Summer 2016

Deep Semantic Image Interpolation

Joshua D. Little
Washington University in St. Louis

Washington University in St. Louis

School of Engineering and Applied Science

Department of Computer Science and Engineering

Thesis Examination Committee:
Robert Pless, Chair
Tao Ju
William Richard

Deep Semantic Image Interpolation

by

Joshua Little

A thesis presented to the
Graduate School of Arts and Sciences
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Science

August 2016
Saint Louis, Missouri

Contents

List of Figures

Acknowledgments

I would like to thank my parents and my advisor for their patience and guidance.

Joshua Little

WASHINGTON UNIVERSITY IN SAINT LOUIS
August 2016

ABSTRACT OF THE THESIS

Deep Semantic Image Interpolation

by

Joshua Little

Master of Science in Computer Science

Washington University in St. Louis, August 2016

Research Advisor: Dr. Robert Pless

Image datasets often live on a continuum: Images from an outdoor scene vary from day to night, across different weather conditions, and over the course of seasons. Faces age and exhibit different expressions. We consider the problem of taking individual images from these datasets and explicitly manipulating those images to change where they lie on the continuum. We focus on a version of this problem that requires as little input as possible, and we build off of previous work using CNN features to construct an intermediate image manifold on which to manipulate the images. We also investigate a novel way of reconstructing images from their CNN features using alpha compositions of the input images. These technique produce convincing semantic interpolations of images and timelapse video from a variety of sources.

Chapter 1

Introduction

Image datasets often live on a continuum. Images from an outdoor scene vary from day to night, across different weather conditions, and over the course of seasons. Faces age and exhibit different expressions. This thesis considers the problem of taking individual images from these datasets and explicitly manipulating those images to change where they lie on this continuum.

We focus on a version of this problem that requires as little input as possible. Specifically, we consider an algorithm where we use a collection of several hundred images on each end of a spectrum (e.g. young faces and old faces) to automatically build a procedure that can take any new input query image and push it towards one end of the spectrum. The non-query images form two sets, the source images and the target images, that on which direction we want to push the query image.

There is an immense amount of research on learning and using image manifolds [9]. Until recently, these approaches explicitly consider manifolds in "pixel space." One class of approaches uses enough images so that linear interpolation between nearby images gave an effective manifold representation [12]. A second class considers image deformations as an operator to move along the manifold [2].

Interpolating linearly in pixel space gives poor results for our datasets. If we, for example, interpolate linearly between day and night images, as in Figure 1.2, the results don't look convincingly night until we're using almost entirely nighttime image. However, by that point, all of the interesting transient features, like the pedestrians in the foreground, have entirely dissappeared. When we attempt linear interpolation on faces, the results end up looking

