Skoltech

Skolkovo Institute of Science and Technology

MASTER'S THESIS

Learning image deformations via deep learning

Master's Educational Program: Data Science

Student_____

Artem Sevastopolskiy
Data Science
May 31, 2019

Research Advisor:_____

Victor S. Lempitsky
Associate Professor

Moscow 2019

All rights reserved.©
The author hereby grants to Skoltech permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.

Skoltech

Skolkovo Institute of Science and Technology

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Обучение деформаций изображений с помощью глубинного обучения

Магистерская образовательная программа: Науки о данных

Студент_____
Артем Севастопольский
Науки о данных
31 мая, 2019

Научный руководитель:_____
Виктор Сергеевич Лемпицкий
Доцент

Москва 2019
Все права защищены.©
Автор настоящим дает Сколковскому институту науки и технологий разрешение на воспроизводство и свободное распространение бумажных и электронных копий настоящей диссертации в целом или частично на любом ныне существующем или созданном в будущем носителе.

Learning image deformations via deep learning

Artem Sevastopolskiy

Submitted to the Skolkovo Institute of Science and Technology
on May 31, 2019

Abstract

In this thesis we propose an approach to learning image deformations parameterized by deep neural networks. A main idea introduced in the work is an imposed decomposition of warping of image pixels and colors refinement for computer vision tasks with inherent geometric and volumetric structure. Towards this end, we propose a *forward warping* layer, which extends existing backward warping layer for differentiable learning of pixels motion and allows one to learn a warping while maintaining a full spatial alignment, implicitly required for fully-convolutional networks.

Our practical contribution is two-fold. First, we apply a neural network based regression of warp field for the task of face rotation, where a significant number of pixels needs to be copied onto the target image. Holes in a warped image is subsequently inpainted by a separate neural network. Here, we additionally propose to use a new architecture of discriminator network which enhances the final result by an adversarial training procedure. Secondly, a we validate our approach over the task of image inpainting for human face and body images, where moving non-hidden pixels onto the occluded part can be performed. For this task, we allow a neural network to regress a large number of warp fields, and then combine them into a merged inpainted image in a differentiable way. In order to enhance the visual realism of the obtained images, we additionally propose a *gap discriminator* – a special auxiliary neural network trained together with the main image generator in an adversarial procedure. Apart from that, we propose a new loss which efficiently regularizes the warp fields and lets one control their non-linearity by a dedicated parameter. The results of this work can have both theoretical and applied impact on the field of image-to-image translation.

Research Advisor:
Name: Victor S. Lempitsky
Degree: Ph.D.
Title: Associate Professor

Обучение деформаций изображений с помощью глубинного обучения

Артем Севастопольский

Представлено в Сколковский институт науки и технологий
31 мая, 2019

Реферат

В данной работе предлагается подход для обучения деформаций изображений, параметризованных с помощью глубоких нейронных сетей. Основная идея, представленная в работе, состоит в декомпозиции смещения пикселей изображения и уточнения их интенсивностей или цветов в тех задачах компьютерного зрения, где неявно присутствует геометрическая или пространственная структура. Для этого вводится слой *прямой деформации*, который основан на широко известном в литературе слое обратной деформации для дифференцируемого предсказания смещения пикселей. Слой прямой деформации позволяет предсказывать величину смещения для каждой точки с помощью полносвёрточных нейронных сетей, при этом сохраняя полное пространственное выравнивание, которое требуется для данных алгоритмов.

Практический вклад работы заключается в исследовании двух задач. В первой из них применяется нейронная сеть, предсказывающая величину смещения пикселей для задачи поворота изображения лица человека. В данной задаче необходимо скопировать большое количество точек из исходной картинки в выходную, содержащую изображение лица человека с изменённого угла. Пропуски в деформированной картинке заполняются отдельной закрашивающей сетью. Во второй из задач мы валидируем метод для закраски неизвестных регионов изображения лица и тела человека. В этой и подобных задачах многие пиксели могут быть перенесены в закрашенные регионы неизвестных цветов. В рамках подхода используется нейронная сеть, предсказывающая большое количество деформаций исходной закрашенной картинки, которые затем специальным образом объединяются в финальное предсказанное изображение. Чтобы уточнить результат и сделать картинку более реалистичной, предлагается использовать *дискриминатор пропусков* — специальную вспомогательную нейронную сеть, обучающуюся совместно с генератором в рамках соревновательной процедуры. Помимо этого, предлагается новая функция потерь, позволяющая эффективно контролировать произвольность предсказанных деформаций изображений величиной определённого параметра. Результаты данной диссертации могут иметь как теоретическое, так и прикладное влияние на область перевода изображений в изображения в компьютерном зрении.

Научный руководитель:
Имя: Виктор Сергеевич Лемпицкий
Ученое звание, степень: Кандидат физико-математических наук
Должность: Доцент

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays convolutional neural networks (CNNs) provide solutions of exceptional quality and generalization ability for a broad diversity of tasks of computer vision. Due to advances in application of CNNs, many fundamental problems of computer vision, such as image classification, detection and segmentation [47], can be considered solved. Speaking of more complex problems of image-to-image translation, CNNs are known to be especially helpful for learning pixel-level features (such as semantic segmentation [36, 10], depth estimation [7, 53], colorization [64, 40]) and for aggregating them into global image-level characteristics (e.g. class of an object depicted on a photo). Typically, convolutional layers are used to obtain complex feature representation for each image pixel. If aggregation is required, all spatial information can be later fused by fully-connected layers or similar techniques. Convolutional layer is a local operation, meaning that if an input image is transformed to an output by applying several consequent convolutions, each pixel of an output image heavily depends on the information in its neighborhood on a corresponding input image, but weakly depends on the information which is farther away. Since convolutional kernels used in deep learning are extremely small, such as 3×3 elements, each output value produced by a convolutional layer depends only on pixels in a small neighborhood of its location on an input image. Despite that convolutional layers can be stacked to enlarge network's *receptive field*, in practice its effective size notoriously remains small [37], and dependence on distant pixels typically remains difficult to model.

Nevertheless, for many tasks of pattern recognition it is required to learn geometric transformation along with learning deep pixel-level features. Examples of such tasks include, but not limited to, the following difficult problems: image inpainting, image resynthesis from another view (e.g. given a face photo, predict, how would a face look from another camera location), predicting of future and intermediate video frames. For instance, in image inpainting it is required to fill holes areas based on texture in adjacent locations. However, a successful image inpainting algorithm should benefit from additional semantic information extracted from distant areas. This is especially important when there are a lot of holes to fill, or when the areas of holes are large and irregular [60, 61].

In this thesis, we intend to propose a general approach for effective learning of spatial warping of various images, which can be valuable for a number of image-to-image translation

tasks where output image is not *spatially aligned* with an input image. The approach is validated on the task of face rotation, which has an inherent geometric structure, and on the image inpainting problem, where one of the suitable approaches would be to find origins of hidden pixels in a non-occluded part.

Chapter 2

Related Work

2.1 General-purpose Methods

2.1.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) represent a family of algorithms which have turned a field of computer vision into a new stream, and are currently among the most popular and powerful machine learning algorithms. Due to advances in application of CNNs, many fundamental problems of computer vision which bothered scientists' minds for years, such as image classification [31], detection [44] and segmentation [36, 10, 16], can be considered fully solved. Furthermore, today researchers discover new problems where CNNs are exceptionally helpful and provide unexpected, surprising results. This family of algorithms is currently driving a new field of deep learning and other fields of machine learning where image processing is essential or just beneficial.

Main operation which CNNs are built upon is a 2D discrete convolution operation (or, more correct, cross-correlation), which applies a set of given filters $\mathbf{K} \in \mathbb{R}^{C_{out} \times C_{in} \times K_h \times K_w}$ to a multi-channel image $\mathbf{I} \in \mathbb{R}^{C_{in} \times H \times W}$ ($C = 3$ in case of RGB encoded image), thus obtaining a filtered (convolved) image $\mathbf{O} \in \mathbb{R}^{C_{out} \times (H - K_h + 1) \times (W - K_w + 1)}$:

$$\mathbf{O}[c', i, j] = \sum_{c=1}^{C_{in}} \sum_{i'=1}^{K_h} \sum_{j'=1}^{K_w} \mathbf{I}\left[c', c, i - \frac{K_h - 1}{2} + i', j - \frac{K_w - 1}{2} + j'\right] \cdot \mathbf{W}[c, i', j'] \tag{2.1}$$

In practice, input image $I \in \mathbb{R}^{C \times H \times W}$ fed to a convolution is padded by zeros or other ways, in order to preserve $C \times H \times W$ image shape after convolution. Various flavours of (2.1) exist, including *strided convolutions*, with filters centered not in every possible location but with a certain skipping interval, *dilated convolutions*, dilated convolutions, with filters of increased size with several skips put between adjacent cells, etc. Slice of an output image $\mathbf{O}$ at specific channel c is usually referred to as a *feature map*.

Both originally proposed CNNs and modern ones are comprised of a stack of described convolutions, usually referred to as convolutional layers, which apply a large number of shared learned filters to incoming images, non-linearities such as ReLU function $\max(0, x)$ and normal-

ization layers such as BatchNorm [23] or InstanceNorm [50]. When spatial dimensions of an image need to be changed, strided convolution or pooling layer is applied in case of resolution decrease, and transpose convolution or upsampling layer is applied in case of resolution increase. If aggregation of values from all the feature map is required, average pooling or reshaping is typically performed.

Extraordinary success of convolutional networks for image processing is usually explained by several facts, including: typical similarity of intensities of adjacent image pixels, strong correlations between close-standing pixels, insights from biology inspired by animal visual cortex. Most often in deep learning, convolutional layers produce a high number of feature maps (the lower the spatial resolution is, the higher is the number of output channels), but spatial dimensions of filters K_h, K_w remain as small as possible. Values of $K_h = K_w = 3$ are the most popular for all convolutional layers in practical applications.

2.1.2 Fully-convolutional Neural Networks

One can notice that convolutions (2.1), especially when K_h are K_w are extremely small, can only transform local regions of size $K_h \times K_w$. Being concatenated, convolutions represent functions depending on a larger region of input image pixels. This way, a convolutional stack applied to an image can be perceived as a simpler CNN applied to a patch at every possible location and returning a single value. This idea is laid into a basis of fully-convolutional neural networks (FCNs), which, given a full input image, apply a set of convolutions and return an output image of the same shape. Each pixel of the resulting image is the same as the result of the same CNN applied to image patches, and thus is a generalization of a memory and time consuming patch-based approach known previously [52]. FCNs only consist of convolutions, normalization layers, non-linearities, down- and upsampling layers, but do not feature any fully-connected ending usually employed for classification and similar tasks (this gives a name for FCNs, since convolutions are usually the only learnable layers in such networks).

Fully-convolutional networks were first proposed as a novel approach to image segmentation [36], but with a remark that they can be also used for dense predictions of arbitrary kind. Later on, FCNs were significantly improved in many different ways [45, 20], mainly featuring skip connections and residual connections [17] at different places, leading to better gradient flows and more lightweight architectures. Today various kinds of FCNs are applied for the tasks of image-to-image translation [54, 4, 19, 57]. Commonly, modern FCNs consist of encoder (resolution-contracting path with convolutions interleaved by poolings or strided convolutions) and decoder (resolution-expanding path with convolutions interleaved by upsamplings or transpose convolutions), and sometimes also feature a heavy feature transformation part between these two [54, 32].

2.1.3 Generative Adversarial Networks

Another breakthrough in a field of image generation and image-to-image translation was brought by Generative Adversarial Networks (GANs), first proposed in [11]. The idea is to combine two networks — a *generator* and a *discriminator* — training in an adversarial procedure. Let us investigate the image generation problem, when one needs to learn sampling new, unseen images from a distribution defined by a collection of images (this is a classical generative problem in machine learning). The generator G_θ, in the most simple variant, aims to transform samples drawn from some predefined distribution $z \sim p(z)$, typically taking a simple form (e.g. uniform or normal on a sphere in $\mathbb{R}^n$), into images $x = G_\theta(z)$, such that their resulting distribution $x = G_\theta(z) \sim p_G(x)$ is as close to the distribution of real data $p_r(x)$ as much as possible. On the contrary, the discriminator D_ϕ learns to predict a probability that x is a real sample, not "fake" (generated one). A loss which penalizes both networks is usually given by

$$\mathcal{L}_{GAN}(\theta, \phi) = \min_{G_\theta} \max_{D_\phi} \mathbb{E}_{x \sim p_r(x)} \log D_\phi(x_r) + \mathbb{E}_{z \sim p(z)} \log(1 - D_\phi(G_\theta(z)),$$

where expectations are replaced by averaging of finite number of samples at each training step. The adversarial procedure makes discriminator trying to distinguish real samples from generated ones, while generator is trying to fool the discriminator. This results in an increase of image generation performance and realism of the resulting images. Despite that GANs are known to be hard to train and experience problems such as mode collapse [46], advanced works exist which modify them to obtain images of an unprecedented realism, quality and semantic consistency [30, 29]. Besides, GANs are known to be highly performing in the area of image-to-image translation, where there are usually treated as auxiliary discriminators which the main, generative network needs to fool [19, 3, 58].

