Deep Feature Interpolation for Image Content Changes

Paul Upchurch[1,2], Jacob Gardner[1,2], Kavita Bala[2], Robert Pless[3], Noah Snavely[2], and Kilian Weinberger[2]

[1]Authors contributed equally
[2]Cornell University
[3]Washington University in St. Louis

Abstract

We propose Deep Feature Interpolation (DFI)*, a new data-driven baseline for automatic high-resolution image transformation. As the name suggests, it relies only on simple linear interpolation of deep convolutional features from pre-trained convnets. We show that despite its simplicity, DFI can perform high-level semantic transformations like "make older/younger", "make bespectacled", "add smile", among others, surprisingly well—sometimes even matching or outperforming the state-of-the-art. This is particularly unexpected as DFI requires no specialized network architecture or even any deep network to be trained for these tasks. DFI therefore can be used as a new baseline to evaluate more complex algorithms and provides a practical answer to the question of which image transformation tasks are still challenging in the rise of deep learning.*

1. Introduction

Generating believable changes in images is an active and challenging research area in computer vision and graphics. Until recently, algorithms were typically hand-designed for individual transformation tasks and exploited task-specific expert knowledge. Examples include transformations of human faces [37, 16], materials [2, 1], color [46], or seasons in outdoor images [22]. However, recent innovations in deep convolutional auto-encoders [29] have produced a succession of more versatile approaches. Instead of designing each algorithm for a specific task, a conditional (adversarial) generator [20, 13] is trained for a set of possible image transformations through supervised learning, for example [44, 39, 48]. Although these approaches can perform a variety of seemingly impressive tasks, in this paper we show that a surprisingly large set of them can be solved via linear interpolation in deep feature space and may not require specialized deep architectures.

How can linear interpolation work? In pixel space, natural images lie on an (approximate) non-linear sub-manifold [40]. Non-linear sub-manifolds are locally Euclidean, but globally curved and non-Euclidean. It is well known that in pixel space linear interpolation between images introduces ghosting artifacts, a sign of the departure from the underlying sub-manifold, and linear classifiers between image categories perform poorly.

On the other hand, deep convolutional neural networks (convnets) are known to excel at classification tasks such as visual object categorization [34, 14, 15]—yet their last layer consists of simple linear classifiers. These linear classifiers can only perform so well because networks map images into new representations in which image classes are *linearly* separable. In fact, previous work has shown that neural networks trained on sufficiently diverse object recognition classes, such as VGG [34] trained on ImageNet [21], learn surprisingly versatile feature spaces and can be used to train linear classifiers for images outside the training set. Bengio *et al*. [3] hypothesize that convnets linearize the manifold of natural images into a (globally) Euclidean subspace of deep features.

Inspired by this hypothesis, we argue that in such deep feature spaces some semantic image editing tasks may no longer be as challenging as previously believed. We propose a simple framework that leverages the notion that in the right feature space, image editing can be performed simply by linear interpolation between images with a certain attribute and images without it. For instance, consider the task of adding facial hair to the image of a male face, given two sets of images: one set with facial hair, and one set without. If convnets can be trained to distinguish between male faces with facial hair and those without, we know that these classes must be linearly separable, and motion along a single linear vector should suffice to move an image from deep features corresponding to "no facial har" to those corresponding to facial hair. Indeed, we will show that even a simple choice of this vector suffices: we average each set of images' features and take the difference.

Figure 1 shows an example of a facial transformation with our method on a 400×400 image. We refer to this method as Deep Feature Interpolation (DFI).

Of course, DFI has limitations: Our method works best with mostly aligned images, for example those that can be lined up using feature points like eyes and mouths in face

Figure 1. **(Zoom in for details.)** Aging a 400x400 face with Deep Feature Interpolation, before and after the artifact removal step, showcasing the quality of our method. In this figure (and no other) a mask was applied to preserve the background. Although the input image was 400x400, all source and target images used in the transformation were only 100x100.

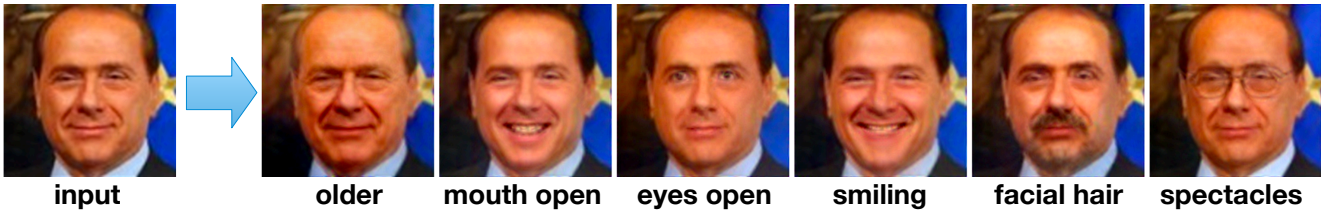

Figure 2. **(Zoom in for details.)** An example Deep Feature Interpolation transformation of a test image (Silvio Berlusconi, left) towards six categories. Each transformation was performed via linear interpolation in deep feature space composed of pre-trained VGG features.

images. It also requires that sample images with and without the desired attribute are otherwise similar to the target image (e.g. in the case of Figure 1 they consist of images of other caucasian males).