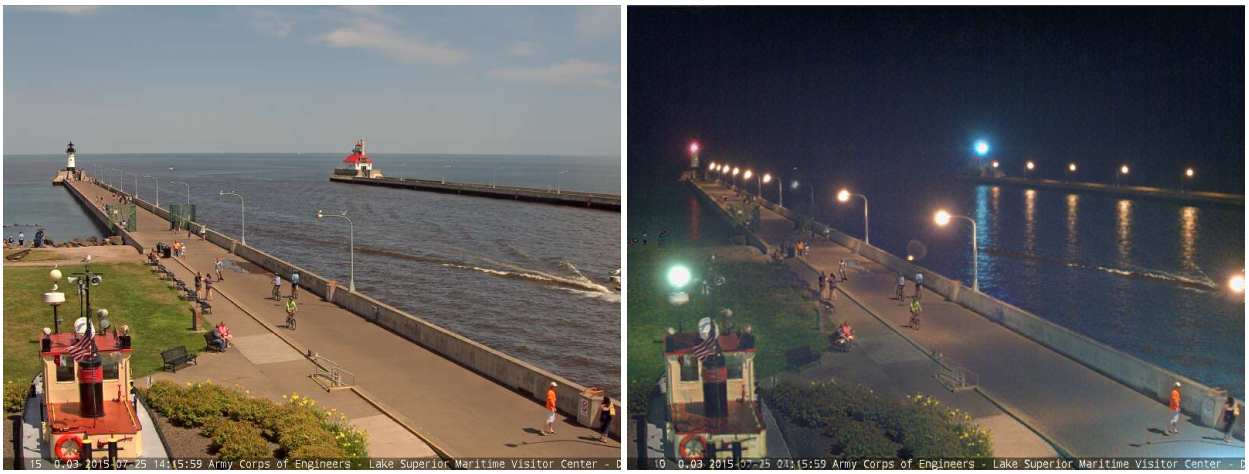
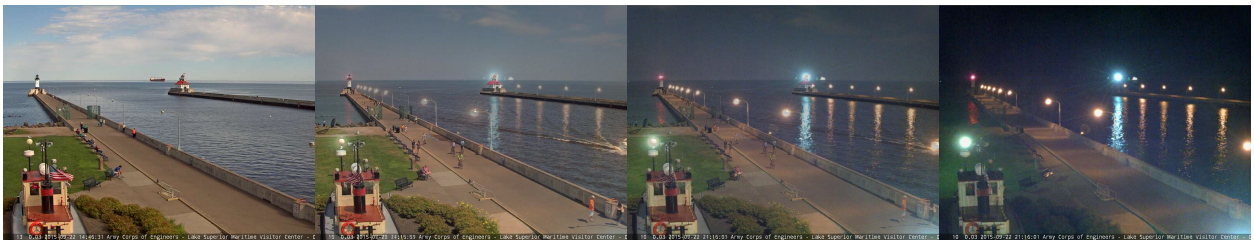


Figure 1.1: Given images from a webcam only labeled day or night, we can take an arbitrary day image (left) and turn it into night image (right) with the same transient features—people walking around, boats in the water—found in that day image.

even less convincing, as in Figure 1.3, since faces almost never line up perfectly. Using more faces in the interpolation may help somewhat. However, as with Figure 1.2, we would still be quickly erasing interesting features unique to the face we're aging.

This shows that these images do not form a linear subspace in the space of all images. However, these images still lie on a manifold. Adding slight deformations or noise to a picture of a face still results in an entirely recognizable image of a face. As the person in the image ages, their appearance changes continuously. Images of a scene like in Figure 1.2



(a) Daytime image. (b) 33% nighttime. (c) 66% nighttime (d) 100% nighttime.

Figure 1.2: Example day-to-night results using linear interpolation. As we add more nighttime image (left to right), the pedestrians and other transient image features slowly disappear. By the time the sky is fully dark (d) the transient features have completely disappeared.

(a) Sean Astin. (b) 50% interpolation. (c) Bob Menendez.

Figure 1.3

change continuously as they transition from day to night and between weather conditions. We cannot model these image manifolds linearly, so we turn to a more expressive model, convolutional neural networks (CNNs).

In the last several years, CNNs have become an important tool for image analysis. When trained to recognize interesting image categories, they create features within their internal representation that have interesting semantic meaning. This work uses this internal representation from a CNN as a representational choice to create a manifold with explicit semantic meaning.

When trained to recognize interesting image categories, CNNs create features within their internal representation that have interesting semantic meaning. By training the CNN to perform semantic tasks like image categorization, the CNN's internal representation is forced to create interesting semantic features in order to perform this task. These semantic features much better follow image manifolds and perform much better for image interpolation, as shown in [4, 1].

Using these intermediate convolutional features as our interpolation space introduces two challenges. First, the mapping from images to convolutional features is not invertible, so we have to solve an optimization problem to convert back to images. Second, the mapping from images to features is many to one, so even if we solve the optimization problem, there are

many possible images that we can get. It is necessary to enforce some prior to encourage this optimization to arrive at a natural image, instead of the many unnatural-looking images that are possible. The work in [4] phrases this optimization problem with two terms, one that encourages the reconstructed image to be consistent with the CNN features, and the other that penalizes large image gradients.