2.2 Methods Suited for Particular Applications

2.2.1 3D Object Rotation

View resynthesis of 3D objects is one of the tasks on the edge of computer vision and computer graphics, highly important for face recognition, telepresence and VR/AR applications. In the most common setting, the task is, given an 2D of a 3D object directed at some angle to the camera, obtain an image of the same object from a different viewpoint. When a single image is provided, the task has an element of prediction, namely, the intensities of pixels not visible in the input image but persistent on the output image need to be inferred. The problem is attracting increasing attention,

and various approaches already exist. Some of them are based on warping field prediction and further image refinement [66, 26, 42]. These works are using a prior knowledge that some pixels can be moved onto new locations. However, usually a problem of non-visible pixels is arising this way [42], and they need to be specially treated. Some works, on the contrary, are making use of more straightforward approaches [6]. A well-posed subtask of generating new views by stereo input has also attracted a number of works [8, 27]. A recent work of [62] is based on a plane sweep in case of several input views with known intrinsic and extrinsic camera parameters. It was shown that a very high prediction fidelity can be achieved in this case, even for new, previously unseen kinds of objects.

2.2.2 Face Rotation

Many works which concentrate on resynthesis task (view and pose change of 3D objects based on a single photo or multiple photos) use faces image as the primary domain. Frontalized face view can be used as a normalized representation useful to simplify and enhance quality of the face recognition, while a view of the rotated faces brings new opportunities to high-quality free-viewpoint telepresence. Several state-of-the-art methods for face frontalization use backward sampler. For instance, HF-PIM [4] predicts a cylindrical texture map and a backward warping field required to transform a texture map to a frontalized facial view, and X2Face [55] fully relies on backward warping for a more complex task of synthesizing a face in new pose and with new mimics based on a driving image. Many other methods which are highly-performing at the moment, such as CAPG-GAN [19], LB-GAN [3], CPF [58], FF-GAN [59], are based on encoder-decoder networks directly performing a desired transformation by representing an image in a low-dimensional latent space. Nevertheless, most existing methods don't provide attempts to learn CNNs in a setting where input and output images are spatially aligned, despite that it is arguably an important property of their successful application. In this regard, it is important to mention a work by Siarohin et al. [49] which pays attention to the problem of spatial misalignment between input and output image. The algorithm proposed in that work is mainly based on Generative Adversarial Networks (GANs) with deformable skip connections. UV-GAN [5] is another algorithm for face rotation which relies on 3D Morphable Model (3DMM) [2] estimation in order to find a facial UV texture map and completes missing regions by U-Net [45]. Despite that 3DMM parameters are typically learned by fully-connected layers and there is no misalignment problem, this 3D model is underparameterized to model all subtleties of an arbitrary face.

2.2.3 Adversarial Image Inpainting

Over the recent years, a number of methods for image inpainting has been produced. Several challenges need to be addressed for this task. Ideally, an inpainting algorithm relevant to the holes fill-in problem addressed by our work must restore the contents visually consistent with the rest of an image, handle holes of arbitrary shape and produce semantically meaningful output. Here we present some of the current state-of-the-art methods which possess these qualities. Deep Image Prior [51] is an elegant optimization-based method which can be applied to several restoration tasks including image inpainting. Despite its universality, it does not support the incorporation of any knowledge about the specific task, either passed as a prior or as a result of training. Many inpainting algorithms based on ConvNets make use of adversarial and perceptual losses and their modifications. Partial Convolutions [33], Gated Convolutions [60] are the examples of such methods. They use convolutions which ignore specific pixels and are well-suited for images with irregular holes; both of these methods are capable of modifying whole images, not only parts with holes. Other flavours of this approach exist, such as [61, 22], mainly based on local refinement of the results by additional adversarial discriminators, and [57], combining FCN with a feature warping. At the same time, a collection of older research exists, such as seminal PatchMatch [1] optimizing for related regions among non-occluded pixels, interpolation-based methods [43], and methods based on dictionary learning [18].

Chapter 3

Method

3.1 Forward Warping

Fully-convolutional networks were initially introduced for semantic segmentation task [36, 45, 20] as a way to generalize a patch-based neural network approach to the problem. By design, these architectures, based on numerous convolutions, can receive images of an arbitrary resolution and apply the same filters centered around every possible location. Convolutional kernels used in deep learning are very tiny, such as 3×3 pixels, and this makes each output value produced by a convolutional layer dependent on an extremely small vicinity of its location on an input image. This way, each output pixel, produced by a fully-convolutional network, is dependent on some region of input image centered in its location, known as the *receptive field*. For the segmentation task, especially with small number of classes, there is usually no need in a large receptive field size, as a segmentation label of a pixel is usually a function of a pixel neighborhood and of the presence of contours and edges [45]. To be noted, even for the networks deep and wide enough, which turn out to have a large receptive field, small spatial dependencies are modeled much better than the long ones, distant spatial dependencies, and effective receptive field is notoriously smaller than the factual one.

While making use of small spatial dependencies is totally justified for the tasks such as image segmentation, it is clear that there are image-to-image translation problems where global features of an image and distant dependencies are the same important as local ones. Despite that, fully-convolutional networks are currently employed for all kinds of computer vision problems where image is translated into another image [54, 4, 19, 57], including those where there is no *spatial alignment* between input and output image. For instance, one can consider a problem of predicting future video frames of a video, where input pixels flow into output pixels positioned differently compared to an input image, and such a motion can fall out of the effective receptive field of a predicting network.

In the following section a *forward sampler* layer is proposed, which, opposed to the backward sampler introduced in STN [25], allows one to regress a deformation of an input image in a full spatial correspondence with an input image. Forward sampler can be implemented as a differentiable neural network layer. From the described point of view, this technique allows for using

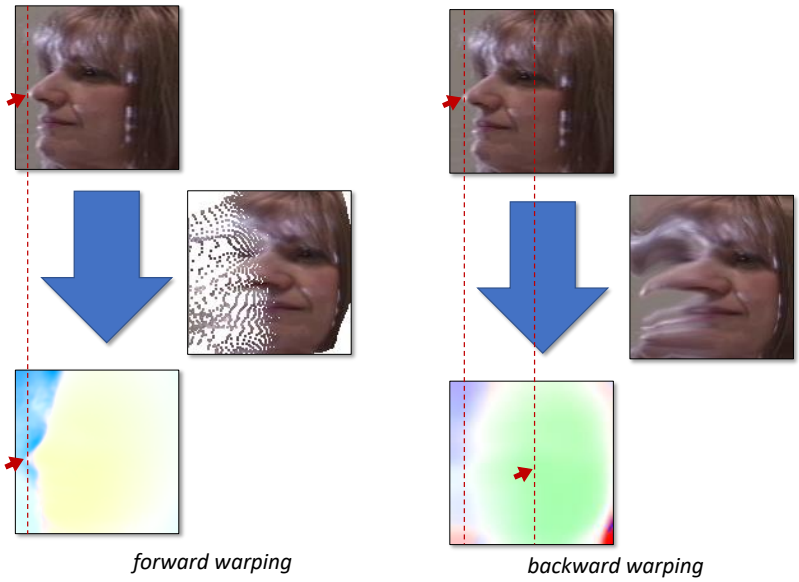


Figure 3.1: The difference between forward and backward warping explained for the task of face frontalization. In both schemes the warping field is predicted (bottom; hue = direction, saturation = magnitude) from the input image (top), and the warping is applied (right). In the case of forward warping the input image and the predicted field are aligned (e.g. the motion of the tip of the nose is predicted at the position of the tip of the nose). Contrary to that, in the case of backward warping the input image and the warping field are misaligned.

an unrestricted class of transformations, including those requiring large displacements of objects depicted on investigated images, without breaking an implicit assumption of local dependency of fully-convolutional networks, described above.

3.1.1 Formulation

Let $\mathbf{x}$ be a source image and let $\mathbf{y}$ be a target image, and let $x[p, q]$ denote the image entry (sample) at the integer location (p, q) (which can be e.g. an RGB value). Let $\mathbf{w}[p, q] = (\mathbf{u}[p, q], \mathbf{v}[p, q])$ be a warping field. In both seminal and applied works [25, 48, 57] it is proposed to predict the warping field based on $\mathbf{x}$ by some convolutional network f_θ, where θ is a vector of some learnable parameters trained on a certain dataset. A standard approach to warping-based resynthesis of images uses backward warping to deform the source image $\mathbf{x}$ to the target image $\mathbf{y}$:

$$\mathbf{y}_{\text{bw}}[p, q] \approx \mathbf{x}[\, p + \mathbf{u}[p, q], q + \mathbf{v}[p, q]\,] \tag{3.1}$$

while sampling at fractional positions $(\mathbf{u}[p, q], \mathbf{v}[p, q])$ is defined bilinearly. More formally, the result of the backward warping is defined as:

$$\mathbf{y}_{\text{bw}}[p, q] = \sum_{i,j} \mathbf{x}[i, j] K(i, j, p + \mathbf{u}[p, q], q + \mathbf{v}[p, q])\,, \tag{3.2}$$

where the bilinear kernel K is defined as follows:

$$K(k,l,m,n) = \max(1-|m-k|,0)\max(1-|n-l|,0)\,, \tag{3.3}$$

so that for each (p,q) the summation in (3.2) is taken over $i = \{\lfloor p+\mathbf{u}[p,q]\rfloor, \lceil p+\mathbf{u}[p,q]\rceil\}$ and $j = \{\lfloor q+\mathbf{v}[p,q]\rfloor, \lceil q+\mathbf{v}[p,q]\rceil\}$.

Originally, the backward warping operation has been popularized for deep image recognition as a part of Spatial Transformer Networks [25] for CNN-based regression of affine deformations and later has been reused extensively for deep image resynthesis [9, 64, 42, 4, 49] becoming a standard layer within deep learning packages along the way. It has been observed, that for resynthesis tasks with significant geometric transformations, the use of warping layers provides significant improvement in the quality and generalization ability compared to architectures that use resynthesis using convolutional layers alone [4].

Backward warping is however limited by the lack of the alignment between the initial image and the warping field. Indeed, as can be seen from (3.1), the vector $(\mathbf{u}[p,q], \mathbf{v}[p,q])$ predicted by the network f_θ for pixel (p,q) defines a motion for the object part that is initially projected onto the pixel $(p+\mathbf{u}[p,q], q+\mathbf{v}[p,q])$. Consider for example the face frontalization task, in which case we want the network to predict the *frontalizing* warp field given the input image containing non-frontal face. Assume that in the initial image the position (p,q) corresponds to the tip of the nose, while for the frontalized face the same position corresponds to the center of the right cheek. When backward warping is used for resynthesis, the prediction of the network f_θ for the position (p,q) has to contain the frontalizing motion of the center of the right cheek. At the same time, the receptive field of the output network unit at (p,q) in the input image corresponds to the tip of the nose. Thus, the network has to predict the motion of the cheek while observing the appearance of a patch centered on the nose (Figure 3.1). When frontalizing motion is small, such misalignment can be handled by deep enough convolutional architecture with large enough receptive fields. However as motions become larger, such mapping becomes progressively harder to learn for a convolutional architecture.