However, these assumptions on the data are surprisingly mild, and in the presence of such data DFI works surprisingly well. We demonstrate its efficacy on several transformation tasks that generative approaches are most commonly evaluated on. Compared to prior work, it is often much simpler, faster and more versatile: It does not require re-training of a convnet, is not specialized on any particular task, and it is able to deal with much higher resolution images. Despite its simplicity we show that on many of these image editing tasks it even outperforms state-of-the-art methods that are substantially more involved and specialized.

2. Related Work

Probably the generative methods most similar to ours are [23] and [28] as these also generate data-driven attribute transformations and rely on deep feature spaces. We use these methods as our primary point of comparison, although they rely on specially trained generative auto-encoders and are fundamentally different in their approaches to learn image transformations. Works by Reed *et al*. [29, 30] propose content change models for challenging tasks (identity and viewpoint changes) but do not demonstrate photo-realistic results. A contemporaneous work [4] edits image content by manipulating latent space variables but their approach fails when applied directly to existing photos. An advantage of our approach is that it works with pre-trained networks and has the ability to run on much higher resolution images. In general, many other uses of generative networks are distinct from our problem setting [13, 6, 47, 33, 26, 7], as they deal primarily with generating novel images rather than changing existing ones.

Gardner *et al*. [9] edits images by minimizing the witness function of the Maximum Mean Discrepancy statistic. The memory needed to find $\mathbf{w}$ by their method grows linearly whereas DFI removes this bottleneck.

Mahendran and Vedaldi [25] recovered visual imagery by inverting deep convolutional feature representations. Gatys *et al*. [11] demonstrated how to transfer the artistic style of famous artists to natural images by optimizing for feature targets during reconstruction. Rather than reconstructing imagery or transferring style, we construct new images with different content class memberships.

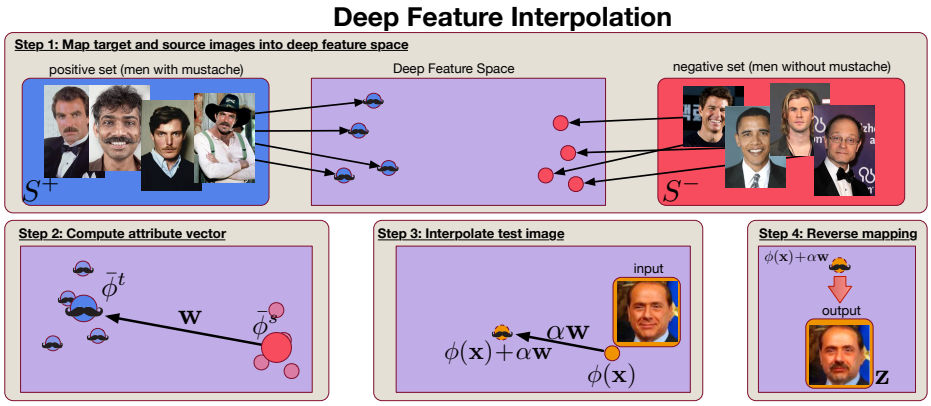

Figure 3. A schematic outline of the four high-level DFI steps.

Many works have used vector operations on a learned generative latent space to demonstrate transformative effects [8, 28, 12, 42]. In contrast, we suggest that vector operations on a discriminatively-trained feature space can achieve similar effects.

In concept, our work is similar to [37, 10, 38, 18, 16] that use video or photo collections to capture the personality and character of one person's face and apply it to a different person (a form of puppetry [35, 41, 19]). This difficult problem requires a complex pipeline to achieve high quality results. For example, Suwajanakorn *et al*. [37] combine several vision methods: fiducial point detection [43], 3D face reconstruction [36] and optical flow [17]. Our method is less complicated and applicable to other domains (e.g., shoes).

While we do not claim to cover all the cases of all techniques above, our approach is surprisingly powerful and effective. We believe investigating and further understanding the reasons for its effectiveness would be useful for better design of image editing with deep learning.

3. Deep Feature Interpolation

In our setting, we are provided with a test image $\mathbf{x}$ which we would like to change with respect to a given attribute in a believable fashion. For example, the image could be a man without a beard and we would like to automatically modify the image to add facial hair, while preserving the identity of the man. We further assume we have access to a set of *target* images *with* the desired attribute $\mathcal{S}^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_n^t\}$ (e.g., *men with facial hair*) and a set of *source* images *without* the attribute $\mathcal{S}^s = \{\mathbf{x}_1^s, ..., \mathbf{x}_m^s\}$ (e.g., *men without facial hair*). Further, we are provided with a pre-trained convnet trained on a sufficiently rich object categorization task, for example the openly available VGG network [34] trained on ImageNet [31]. We can use this convnet to obtain a new representation of the image, which we denote as $\mathbf{x} \rightarrow \phi(\mathbf{x})$. The vector $\phi(\mathbf{x})$ consists of concatenated activations of the convnet when applied to image $\mathbf{x}$ and we refer to it as the *deep feature representation* of $\mathbf{x}$.