The contributions of this thesis are threefold. First, we pose an alternate image reconstruction optimization to the one proposed in [4]. We attempt to reconstruct images from the output CNN features as a alpha composite of the example source and target images for that dataset, rather than attempt to optimize for the image pixels directly. Second, we are often forced to perform the manifold traversal on lower-resolution images due to time and memory constraints. We show how to use this alternate reconstruction method to reconstruct higher-resolution images, up to the resolution of the original images. Third, we show how to significantly speed up image reconstruction for video inputs by exploiting interframe similarities in the video.

The problem of semantic image interpolation is interesting because it provides a new set of tools manipulating images. Manipulating images across a continuum is important for many applications and currently it is necessary to spend great effort to create algorithms for each application domain. For example, facial aging is used to predict current appearance of children that have been missing for a long time, and there is immense literature about how to age faces properly [3]. In the graphics community there is interest in the variation of outdoor scenes as a function of lighting conditions, and substantial work in how to model those variations [8].

The semantic image interpolation line of work seeks to replace application specific algorithms with an learning framework. We give a user the ability to define two ends of a spectrum (e.g. young faces and old faces) just by giving example data sets. These datasets are increasingly available, for example, LFW annotates a large set of faces with attributes like age, demographics, gender, and so forth [5, 7]. The tools developed in this thesis make image manipulation possible for anyone with access to these datasets, and does not require the development of new algorithms for each possible application domain.

Chapter 2

Image Manifold Traversal

2.1 Baseline Algorithm

The baseline algorithm, devised in [4], comprises two main stages: manifold traversal and image reconstruction. In the manifold traversal stage, we move an image along the image manifold away from our source images and towards our target images. In the image reconstruction stage, we take the new image features produced by the traversal stage and construct a natural-looking image that reproduces those features as closely as possible.

2.1.1 Manifold Traversal

We want to make meaningful semantic changes to our image. Standard pixel space is very poor at representing the semantic properties at an image, so we first map the image into a more useful feature space. While there has been much work on hand-crafted feature representations of images, we instead use the very dense represented afforded by convolutional neural networks (CNNs).

Network Architecture

We extract our feature representation of the image from the middle convolutional layers of a deep CNN. In our case we use VGG19 [11], trained for image object classification on ImageNet (ILSVRC-2014) [10], with 16 convolutional layers and 3 fully-connected layers at

the end We extract our features specifically from the 5th, 9th, and 13th convolutional layers, giving features at three different physical and semantic scales. Each of these three layers has a significantly smaller width and height than the original image due to the network's pooling layers. They have 256, 512, and 512 image channels, respectively, so we still end up with an image representation many-times larger than the original image.

Feature Interpolation

Using these deep CNN features allows us to approximate the image manifold in a more linear fashion. The hope is this allows us to travel along the manifold a short distance linearly to arrive at a set of output features. We compute these output features as an optimization over linear combinations of the source, target, and input image features. We attempt to maximize similarity between the output features and target features, while minimizing similarity between the output features and source features. We control how far the output features move by applying an l_2-regularizer to the feature change. Our objective is then the following minimization:

$$z = \arg\min_{\delta} L(x_0 + X\delta) + \lambda_\delta \|X\delta\|^2, \quad \text{where} \tag{2.1}$$

$$L(x) = \sum_{x_s \in X_s} k(x, x_s) - \sum_{x_t \in X_t} k(x, x_t) \tag{2.2}$$

and z is our final output feature vector, x_0 is our starting input image feature vector, δ is the vector of coefficients we're adding, X is the matrix of source, target, and output features, X_s is the set of source features, X_t is the set of target features, and k is the similarity kernel function. We use an RBF kernel to measure image feature similarity, which allows fast computation of the optimization's score function. λ_δ is the regularizer coefficient and corresponds to how much we are allowing our output image features to change, demonstrated in Figure 2.1.

Figure 2.1: Increasing traversal distance (left to right) strengthens the aging effect. Going too far yields unconvincing results (far right). See Section 3.1.1 for a description of this dataset.

2.1.2 Image Reconstruction

The manifold traversal optimization process gives us a new set of image features, but we still need to map these features back to an image. The CNN we use is not directly invertible, due to the pooling layers, so we treat reconstructing the output image as another optimization problem. We want to find a natural-looking image that reproduces our output features as closely as possible.