Therefore, in our resynthesis architectures, we replace backward warping with forward warping. The forward warping operation is defined such that the following equality holds approximately for the output image $\mathbf{y}_{\text{fw}}$:

$$\mathbf{y}_{\text{fw}}[\,p+\mathbf{u}[p,q], q+\mathbf{v}[p,q]\,] \approx \mathbf{x}[p,q]\,. \tag{3.4}$$

Thus, in the case of the forward warping, the warp vector at pixel $[p,q]$ defines the motion of this pixel. To implement forward warping, we use the bilinear kernel to rasterize the source pixels onto the target image as follows. First, all contributions from all pixels are aggregated using

convolutional kernel into the aggregator map **a**:

$$\mathbf{a}[i,j] = \sum_{p,q} \mathbf{x}[p,q] K(i,j,p+\mathbf{u}[p,q], q+\mathbf{v}[p,q]) \,. \tag{3.5}$$

Simultaneously, the total weight of all contributions to each pixel is accumulated in a separate aggregator **w**:

$$\mathbf{w}[i,j] = \sum_{p,q} K(i,j,p+\mathbf{u}[p,q], q+\mathbf{v}[p,q]) \,. \tag{3.6}$$

Finally, the value at pixel is defined by normalizing:

$$\mathbf{y}_{\text{fw}}[i,j] = \mathbf{a}[i,j] \;/\; (\mathbf{w}[i,j] + \epsilon) \,, \tag{3.7}$$

where the small constant ϵ prevents numerical instabilities. Formally, for every target position (i,j) the summation in (3.5) and (3.6) runs over all source pixels (p,q). However, since for every source pixel (p,q), the bilinear kernel $K(\cdot,\cdot,p+\mathbf{u}[p,q], q+\mathbf{v}[p,q])$ can take non-zero values at only four positions in the target image, the summations above can be computed efficiently using four passes over the pixels of the source image. Note that similar techniques are used for partial convolutions [33]. As operations (3.5)-(3.7) are subdifferentiable w.r.t. both the input image **x** and the warping field $(\mathbf{u}, \mathbf{v})$, one can perform a backpropagation of gradients through the forward warping operation during training.

Importantly, a principal difference in formulations of (3.4) and (3.1) is in the presence of a denominator responsible for weights normalization. The idea behind is the following: if backward warping is used, output pixel has exactly 4 counterparts on an input image, and their weighted contributions sum up to 1 in a case of the bilinear kernel chosen. Indeed, if a fractional coordinate for output pixel on an input image is (i,j), then sum of weights will be:

$$\begin{aligned}
&K(i,j,\lfloor i \rfloor, \lfloor j \rfloor) + K(i,j,\lfloor i \rfloor, \lceil j \rceil) + K(i,j,\lceil i \rceil, \lfloor j \rfloor) + K(i,j,\lceil i \rceil, \lceil j \rceil) \\
&= \{\text{let } w_i = |i - \lfloor i \rfloor|, w_j = |j - \lfloor j \rfloor|\} \\
&= w_i w_j + w_i(1-w_j) + (1-w_i)w_j + (1-w_i)(1-w_j) \\
&= 1
\end{aligned}$$

On the contrary, in the case of forward warping, an indefinite number of pixels define a color for each output pixel, which makes a sum of weights summing up not to 1 but to an arbitrary quantity.

The main advantage of forward warping over backward warping is that the input image and the predicted warping field are now aligned, as the prediction of the network at pixel (p,q) now

corresponds to the 2D motion of the object part that is projected onto (p, q) in the input image. In the frontalization example above, the network has to predict the frontalizing motion of the tip of the nose, based on the receptive field centered on the tip of the nose. This is an easier mapping to learn than in the case of backward warping, and our experiments demonstrate this effect. On the downside, the output $\mathbf{y}_{\text{fw}}$ of the forward-warping operation in most circumstances contains a number of empty pixels, into which no source pixels were mapped. We denote with $\mathbf{m}$ the binary mask of pixels that are non-empty, i.e. $\mathbf{m}[i, j] = [\mathbf{w}[i, j] > 0]$. A separate *inpainting* stage is then needed to remove (complete) such holes.

Despite that some modern frameworks can make use of sub-differentiability of (3.7) and backpropagate through such an operation with automatic gradient computation (e.g. PyTorch), we provide expressions for gradients and guidelines for their efficient calculation in Appendix A.

3.2 Inpainting with Gap Discriminators

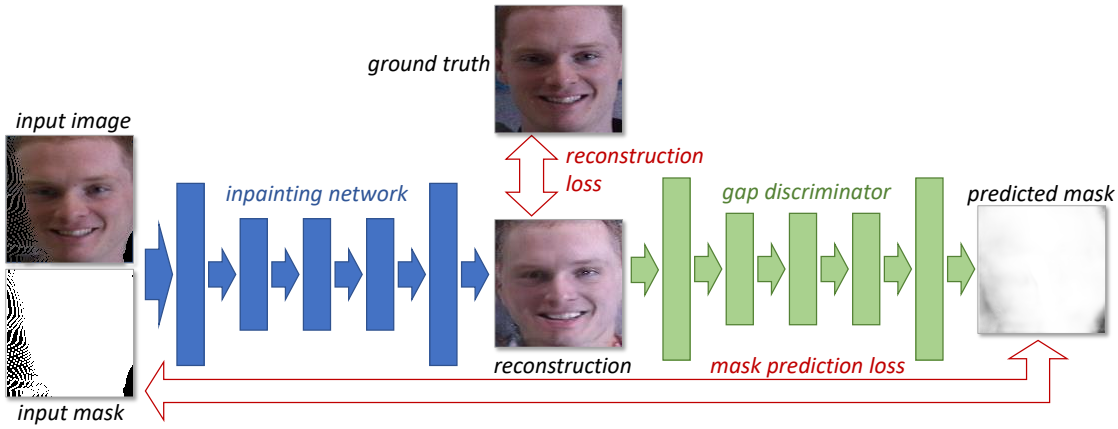


Figure 3.2: Learning to inpaint with gap discriminators. We learn an inpainting network to fill-in holes in the input image (where the known pixels are specified by the mask) by minimizing the reconstruction loss w.r.t. to the ground truth. In parallel, we learn a segmentation network (gap discriminator) that predicts the mask from the result of the fill-in operation by minimizing the mask prediction loss. The inpainting network is trained adversarially to the gap discriminator by maximizing the mask prediction loss, which forces the filled-in parts be indiscernible from the original parts in the reconstructed image.

The image completion function g_ϕ with learnable parameters ϕ maps the image $\mathbf{y}_{\text{fw}}$ and the mask $\mathbf{m}$ to a complete (inpainted) image $\mathbf{y}_{\text{inp}}$:

$$\mathbf{y}_{\text{inp}} = g_\phi(\mathbf{y}_{\text{fw}}, \mathbf{m})\,. \tag{3.8}$$

A significant effort has gone recently into designing good architectures for inpainting. In our experiments, we follow [60] that suggested the use of deep network with *gated convolutions* to handle inpainting tasks.

Irrespective of the architecture of g_ϕ, the choice of the loss function for its learning plays a crucial role. Most often, learning ϕ is done in supervised setting, where a dataset of complete images is taken, a random process that occludes parts of those images is designed, and the network is trained to reverse this random process. Minimization of the following loss is then performed at training time:

$$L_{\text{pix}}(\phi) = -\sum_i \|g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i) - \mathbf{y}^i_{\text{gt}}\|\,, \tag{3.9}$$

where i iterates over training examples and $\mathbf{y}^i_{\text{gt}}$ denotes complete images. The norm in (3.9) can be chosen as an L1 norm or as a more complex *perceptual* loss [28].

When empty pixels form large contiguous holes, the results of learning with pixel-wise or perceptual losses is usually suboptimal and lack plausible large-scale structure due to the inherent multimodality of the task. The use of adversarial learning [11] gives significant boost in this case. Adversarial learning trains a separate classification network d_ψ in parallel with the network g_ϕ. The training objective for d_ψ is the discrimination between the inpainted and the original (uncorrupted) images [22]:

$$L_{\text{disc}}(\psi) = -\sum_i \log d_\psi(\mathbf{y}^i_{\text{gt}}) + \log\left(1 - d_\psi(g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i))\right)\,, \tag{3.10}$$

The training objective for g_ϕ is then augmented with a separate term that measures the probability of the discriminator to classify the inpainted image as the real one:

$$L_{\text{adv}}(\phi) = -\sum_i \log d_\psi(g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i))\,. \tag{3.11}$$

Several works on adversarial inpainting suggested the use of two discriminators that both work on the same principle, but focus on different parts of the images [61, 22]. Usually, the *global* discriminator focuses on the entire image, while the *local* discriminator focuses on the most important part such as the immediate vicinity of the hole [22] or the central part of the face [5].

In our work, we introduce a different kind of discriminators (*gap discriminators*) for the inpainting tasks. We are motivated by the following simple observation. Humans judge the success of the inpainting operation by their (in)ability to identify the hole regions while looking at the inpainted image. Interestingly, they do not need to know any sort of the "ground truth" for such judgment. To mimic this idea, we train the gap discriminator h_ξ to predict the mask $\mathbf{m}$ from the inpainted image by minimizing the weighted cross-entropy loss for binary segmentation:

$$L_{\text{gap}}(\phi, \xi) = -\sum_i \frac{\mathbf{m}^i}{|\mathbf{m}^i|} \odot \log h_\xi(g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i)) + \frac{1-\mathbf{m}^i}{|1-\mathbf{m}^i|} \odot \log\left(1 - h_\xi(g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i))\right) \tag{3.12}$$

Here, $\odot$ denotes the element-wise product (the summation over all pixels) and $|\mathbf{m}|$ denotes the number of non-zero pixels in $\mathbf{m}$. As the training of the gap discriminator progresses, the inpaint-

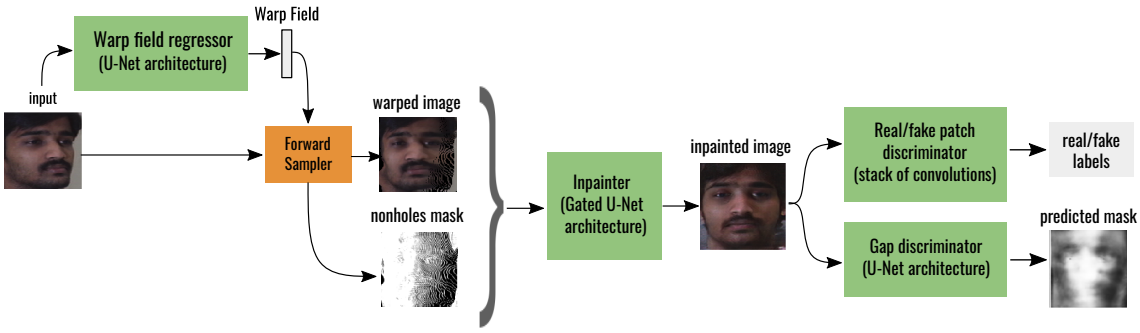


Figure 3.3: Our pipeline for face rotation is based on two generative networks: a warp field regressor f_ω and an inpainter g_ϕ, both based on U-Net architecture (while the latter contains gated convolutions instead of plain ones). Given an input image, a warp field regressor produces offsets, which, summed up with coordinate mesh grid, form a forward warp field. By passing image and a warp field, which encodes the deformation, to a forward sampler, a warped image and its non-holes mask are produced. Inpainter receives the resulting warped image and a nonholes mask and refines the former by filling in the non-visible part. Two discriminators are participating in the learning procedure to make the resulting image look more natural.

ing network is trained to confuse the gap discriminator by maximizing the same cross-entropy loss (3.12) (thus playing a zero-sum game). The new loss can be used alongside the "traditional" adversarial loss (3.11) as well as any other losses. The proposed new loss is applicable to any inpainting/completion problem, not necessarily in conjunction with forward warping.

Learning with incomplete ground truth. In some circumstances, such as texture inpainting tasks, complete ground truth images are not available. Instead, each ground truth image $\mathbf{y}^i_{\text{gt}}$ comes with the binary mask $\mathbf{m}^i_{\text{gt}}$ of known pixels. This mask has to be different from the input mask $\mathbf{m}^i$ (otherwise, the training process may converge to a trivial identity solution for the inpainting network). In such circumstances, the losses (3.9)-(3.11) are adapted so that $\mathbf{y}^i$ and $g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i)$ are replaced by $\mathbf{y}^i \odot \mathbf{m}^i_{\text{gt}}$ and $g_\phi(\mathbf{y}^i_{\text{fw}}, \mathbf{m}^i) \odot \mathbf{m}^i_{\text{gt}}$ correspondingly. Interestingly, the new adversarial loss does not look at the ground truth complete images. Therefore, even when complete ground truth is unavailable, the loss (3.12) may still be applicable without modification (both for the gap discriminator training and as a loss for the inpainting network training).

