Deep Manifold Traversal: Changing Labels with Convolutional Features

Abstract. Many tasks in computer vision can be cast as a "label changing" problem, where the goal is to make a semantic change to the appearance of an image or some subject in an image in order to alter the class membership. Although successful task-specific methods have been developed for some label changing applications, to date no general purpose method exists. Motivated by this we propose *deep manifold traversal*, a method that addresses the problem in its most general form: it first approximates the manifold of natural images then morphs a test image along a traversal path away from a source class and towards a target class while staying near the manifold throughout. The resulting algorithm is surprisingly effective and versatile. It is completely data driven, requiring only an example set of images from the desired source and target domains. We demonstrate deep manifold traversal on highly diverse label changing tasks: changing an individual's appearance (age and hair color), changing the season of an outdoor image, and transforming a city skyline towards nighttime.

1 Introduction

Many tasks in computer vision can be cast as a *label changing* problem: given an input image, change the label of that image from some label y^s to some target label y^t. Recent examples of this general task include changing facial expressions and hairstyle [1, 2], example-based image colorization [3, 4], aging of faces [5, 6], material editing [6], editing of outdoor scenes [7] changing seasons [8], and image morphs [9], relighting of photos [10, 11] or hallucinating a night image from a day image [12]. A variety of specialized algorithms exist for each of these tasks. However, these algorithms often incorporate substantial domain-specific prior knowledge relevant to the task and may require hand annotation of images, rendering them unable to perform any other task. For example, it is unlikely that a facial aging algorithm would be able to change the season of an outdoor scene.

This motivates research into the most *general* form of changing image appearances. Our goal is to design a method that takes as input a set of source and target images (e.g. images of young and old people) and changes a given test image to be semantically more similar to the target than the source images.

A given image could be transformed into a target image through linear interpolation in pixel space. However, intermediate images would not be meaningful because the set of natural images does not span a linear subspace in the pixel space. Instead, it is believed to constitute a low dimensional sub-manifold [13].

In order to make meaningful changes, the image traversal path must be confined to the underlying manifold throughout.

Bengio et al. 2012 [14] hypothesizes that deep convolutional networks linearize the manifold of natural images into a subspace of deep features. This suggests that convolutional networks, and in particular the feature space learned by such networks, may be a natural choice for solving the label changing problem. However, recent work [15, 16] has demonstrated that this problem can be surprisingly hard for machine learning algorithms. In the context of object classification through convolutional neural networks, it has been shown possible to change the *prediction* of an image with tiny alterations that can be *imperceptible to humans*. Such changes do not affect the appearance of the image and leave the *class label* untouched [15]. In fact, the problem of changing class labels persists for most discriminative machine learning algorithms [17] and is still an open problem to date.

In this paper we investigate how to make *meaningful* changes to input images while staying on the underlying manifold. We follow the intuition by Bengio et al. 2012 [14] and utilize a deep convolutional network trained on 1.2 million images [18] to simplify the manifold of natural images to a linear feature space. We avoid the difficulty pointed out by Szegedy et al [15] by using kernel Maximum Mean Discrepancy (MMD) [19] to estimate the distributions of source and target images in this feature space to guide the traversal. The traversal stays on the manifold, because it is confined to the subspace of deep features and is forced by the MMD guide to regions that correspond to likely images. Each point along the path can be mapped back to an image with reverse image reconstruction [20]. Furthermore, our method is linear in space and time so it naturally scales to large images (e.g., 900×600), which is much larger than most results demonstrated by generative models.

In a nutshell, our algorithm works in three steps: 1. Source, target and the test images are forward propagated through a convolutional network and mapped into a deep feature space; 2. MMD is used to guide the traversal of the test image in the deep feature space towards the target and away from the source distribution while staying close to the manifold of natural images; 3. a point along the traversal path is specified and a corresponding image is generated through reverse image reconstruction.

The resulting algorithm allows us to traverse the manifold of natural images freely in a completely data-driven way. We only require labeled images from the source and target classes, and no hand annotation (e.g., correspondences or strokes). While this method certainly does not replace specialized methods, it may function as a baseline for a wide variety of tasks, or perhaps enable some tasks for which no specialized algorithms have been derived. Our results indicate that our method is highly general and performs better than current general methods (which make use of image morphing) on a number of different tasks.

2 Related Work

Szegedy et al. [15] were the first to show that deep networks can be 'easily convinced' that an input is in a different class, by making subtle, imperceptible changes to the input. Such changed inputs were termed 'adversarial examples' and Goodfellow, et al. [17] showed that these examples are generally problematic for high-dimensional linear classifiers. These results indicate it is inherently difficult to meaningfully change the label of an input with small changes.

In general, generative networks are somewhat orthogonal to our problem setting, as they [21, 22], (a) deal primarily with generating novel images rather than changing existing ones, and (b) are typically restricted to very low resolution images, such as 32×32.