Deep Feature Interpolation can be summarized in four high-level steps (illustrated in Figure 3):

1. We map the images in the target and source sets $\mathcal{S}^t$ and $\mathcal{S}^s$ into the deep feature representation through the pre-trained convnet ϕ (e.g., VGG-19 trained on ILSVRC2012).
2. We compute the mean feature values for each set of images, $\bar{\phi}^t$ and $\bar{\phi}^s$, and define their difference as the *attribute vector*

$$\mathbf{w} = \bar{\phi}^t - \bar{\phi}^s.$$

3. We map the test image $\mathbf{x}$ to a point $\phi(\mathbf{x})$ in deep feature space and move it along the attribute vector $\mathbf{w}$, resulting in $\phi(\mathbf{x}) + \alpha\mathbf{w}$.
4. We can reconstruct the transformed output image $\mathbf{z}$ by solving the reverse mapping into pixel space w.r.t. $\mathbf{z}$

$$\phi(\mathbf{z}) = \phi(\mathbf{x}) + \alpha\mathbf{w}.$$

Although this procedure may appear deceptively simple, we show in section 3 that it can be surprisingly effective. In the following we will describe some important details to make the procedure work in practice.

Selecting $\mathcal{S}^t$ and $\mathcal{S}^s$. DFI assumes that the attribute vector $\mathbf{w}$ isolates the targeted transformation, i.e., it points towards the deep feature representation of image $\mathbf{x}$ with the desired attribute change. If such an image $\mathbf{z}$ was available (e.g., the same image of Mr. Berlusconi with beard), we could compute $\mathbf{w} = \phi(\mathbf{z}) - \phi(\mathbf{x})$ to isolate exactly the difference induced by the change in attribute. In the absence of the exact target image, we estimate $\mathbf{w}$ through the target and source sets. It is therefore important that both sets are as similar as possible to our test image $\mathbf{x}$ and there is no systematic

attribute bias across the two data sets. If for example, all target images in $\mathcal{S}^t$ were images of more senior people, and source images in $\mathcal{S}^s$ of younger individuals the vector $\mathbf{w}$ would also capture the change involved in aging. Also, if the two sets are too different from the test image (e.g., a different race) the transformation would not look believable. To ensure sufficient similarity we restrict $\mathcal{S}^t$ and $\mathcal{S}^s$ to the K nearest neighbors. Let $\mathcal{N}_K^t$ denote the K nearest neighbors of $\mathcal{S}^t$ to $\phi(\mathbf{x})$ and we define

$$\bar{\phi}^t = \frac{1}{K} \sum_{\mathbf{x}^t \in \mathcal{N}_K^t} \phi(\mathbf{x}_i^t) \text{ and } \bar{\phi}^s = \frac{1}{K} \sum_{\mathbf{x}^s \in \mathcal{N}_K^s} \phi(\mathbf{x}_i^s).$$

These neighbors can be selected in two ways, depending on the amount of information available. When attribute labels are available we find the nearest images by counting the number of matching attributes (e.g., matching gender, race, age, hair color). When attribute labels are unavailable, or as a second selection criterion, we take the nearest neighbors by cosine distance in deep feature space.

Deep feature mapping. There are many choices for a mapping into deep feature space $\mathbf{x} \to \phi(\mathbf{x})$. We use the convolutional layers of the normalized VGG-19 network pre-trained on ILSVRC2012, which has proven to be effective at semantic style editing [11]. We need the deep feature space to be suitable for two very different tasks, the linear interpolation and the reverse mapping back into pixel space. For the interpolation, it is advantageous to pick deep layers that are further along the linearization process of deep convnets [3]. In contrast, for the reverse mapping, earlier layers capture more details of the image to be constructed [25]. The VGG network is divided into five pooling regions (with increasing depth). As an effective compromise we pick the first layers from the last three regions, `conv3_1`, `conv4_1` and `conv5_1` layers, flattened and concatenated. As the pooling layers of VGG reduce the dimensionality of the input image, DFI works best on medium high resolution images. To still apply DFI on low-resolution images we *increase* the image resolution to 200×200 before applying ϕ.

Image transformation. Due to the ReLU activations used in most convnets (including VGG), all dimensions in $\phi(\mathbf{x})$ are non-negative and the vector is sparse. As we average over K images (instead of a single image as in [3]), we expect $\bar{\phi}^t, \bar{\phi}^s$ to have very small components in most features. As the two data sets $\mathcal{S}^t$ and $\mathcal{S}^s$ only differ in the target attribute, features corresponding to visual aspects unrelated to this attribute will average out to very small values and be approximately subtracted away in the vector $\mathbf{w}$. A crucial element of the linear interpolation in deep feature space is the parameter α. The dimensionality and sparseness of the deep feature representation is affected by the resolution of the images and the transformation task to be performed. To make the choice of α more robust across these variations we L2-normalize the attribute vector $\mathbf{w} \to \frac{\mathbf{w}}{\|\mathbf{w}\|}$.