3.3 Face rotation

We apply our methodology to the task of face rotation, which is the following: given an input image of a face directed at some angle w.r.t. to a camera, obtain an image of the face of the same person in the same environment which would be directed at a different angle w.r.t. to the camera.

Our pipeline is shown in Figure 3.3 and consists of two main networks: a *warp field regressor* and an *inpainter*. A warp field regressor is a ConvNet f_ω with learnable parameters ω which follows a U-Net [45] architecture. Receiving input image (and two additional *mesh grid* arrays encoding pixels rows and columns), it produces a field of offsets, encoded by two 2D arrays $\mathbf{w}_\delta[p,q] = (\mathbf{u}_\delta[p,q],\ \mathbf{v}_\delta[p,q])$. This field is subsequently transformed to a complete forward warp field by a simple addition $\mathbf{w}[p,q] = (\mathbf{u}[p,q], \mathbf{v}[p,q]) = (p + \mathbf{u}_\delta[p,q], q + \mathbf{v}_\delta[p,q])$ and passed to a forward grid sampler. In the described case, $\mathbf{w}_\delta[p,q]$ encodes a motion of a pixel (p,q) on an input image. Note, however, that the same construction could be potentially used to regress a backward warp field, if backward sampler is applied.

The second part, an *inpainter*, is a network g_ϕ with learnable parameters ϕ also based on U-Net architecture (without skip connections, however) with all convolutions replaced to *gated convolutions*. These are the attentive layers first proposed in [60] to efficiently handle difficult inpainting tasks. We employ a gated convolution as defined in [60]:

$$\begin{aligned}\mathsf{Gating} &= conv(I, W_g),\\ \mathsf{Features} &= conv(I, W_f),\\ \mathsf{Output} &= ELU(\mathsf{Features}) \cdot \sigma(\mathsf{Gating}),\end{aligned}$$

where $I \in \mathbb{R}^{I_h \times I_w \times C}$ is an input image, $W_g,\ W_f \in \mathbb{R}^{K_h \times K_w \times C' \times C}$ are weight tensors, and σ and ELU are sigmoid and Exponential Linear Unit activation functions, respectively. The inpainter receives a warped image with holes, a holes mask, and meshgrid tensor encoding positions of the pixels, and predicts an inpainted image.

We train the model in GAN setting and add two discriminator networks. The first, real/fake discriminator, aims to tell ground truth output images from inpainted images, produced by the generative inpainting network. Similar to [60], our real/fake discriminator d_ψ is organized as a stack of plain and strided convolutions followed by average pooling and sigmoid. The resulting number indicates the predicted probability of an image being a "real" one. The second discriminator is a gap discriminator h_ξ, which aims to recover the holes mask from an inpainted image by solving a segmentation problem. Generator, on the contrary, tries to fool the gap discriminator by producing images with the inpainted areas indistinguishable from the non-inpainted areas.

Loss functions. End-to-end learning of the pipeline is a difficult task, requiring a careful balance between various loss components. We optimize a loss $L_{\text{generator}}$ for a generative ConvNet, which comprises a warp field regressor followed by an inpainter:

$$\mathcal{L}_{\text{generator}}(\omega, \phi) = \mathcal{L}_{\text{warping}}(\omega) + \mathcal{L}_{\text{inpainter}}(\omega, \phi) + \alpha_{\text{adv}}\mathcal{L}_{\text{adv}}(\omega, \phi) + \alpha_{\text{gap}}\mathcal{L}_{\text{gap}}(\omega, \phi),$$

where $\mathcal{L}_{\text{warping}}(\omega)$ penalizes a warped image and a warp field, and $\mathcal{L}_{\text{inpainter}}(\omega, \phi)$ penalizes only an inpainted image, $\mathcal{L}_{\text{adv}}(\omega, \phi)$ and $\mathcal{L}_{\text{gap}}(\omega, \phi)$ are generator penalties corresponding to the adversarial learning with the first, real/fake discriminator, and the second, gap discriminator, respectively. Consequently, these components decompose into the following basic loss functions:

$$\mathcal{L}_{\text{warping}}(\omega) = \alpha_{\text{pix}_1}\mathcal{L}_{\text{pix}_1}(\omega) + \alpha_{\text{rigidity}}\mathcal{L}_{\text{rigidity}}(\omega),$$

$$\mathcal{L}_{\text{pix}_1}(\omega) = \frac{1}{3|\mathbf{m}|} \left\| \mathbf{m} \odot (f_\omega(\mathbf{x}) - \mathbf{y}_{\text{gt}}) \right\|_1,$$

$$\mathcal{L}_{\text{rigidity}}(\omega, \eta) = \frac{1}{K}\sum_{i=1}^{K} \left(\left| \sqrt{(\mathbf{u}^i_\omega[p+1, q] - \mathbf{u}^i_\omega[p, q])^2 + (\mathbf{v}^i_\omega[p+1, q] - \mathbf{v}^i_\omega[p, q])^2 + \varepsilon} - \frac{1}{H-1} \right| \right.$$

$$\left. + \left| \sqrt{(\mathbf{u}^i_\omega[p, q+1] - \mathbf{u}_\omega[p, q])^2 + (\mathbf{v}^i_\omega[p, q+1] - \mathbf{v}^i_\omega[p, q])^2 + \varepsilon} - \frac{1}{W-1} \right| \right)$$

where x is the input image, $\mathbf{w} = f_\omega(x)$ is a forward warp field, $(\mathbf{y}_{\text{fw}}, \mathbf{m}) = \mathbf{x} \otimes_{\text{F}} \mathbf{w}$ are a warped image and a nonholes mask, respectively, obtained by a forward sampler $\otimes_{\text{F}}$. H, W correspond to the image shape. Here and below we omit meshgrid as an input to a warp field regressor and an inpainter for the sake of notation clarity.

Rigidity loss defined as $\mathcal{L}_{\text{rigidity}}$ is our essential contribution, which has not been employed for tasks of such kind before, to the best of our knowledge. Illustration is given in Figure 3.4. In order to understand the motivation behind this loss, it should be imagined that springs are placed between each pair of horizontally and vertically adjacent points in a coordinate mesh grid, corresponding to input image. Given a forward warp field $(\mathbf{u}, \mathbf{v})$, rigidity loss constraints it, s.t. lengths of all springs after forward sampling must be close to initial lengths of springs (namely, $\frac{1}{H-1}$ for horizontal springs, $\frac{1}{W-1}$ for vertical springs). This way, deformation is made to be as rigid as possible, and $\alpha_{rigidity}$ controls the degree of warp field rigidity. The loss expression was constructed as a sum of spring forces, where $\alpha_{rigidity}$ can be considered an elasticity coefficient. Optima of this loss alone correspond to all rigid movements of a coordinate grid (translations, rotations and reflections). In the equation for rigidity loss (3.13), forward warp fields $(\mathbf{u}^i_\omega, \mathbf{v}^i_\omega)_{i=1}^K = f_\omega(\mathbf{x}, \mathbf{m})$ returned by a warp field regressor are constrained in the described way.

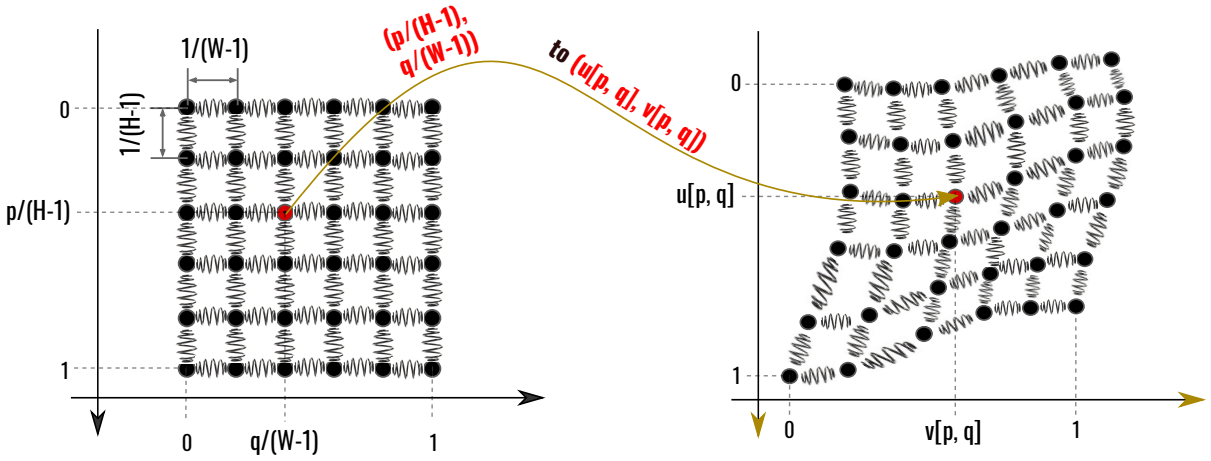


Figure 3.4: Rigidity loss. Plot on the left illustrates a set of input image points; plot on the right shows fractional locations of these points on an output image defined by a forward warp field (i.e. where these points arrive after forward sampling). Here, input point with coordinates $(\frac{p}{H-1}, \frac{q}{W-1})$ will define a color for a point at location $(\mathbf{u}_\omega[p,q], \mathbf{v}_\omega[p,q])$ in the output image. It should be imagined that springs are placed between each pair of horizontally and vertically adjacent points. Rigidity loss constrains each of the warp fields $(\mathbf{u}_\omega^i, \mathbf{v}_\omega^i)$, $i = \overline{1,..,K}$, s.t. lengths of all springs after forward sampling must be close to initial lengths of springs (namely, $\frac{1}{H-1}$ for horizontal springs, $\frac{1}{W-1}$ for vertical springs, where $H \times W$ is a resolution of input image).

$$\mathcal{L}_{\text{inpainter}}(\omega, \psi) = \alpha_{\text{pix}_2} L_{\text{pix}_2}(\omega, \psi) + \alpha_{\text{identity}} L_{\text{identity}}(\omega, \psi),$$
$$\mathcal{L}_{\text{pix}_2}(\omega, \psi) = \frac{1}{3 \cdot HW} \left\| g_\psi(f_\omega(\mathbf{x}), \mathbf{m}) - \mathbf{y}_{\text{gt}} \right\|_1,$$
$$\mathcal{L}_{\text{identity}}(\omega, \psi) = \frac{1}{K} \left\| \mathbf{v}(g_\psi(f_\omega(\mathbf{x}), \mathbf{m})) - \mathbf{v}(\mathbf{y}_{\text{gt}}) \right\|_2,$$

where $\mathbf{v}$ is an identity feature extractor. We employ the first fully-connected layer of Light-CNN-29 [56] pretrained on MS-Celeb-1M [15] dataset as a source of identity-invariant embedding of dimensionality $K = 256$. Weights of $\mathbf{v}$ are fixed during training.

$L_{\text{adv}}(\omega, \phi)$ follows the expression (3.11), and $L_{\text{gap}}(\omega, \phi)$ is defined similarly:

$$L_{\text{gap}}(\omega, \phi) = -\frac{1}{|1 - \mathbf{m}|} \left\| (1 - \mathbf{m}) \odot \log h_\xi(g_\psi(\mathbf{y}_{\text{fw}}, \mathbf{m})) \right\|_1.$$

Along with the generator, both discriminators are updated by the aforementioned losses (3.10) and (3.12).