Mahendran and Vedaldi [20] recovered visual imagery by inverting deep convolutional feature representations. Their goal was to reveal invariance by comparing a reconstructed image to the original image. Gatys, et al. [23] demonstrated how to transfer the artistic style of famous artists to natural images by optimizing for feature targets during reconstruction. We draw upon these works as means to demonstrate our framework in the image domain. Yet, rather than reconstructing imagery or transferring style, we construct new images which have the qualities of a different class.

A few methods in the machine learning literature also deal with data-driven changes to images. Reed et al. [24, 25] propose to learn a model to disentangle factors of variation (e.g., identity and viewpoint). In our work, we directly minimize the discrepancy between an image and a target sub-manifold inside the semantic space learned by a convolutional network trained on millions of images. An advantage of our approach is the ability to run on much higher resolution images up to 900x600 in this paper, compared to 48x48 images in [24].

Analogical reasoning methods [26–29, 25, 30] solve for D in the expression: A is to B as C is to D. Other methods generate images in a controlled fashion [31, 32]. Our method also has multiple inputs but we do not solve for analogies nor do we learn a disentangled model.

In concept, our work is similar to methods [33–35] which 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 [36–38]). This difficult problem requires a complex pipeline to achieve high quality results. For example, Suwajanakorn et al. [33] combines several vision methods: fiducial point detection [39], 3D face reconstruction [40], optical flow [41] and texture mapping. Our work is conceptually similar because we also use photo collections to define the source and target. However, we can produce plausible results without any additional machinery and our usage of a high-level semantic CNN feature space makes our method applicable to a wide-variety of domains.

Our task is related to a large body of image morphing work (survey by Wolberg [42]). Image morphing warps images into an alignment map then color interpolates between mapped points. Unlike image morphing, we do not warp images to a map. A recent work by Liao et al. [9] aligns based on structural similarity [43]. Their goal is to achieve semantic alignment partially invariant to

lighting, shape and color. We achieve this with a high-level semantic CNN feature space. Their method also requires manual annotations to refine the mapping whereas our method is fully automated.

Kemelmacher et al. [44] creates plausible transformations between two images of the same person by selecting an ordered sequence of photos from a large photo collection. Qualitatively, the person may appear to change expression as if the image was changing. Unlike their method, we actually change the original image while preserving the clothing and background.

3 Background: Maximum Mean Discrepancy

The *Maximum Mean Discrepancy* [45] (MMD) statistic tests whether two probability distributions, source P^s and target P^t, are the same. The MMD metric measures the maximum difference between the mean function values:

$$\text{MMD}(P^s, P^t, \mathcal{F}) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}\left[f(\mathbf{z}^s)\right]_{\mathbf{z}^s \sim P^s} - \mathbb{E}\left[f(\mathbf{z}^t)\right]_{\mathbf{z}^t \sim P^t} \right) \tag{1}$$

given some function class $\mathcal{F}$. MMD can be thought of as producing a test function that distinguishes samples from these two distributions. In particular, the MMD test function is large when evaluated on samples drawn from a source distribution P^s, and small when evaluated on samples drawn from a target distribution P^t. When $\mathcal{F}$ is a reproducing kernel Hilbert space, the function maximizing this difference can be found analytically [19], and is called the *witness function*:

$$f^*(\mathbf{z}) = \mathbb{E}\left[k(\mathbf{z}^s, \mathbf{z})\right]_{\mathbf{z}^s \sim P^s} - \mathbb{E}\left[k(\mathbf{z}^t, \mathbf{z})\right]_{\mathbf{z}^t \sim P^t} \tag{2}$$

The MMD using this function is a powerful measure of discrepancy between two probability distributions. For example, it is easy to show that if $\mathcal{F}$ is universal, then $P^s = P^t$ if and only if $\text{MMD}(P^s, P^t, \mathcal{F}) = 0$ [19].

Given finite samples $\mathbf{z}_1^s, ..., \mathbf{z}_m^s \overset{iid}{\sim} P$ and $\mathbf{z}_1^t, ..., \mathbf{z}_n^t \overset{iid}{\sim} P^t$, the witness function can be estimated empirically:

$$f^*(\mathbf{z}) \approx \frac{1}{m} \sum_{i=1}^{m} k(\mathbf{z}_i^s, \mathbf{z}) - \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{z}_i^t, \mathbf{z}) \tag{3}$$

Intuitively, $f^*(\mathbf{z})$ measures the degree to which $\mathbf{z}$ is representative of either P^s — by taking a positive value — or P^t — by taking a negative value. In this work, we will make use of the Gaussian kernel, defined as $k(\mathbf{z}, \mathbf{z}') = e^{-\frac{1}{2\sigma}|\mathbf{z}-\mathbf{z}'|^2}$, where σ is the kernel bandwidth. While kernel methods often generalize poorly on images in pixel space because of violated smoothness assumptions, we expect that these assumptions hold after deep visual feature extraction [46]. For a more thorough review of the MMD statistic, see [19].