Reverse mapping. The final step of DFI is to reverse map the vector $\phi(\mathbf{x}) + \alpha\mathbf{w}$ back into pixel space to obtain an output image $\mathbf{z}$. Intuitively, $\mathbf{z}$ is an image, which when mapped into deep feature space will give $\phi(\mathbf{z}) \approx \phi(\mathbf{x}) + \alpha\mathbf{w}$. Although no closed-form inverse function exists for the VGG mapping we can obtain a color image by adopting the approach of [25] and find $\mathbf{z}$ with gradient descent:

$$\mathbf{z} = \arg\min_{\mathbf{z}} \frac{1}{2} \|(\phi(\mathbf{x}) + \alpha\mathbf{w}) - \phi(\mathbf{z})\|_2^2 + \lambda_{V^\beta} R_{V^\beta}(\mathbf{z}), \quad (1)$$

where R_{V^β} is the Total Variation regularizer [25] which encourages smooth transitions between neighboring pixels,

$$R_{V^\beta}(\mathbf{z}) = \sum_{i,j} \left((z_{i,j+1} - z_{i,j})^2 + (z_{i+1,j} - z_{i,j})^2 \right)^{\frac{\beta}{2}}.$$

Here, $z_{i,j}$ denotes the pixel in location (i, j) in image $\mathbf{z}$. Throughout, we set $\lambda_{V^\beta} = 0.001$ and $\beta = 2$. We solve (1) with the standard hill-climbing algorithm L-BFGS [24].

Artifact removal. The VGG reverse mapping is inherently under-constrained. Even with regularization this can lead to color distortion and spurious artifacts, which become particularly visible with high-resolution images. We can correct these by utilizing the similarities between $\mathbf{z}$ and the original input $\mathbf{x}$. We can use $\mathbf{x}$ to correct the color distribution of $\mathbf{z}$ by adjusting their channel mean and stddev to match. We can also use $\mathbf{x}$ to estimate spurious artifacts of the VGG reverse mapping by reconstructing $\mathbf{x}$ from its own deep feature representation $\phi(\mathbf{x})$ [5]. Let this reconstruction be $\hat{\mathbf{x}}$ and its residual $\mathbf{r}_\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$. If the images $\mathbf{x}$ and $\mathbf{z}$ are sufficiently similar, the reconstruction residuals are highly correlated and we can denoise the output by subtracting $\mathbf{r}_\mathbf{x}$, i.e. $\mathbf{z} \to \mathbf{z} - \mathbf{r}_\mathbf{x}$.

4. Experimental Results

We evaluate DFI on a variety of tasks and data sets. For perfect reproducibility our code will be made available on GitHub.

4.1. Changing Face Attributes

We compare DFI to AEGAN [23], a generative adversarial autoencoder on several face attribute modification tasks. We use the Labeled Faces in the Wild (LFW) data set, which contains 13,143 images of faces with predicted annotations for 73 different attributes (e.g., SUNGLASSES, GENDER, ROUND FACE, CURLY HAIR, MUSTACHE, etc.). We use these annotations as attributes for our experiments.