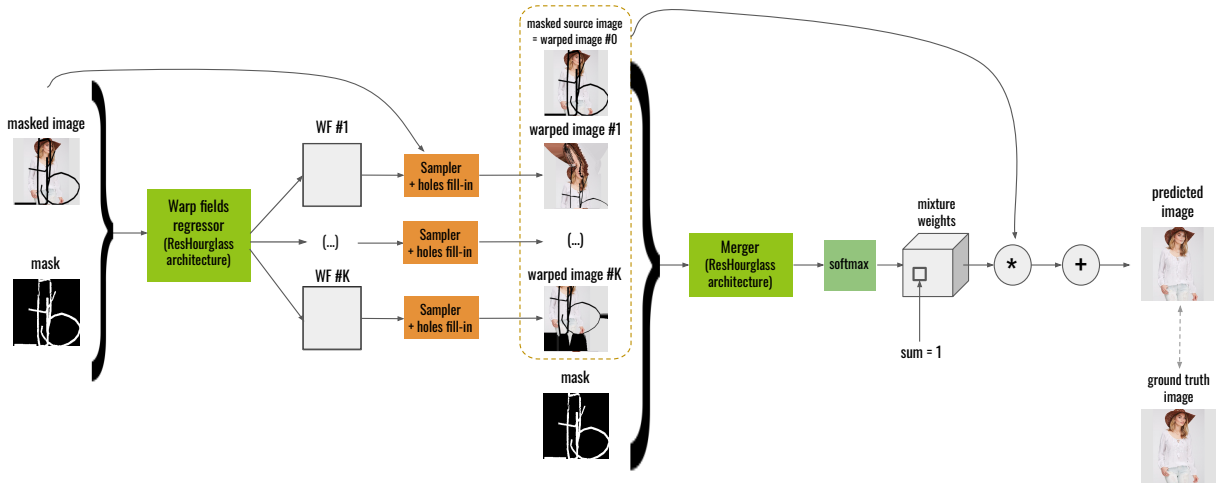


Figure 3.5: Our pipeline for image inpainting consists of two main networks — a warp fields regressor f_ω which regresses multiple warp fields based on an input image and its mask. Next, we obtain a set of warped images by a forward sampler and trivial holes fill-in. Stacking them together with the source image and a mask, we use a merger network p_η followed by softmax along depth axis to regress a mixture weights tensor $\mathbf{M}$. In this tensor, $\mathbf{M}(i, j, k)$ defines a probability of taking output pixel (i, j) from warped image $\#k$. By multiplying the tensor of warped images $\mathbf{W} = [\mathbf{x}\ \mathbf{w}_1\ \ldots\ \mathbf{w}_K]$ with the tensor $\mathbf{M}$ and summing it up by axis of images, we obtain a merged final image.

3.4 Image inpainting

We also validate our approach on a well-known image inpainting problem. Let $\mathbf{y}$ be an original image and $\mathbf{m}$ be a mask of random holes occluding some pixels of an image (value of 0 in the mask denotes an occlusion, value of 1 denotes an intact pixel). The task is to recover an origin image $\mathbf{y}$ based on a masked image $\mathbf{x} = \mathbf{y} \odot \mathbf{m}$.

One of the possible ideas is to use non-occluded pixels as a source of colors for the occluded ones. Our pipeline is depicted in Figure 3.5. The first network in the pipeline is a *warp fields regressor* f_ω, which receives $\mathbf{x}$ and $\mathbf{m}$ and outputs a set of K forward warp fields. Later on, each of the warp fields is passed to a forward sampler together with $\mathbf{x}$, followed by a trivial holes fill-in operation. This operation changes color of each hole in a warped image with its nearest neighbor on the image from a non-hole part, according to a euclidean distance between pixels, which is performed to make the subsequent inpainter network receive an input without regions of holes. Resulting warped images $\mathbf{w}_1, \ldots, \mathbf{w}_K$ reflect deformations of $\mathbf{x}$, which will be subsequently fused into a final predicted image.

Let us define a tensor $\mathbf{W}$, which is based on concatenation of $\mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_K$ along a new axis. The resulting tensor will have a shape of $(K + 1) \times 3 \times H \times W$, where $H \times W$ is a spatial resolution of $\mathbf{x}$. By squeezing the first two axes we obtain a multi-channel tensor $\overline{\mathbf{W}}$ of shape

$3(K+1) \times H \times W$, which, concatenated along first (channel) dimension with **m**, is a passed to a *merger network* p_η. Output of the latter, followed by softmax operation $\sigma(\boldsymbol{\alpha}) = \left(\frac{\exp(\alpha_i)}{\sum_k \exp(\alpha_k)}\right)_{i=1}^{n}$ along the channel axis, results in a tensor **M** of shape $(K+1) \times H \times W$. Each cell $\mathbf{M}(i,j,k)$ of **M** is now interpreted as a probability that the output pixel (i,j) from warped image $\#k$. Due to the applied softmax, a vector $(\mathbf{M}(i,j,k))_{k=1}^{K+1}$ defines a distribution. However, instead of subsequent sampling from this distribution, we employ a differentiable way of fusing images, namely, we multiply each color channel of **W** by **M**, thus obtaining a tensor of weighted images **A**, where $\mathbf{A}[:,c] = \mathbf{W}[:,c] \odot \mathbf{M}$, $c = \overline{1,..,3}$, of shape $(K+1) \times 3 \times H \times W$. Summing it up by the first (images) dimension, we receive a final output image. Another words, we "mix" warped images contained in **W** with mixture weights provided by **M**. As a result, the full pipeline is end-to-end differentiable, and an output image is a combination of pixels of a masked source image, deformed many different ways.

Additionally, the system features the same two discriminators – real/fake discriminator d_ψ and gap discriminator h_ξ – mentioned in Section 3.3, trained with the generator altogether.

Image inpainting does not have an underlying geometric structure, and pixels on an masked input image do not necessarily have counterparts on an unmasked output image. However, for masks which are tiny enough, pixels can be copied from their nearest neighbors or semantically related regions. Such an approach is known to be quite popular in the literature [1], and it motivates the use of differentiable image deformations parameterized by neural networks as an approach to this task.

Loss functions. The whole generating network $F_{\omega,\eta}(\mathbf{x}, \mathbf{m})$, transforming an input image **x** and a mask **m** into a final predicted image, is trained along with two discriminators mentioned above, being constrained by a loss $\mathcal{L}(\omega, \eta)$ (3.13). There, P corresponds to a network of VGG-16 architecture with the weights pretrained on ImageNet, and P_k denotes flattened activations of **relu1_2**, **relu2_2**, **relu3_3**, **relu4_3** layers of P for $k = 1,..,4$, respectively. $G_{ij}(P_k(F_{\omega,\eta}(\cdot,\cdot)))$ is a Gram matrix of a feature vector corresponding to (i,j) position in a feature tensor returned by P_k, and L_k refers to a number of elements in G_{ij} for any i, j.

$$
\begin{aligned}
\mathcal{L}(\omega, \eta) &= \alpha_{L_1}\mathcal{L}_{L_1} + \alpha_{content}\mathcal{L}_{content} + \alpha_{style}\mathcal{L}_{style} + \alpha_{rigidity}\mathcal{L}_{rigidity} + \alpha_{adv}\mathcal{L}_{adv} \\
&+ \alpha_{gap}\mathcal{L}_{gap} \\
\mathcal{L}_{L_1}(\omega, \eta) &= \frac{1}{3 \cdot HW}\|F_{\omega,\eta}(\mathbf{x}, \mathbf{m}) - \mathbf{y}\|_1 \\
\mathcal{L}_{content}(\omega, \eta) &= \frac{1}{8 \cdot HW}\sum_{k=1}^{4}\|P_k(F_{\omega,\eta}(\mathbf{x}, \mathbf{m})) - P_k(\mathbf{y})\|_2 \\
\mathcal{L}_{style}(\omega, \eta) &= \frac{1}{HW}\sum_{k=1}^{4}\frac{1}{2^{6-k}}\sum_{i,j}\left\|\frac{1}{L_k}\left[G_{ij}(P_k(F_{\omega,\eta}(\mathbf{x}, \mathbf{m}))) - G_{ij}(P_k(\mathbf{y}))\right]\right\|_F^2 \\
\mathcal{L}_{rigidity}(\omega, \eta) &= \frac{1}{K}\sum_{i=1}^{K}\left(\left|\sqrt{(\mathbf{u}^i_\omega[p+1, q] - \mathbf{u}^i_\omega[p, q])^2 + (\mathbf{v}^i_\omega[p+1, q] - \mathbf{v}^i_\omega[p, q])^2 + \varepsilon} - \frac{1}{H-1}\right|\right. \\
&\left. + \left|\sqrt{(\mathbf{u}^i_\omega[p, q+1] - \mathbf{u}_\omega[p, q])^2 + (\mathbf{v}^i_\omega[p, q+1] - \mathbf{v}^i_\omega[p, q])^2 + \varepsilon} - \frac{1}{W-1}\right|\right)
\end{aligned}
\tag{3.13}
$$

Content and style losses are examples of perceptual losses first introduced in [28] to less penalize images which are closer according to activations of a network trained on a large corpus of images for their better classification. Adversarial and perceptual losses are known to be especially helpful for realistic image synthesis [42].

$$
\begin{aligned}
\mathcal{L}^G_{\text{adv}}(\omega, \eta) &= \frac{1}{HW}\|1 - F_{\omega,\eta}(\mathbf{x}', \mathbf{m}')\|_2^2 \\
\mathcal{L}^G_{\text{gap}}(\omega, \eta) &= -\frac{1}{HW}\|\log(1 - h_\xi(F_{\omega,\eta}(\mathbf{x}', \mathbf{m}')) + \varepsilon)\|_1
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}^D_{\text{adv}}(\psi) &= \alpha_{adv} \cdot \frac{1}{2 \cdot H_D W_D}\left(\|d_\psi(F_{\omega,\eta}(\mathbf{x}', \mathbf{m}'))\|_2^2 + \|1 - d_\psi(F_{\omega,\eta}(\mathbf{x}', \mathbf{m}'))\|_2^2\right) \\
\mathcal{L}^D_{\text{gap}}(\xi) &= -\alpha_{gap} \cdot \frac{1}{HW}\left(\|(1 - \mathbf{m}) \odot \log(1 - h_\xi(F_{\omega,\eta}(\mathbf{x}', \mathbf{m}')) + \varepsilon)\|_1\right. \\
&\left. + \|\mathbf{m} \odot \log(h_\xi(F_{\omega,\eta}(\mathbf{x}', \mathbf{m}') + \varepsilon)\|_1\right),
\end{aligned}
$$

where (H_D, W_D) is a resolution of output tensor of patch discriminator d_ψ. Here we used LS-GAN [38] losses construction for more stable and robust training with d_ψ.

Chapter 4

Experiments

4.1 Face rotation

As a first task, we consider face frontalization approach, where the task is to warp the non-frontally oriented face into a frontalized one, while preserving the identity, the facial expression, and the lighting.

4.1.1 Datasets

We train and evaluate our method on Multi-PIE [14], which is a dataset of upper body photos of more than 750,000 images of 337 people. For each subject, 15 views were taken simultaneously by multiple cameras, 13 of which were placed around the subject at regular intervals of $15°$, ranging from $-90°$ to $90°$, in the same horizontal plane, and 2 on an elevated level. Each multi-view collection was taken with 19 various illumination conditions, up to 4 sessions and 4 facial expressions. In our experiments, we used only 13 cameras placed around the subject in the same horizontal plane. For the upper body experiments, raw images were used, while for the experiments with face we used MTCNN [63] face detector to find a corresponding bounding box and crop it with a gap of 10 pixels. 128×128 was the standard resolution we were working with, and all images were finally resized to that resolution before passing to the learning algorithm. We consider frontalization the most important particular case of the rotation task for our experiments (the method, however, can be adapted to arbitrary rotations).

4.1.2 Compared Models

Our pipeline was implemented in 3 different ways: with forward warping and gap discriminator, with forward warping alone and with conventional backward warping instead of a forward one. We chose architecture of U-Net [45] kind for a warp field regressor f_ω, which is an FCN with an encoder and a decoder. Encoder consists of 6 convolutional blocks interleaved by 2x max pooling, while each block is a concatenation of 3 convolutions with BatchNorm and ReLU after each one. This way, output of an encoder is a 32x less image by each side. Decoder is symmetric to the encoder, except that each max pooling is replaced by a 2x bilinear upsampling. An inpainter

g_ϕ is a network of an architecture similar to U-Net but built upon gated convolutions described earlier, not plain ones. Here, we fully adopt the architecture described in [60] in order to maintain the same image inpainting performance level. A real/fake discriminator d_ψ is a concatenation of 6 convolutions with a stride of 2 interleaved by InstanceNorm and Leaky ReLU. This is a patch discriminator, which means that the output tensor is average pooled, and sigmoid is applied afterwards. Gap disciminator h_ξ also follows U-Net style, namely, its encoder consists of 4 convolutions with subsequent InstanceNorm and Leaky ReLU, interleaved by 2x max poolings, and its decoder is symmetric to the encoder, where each max pooling is replaced by 2x bilinear upsampling. The warp field regressor f_ω contains 3'020'993 learning parameters, the inpainter g_ϕ — 5'640'742 parameters, real/fake patch discriminator d_ψ — 536'000 parameters, gap discriminator h_ξ — 869'025 parameters. All networks are trained end-to-end by optimizing all losses described in Sec. 3.3 via Adam optimization method. The following loss weights were taken: $\{\alpha_{pix_1} = 0.1,\ \alpha_{rigidity} = 0.01,\ \alpha_{pix_2} = 1.0,\ \alpha_{adv} = 0.1, \alpha_{gap} = 5 \cdot 10^{-2}, \alpha_{identity} = 0.1\}$.