4 Deep Manifold Traversal

In this section, we will discuss our method for manifold traversal from one class into another. Importantly, any transformation should preserve the class-independent aspects of the original image, only changing the class-identifying features. In our setting, we are given a labeled set of images from a *source domain*, $\mathbf{x}_1^s, ..., \mathbf{x}_m^s$ each with source label y^s, and a set of labeled images from a *target domain*, $\mathbf{x}_1^t, ..., \mathbf{x}_n^t$ each with target label y^t. We are also given a specific *input image* $\bar{\mathbf{x}}^s$ with label y^s. Informally, our goal is to change $\bar{\mathbf{x}}^s \rightarrow \bar{\mathbf{x}}^t$ in a meaningful way such that $\bar{\mathbf{x}}^t$ has true label y^t. Figure 1 provides an overview of our approach.

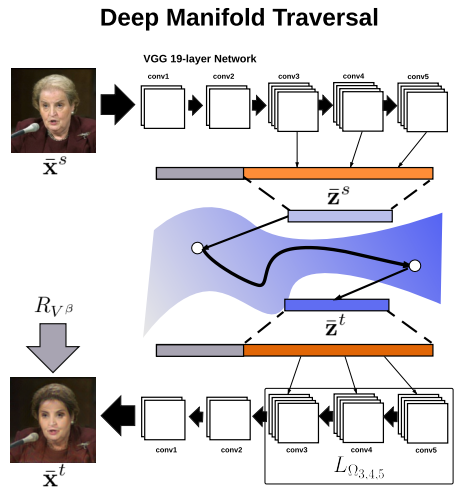

Fig. 1. Top: Input image $\bar{\mathbf{x}}^s$ is transformed by a ConvNet to deep features (orange). **Middle:** The manifold is traversed (black arrow) from source, $\bar{\mathbf{z}}^s$, to target, $\bar{\mathbf{z}}^t$, in feature space. **Bottom:** $\bar{\mathbf{z}}^t$ is inverted to recover $\bar{\mathbf{x}}^t$, subject to total variation regularizer R_{V^β}.

Manifold representation. The first step of our approach is to approximate the manifold of natural images and obtain a mapping from input images in pixel space, $\mathbf{x}$, to a high-level feature representation, $\mathbf{x}_i \longrightarrow \phi_i$. By modifying these deep visual features rather than the raw pixels of $\mathbf{x}$ directly, we make changes to the image in a space in which the manifold of natural images is simplified, which more easily allows for images to remain on the manifold.

Network details. Following the method of [23] we use the feature representations from deeper layers of a normalized, 19-layer VGG [18] network. Specifically, we use layers conv3_1 ($256 \times 63 \times 63$), conv4_1 ($512 \times 32 \times 32$) and conv5_1 ($512 \times 16 \times 16$), which have the indicated dimensionalities when the color input is 250×250. These layers are the first convolutions in the 3rd, 4th and 5th pooling regions. After ReLU, flattening and concatenation, a feature vector has 1.67 million dimensions for a 250×250 input image.

Image transformation. Our approach to image transformation will be to change the deep visual features $\bar{\mathbf{z}}^s = \phi(\bar{\mathbf{x}}^s)$ to look more like the deep visual features characteristic of label y^t. Because the deep convolutional network has mapped the original images in to a more linear subspace, we move linearly away from source high-level features and towards target high-level features. Specifically, we seek to add some linear combination of the source, target, and test images' deep features:

$$\bar{\mathbf{z}}^t = \phi(\bar{\mathbf{x}}^t) = \bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta}. \tag{4}$$

where $\mathbf{V} \in \mathbb{R}^{K \times D}$ is the matrix of deep convolutional features for the source, target, and test images: $[\phi_1^t, ..., \phi_n^t, \phi_1^s, ..., \phi_m^s, \bar{\mathbf{z}}^s]$. This linear combination should produce a set of deep features less like the source domain, more like the target domain, but still strongly like the original image. Thus, $\boldsymbol{\delta}$ should ideally contain negative values in most source indices, and positive values in most target indices.

To obtain this transformation, we propose an optimization guided by the MMD witness function from section 3. We make use of the empirical witness function $f^*(\bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta})$ to measure the degree to which the transformed VGG features $\bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta}$ resembles objects with source label y^s or those with target label y^t:

$$f^*(\bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta}) = \frac{1}{m}\sum_{i=1}^{m} k(\phi_i^s, \bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta}) - \frac{1}{n}\sum_{j=1}^{n} k(\phi_j^t, \bar{\mathbf{z}}^s + \mathbf{V}\boldsymbol{\delta}). \tag{5}$$

Observing that–given the definition of $\mathbf{V}$–each ϕ_i^s and ϕ_j^t and $\bar{\mathbf{z}}^s$ can be themselves written as $\mathbf{V}e_i$, $\mathbf{V}e_j$ and $\mathbf{V}e_K$ for one-hot vectors e_i, e_j, and e_K we rewrite the above as:

$$f^*(\mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}) = \frac{1}{m}\sum_{i=1}^{m} k(\mathbf{V}e_i^s, \mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}) - \frac{1}{n}\sum_{j=1}^{n} k(\mathbf{V}e_j^t, \mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}). \tag{6}$$

