.

[196] S. Sanyal *et al.*, "Learning to regress 3d face shape and expression from an image without 3d supervision", in *CVPR*, 2019.

[197] A. Tewari *et al.*, "Fml: Face model learning from videos", in *CVPR*, 2019.

[198] J. Roth *et al.*, "Adaptive 3d face reconstruction from unconstrained photo collections", in *CVPR*, 2016.

[199] A. Tewari *et al.*, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz", in *CVPR*, 2018.

[200] J. Booth *et al.*, "3d face morphable models "in-the-wild"", in *CVPR*, 2017.

[201] H. Kim *et al.*, "Inversefacenet: Deep monocular inverse face rendering", in *CVPR*, 2018.

[202] K. Genova *et al.*, "Unsupervised training for 3d morphable model regression", in *CVPR*, 2018.

[203] H. Kato *et al.*, "Neural 3d mesh renderer", in *CVPR*, 2018.

[204] A. Szabó *et al.*, "Unsupervised generative 3d shape learning from natural images", *arXiv:1910.00287*, 2019.

[205] W. Zhu *et al.*, "Reda:reinforced differentiable attribute for 3d face reconstruction", in *CVPR*, 2020.

[206] X. Xu *et al.*, "View independent generative adversarial network for novel view synthesis", in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[207] Y. Hu *et al.*, "Pose-guided photorealistic face rotation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[208] I. Masi *et al.*, "Deep face recognition: A survey", in *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, IEEE, 2018.

[209] T. Hassner *et al.*, "Effective face frontalization in unconstrained images", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[210] Z. Zhang *et al.*, "Face frontalization using an appearance-flow-based convolutional neural network", *IEEE Transactions on Image Processing*, vol. 28, no. 5, 2018.

[211] R. Huang *et al.*, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[212] J. Zhao *et al.*, "Dual-agent gans for photorealistic and identity preserving profile face synthesis", in *Advances in neural information processing systems*, 2017.

[213] J. Shen *et al.*, "The first facial landmark tracking in-the-wild challenge: Benchmark and results", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2015.

[214] T. Karras *et al.*, "Analyzing and improving the image quality of StyleGAN", in *Proc. CVPR*, 2020.

[215] G. Meishvili *et al.*, "Learning to have an ear for face super-resolution", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[216] Q. Cao *et al.*, "VGGFace2: A dataset for recognising faces across pose and age", in *International Conference on Automatic Face and Gesture Recognition*, 2018.

[217] K. Simonyan *et al.*, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[218] "Statistical color models with application to skin detection", *International Journal of Computer Vision*, vol. 46, no. 1, 2002.

[219] M. Jin *et al.*, "Learning to extract a video sequence from a single motion-blurred image", in *CVPR*, 2018.

[220] A. Bulat *et al.*, "How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)", in *International Conference on Computer Vision*, 2017.

[221] D. P. Kingma *et al.*, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[222] H. Zhou *et al.*, "Rotate-and-render: Unsupervised photorealistic face rotation from single-view images", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[223] G. G. Chrysos *et al.*, "Offline deformable face tracking in arbitrary videos", in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.

[224] T. Karras *et al.*, "A style-based generator architecture for generative adversarial networks", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[225] W. Wang *et al.*, "What makes training multi-modal networks hard?", *CoRR*, vol. abs/1905.12681, 2019. arXiv: 1905.12681. [Online]. Available: http://arxiv.org/abs/1905.12681.

[226] T. Karras *et al.*, "A style-based generator architecture for generative adversarial networks", *arXiv preprint arXiv:1812.04948*, 2018.

[227] Y. Song *et al.*, "Talking face generation by conditional recurrent adversarial network", *arXiv preprint arXiv:1804.04786*, 2018.

[228] H. Zhu *et al.*, "High-resolution talking face generation via mutual information approximation", *arXiv preprint arXiv:1812.06589*, 2018.

[229] R. Arandjelovic *et al.*, "Objects that sound", in *The European Conference on Computer Vision (ECCV)*, 2018.

[230] Y. Tian *et al.*, "Audio-visual event localization in unconstrained videos", in *The European Conference on Computer Vision (ECCV)*, 2018.

[231] A. Owens *et al.*, "Audio-visual scene analysis with self-supervised multisensory features", in *The European Conference on Computer Vision (ECCV)*, 2018.

[232] H. Zhao *et al.*, "The sound of pixels", in *The European Conference on Computer Vision (ECCV)*, 2018.

[233] R. Gao *et al.*, "Learning to separate object sounds by watching unlabeled video", in *The European Conference on Computer Vision (ECCV)*, 2018.

[234] A. Ephrat *et al.*, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation", *arXiv preprint arXiv:1804.03619*, 2018.

[235] A. Sterling *et al.*, "Isnn: Impact sound neural network for audio-visual object classification", in *The European Conference on Computer Vision (ECCV)*, 2018.

[236] E. Shlizerman *et al.*, "Audio to body dynamics", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[237] D. Bau *et al.*, "Seeing what a gan cannot generate", in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[238] D. Ulyanov *et al.*, "Instance normalization: The missing ingredient for fast stylization", *arXiv preprint arXiv:1607.08022*, 2016.