Apart from that, we compare with the stack of works claiming state-of-the-art results for the same task of face frontalization [21, 19, 65, 58, 59] and approaching it with various methods, mainly based on modifications of FCNs, GANs and feature encoding and aggregation tricks.

4.1.3 Metrics and Results

Results of the experiments can be evaluated based on the following data. Fig. 4.1 depicts results of warping and inpainting for the network with forward sampler and gap discriminator. Fig. 4.2 shows a comparison between investigated approaches and demerits of each of them at distinct rotation angles. Table 4.1 supports the latter figure with the quantitative assessment. First of all, the experiments support the methodology as a competitive one for the investigated task. Despite that the table shows backward warping winning according to the majority of metrics, it is hard to decide on the best method. Indeed, forward warping with gap discriminator seems to be dealing with some of the extreme angles relatively well.

It is important to emphasize that the dataset contained only 13 rotation angles which could be remembered by the algorithm, and such a knowledge can be misused by the algorithm to approximate the answer by initial angle guessing. This suggests a potential use of the method for datasets with continuous set of objects rotations in the data.

Additionally, we compare against state-of-the-art methods and report the results in Table 4.2 according to Rank-1 face recognition accuracy, based on the numbers reported in the corresponding works. A conventional testing protocol followed in [21, 19, 65] was used (usually referred to Setting-2 in these works). This protocol is concentrated on pose, illumination and session variations, and is considered the hardest version among two protocols employed for the task. Overall,

images with neutral expression from all four sessions, which contains 337 identities, were filtered out, as required by the protocol. The images with 11 poses within all the rotation range $\pm 90^\circ$ and 20 illumination levels of the first 200 identities are used for training. During testing, each predicted image is compared against a frontal view of the same person with neural illumination. Apart from standard methods suited for the task, Light CNN was additionally included as a baseline, as preferred by the other works. We observe superiority of our method equipped with backward warping at several angles, including extreme $\pm 90^\circ$ and ones corresponding to a small rotation. At the same time, forward warping based method also delivers high recognition rates.

Figure 4.1: Face frontalization via forward warping. Here, the algorithm equipped with forward sampler and gap discriminator is trained on samples of 80% of randomly taken subjects from Multi-PIE is visually assessed based on two randomly picked subjects from validation part. Each input photo (1^{st} row — *input*) is independently passed through the warp field regressor yielding a warped image (2^{nd} row — *warped*; holes are painted black) and then through the inpainter yielding an inpainted image with holes filled in, warping mistakes corrected, and lighting normalized (3^{rd} row — *inpainted*). Target image is in the centre of each row and corresponds to 0° rotation angle (encircled in a red square). Additionally, for the second sample a set of nonholes masks (4^{th} row — *nonholes mask*; black = hole, white = non-hole) and a set of corresponding masks predicted by gap discriminator are reported. Clearly, here generator "wins" in an adversarial game with a discriminator, however, the latter makes inpainted regions indistinguishable from transferred, non-inpainted ones. *Electronic zoom-in recommended.*

	MSE			IS			FID		
	FW+GAP	FW	BKWD	FW+GAP	FW	BKWD	FW+GAP	FW	BKWD
$-90°$	0.0112	0.0108	**0.0105**	1.630	1.673	**1.717**	**15.541**	16.494	15.973
$-75°$	0.0107	0.0102	**0.0099**	1.668	1.753	**1.777**	14.824	15.441	**14.453**
$-60°$	0.103	0.010	**0.0093**	1.650	1.698	**1.754**	14.398	12.710	**11.689**
$-45°$	0.0097	0.0090	**0.0087**	1.690	1.749	**1.757**	14.250	11.858	**9.824**
$-30°$	0.0089	0.0082	**0.0080**	1.713	1.774	**1.790**	11.128	11.308	**9.1824**
$-15°$	0.0074	0.0070	**0.0066**	**1.812**	1.774	1.781	11.769	10.236	**7.8085**
$0°$	0.0010	0.0009	**0.0007**	**1.879**	1.815	1.860	3.450	**2.4101**	3.0101
$15°$	0.0077	0.0070	**0.0068**	1.783	**1.880**	1.864	11.063	9.9621	**7.6996**
$30°$	0.0085	0.0081	**0.0079**	1.791	1.830	**1.855**	12.348	12.859	**9.8979**
$45°$	0.0093	0.0089	**0.0086**	1.765	1.753	**1.779**	13.936	14.357	**10.430**
$60°$	0.0099	0.0095	**0.0093**	**1.714**	1.681	1.710	15.317	15.616	**11.660**
$75°$	0.0105	0.0102	**0.0101**	**1.739**	1.684	1.712	15.093	17.367	**12.733**
$90°$	0.0107	**0.0104**	**0.0104**	**1.691**	1.615	1.650	16.233	19.547	**15.515**

Table 4.1: Quantitative comparison of 3 methods based on the proposed approach: the one with forward warping and gap discriminator (*FWD + GAP*), with forward warping alone (*FWD*), and with backward warping instead of forward one (*BKWD*). Three metrics were employed: Mean Squared Error (MSE) and two perceptual metrics: Inception Score (IS) and Fréchet Inception Distance (FID).

Rank-1 recognition accuracy	$\pm 90°$	$\pm 75°$	$\pm 60°$	$\pm 45°$	$\pm 30°$	$\pm 15°$
Light CNN [56]	5.5	24.2	62.1	92.1	97.4	98.6
CPF [58]	—	—	61.9	79.9	88.5	95.0
TP-GAN [21]	64.64	77.43	87.72	95.38	98.06	98.68
FF-GAN [59]	61.2	77.2	85.2	89.7	92.5	94.6
CAPG-GAN [19]	66.1	83.1	90.3	97.3	99.6	99.8
PIM [65]	86.5	**95.0**	**98.1**	98.5	99.0	99.3
Our method w/ FWD	69.3	70.4	71.1	73.7	75.7	97.7
Our method w/ BKWD	**88.3**	92.6	96.8	**99.4**	**99.9**	**100.0**

Table 4.2: Quantitative comparison of our method with forward warping (FWD) and with backward warping (BKWD) with other state-of-the-art methods for face rotation on Multi-PIE according to Rank-1 face recognition accuracy. A conventional testing protocol followed in [21, 19, 65] and other works was used (usually referred to Setting-2 in these works). Overall, images with neutral expression from all four sessions, which contains 337 identities, were filtered out. The images with 11 poses within all the rotation range $\pm 90°$ and 20 illumination levels of the first 200 identities are used for training. During testing, each predicted image is compared against a frontal view of the same person with neural illumination. We observe superiority of our method equipped with backward warping at several angles, including extreme $\pm 90°$ and ones corresponding to a small rotation. At the same time, forward warping based method also delivers high recognition rates.

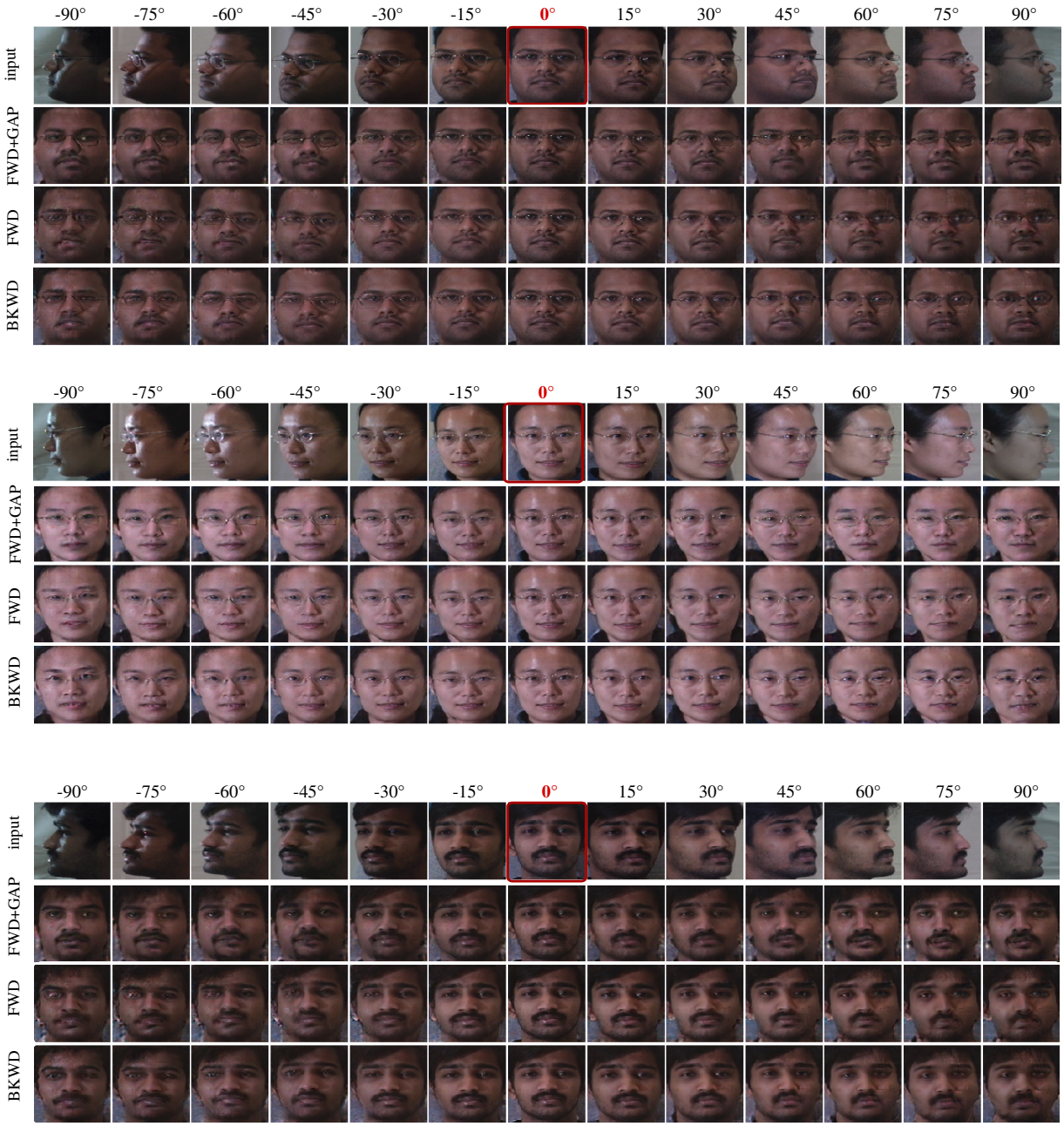


Figure 4.2: Comparison of face frontalization approaches: the one with forward warping and gap discriminator (*FWD + GAP*), with forward warping alone (*FWD*), and with backward warping instead of forward one (*BKWD*). Three validation samples were taken; for each sample, first row contains input image given to each network (*input*), and target image is in the centre of each row and corresponds to $0°$ rotation angle (encircled in a red square). Note the different performance of algorithms on the extreme angles, such as $-90°$ and $90°$. For the model equipped with gap discriminator, the desired effect was obtained, as it is hard to find any artifacts or inaccuracies which would reveal inpainted pixels compared to non-inpainted ones. *Electronic zoom-in recommended.*

4.2 Image inpainting

4.2.1 Datasets

We experimented on several datasets, which let us evaluate the performance of the proposed inpainting algorithm in several situations. In this work, the main focus was to test the approach on

Figure 4.3: Examples of masks from QD-IMD dataset.

data with semantically similar images, such as images of human faces and full-length photographs of human body. This allows the method to learn semantic correspondences between various parts of an image. Towards this end, we selected 2 different datasets, described below.