When using the squared exponential kernel, we can factor $\mathbf{V}$:

$$\begin{aligned} f^*(\mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}) = &\frac{1}{m}\sum_{i=1}^{m} \exp\left\{-\frac{1}{\sigma}(e_i^s - (e_K + \boldsymbol{\delta}))\mathbf{V}^\top\mathbf{V}(e_i^s - (e_K + \boldsymbol{\delta}))\right\} \\ &- \frac{1}{n}\sum_{j=1}^{n} \exp\left\{-\frac{1}{\sigma}(e_j^t - (e_K + \boldsymbol{\delta}))\mathbf{V}^\top\mathbf{V}(e_j^t - (e_K + \boldsymbol{\delta}))\right\}. \end{aligned} \tag{7}$$

If the $K \times K$ matrix $\mathbf{V}^\top\mathbf{V}$ is precomputed for a dataset, this function can be computed in time *independent* of the number of convolutional features, and therefore original image resolution.

The witness function $f^*(\mathbf{V}e_K + \mathbf{V}\delta)$ has a negative value if the transformed visual features $\mathbf{V}e_K + \mathbf{V}\delta$ are more characteristic of label y^t than of label y^s. To transform $\bar{\mathbf{x}}^s$ to have target label y^t, we therefore wish to minimize $f^*(\mathbf{V}\phi(\bar{\mathbf{z}})^s + \mathbf{V}\boldsymbol{\delta})$ in $\boldsymbol{\delta}$. However, when performed unbounded, this optimization moves too far along the manifold to a mode of the target domain, preserving little of the information contained in $\bar{\mathbf{z}}^s$. We therefore follow the techniques used in [15] and enforce a *budget* of change, and instead obtain $\bar{\mathbf{z}}^t$ by minimizing:

$$\phi(\bar{\mathbf{z}}^t) = \mathbf{V}(e_K + \boldsymbol{\delta}) \quad \text{where:} \quad \boldsymbol{\delta} = \arg\min_{\boldsymbol{\delta}} f^*(\mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}) + \lambda\|\mathbf{V}\boldsymbol{\delta}\|_2^2 \tag{8}$$

Minimizing the witness function encodes two "forces": $\phi(\bar{\mathbf{z}}^s)$ is pushed *away* from visual features characteristic of the source label y^s and simultaneously pulled *towards* visual features characteristic of the target label y^t.

Reconstruction. The optimization results in the transformed representation $\bar{\mathbf{z}}^t = \mathbf{V}e_K + \mathbf{V}\boldsymbol{\delta}$. In order to obtain our corresponding target image $\bar{\mathbf{x}}^t = \phi^{-1}(\bar{\mathbf{z}}^t)$, we need to "invert" the CNN. The deep CNN mapping is not invertible, so we cannot obtain the image in pixel space $\bar{\mathbf{x}}^t$ from $\bar{\mathbf{z}}^t$ directly. The mapping is however differentiable and we can adopt the approaches of [20] and [23] to find $\bar{\mathbf{x}}^t$ with gradient descent by minimizing the loss function

$$L_{\Omega_{3,4,5}}(\bar{\mathbf{x}}^t) = \frac{1}{2}\|\Omega_{3,4,5}(\bar{\mathbf{x}}^t) - \bar{\mathbf{z}}^t\|^2. \tag{9}$$

Regularization. Following the method of [20], we add a total variation regularizer

$$R_{V^\beta}(\bar{\mathbf{x}}^t) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2\right)^{\frac{\beta}{2}}. \tag{10}$$

Here, $x_{i,j}$ refers to the pixel with i,j coordinate in image $\mathbf{x}$. The addition of this regularizer greatly improves image quality. The final optimization problem becomes

$$\bar{\mathbf{x}}^t = \arg\min_{\bar{\mathbf{x}}^t} L_{\Omega_{3,4,5}}(\bar{\mathbf{x}}^t) + \lambda_{V^\beta} R_{V^\beta}(\bar{\mathbf{x}}^t). \tag{11}$$

We minimize (11) with bounded L-BFGS initialized with $\boldsymbol{\delta} = 0$. We set $\lambda_{V^\beta} = 0.001$ and $\beta = 2$ in our experiments. After reconstruction we have completed the manifold traversal from source to the target: $\bar{\mathbf{x}}^s \rightarrow \bar{\mathbf{z}}^s \rightarrow \bar{\mathbf{z}}^t \rightarrow \bar{\mathbf{x}}^t$. We will provide source code for our method on GitHub at http://anonymized.