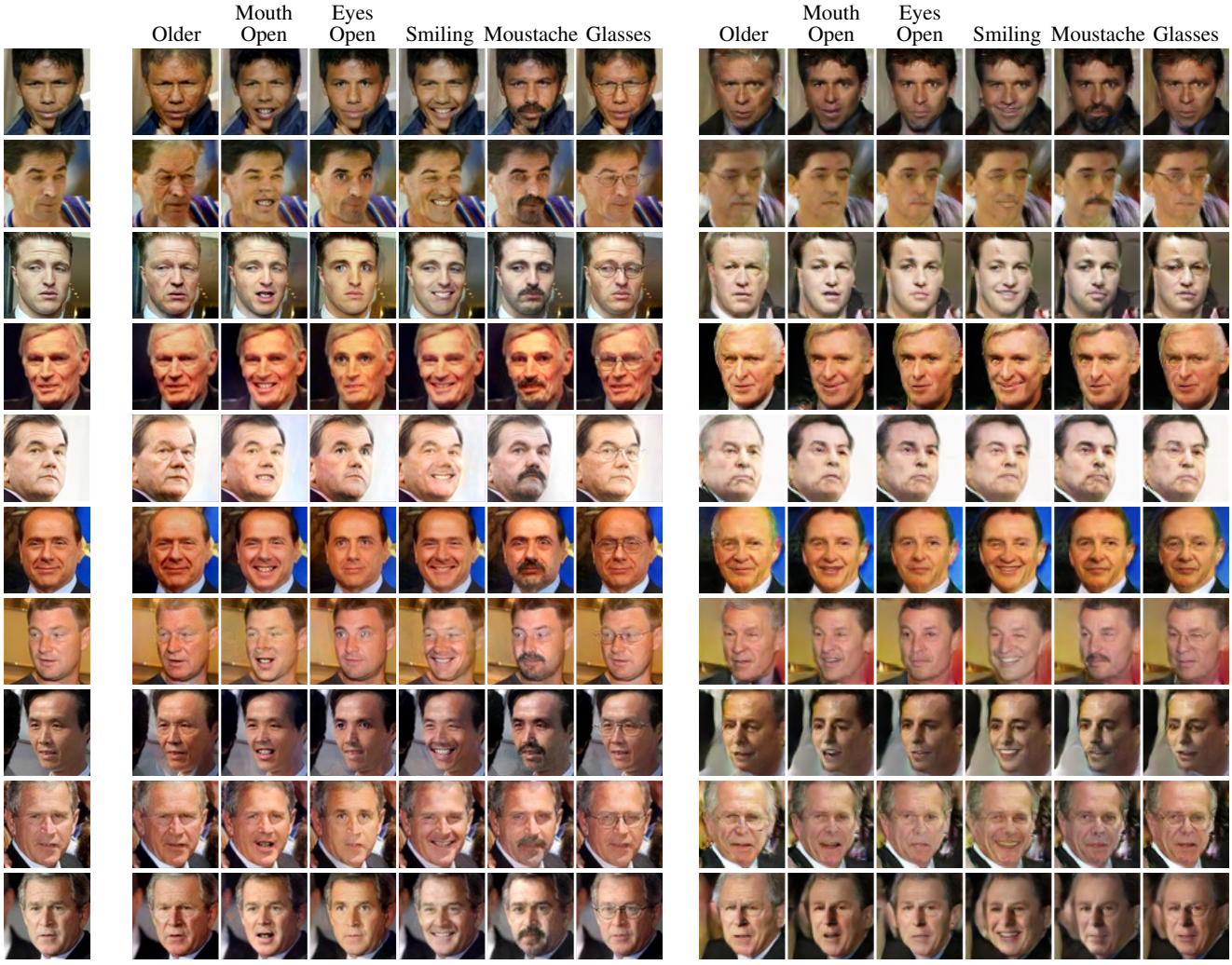

Figure 4. **(Zoom in for details.)** Adding different attributes to the same person (random test images). **Left.** Original image. **Middle.** DFI. **Right.** AEGAN. The goal is to add the specified attribute while preserving the identity of the original person. For example, when adding a moustache to Ralf Schumacher (3rd row) the hairstyle, forehead wrinkle, eyes looking to the right, collar and background are all preserved by DFI. No foreground mask or human annotation was used to produce these test results.

Similar to DFI, AEGAN also makes changes to faces by vector operations in a feature space. We use color matching in all results in this paper except inpainting since the large gray mask artificially distorts the color statistics of the input image. We remove the spurious artifacts only in Figure 1. We chose six attributes for testing: SENIOR, MOUTH SLIGHTLY OPEN, EYES OPEN, SMILING, MOUSTACHE and EYEGLASSES. (The negative attributes are YOUTH, MOUTH CLOSED, NARROW EYES, FROWNING, NO BEARD, NO EYEWEAR.) These attributes were chosen because it would be plausible for a single person to be changed into having each of those attributes. Our test set consists of images that did not have any of the six target attributes, were not WEARING HAT, had MOUTH CLOSED, NO BEARD and NO EYEWEAR. There are 38 images which meet this criteria. As LFW is highly gender imbalanced, we only used images of the more common gender, men, as target, source, and test images.

Matching the approach of [23], we align the face images and crop the outer pixels leaving a 100×100 face image, which we resize to 200×200, and set $\alpha = 0.4$ (taking 80 seconds per image). image collections are the images which have the source (target) attributes. From each collection we take the $K = 100$ nearest neighbors (by number of matching attributes).

Comparisons are shown in Figure 4. Looking down each column, we expect each image to express the target attribute. Looking across each row, we expect to see that the identity of the person is preserved. Although AEGAN often produces the right attributes, it does not preserve identity as well as the much simpler DFI.