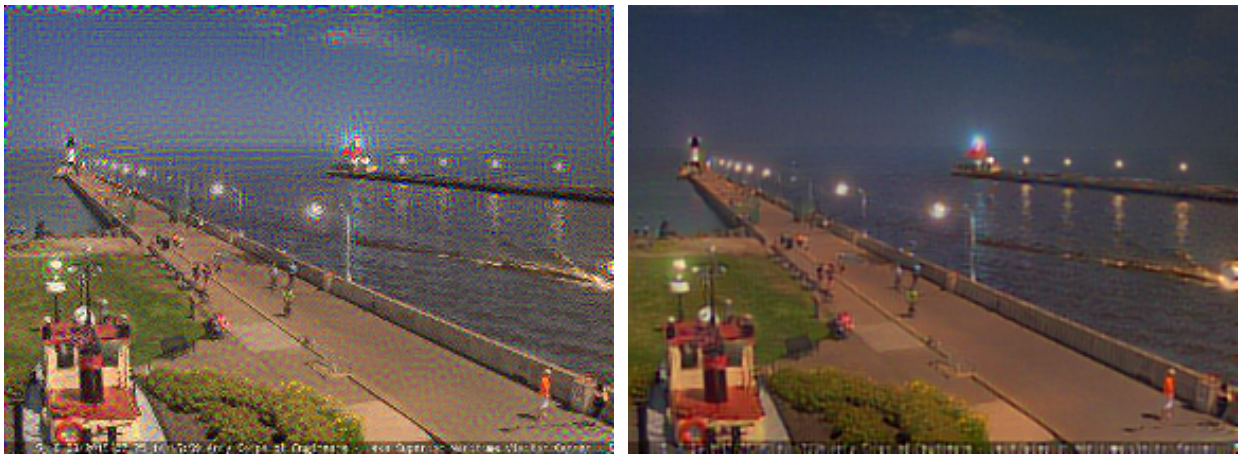
Optimization Criteria

In the baseline algorithm, we directly optimize for the output RGB image pixels. Our primary optimization objective is to minimize the difference between this image's CNN features and the output features from the manifold traversal stage, using a standard l_2-norm:

$$L(I, z) = \frac{1}{2} \|F(I) - z\|^2 \tag{2.3}$$

where I is the image we're optimizing, z is our output feature vector, and $F(I)$ is the feature vector currently produced by I. Using this as our only optimization criterion unfortunately leads to significant color artifacts that are not represented in the CNN features, shown in Figure 2.2a. To deal with these and other miscellaneous artifacts, we apply a total variation norm on the image pixel values, yielding cleaner images as in 2.2b.

$$TV(I) = \sum_{i,j} \left((p_{i,j+1} - p_{i,j})^2 + (p_{i+1,j} - p_{i,j})^2 \right) \tag{2.4}$$



(a) Without regularization. (b) With regularization.

Figure 2.2: Optimizing to the CNN features creates significant color artifacts in the reconstructed image (a) that the total variation regularizer mostly removes (b). See Section 3.1.2 for a description of this dataset.