5 Experimental Results - LFW

We evaluate our method on several manifold traversal tasks using the Labeled Faces in the Wild (LFW) dataset. This dataset contains 13,143 images (250×250) of faces with predicted annotations for 73 different attributes (e.g., "sunglasses", "soft lighting", "round face", "curly hair", "mustache", etc.). We use these annotations as labels for our manifold traversal experiments. Because the predicted annotations [47] have label noise, we take the 2,000 most confidently labeled images to construct an image set. For example, in our aging task below, we take the bottom (i.e., most negative) and top (i.e., most positive) 2,000 images in the "senior" class as our source and target image sets.

All single transformation image results shown for LFW use the same λ value of $\lambda =$ 4e-8. All experiments were run with RBF kernel width $\sigma =$ 7.7e5. In the tasks below, test images were chosen at random, with the exception of Aaron Eckhart (the first image in LFW), who we included in all tests in order to show multiple tasks on the same image. Due to space constraints we only show a small number of results per experiment. More results are in the supplemental.

5.1 Aging faces via manifold traversal.

To demonstrate the ability of our algorithm to make meaningful changes to the true label of a test instance, we first consider the task of computationally

Fig. 2. (Zoom in for details.) Face aging via manifold traversal on random (except Aaron Eckhart) 250x250 test images from LFW. All aging results shown were run with the same value of λ.

aging faces. To do this, we first follow our procedure above for selecting 2,000 source (young) and target (old) images. We select 7 test images at random plus Aaron Eckhart from the remainder of LFW. We then perform manifold traversal towards "senior" on these 8 images, using the same value of λ for each traversal.

The results of our aging experiment are shown in figure 2. In each case, deep manifold traversal generates an older-appearing version of the original image by adding wrinkles, graying hair and adding bags under eyes. Note that the images remain sharp despite the relatively high resolution compared to existing purely learning-based approaches for facial morphing [24].

One important aspect of the transformations made by deep manifold traversal is that changes are localized to the face and hair. Clothing, background, lighting, and other features of the image irrelevant to the desired label change were not significantly affected. Thus, our algorithm succeeds in preserving as much character of the original image as possible while still changing the true label of the image.

Finally, we note that an advantage of our technique over many other approaches is that we do not need a photocollection of the test individual. For example, Aaron Eckhart and Mark Rosenbaum (first and 7th column in the figure) only have one image in the dataset.

Comparison on aging. In this section, we compare several methods to deep manifold traversal on the aging task. We compare to two alternative data driven approaches that motivate the need for performing traversal with deep features. First, we compare to "shallow" manifold traversal, where we perform our linear traversal algorithm, but in the original pixel space rather than after extracting deep convolutional features. We also compare to interpolation in pixel space between the original input image and the average "senior" image. We also compare to a state-of-the-art technique for image morphing [9], which only requires one target image but requires manual annotation of correspondances between the test and target image.

In the case of aging, the image morphing algorithm requires both a young and an aged photo of the same person, which would not typically be available. Therefore, we chose to evaluate the aging task on Harrison Ford, as young and old images of him are both readily available from Google image search. For

Fig. 3. (Zoom in for details.) Several methods used to change the age of an input image of Harrison Ford.

the image morphing baseline, we show the "halfway" image. The annotationed correspondences are shown as red dots on the original and target image.

The results of our experiment are shown in figure 3. Deep manifold traversal clearly perorms better than both of the other data-driven baselines, producing a sharp image with characteristic aging features. This suggests that traversal in the deep convolutional feature space is indeed necessary. When compared to the image morphing task, the visual clarity of the face are comparable. However, the image morphing algorithm introduces some warping of the face in the intermediate stages. Perhaps the most obvious difference between the two methods is that deep manifold traversal preserves both the background and the clothing of the original image, thus avoiding changes that are irrelevant to the desired change.

Comparison with Szegedy et al. 2014. Existing work has shown that it is possible to make imperceptible changes to images so that deep convolutional networks make high-confidence misclassifications [15]. In this section, we demonstrate that when we vary λ in our manifold traversal algorithm, we can gradually change both the class label of an image *and* a machine learning classifier's prediction, not just the prediction alone.

To do this, we use the convolutional layers of VGG as a feature extractor, and train an SVM using the top 2000 "senior" and "non-senior" faces from LFW to distinguish between VGG features extracted from images with positive "senior" attribute values and negative ones. We then use Platt scaling to transform the SVM decision values into probabilities [48] ranging between 0 and 1, where lower probability value indicates the likelihood for being more "senior".

We construct adversarial "senior" images–which we display on the left in figure 4–as well as perform manifold traversal with three different lambdas, which we display on the right in figure 4. All manifold traversal results were generated using the same set of lambda values: 6e-8, 5e-8, and 4e-8.

Below each image is the class probability of "not senior" assigned to that image by the Platt-scaled SVM. In order to make outputs on both sides comparable, we set the adversarial regularizer so that the adversarial images have comparable decision values to those generated by manifold traversal.