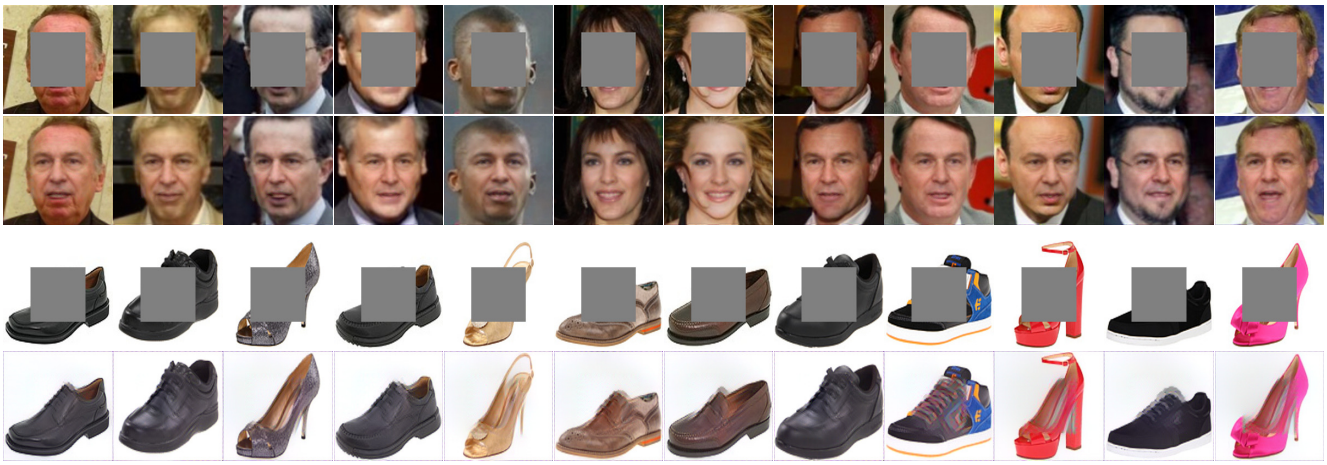

Figure 5. **(Zoom in for details.)** Filling missing regions. **Top.** LFW faces. **Bottom.** UT Zappos50k shoes. Inpainting is an interpolation from masked to unmasked images. Given any dataset we can create a source and target pair by simply masking out the missing region. DFI uses $K = 100$ such pairs derived from the nearest neighbors (excluding test images) in feature space. The face results match wrinkles, skin tone, gender and orientation (compare noses in 3rd and 4th images) but fail to fill in eyeglasses (3rd and 11th images). The shoe results match style and color but exhibit silhouette ghosting due to misalignment of shapes. Supervised attributes were not used to produce these results.

older	mouth open	eyes open	smiling	moustache	glasses
4.57	7.09	17.6	20.6	24.5	38.3

Table 1. Perceptual study results. Each column shows the ratio at which workers preferred DFI to AEGAN on a specific attribute change (see Figure 4 for images).

Perceptual Study. Judgments of visual image changes are inherently subjective. In order to obtain an objective comparison between DFI and AEGAN we therefore conducted a blind perceptual study with Amazon Mechanical Turk workers. We asked workers to pick the image which best expresses the target attribute while preserving the identity of the original face. This is a nuanced task so we required workers to complete a tutorial before participating in the study. The task was a forced choice between AEGAN and DFI (shown in random order) for six attribute changes on 38 test images. We collected an average of 29.6 judgments per image from 136 unique workers and found that DFI was preferred to AEGAN by a ratio of 12:1. The least preferred transformation was Senior at 4.6:1 and the most preferred was Eyeglasses at 38:1 (see Table 1).

High resolution and artifact removal. A big advantage of DFI over related work is that it naturally scales to higher resolution images. To showcase this ability we download an out-of-sample 400×400 test-image of Ralf Schumacher (see the left image in Figure 1). In the absence of a higher resolution face image collection we use the same low resolution sets $\mathcal{S}^t$ and $\mathcal{S}^s$ as for Figure 4, which we enlarge to match the resolution of the test image. Next we create a mask for the face which is used to restrict changes to only the foreground pixels (shown behind the original image). For the final output we also use artifact removal, as described in section 3. We use aging as our transformation, as this produces changes which are low-frequency and can be achieved with the enlarged 100×100 LFW images. The result (right image) is compelling—wrinkles deepen, the iris become paler, the skin becomes rougher and hair becomes lighter.

4.2. Inpainting Without Attributes

Inpainting fills missing regions of an image with plausible pixel values. There can be multiple correct answers. Inpainting is hard when the missing regions are large (see Figure 5 for our test masks). Since attributes cannot be predicted (e.g., eye color when both eyes are missing) we use distance in feature space to select the nearest neighbors.

Inpainting may seem like a very different task from changing face attributes, but it is actually a straightforward application of DFI. All we need are source and target pairs which differ only in the missing regions. Such pairs can be generated for any dataset by taking an image and masking out the same regions that are missing in the test image. The images with mask become the source set and those without the target set. We then find the $K = 100$ nearest neighbors in the masked dataset (excluding test images) by cosine distance in VGG-19 pool5 feature space. We experiment on two datasets: all of LFW (13,143 images, including male and female images) and the Shoes subset of UT Zappos50k (29,771 images) [45, 27]. For each dataset we find a single α that works well (1.6 for LFW and 2.8 for UT Zappos50k).

We show our results in Figure 5 on 12 test images (more in supplemental) which match those used by disCVAE [44] (see Figure 6 of their paper). Qualitatively we observe that the DFI results are plausible. The filled face regions match skin tone, wrinkles, gender, and pose. The filled shoe re-

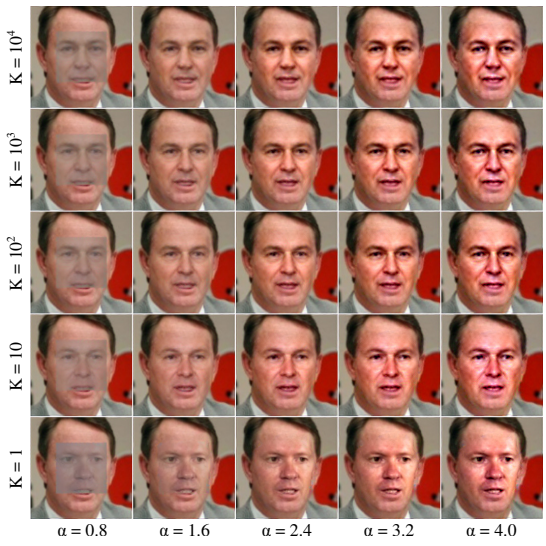

Figure 6. Inpainting and varying the free parameters. **Rows:** K, the number of nearest neighbors. **Columns:** α, higher values correspond to a larger perturbation in feature space. When K is too small the generated pixels do not fit the existing pixels as well (the nose, eyes and cheeks do not match the age and skin tone of the unmasked regions). When K is too large a difference of means fails to capture the discrepancy between the distributions (two noses are synthesized). When α is too small or too large the generated pixels look unnatural. We use $K = 100$ and $\alpha = 1.6$.