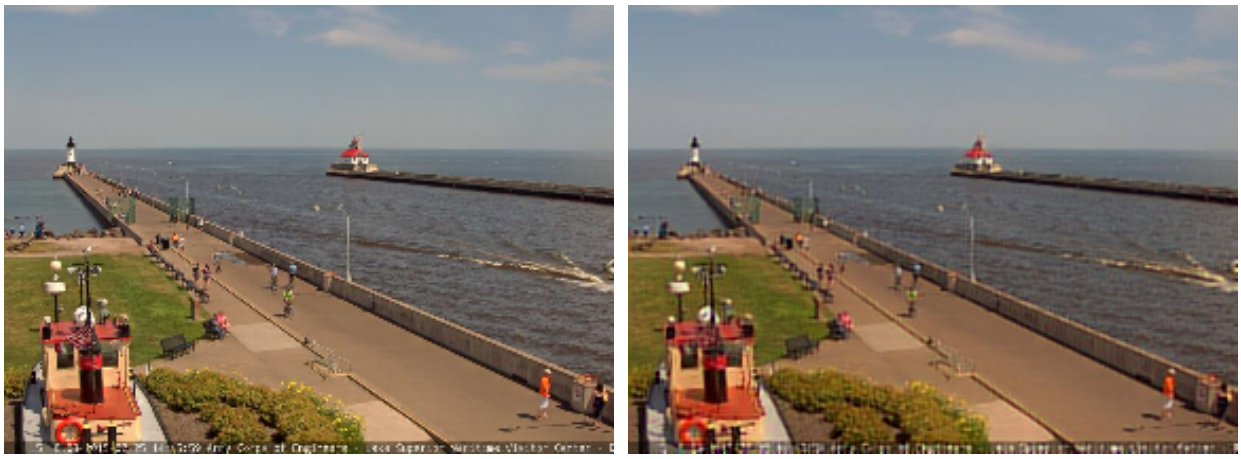
where $p_{i,j}$ is the pixel value of I at position i, j. This norm discourages unnecessary gradients in the image, usually resulting in a more natural looking image. Our final image reconstruction objective is the following:

$$\tilde{I} = \arg\min_{I} \; L(I, z) + \lambda_{tv} TV(I) \tag{2.5}$$

where $\tilde{I}$ is our final reconstructed image.

2.2 Alpha Reconstruction

While optimizing the image reconstruction in RGB space, as in [4], works well for many images, it has a few flaws. The total variation norm causes noticeable color shifting in the image (as in Figure 2.3). The CNN features also do not properly reconstruct fine detail not found in the starting image. Both of these problems can be addressed if we restrict our reconstruction optimization from the space of all RGB images to the space of all alpha composites of images in our dataset, using many of our existing images during the reconstruction instead of just the query image.



(a) Original image. (b) With regularization applied.

Figure 2.3: The total variation regularizer can lead to a noticeable color shift (a reddening in this case) in the image reconstruction, even when the image features are not changed.

2.2.1 Alpha Composition

Instead of searching over all RGB images I for our reconstruction, we search over all sets of alpha coefficients A, where the output image I is computed as the sum of our dataset images weighted by the coefficients A. Each pixel i, j of each input image I_k gets its own alpha coefficient $(A_k)_{i,j} \in [0, 1]$, but each color channel of an image uses the same set of alpha coefficients. The image optimization for this alpha reconstruction then is

$$\tilde{A} = \arg\min_{A} \; L(I(A), z) + \lambda_{tv} TV(I(A)) \tag{2.6}$$

$$I(A) = \sum_{k} I_k \circ A_k \tag{2.7}$$

where loss L and regularizer TV are the same as before, and our output image is $I(\tilde{A})$.

Images to Compose

Using every source and target image during this alpha reconstruction would slow the optimization process to a crawl and heavily increase memory usage. To keep things tractable, we instead use only the starting image and the 10-20 source or target images with features

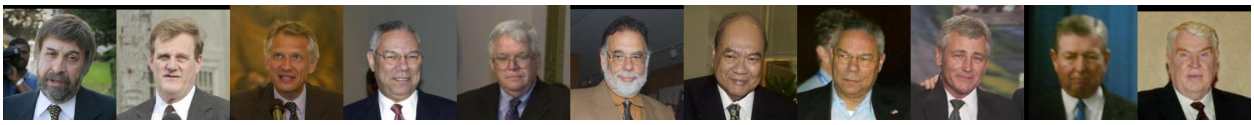


Figure 2.4: Example input images for alpha composition. Left to right is the starting image and then the 10 target images in increasing distance from the output features. These were used to reconstruct the 3rd image in Figure 2.1.

closest to the output features. Even with this simplification, we find improvements for the color shifting and fine details when compared to output from the original RGB regression.