We note several important features of this result. First, the original images all have very high probability of being "not senior". However, after both the adversarial and the DMT modifications, we were able to change the SVM prediction

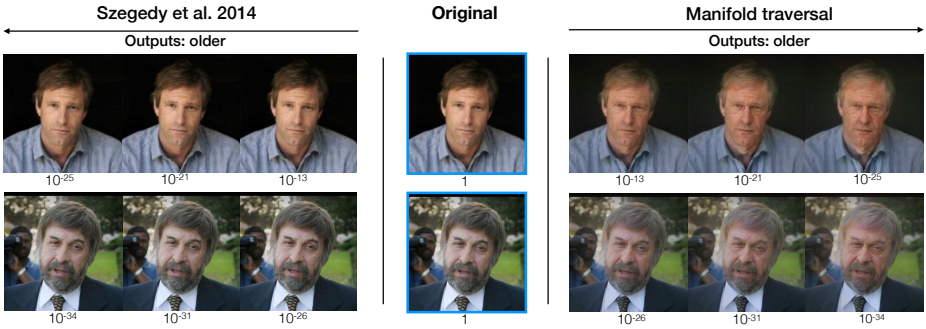

Fig. 4. (Zoom in for details.) Left: "Aging" images generated using the method of [15]. **Right:** "Aging" images generated by deep manifold traversal. The image progression towards the right was generated by gradually decreasing the value of λ. Numbers below each image show the Platt scaled probabilities of an SVM trained on VGG features to distinguish old age, where lower values indicate more "senior".

to be completely confident that the transformed images were of seniors. We find that deep manifold traversal makes meaningful change to the *true label* of the images as well, clearly aging the person in each image. In contrast, the comparable adversarial images fail to change the original images in a human-perceptibly meaningful way.

5.2 Changing hair color via manifold traversal.

To show the versatility of manifold traversal, we also perform manifold traversal to change hair color. This task is different from aging because different hair styles require manifold traversal to focus on a larger variety of shapes than aging does. We perform two traversals: one towards blonde (lighter) hair, and one towards black (darker) hair. To help ensure that the randomly selected test images did not already have blonde or black hair, we selected our 8 random test images from among the top 90th percentile of the "brown hair" attribute.

The results of our hair color experiment are shown in figure 5. The middle row displays the original images in LFW. The top and bottom row show the results of manifold traversal towards lighter hair ("blonde hair") and darker hair ("black hair") respectively.

We note that the hair color traversal generally succeeded despite the variety of hair styles, while again preserving features of the image like clothing and background. The varying hair styles suggest that manifold traversal is able to transform more complex shapes than simply faces.

Of particular interest in this experiment are the changes made other than the color of hair on the top of the head. In most cases facial hair such as eyebrows and beard hair was changed to the appropriate color as well (for example in the first column). Furthermore, when traversing to blonde hair, eye color was occasionally also changed to blue to match (for example, in the 2nd, 5th, and 6th columns).

Fig. 5. (Zoom in for details.) Changing hair color of random (except Aaron Eckhart) 250x250 images from LFW with manifold traversal. **Top.** Manifold traversal to lighter hair. **Middle.** Original image. **Bottom.** Manifold traversal to darker hair. All traversals were performed with the same value of lambda.

6 Experimental Results - AMOS

Does our technique work outside the context of faces? To test this, we also evaluate our method on two tasks using data from the Archive of Many Outdoor Scenes (AMOS) collection of webcams [49]. This dataset contains images from thousands of webcams taken nearly hourly (with some missing data) over the course of several years. While this data lacks the rich set of annotations that LFW has, we are able to construct two tasks based on image timestamps–traversing from winter to summer and traversing from day to dusk.

6.1 Changing from winter to summer.

In this section, we look at if we can learn to transform images from winter to summer given a specific webcam. We collect 2762 images from January and February to form the source "winter" set, and 2858 images from June and July form the target "summer" set. We then select two winter test images which do not occur in either the source or target set and perform deep manifold traversal.

The results of both deep manifold traversal and the image morphing algorithm of [9] on this task are shown in figure 6. In both test images, deep manifold traversal adds leaves to the trees in the foreground, as well as dense foliage to the forest in the background. In the second test image, the grass is made significantly greener, and the snow on the ground begins to fade. We notice that during partial traversal a tree trunk is added in the second experiment (likely due to a viewpoint change), which fades upon complete traversal.

The image morphing algorithm also performs reasonably well when adding leaves to trees, producing leaves of comparable quality to deep manifold traversal. However, we note two notable image artifacts in the morphing algorithm results. First, the trunk of the foreground tree is clearly still visible, despite the fact that there is dense foliage. Second, while the image morphing algorithm did not duplicate the trunk of the tree on the right, there is significant image warping

Fig. 6. (Zoom in for details.) Changing from winter to summer with deep manifold traversal (1st and 3rd row). Tree branches are replaced with leaves (red arrows), dirt appears at the base of a large tree (yellow arrow). At a partial traversal a tree trunk (blue arrow) is duplicated. This may be due to a viewpoint change. For comparison we show image morphing [9] (2nd and 4th row). [9] requires manual annotations (red dots) and uses a single target image rather than a photo collection.

near that tree and the bank of the lake. One possible reason for this may be due to the fact that the image morphing technique relies on a single target image. This means that, if a natural event causes the camera viewpoint to change slightly, the algorithm must also morph the viewpoint, which may be the cause of the odd riverbank location in the morphed image. Deep manifold traversal, however, is robust to such changes during full traversal as such small irregularities are not vital to the label change.