1. CelebA [35] — a large-scale dataset of diverse face images of celebrities with auxiliary information (landmarks and facial attributes). We use an aligned version, where each image was aligned using similarity transformation according to the two eye locations, such that eyes, mouth and other face parts are located in similar positions across the dataset. We used a random separation of the dataset into train and validation in 70% / 30% proportions.

2. DeepFashion [34] — a dataset of full-height body images of people, s.t. for each person wearing specific clothes there are several (4–8) photos in distinct poses. An in-shop clothes part was used, which contains 7'982 clothing items, 52'712 in-shop clothes images, and over 200'000 cross-pose/scale pairs.

In addition, we use QD-IMD [24] dataset as a source of random masks for inpainting. These masks are based on sketches drawn by various people collected from Quick-Draw dataset by Google [12], which comprises more than 50 million human drawings. Each mask is a combination of strokes drawn with uniformly selected brush width fit in a bounding box with a uniformly sampled side length. Consequently, such a mask is close to a collection of human-made, natural brush movements, and training with set of masks of this kind is known to be helpful for general improvement of visual quality of inpainted images [24]. Examples of masks are shown in Fig. 4.3. A publicly available subset of 50'000 training masks and 50'000 testing masks was used for training and testing the algorithms, respectively.

Inpainting of unknown regions on human body and face images is an important auxiliary task for pose and view resynthesis of human images. For instance, the task of obtaining images of human body in a different pose is solved in Dense Pose Transfer [41], which at some point comes up with partially known predictions and performs the subsequent inpainting of occluded regions.

4.2.2 Compared Models

Here, the approach described in Section 3.4 is applied in 2 variants. The first one is the method without changes, i.e. the one which employs a forward sampler for applying to deformations predicted

by f_ω to a masked source image $\mathbf{x}$. Another one is using backward sampler instead of a forward one. For the latter, all losses can be preserved, including the introduced rigidity loss, for which the same expression is used. The only difference is that the warp fields $(\mathbf{u}^i_\omega, \mathbf{v}^i_\omega)^K_{i=1} = f_\omega(\mathbf{x}, \mathbf{m})$ are interpreted as the backward warp fields, and thus, the rigidity loss will constrain the length of deformed springs in a space of input image, not the output image (see Fig. 3.4). Additionally, we compare against a method based on fully-convolutional network with gated convolutions [60] which is considered one of state-of-the-art image inpainting of general kind, additionally known to be well-performing on facial images. For fairness of comparison, both our networks and a baseline trained on the same dataset in each experiment, which might limit the potential capabilities of the network based on gated convolutions prepared to be a universal inpainter. A full solution described in the paper [60] was implemented, including patch discriminator and perceptual losses employed.

4.2.3 Metrics and Results

It should be particularly noted that such an approach is mostly valuable when the masks are mainly tiny, but there is a need to preserve image sharpness as much as possible.

We chose a working resolution of 256×256 for DeepFashion, and of 128×128 for CelebA (the method, however, can be used at larger resolutions). All images were resized to the chosen resolutions in advance.

Architecture of a warp field regressor f_ω is a *residual hourglass* network, which consists of an encoder, several residual blocks and a decoder, all built upon convolutions and transposed convolutions. More formally, the network contains 6 convolutional blocks in the encoder (two of them contain stride-2 convolutions), 12 — in the residual part (here, input to each convolutional block is subsequently added to its output by a residual connection), 6 — in the decoder part (there is a bilinear 2x upsampling before two of them). Each convolutional block contains one 3×3 convolution, InstanceNorm normalization layer, and Leaky ReLU activation. Conceptually, such kind of architecture is well-proven to be powerful for image-to-image translation tasks [54, 13]. Merger network p_η is based on the same type of architecture but with a 4x larger number of feature maps in each layer. Gap discriminator follows more simple *hourglass* concept, which can be represented as a stack of 16 aforementioned convolutional blocks, such that two of them in the beginning have stride of 2, and with two bilinear 2x upsamplings in the latest layers. Patch discriminator contains 4 convolutions with Spectral Normalization [39] on top of it, which makes GAN training a more stabilized process, BatchNorm normalization and Leaky ReLU, followed by 1 convolution with Spectral Normalization and sigmoid; here we borrow insights from [60]. Method is trained for 100'000 steps with a learning rate of $2 \cdot 10^{-5}$ for generator and 10^{-5} for each of the discriminators. Loss weights were selected as $\{\alpha_{L_1} = 10,\ \alpha_{content} = 2{\cdot}10^{-2},\ \alpha_{style} = 5{\cdot}10^{4},\ \alpha_{rigidity} = 1,\ \alpha_{adv} =$

10^{-1}, $\alpha_{gap} = 10^{-5}\}$. Number of warp fields was chosen to be 10; still, some of them might be excessive for many input samples. Given the described setting, warp fields regressor f_ω contains 981'140 learning parameters, merger network p_η — 15'584'523 parameters; discriminators are generally more lightweight, namely, real/fake discriminator d_ψ contains 2'783'797 parameters, gap discriminator h_ξ contains 869'025 parameters.

Our comparison against the baseline includes both visual assessment (see Fig. 4.4, Fig. 4.6) and quantitative experiments w.r.t. several perceptual metrics of image quality and similarity, which are popular in the field (see Table 4.3, Table 4.4). In addition, we report the dependency of warped images on the rigidity loss weight $\alpha_{rigidity}$ in Fig. 4.5. Clearly, the proposed models outperform a baseline by both metrics and visual quality. According to the model design, it can only copy or mix pixels persistent on an input image, which makes the results the same sharp as the input images themselves. On the contrary, baseline converts an input image and a mask to a internal latent representation and recovers an output image based on it, as expected. This yields semantically correct but more blurry results, and artefacts and imprecise details rendering can take place. Interestingly, backward warping performs a little better than forward warping according to the most metrics, and on a similar level according to the visual quality.

Investigation of the results in comparison to the general-purpose baseline further suggests that the proposed methods are better suited for masks with narrow strokes but when the sharpness of an image needs to be preserved as much as possible. Nevertheless, we observe that the methods deal with spacious masks reasonably well.

	MSE ↓	SSIM ↑	IS ↑	FID ↓
Ours w/ forward warping with gap discriminator	$\mathbf{6.5 \cdot 10^{-4}}$	0.983	**3.841**	7.532
Ours w/ forward warping	$6.8 \cdot 10^{-4}$	0.983	3.788	7.741
Ours w/ backward warping	$\mathbf{6.5 \cdot 10^{-4}}$	**0.984**	3.728	**6.817**
Baseline: GatedConvNet [60]	$4.9 \cdot 10^{-3}$	0.891	3.582	32.202

Table 4.3: Comparison of approaches with an interchangeable sampler against the baseline on the validation part of DeepFashion dataset. Arrow indicates if the score is the more the better or the lower the better.

	MSE ↓	SSIM ↑	IS ↑	FID ↓
Ours w/ forward warping & gap discriminator	$3.4 \cdot 10^{-4}$	0.982	3.613	3.501
Ours w/ forward warping	$3.3 \cdot 10^{-4}$	0.982	3.623	3.489
Ours w/ backward warping	$\mathbf{3.1 \cdot 10^{-4}}$	**0.984**	**3.654**	**3.296**
Baseline: GatedConvNet [60]	$5 \cdot 10^{-3}$	0.828	3.522	18.580

Table 4.4: Comparison of approaches with an interchangeable sampler against the baseline on the validation part of CelebA dataset. Arrow indicates if the score is the more the better or the lower the better.

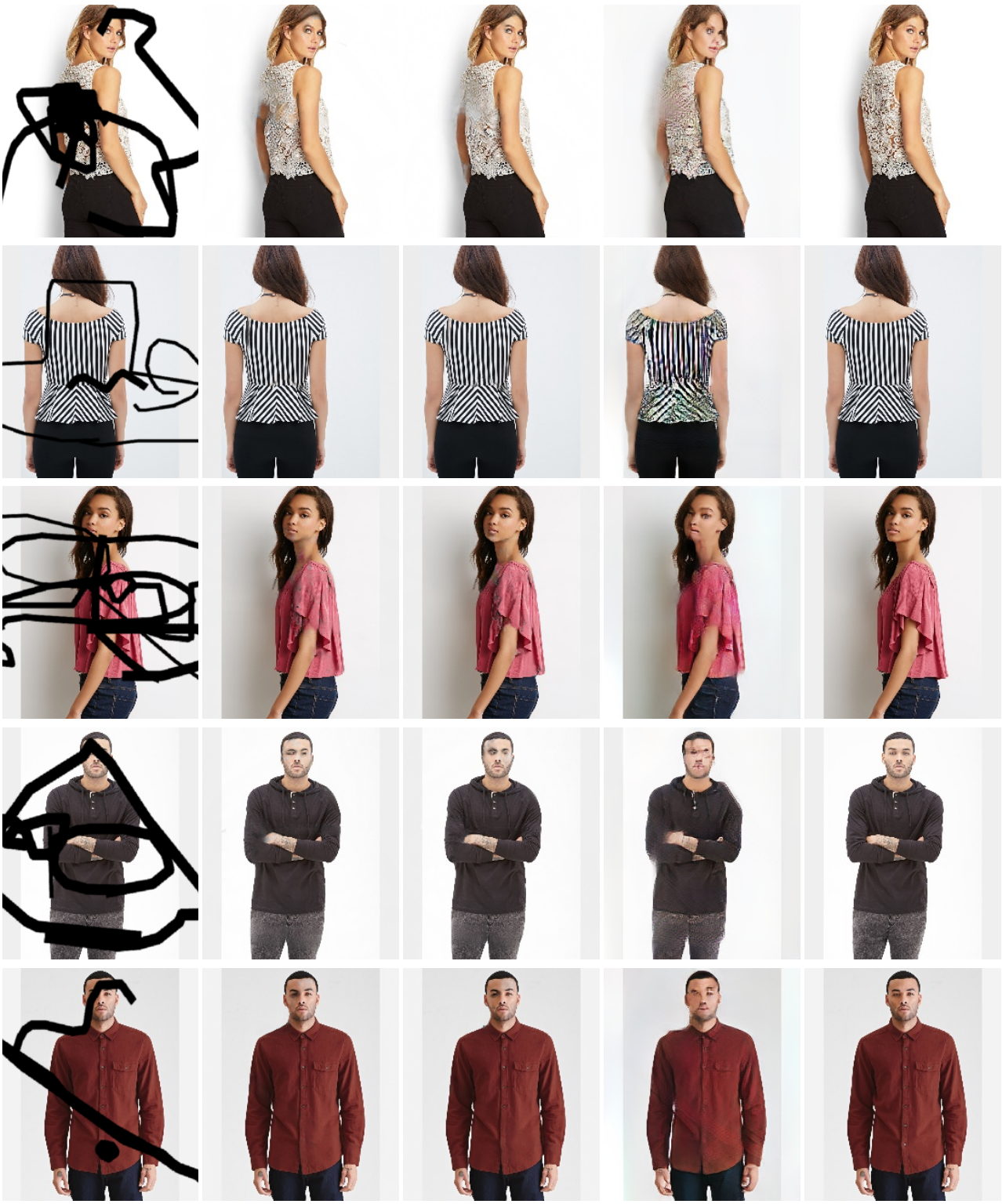
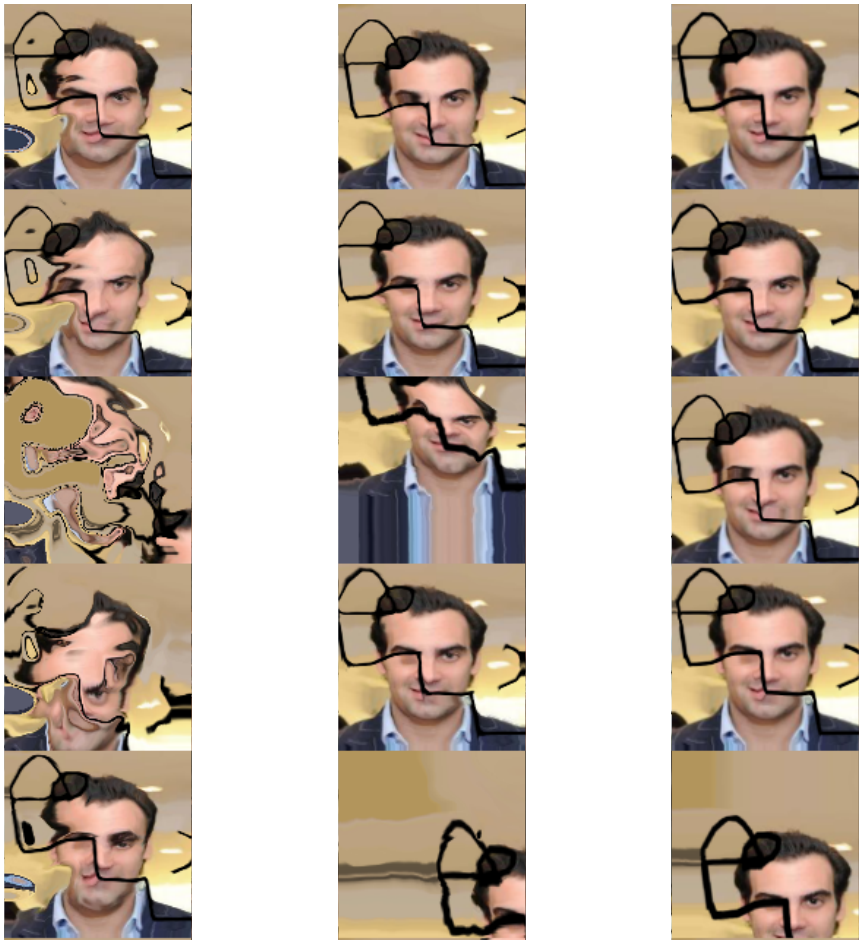