2.2.2 Upscaling Output Images

The manifold traversal stage is very memory intensive, limiting the size of the images we may use. However, we have the fullsize versions of the images we're using for the alpha reconstruction. By using a larger alpha coefficients map A and downsampling before applying the loss function, we can exploit these fullsize input images to reconstruct a larger output image than the output feature vector allows for on its own. The new loss function is

$$\tilde{A} = \arg\min_{A} \; L(DS(I(A)), z) + \lambda_{tv} TV(I(A)) \tag{2.8}$$

where DS downsamples the image (bilinearly) to match the size of our output features. By downsampling only a small amount (around 1.5x or less) and applying the total variation regularizer to the large image, we can maintain reasonable results, like shown in Figure 2.6a. If we try to downsample too much, the optimization only updates a subset of the pixels to match, as in 2.6b.

2.3 Video

One of the main benefits of automated image editing methods like our CNN manifold traversal is that they can be applied to many images at once. This allows manifold traversal to be applied to video about as easily as to a single image. However, video has a property we can exploit to significantly improve the image reconstruction stage.

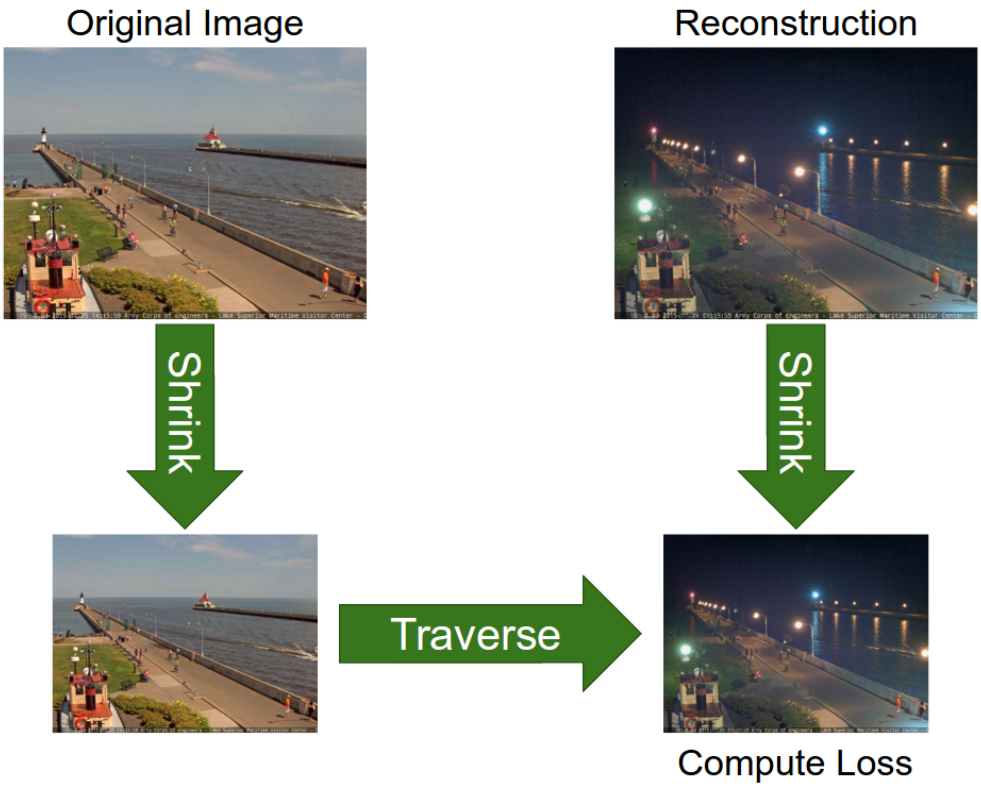


Figure 2.5: The upscaling process. In order to follow memory and time constraints, we often have to downsample our original images (left) before performing the manifold traversal and reconstructing in this smaller image space (bottom right). When performing alpha reconstruction, we can reconstruct in a larger image space (top right) and only downsample to compute the reconstruction loss function.

(a) Resizing by 1.33x. (b) Resizing by 2.8x.

Figure 2.6: Upscaling the alpha coefficient map can lead to a higher-resolution output image (a), but attempting it to upscale it by too much can lead to degenerate reconstruction results. See Section 3.1.2 for a description of this dataset.