6.2 Scalability

How well does our method scale to larger images? As a demonstration, we performed a manifold traversal on a 4k resolution AMOS webcam. The images were downsampled to 900×600 then a manifold traversal was performed on a random test image. The traversal was from 2051 day images toward 1507 night images — day and night selected by timestamp, dawn and dusk excluded. The test image was not one of the source or target images. Figure 7 shows the traversal result.

Fig. 7. (Zoom in for details.) Deep manifold traversal at 900×600 pixels. The city (left) is changed to make it more similar to nighttime (right). Our data-driven method selects multiple factors to change. The tone of the buildings changes from daytime gray to nighttime blue and nighttime artificial lighting appears in windows (red insets). The waterfront pavilion light and car headlamps are reflected on the water (blue insets).

Our method found that changing tone, adding artificial lighting and reflections of light off the water (see insets in the figure) are the cues which make the image more like nighttime. Interestingly, the sky remains blue as it would during the day. One hypothesis for this is that, because VGG was trained on an object recognition dataset, the sky is treated as background and not represented in the high-level feature space–for example, when classifying birds or airplanes, the sky is background.

The feature matrix is 3559×14088192, which requires 186 GB of storage. Manifold traversal takes 132 minutes and reconstruction takes 43 minutes. In comparison, LFW (250×250, 2000 source and 2000 target images) requires 25 GB (feature matrix is 4001×1671424) and 18 minutes to transform. Our method can transform large images (larger than most generative model demonstrations) and is primarily limited by memory constraints. Furthermore, the manifold traversal time is linear in image size.

7 Discussion and Future Work

In the LFW experiments we use 2000 source and 2000 target images to define the manifold. It is possible to use fewer images at the cost of reduced output quality (figure 8). There are ways to address this limitation. Video sources can generally produce thousands of images easily. Data augmenation could increase the effective size of a small image set. Exploring ways to reduce the number of images while maintaining quality would be future research.

We find that images must be well aligned. For example, in figure 6 the small tree on the right is displaced between the source and target image sets. As a result, a ghostly tree trunk appears at some lambdas (but disappears when lambda is sufficiently small). We note that only the subject needs to be aligned. For example, there is variety in the LFW backgrounds yet this does not prevent our

method from operating on the aligned faces. It may be possible to overcome this limitation by incorporating an image alignment mapping [33] or to automatically identify photos taken from the same viewpoint [50].

Although we gain much from using VGG features, those features are roughly 10x larger than the input image. As a result, holding thousands of 960x540 image feature vectors requires over 128 GB of main memory. These limitations can be overcome by out-of-core methods at the cost of speed. Reducing the size of the deep neural network feature space is future research.

Fig. 8. The effect of varying the number images used to define a manifold.

Many of the best state-of-the-art methods are computational pipelines which combine domain-specific knowledge and specialized algorithms to solve sub-problems of a larger problem. An exciting direction of future research is to see if our generic method can simplify existing state-of-the-art methods by replacing pieces of the pipeline with our data-driven approach.

One possible use case for deep manifold traversal is in data augmentation. Typical data augmentation involves transforming images with label-invariant changes such as horizontal flipping, with the goal of constructing a larger dataset. If we seek to train a deep neural network that, for examples, distinguishes between young and old faces, we could augment our data by performing manifold traversal on other aspects–such as facial expressions or hair color.

8 Conclusion

We introduced a single general purpose approach to make semantically meaningful changes to images in an automated fashion. In contrast to prior work, our approach is not specific for any given task. We leverage the combination of MMD and deep features from convolutional networks to naturally confine the traversal path onto the manifold of natural images. The resulting algorithm scales linearly in space and time (after pre-processing), is extremely general and only requires minimal supervision through example images from source and target domains. However, we believe that the true power of our method lies in its versatility. Without modifications it can be applied to changing the appearance of faces, city skylines or nature scenes. As future work we plan to investigate the use of manifold traversal for active learning and automated image augmentation as pre-processing for supervised computer vision tasks. We hope that our work will be used as a baseline for a variety of computer vision tasks and will enable new application in areas where no specialized algorithms exist.