gions match color and shoe style. However, DFI failed to produce eyeglasses when stems are visible in the input and some shoes exhibit ghosting since the dataset is not perfectly aligned. DFI performs well when the face is missing (i.e., the central portion of each image) but we found it performs worse than disCVAE when half of the image is missing (Figure 8). Overall, DFI works surprisingly well on these inpainting tasks. The results are particular impressive considering that, in contrast to disCVAE, it does not require attributes to describe the missing regions.

4.3. Varying the free parameters

Figure 6 illustrates the effect of changing α and K. As α increases task-related visual elements change more strongly (Figure 7). If α is low then ghosting can appear. If α is too large then we may jump to a point in feature space which leads to an unnatural reconstruction. K controls the variety of images in the source and target sets. A lack of variety can cause artifacts where changed pixels do not match nearby unchanged pixels (e.g., see the lower lip when $K = 1$). However, too much variety can cause $\bar{\phi}^s$ and $\bar{\phi}^t$ to contain distinct subclasses and the set mean may no longer describe either subclass (e.g., when $K = 10^4$, the nose has two tips, reflecting the presence of left-facing and right-facing subclasses). In practice, we pick an α and K which work well for a variety of images and tasks rather than choosing per-case.

5. Discussion

In the previous section we have shown that Deep Feature Interpolation is surprisingly effective on several image transformation tasks. This is very promising and may have implications for future work in the area of automated image transformations. However, DFI also has clear limitations and requirements on the data. We first clarify some of the aspects of DFI and then focus on some general observations.

Image alignment is a necessary requirement for DFI to work. We use the difference of means to cancel out the contributions of convolutional features that are unrelated to the attribute we wish to change, particularly when this attribute is centered in a specific location (adding a mustache, opening eyes, adding a smile, etc). For example, when adding a mustache, all target images contain a mustache and therefore the convolutional features with the mustache in their receptive field will not average out to zero. While max-pooling affords us some degree of translation invariance, this reasoning breaks down if mustaches appear in highly varied locations around the image, because no specific subset of convolutional features will then correspond to "mustache features". Similarly, when opening eyes, if the eyes of the input image are not well aligned to the eyes of the target set, DFI may add a second set of eyes entirely.

Neighborhood size. The nearest neighbor search for the sets $\mathcal{S}^t, \mathcal{S}^s$ are an important aspect of DFI. If the the data is multi-modal, restricting the interpolation to a sufficiently small set of K nearest neighbors in feature space automatically selects the closer mode for the transformation. The choice of K represents a trade-off of how varied the transformations should be across input images. On one extreme (K too high), we apply nearly the same transformation to all input images. For example, inpainting with K too high produces very similar faces on all images. On the other extreme (K too low), DFI just copies convolutional features directly from the nearest target image, pasting the nearest neighbor's face on. For example, in Figure 6, this leads to significant artifacts around the boundary of the inpainting mask for $K = 1$, as the input image's face does not exactly match the nearest neighbor.

Time and space complexity. A significant strength of DFI is that it is very lean. The biggest resource footprint is GPU memory for the convolutional layers of VGG-19 (the large fully-connected layers are not needed). A 1280×960 image requires 3.5 GB and takes 32 minutes to reconstruct. The time and space complexity are linear. In comparison, many generative models only demonstrate 64×64 images.

Figure 7. Morphing a face to make it appear older. The transformation becomes more pronounced as the value of α increases.

Although DFI does not require the training of a specialized architecture, it is also fair to say that during test-time it is significantly slower than for example generative auto-encoders. In our code-base a 200×200 image takes 80s to process. As future work it may be possible to incorporate techniques from real-time style-transfer [32] to speed-up DFI in practice.

Figure 8. Example of a hard task for DFI: inpainting an image with the right half missing.

Resolution. Although there exists work on high-resolution style transfer [11, 25, 32], to our knowledge, DFI is the first algorithm to enable automated high resolution content transformations. The fact that DFI is so simple may inspire more sophisticated follow-up work that exploits similar mechanisms to enable more general high-resolution transformations. High-resolutions are important and scaling up current generative architectures to higher resolutions may unlock a wide range of new applications and use cases.

DFI's simplicity. It is possible that the generative models are much more powerful than DFI, but the current problems (in particular, face attribute editing) which are used to showcase generative approaches are too simple. Indeed, we do find many problems where generative models outperform DFI. In the case of inpainting we find DFI to be lacking when the masked region is half the image (Figure 8). DFI is also incapable of shape [49] or rotation [30] transformations since those tasks require aligned data. Finding more of these difficult tasks where generative models outshine DFI would help us better evaluate generative models. Ultimately DFI can be used as a first test for whether a task is interesting: problems that can easily be solved by DFI are unlikely to require the complex machinery of generative networks.