[239] J. S. Chung *et al.*, "Voxceleb2: Deep speaker recognition", in *INTERSPEECH*, 2018.

[240] O. M. Parkhi *et al.*, "Deep face recognition", in *British Machine Vision Conference*, 2015.

[241] R. Rothe *et al.*, "Deep expectation of real and apparent age from a single image without facial landmarks", *International Journal of Computer Vision*, vol. 126, no. 2, pp. 144–157, 2018, ISSN: 1573-1405. DOI: 10.1007/s11263-016-0940-3. [Online]. Available: https://doi.org/10.1007/s11263-016-0940-3.

[242] H. Huang *et al.*, "Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution", in *CVPR*, 2017.

[243] M. Tassano *et al.*, "Dvdnet: A fast network for deep video denoising", in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 1805–1809.

[244] J. He *et al.*, "Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration", in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, Springer, 2020, pp. 53–68.

[245] F. Stanco *et al.*, "Towards the automated restoration of old photographic prints: A survey", Oct. 2003, 370 –374 vol.2, ISBN: 0-7803-7763-X. DOI: 10.1109/EURCON.2003.1248221.

[246] V. Bruni *et al.*, "A generalized model for scratch detection", *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 44–50, 2004. DOI: 10.1109/TIP.2003.817231.

[247] R.-C. Chang *et al.*, "Photo defect detection for image inpainting", in *Seventh IEEE International Symposium on Multimedia (ISM'05)*, 2005, 5 pp.–. DOI: 10.1109/ISM.2005.91.

[248] I. Giakoumis *et al.*, "Digital image processing techniques for the detection and removal of cracks in digitized paintings", *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 178–188, 2006. DOI: 10.1109/TIP.2005.860311.

[249] K. Yu *et al.*, "Crafting a toolchain for image restoration by deep reinforcement learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[250] M. Suganuma *et al.*, "Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[251] Z. Wan *et al.*, "Bringing old photos back to life", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2747–2757.

[252] X. Hu *et al.*, "Meta-sr: A magnification-arbitrary network for super-resolution", 2019.

[253] D. Tran *et al.*, "A closer look at spatiotemporal convolutions for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[254] Y. Zhang *et al.*, "Image super-resolution using very deep residual channel attention networks", in *ECCV*, 2018.

[255] W. Shi *et al.*, "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 (CVPR)*, 2016.

[256] Z. Hu *et al.*, "Learning good regions to deblur images", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 345–362, 2015. DOI: 10.1007/s11263-015-0821-1.

[257] J. Pan *et al.*, "Deep blind video super-resolution", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 4811–4820.

[258] K. Zhang *et al.*, "Designing a practical degradation model for deep blind image super-resolution", in *IEEE International Conference on Computer Vision*, 2021, pp. 4791–4800.

[259] D. Y. Sheth *et al.*, "Unsupervised deep video denoising", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[260] E. R. Chan *et al.*, "Efficient geometry-aware 3D generative adversarial networks", in *arXiv*, 2021.

[261] K. Park *et al.*, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields", *ACM Trans. Graph.*, vol. 40, no. 6, 2021.

[262] P. Hedman *et al.*, "Baking neural radiance fields for real-time view synthesis", *ICCV*, 2021.

[263] S. Liu *et al.*, "Editing conditional radiance fields", in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[264] K. Park *et al.*, "Nerfies: Deformable neural radiance fields", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5865–5874.

[265] B. Mildenhall *et al.*, "Nerf: Representing scenes as neural radiance fields for view synthesis", in *ECCV*, 2020.

[266] A. Radford *et al.*, "Learning transferable visual models from natural language supervision", in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html.

[267] O. Patashnik *et al.*, "Styleclip: Text-driven manipulation of stylegan imagery", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2085–2094.

Declaration of consent

on the basis of Article 18 of the PromR Phil.-nat. 19

Name/First Name: Meishvili Givi

Registration Number: 12-338-224

Study program: Computer Science

Bachelor ☐ Master ☐ Dissertation ☑

Title of the thesis: Learning Representations for Controllable Image Restoration

Supervisor: Prof. Dr. Paolo Favaro

I declare herewith that this thesis is my own work and that I have not used any sources other than those stated. I have indicated the adoption of quotations as well as thoughts taken from other authors as such in the thesis. I am aware that the Senate pursuant to Article 36 paragraph 1 litera r of the University Act of September 5th, 1996 and Article 69 of the University Statute of June 7th, 2011 is authorized to revoke the doctoral degree awarded on the basis of this thesis.
For the purposes of evaluation and verification of compliance with the declaration of originality and the regulations governing plagiarism, I hereby grant the University of Bern the right to process my personal data and to perform the acts of use this requires, in particular, to reproduce the written thesis and to store it permanently in a database, and to use said database, or to make said database available, to enable comparison with theses submitted by others.

Wangen bei Olten 02.02.2022

Place/Date

Signature