References

1. Chai, M., Wang, L., Weng, Y., Yu, Y., Guo, B., Zhou, K.: Single-view hair modeling for portrait manipulation. ACM Transactions on Graphics (TOG) **31**(4) (2012) 116

2. Kemelmacher-Shlizerman, I.: Internet based morphable model. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 3256–3263
3. Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Eurographics Symp. on Rendering. Volume 2., Citeseer (2005)
4. Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: Proceedings of the 20th ACM international conference on Multimedia, ACM (2012) 369–378
5. Kemelmacher-Shlizerman, I., Suwajanakorn, S., Seitz, S.M.: Illumination-aware age progression. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 3334–3341
6. Boyadzhiev, I., Bala, K., Paris, S., Adelson, E.: Band-sifting decomposition for image-based material editing. ACM Trans. Graph. **34**(5) (November 2015) 163:1–163:16
7. Laffont, P.Y., Ren, Z., Tao, X., Qian, C., Hays, J.: Transient attributes for high-level understanding and editing of outdoor scenes. ACM Transactions on Graphics (TOG) **33**(4) (2014) 149
8. Neubert, P., Sunderhauf, N., Protzel, P.: Appearance change prediction for long-term navigation across seasons. In: Mobile Robots (ECMR), 2013 European Conference on, IEEE (2013) 198–203
9. Liao, J., Lima, R.S., Nehab, D., Hoppe, H., Sander, P.V., Yu, J.: Automating image morphing using structural similarity on a halfway domain. ACM Transactions on Graphics (TOG) **33**(5) (2014) 168
10. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: Model-based photograph enhancement and viewing. In: ACM Transactions on Graphics (TOG). Volume 27., ACM (2008) 116
11. Laffont, P.Y., Bousseau, A., Paris, S., Durand, F., Drettakis, G.: Coherent intrinsic images from photo collections. ACM Transactions on Graphics **31**(6) (2012)
12. Shih, Y., Paris, S., Durand, F., Freeman, W.T.: Data-driven hallucination of different times of day from a single outdoor photo. ACM Transactions on Graphics (TOG) **32**(6) (2013) 200
13. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision **70**(1) (2006) 77–90
14. Bengio, Y., Mesnil, G., Dauphin, Y., Rifai, S.: Better mixing via deep representations. arXiv preprint arXiv:1207.4404 (2012)
15. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. International Conference on Learning Representation (2014)
16. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. arXiv preprint arXiv:1412.1897 (2014)
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. (2015)
19. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1) (2012) 723–773
20. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2015)

21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. (2014) 2672–2680
22. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems. (2015) 1486–1494
23. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
24. Reed, S., Sohn, K., Zhang, Y., Lee, H.: Learning to disentangle factors of variation with manifold interaction. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). (2014) 1431–1439
25. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: Advances in Neural Information Processing Systems. (2015) 1252–1260
26. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural computation **12**(6) (2000) 1247–1283
27. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM (2001) 327–340
28. Memisevic, R., Hinton, G.: Unsupervised learning of image transformations. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119
30. Sadeghi, F., Zitnick, C.L., Farhadi, A.: Visalogy: Answering visual analogy questions. In: Advances in Neural Information Processing Systems. (2015) 1873–1881
31. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1538–1546
32. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: Advances in Neural Information Processing Systems. (2015) 2530–2538
33. Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I.: What makes tom hanks look like tom hanks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3952–3960
34. Garrido, P., Valgaerts, L., Rehmsen, O., Thormahlen, T., Perez, P., Theobalt, C.: Automatic face reenactment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 4217–4224
35. Thies, J., Zollhöfer, M., Nießner, M., Valgaerts, L., Stamminger, M., Theobalt, C.: Real-time expression transfer for facial reenactment. ACM Transactions on Graphics (TOG) **34**(6) (2015) 183
36. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. In: ACM Transactions on Graphics (TOG). Volume 23., ACM (2004) 399–405
37. Weise, T., Li, H., Van Gool, L., Pauly, M.: Face/off: Live facial puppetry. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation, ACM (2009) 7–16
38. Kholgade, N., Matthews, I., Sheikh, Y.: Content retargeting using parameter-parallel facial layers. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM (2011) 195–204

39. Xiong, X., Torre, F.: Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2013) 532–539
40. Suwajanakorn, S., Kemelmacher-Shlizerman, I., Seitz, S.M.: Total moving face reconstruction. In: Computer Vision–ECCV 2014. Springer (2014) 796–812
41. Kemelmacher-Shlizerman, I., Seitz, S.M.: Collection flow. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1792–1799
42. Wolberg, G.: Image morphing: a survey. The visual computer **14**(8) (1998) 360–372
43. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on **13**(4) (2004) 600–612
44. Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., Seitz, S.M.: Exploring photobios. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 61
45. Fortet, R., Mourier, E.: Convergence de la répartition empirique vers la répartition théorique. Annales scientifiques de l'École Normale Supérieure **70**(3) (1953) 267–285
46. Bengio, Y., LeCun, Y., et al.: Scaling learning algorithms towards ai. Large-scale kernel machines **34**(5) (2007)
47. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: IEEE International Conference on Computer Vision (ICCV). (Oct 2009)
48. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3) (1999) 61–74
49. Jacobs, N., Roman, N., Pless, R.: Consistent temporal variations in many outdoor scenes. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–6
50. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. International Journal of Computer Vision **80**(2) (2008) 189–210