Figure 4.4: Side-by-side comparison with state-of-the-art (first eight samples from the test set). In each row we show source image (Source), predicted by a network based on forward sampler (FWD), predicted by a network based on conventional backward sampler (BKWD), predicted by a GatedConvNet baseline [60], ground truth in the target pose (GT). Consistently with the rest of validation samples, our method is more robust and has less artifacts than one of the state-of-the-art general-purpose image inpainting networks [60] used as a baseline. *Electronic zoom-in recommended.*



$\alpha_{rigidity} = 0$ $\alpha_{rigidity} = 10^{-1}$ $\alpha_{rigidity} = 1$

Figure 4.5: Examples of warped images $\mathbf{w}_1, \ldots, \mathbf{w}_K$ which occur after neural network training on CelebA dataset with 3 different values of rigidity loss weight $\alpha_{rigidity}$: 0, 10^{-1}, and 1. One can see how $\alpha_{rigidity}$ controls the allowed arbitrariness and non-linearity of the predicted warpings.

Figure 4.6: Side-by-side comparison with state-of-the-art (first eight samples from the test set) on a CelebA dataset of facial images. In each row we show source image (Source), predicted by a network based on forward sampler with gap discriminator used (FWD + GAP), predicted by a network based on forward sampler alone (FWD), predicted by a network based on conventional backward sampler (BKWD), predicted by a GatedConvNet baseline [60], ground truth in the target pose (Target). Consistently with the rest of validation samples, our method is more robust and has less artifacts than one of the state-of-the-art general-purpose image inpainting networks [60] used as a baseline. *Electronic zoom-in recommended.*

Chapter 5

Discussion

Our work suggests several ways to adapt warpings to standard computer vision problems, either to the ones based on pixels shift or the ones where this can be beneficial. We show that these warpings can be successfully parameterized and predicted by neural networks without any direct supervision. Deformations can be either predicted without a spatial alignment with an input image (in a conventional way well-known to computer vision community), or with a full spatial alignment with an input image but with inevitable holes produced (in a conceptually new way proposed in this thesis). Additionally, we give a way to control non-linearity of deformations. A gap discriminator was suggested as a new universal tool for hiding artifacts in holes regions obtained by forward warping and other transformations; apart from that, it also contributes to adversarial image inpainting.

The proposed methodology can be applied to various computer vision tasks, either lacking spatial alignment or implicitly requiring pixels shift. We would like to enlist several problems of such kind, practically important at the moment, as our future work directions.

- **Face-to-texture for 3D reconstruction.** Face photo can be geometrically deformed to a part of UV texture – canonical pose-invariant representation of a face surface. Inferred face texture and face shape define a 3D model, which can be used to model 3D face avatars, valuable for VR/AR and mobile technologies.

- **Resynthesis of human body images with changed pose.** The task is to transform a photo of a person to another photo of this person standing in another pose. This application can be very valuable in VR/AR technologies and telepresence.

- **Video frames prediction.** The task is to predict future or intermediate frames of a video based on given source frames, which would be useful for many applications of computer vision in biometry, cinema and realistic rendering of 3D shapes.

- **Unsupervised visual attention.** The proposed ideas can be employed to find warpings of an image which make the most important information more visible and positionally normalized. This way, spatial alignment can be introduced to algorithms which implicitly encourage that.

For instance, it would be beneficial to find important symmetries of clothes on a human photo for the task of image inpainting and human pose change.

- **Prediction of pixels movement for high-resolution images.** Deformations parameterized by forward warp fields can be predicted the most efficiently by fully-convolutional networks, and thus, a quality gap might be observed for high-resolution images processed by networks with relatively small effective receptive field.

A publication touching upon the related tasks of face pose and direction change, human pose change and human garment transfer was accepted to CVPR 2019 conference and will be published in the conference proceedings [13]. In particular, the author's contribution was concentrated on face resynthesis in other head positions and with different mimics.

Appendix A

Expression for Gradient of Forward Sampling Operation Result

In this appendix we revisit forward warping operation and provide implicit expression for gradients of forward warping result w.r.t. its inputs (an input image and a warp field). This is required for performing backpropagation of gradients through this operation, in case it is integrated into the neural network based constructions as a layer. Since forward warping can be seen as a differentiable indexing operation, the main non-trivial element of this operation is a non-zero gradient w.r.t. a warp field.

More formally, we perceive forward warping as a function of an input image and a warp field, returning another image. In this regard, it would be convenient to use a function-style notation: $\mathcal{W}_F : \mathbb{R}^{C \times H \times W} \times (\mathbb{R}^{H \times W}, \mathbb{R}^{H \times W}) \to \mathbb{R}^{C \times H \times W}$. Let $\mathbf{y} = \mathcal{W}_F(\mathbf{x}, (\mathbf{u}, \mathbf{v})) \in \mathbb{R}^{C \times H \times W}$.

Then, the expression which defines $\mathbf{y}$ is given by

$$\mathbf{y}^c_{pl} = \frac{\sum\limits_n^H \sum\limits_m^W \mathbf{x}^c_{nm} \max(0, 1 - |\mathbf{u}_{nm} - p|) \max(0, 1 - |\mathbf{v}_{nm} - l|)}{\sum\limits_n^H \sum\limits_m^W \max(0, h - |\mathbf{u}_{nm} - p|) \max(0, 1 - |\mathbf{v}_{nm} - l|)} = \frac{\mathbf{A}^c_{pl}}{\mathbf{w}_{pl}},$$

where $\mathbf{x}^c_{nm}$ defines an intensity of (n, m) point in image $\mathbf{x}$ on a channel c, and tensors $\mathbf{A}$ and $\mathbf{w}$ were introduced as a numerator and denominator of the fraction, respectively.

$$\frac{\partial \mathbf{A}^c_{pl}}{\partial \mathbf{x}^c_{nm}} = \sum_n^H \sum_m^W \max(0, 1 - |\mathbf{u}_{nm} - p|) \max(0, 1 - |\mathbf{v}_{nm} - l|)$$

$$\frac{\partial \mathbf{A}^c_{pl}}{\partial \mathbf{u}_{nm}} = \sum_n^H \sum_m^W \mathbf{u}_{nm} \max(0, 1 - |\mathbf{v}_{nm} - l|) \cdot \begin{cases} 0, & \text{if } |p - \mathbf{u}_{nm}| \geq 1 \\ 1, & \text{if } |p - \mathbf{u}_{nm}| < 1 \text{ and } \mathbf{u}_{nm} \geq p \\ -1, & \text{if } |p - \mathbf{u}_{nm}| < 1 \text{ and } \mathbf{u}_{nm} < p \end{cases}$$

$$\frac{\partial \mathbf{w}_{pl}}{\partial \mathbf{x}^c_{nm}} = 0$$

$$\frac{\partial \mathbf{w}_{pl}}{\partial \mathbf{u}_{nm}} = \sum_{n}^{N} \sum_{m}^{N} \max(0, 1 - |\mathbf{v}_{nm} - p|) \begin{cases} 0, & \text{if } |p - \mathbf{u}_{nm}| \geq 1 \\ 1, & \text{if } |p - \mathbf{u}_{nm}| < 1 \text{ and } \mathbf{u}_{nm} \geq p \\ -1, & \text{if } |p - \mathbf{u}_{nm}| < 1 \text{ and } \mathbf{u}_{nm} < p \end{cases}$$

Derivatives of **A** and **w** over $\mathbf{v}_{nm}$ are defined similarly to the derivatives over $\mathbf{u}_{nm}$.

The summations in all formulas (both in the gradients and in the expression itself) are making time complexity of straightforward calculation $\mathcal{O}((HW)^2)$. Nevertheless, one can notice that bilinear kernels are talking non-zero values only in a neighborhood of radius 1 within $(\mathbf{u}_{nm}, \mathbf{v}_{nm})$ point in an output image, which can contain up to 4 points. Hence, it is possible to implement forward sampling and its gradients in 4 passes over the input image, making the resulting complexity more much affordable $\mathcal{O}(HW)$.

The operation itself is sub-differentiable and can be accurately implemented via automatic gradient computation in some computational frameworks, if a necessary set of operations is available (e.g. in PyTorch).

Bibliography

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

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

[3] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. Load balanced gans for multi-view face image synthesis. *arXiv preprint arXiv:1802.07447*, 2018.

[4] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *arXiv preprint arXiv:1806.08472*, 2018.

[5] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: adversarial facial uv map completion for pose-invariant face recognition. In *Proc. CVPR*, pages 7093–7102, 2018.

[6] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[8] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[9] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European Conference on Computer Vision*, pages 311–326. Springer, 2016.

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

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] Google. Quick, draw! dataset. https://quickdraw.withgoogle.com/data.

[13] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided image generation. *arXiv preprint arXiv:1811.11459*, 2018.

[14] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[15] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[18] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5135–5143, 2015.

[19] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. Pose-guided photorealistic face rotation. In *Proc. CVPR*, 2018.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[21] Rui Huang, Shu Zhang, Tianyu Li, Ran He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv preprint arXiv:1704.04086*, 2017.

[22] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[24] Karim Iskakov. Semi-parametric image inpainting. *arXiv preprint arXiv:1807.02855*, 2018.

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proc. NIPS*, pages 2017–2025, 2015.

[26] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2155–2163, 2017.

[27] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017.

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711, 2016.

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

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[33] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018.

[34] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proc. CVPR*, pages 1096–1104, 2016.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, pages 3730–3738, 2015.

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[37] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[38] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[40] Kamyar Nazeri and Eric Ng. Image colorization with generative adversarial networks. *arXiv preprint arXiv:1803.05400*, 2018.

[41] Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[42] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proc. CVPR*, pages 702–711. IEEE, 2017.

[43] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2015.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

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

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[47] P. Sharma and A. Singh. Era of deep neural networks: A review. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5, July 2017.

[48] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. *arXiv preprint arXiv:1806.06503*, 2018.

[49] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[50] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

[52] Juan P Vigueras-Guillén, Busra Sari, Stanley F Goes, Hans G Lemij, Jeroen van Rooij, Koenraad A Vermeer, and Lucas J van Vliet. Fully convolutional architecture vs sliding-window cnn for corneal endothelium cell segmentation. *BMC Biomedical Engineering*, 1(1):4, 2019.

[53] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015.

[54] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[55] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–686, 2018.

[56] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[57] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–17, 2018.

[58] Junho Yim, Heechul Jung, ByungIn Yoo, Changkyu Choi, Dusik Park, and Junmo Kim. Rotating your face using multi-task deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 676–684, 2015.

[59] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *Proc. ICCV*, pages 1–10, 2017.

[60] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[61] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018.

[62] Xu Zexiang, Bi Sai, Sunkavalli Kalyan, Hadap Sunil, Su Hao, and Ramamoorthi Ravi. Deep view synthesis from sparse photometric images. In *Proceedings of SIGGRAPH (Conditionally accepted)*.

[63] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[64] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[65] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2207–2216, 2018.

[66] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *Proc. ECCV*, pages 286–301, 2016.