Generative vs. Discriminative networks. Autoencoders have specialized layers and loss functions specifically designed to allow us to recover images from their latent spaces. AE architectures have continually improved by each new architecture being compared against previous AE architectures. To our knowledge, this work is the first cross-architectural comparison of an AE against a method that uses features from a discriminatively trained network. To our great surprise, we find that the discriminative model has a convolutional latent space which appears to be as good as an AE model at disentangling semantic visual elements. One possibility is that the AE architecture could organize a better latent space but it has not yet been demonstrated since AE are typically trained on small datasets of 10,000s of samples with very little variety compared to the richness of recognition datasets. The richness of ImageNet seems to be an important factor since we found in early experiments that the convolutional feature spaces of VGG-19 outperformed those of VGG-Face on face attribute change tasks.

Image editing tasks. We need to find out on which tasks generative models beat a linear interpolation baseline and focus research on those problems. We see DFI as a tool for guiding research towards the frontiers of semantic image editing, and away from simple tasks. There is value in producing face attribute changes as a useful diagnostic since there is a wealth of previous work to compare against. But we hope that task will be viewed as something to verify a model rather than showcase it. We propose DFI to be the linear interpolation baseline because it is very easy to compute, it will scale to future high-resolution models, it does not require supervised attributes, and it can be applied to nearly any aligned class-changing problems since it only needs empirical distributions in the form of two image collections.

6. Conclusion

Overall, DFI performs surprisingly well given the method's simplicity. It is able to produce high quality images over a variety of tasks, in many cases of higher quality than existing state-of-the-art methods. This suggests that, given the ease with which DFI can be implemented, it should serve as a highly competitive baseline for certain types of image transformations on aligned data. Given the performance of DFI, we hope that this spurs future research in to image transformation methods that outperform this approach.

References

[1] M. Aittala, T. Aila, and J. Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (TOG)*, 35(4):65, 2016. 1

[2] R. Bellini, Y. Kleiman, and D. Cohen-Or. Time-varying weathering in texture space. *ACM Transactions on Graphics (TOG)*, 35(4):141, 2016. 1

[3] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai. Better mixing via deep representations. In *ICML (1)*, pages 552–560, 2013. 1, 4

[4] A. Brock, T. Lim, J. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 2

[5] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008. 4

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1486–1494, 2015. 2

[7] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[8] A. Dosovitskiy, J. Tobias Springenberg, and T. 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. 3

[9] J. R. Gardner, P. Upchurch, M. J. Kusner, Y. Li, K. Q. Weinberger, K. Bala, and J. E. Hopcroft. Deep Manifold Traversal: Changing labels with convolutional features. *arXiv preprint arXiv:1511.06421*, 2015. 2

[10] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt. Automatic face reenactment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4217–4224, 2014. 3

[11] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 2, 4, 8

[12] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. *arXiv preprint arXiv:1603.08637*, 2016. 3

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016. 1

[15] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016. 1

[16] I. Kemelmacher-Shlizerman. Transfiguring portraits. *ACM Transactions on Graphics (TOG)*, 35(4):94, 2016. 1, 3

[17] I. Kemelmacher-Shlizerman and S. M. Seitz. Collection flow. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1792–1799. IEEE, 2012. 3

[18] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, and S. M. Seitz. Exploring photobios. In *ACM Transactions on Graphics (TOG)*, volume 30, page 61. ACM, 2011. 3

[19] N. Kholgade, I. Matthews, and Y. Sheikh. Content retargeting using parameter-parallel facial layers. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 195–204. ACM, 2011. 3

[20] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014. 1

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[22] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)*, 33(4):149, 2014. 1

[23] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 2, 4, 5

[24] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. 4

[25] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4, 8

[26] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016. 2

[27] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *arXiv preprint arXiv:1604.07379*, 2016. 6

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 2, 3

[29] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1431–1439, 2014. 1, 2

[30] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, pages 1252–1260, 2015. 2, 8

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

[32] F. Sadeghi, C. L. Zitnick, and A. Farhadi. Visalogy: Answering visual analogy questions. In *Advances in Neural Information Processing Systems*, pages 1873–1881, 2015. 8

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016. 2

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 3

[35] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 399–405. ACM, 2004. 3

[36] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz. Total moving face reconstruction. In *Computer Vision–ECCV 2014*, pages 796–812. Springer, 2014. 3

[37] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. What makes tom hanks look like tom hanks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3952–3960, 2015. 1, 3

[38] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6):183, 2015. 3

[39] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. *arXiv preprint arXiv:1603.05631*, 2016. 1

[40] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006. 1

[41] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 7–16. ACM, 2009. 3

[42] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *arXiv preprint arXiv:1610.07584*, 2016. 3

[43] X. Xiong and F. Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013. 3

[44] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2Image: Conditional image generation from visual attributes. In *Computer Vision–ECCV 2016*. 2016. 1, 6

[45] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*, June 2014. 6

[46] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *arXiv preprint arXiv:1603.08511*, 2016. 1

[47] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[48] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. *arXiv preprint arXiv:1605.03557*, 2016. 1

[49] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 8