2.3.1 Incremental Reconstruction

For stable or high-framerate video, each video frame is going to be very similar to its adjacent frames. We can significantly speed up the image reconstruction stage by performing the optimization incrementally, using the optimization output for each frame, the RGB image or the alpha coefficients, as the starting point for reconstructing the next frame. Not only does this allow the reconstruction process to converge much more quickly for video frames after the first, it encourages adjacent frames to arrive at the same reconstruction for regions in the video that are not changing from frame to frame, like in Figure 3.10.

The downside of using an incremental approach with alpha reconstruction is that we have to use mostly the same images in the alpha composition for each frame. If we use a completely new set of images each time, we have to throw out the alpha coefficients instead of reusing them. Output image quality does not seem adversely affected, however, as long as we always use the current input frame in the alpha composition. If the video viewpoint or scene changes drastically during a video, it may require an excessively large number of input images to perform the alpha composition, though.

Chapter 3

Experimental Results and Applications

We show visual results for both still imagery and video/timelapse imagery across several different categories of image content.

3.1 Still Images

3.1.1 Labeled Faces

The first set of images we use is the Labeled Images in the Wild (LFW) dataset [5, 7]. LFW is a collection of 13k roughly aligned images of public figures' faces. Each image is labeled

Figure 3.1: Four example LFW images, of Aaron Eckhart, Mark Rosenbaum, Elizabeth Dole, and George W. Bush.

Figure 3.2: Mark Rosenbaum, aged increasing amounts (left to right) via our manifold traversal. The top row uses RGB reconstruction, while the bottom row uses alpha reconstruction.

with the subject's name and annotations of 73 different attributes ranging from "age" to "smiling" and "mustache," and so forth. We follow the example in [4] and use the 2000 youngest rated images in the "age" category as our source images and the 2000 oldest ratest images as our target images. The intent is to age our input images to make the subject look older, while keeping the face recognizably the same person. We performed the manifold traversal on the original 250x250 pixel images, so no upsampling is used.

Figure 3.2 shows on top reconstruction by directly optimizing the RGB values (RGB reconstruction from from Section 2.1.2) on top, and on bottom reconstruction by optimizing based on the alpha channel (alpha reconstruction from Section 2.2.1). The example images and output alpha masks for these alpha reconstructions can be found in Figure 3.3. The manifold traversal (and both reconstruction methods) slowly add wrinkles to Mark's face and turn his hair and beard white, while leaving the background and his clothes largely intact. Traversing too far in feature space (like in the far right result) causes the image too look unnatural, as the CNN features are an imperfect approximation of the image manifold, or linear interpolation is valid only for shorter traversal distances.

The RGB reconstruction has a redding effect on Mark's face, getting worse as we age him more. This is a common effect with the RGB reconstruction and largely stems from the total variation regularizer. The alpha reconstruction doesn't have this reddening effect, since it is constrained to the colors in the selected input images. See Figure 2.4 for example input images for this face. Notably, other than avoiding the color shift, the alpha reconstruction is

nearly identical to the RGB reconstruction. Alpha reconstruction is not much less powerful, even when we're using only 11 images to construct the output image.

3.1.2 Lake Superior Marina

The second set of images we use is from a webcam at the Lake Superior Maritime Visitor Center in Minnesota (images archived in AMOS [6]). The full dataset used consists of 12k images from all of 2015 and the first three months of 2016. We use 2000 randomly sampled day images as the source images and 2000 randomly sampled night images as the target images. The intent is to take an arbitrary day image and turn it into a night image with the same transient features—people walking around, boats in the water—found in that day image. We performed the manifold traversal on 600x450 images and upsampled to the original 600x800 image size. With all examples, upsampling is performed naively for the RGB reconstructions and via our alpha scaling for the alpha reconstructions.

Figure 3.5 shows an original day image and the nighttime version of that image for three different traversal distances (from three different regularizer strengths) using RGB reconstruction. The manifold traversal successfully makes the sky dark, turns all of the lamps on, and adjusts the ground and water reflections to match the lights. Unfortunately, we still have the shadows from the sun on all of the people, and the people themselves seem too bright in many instances. As with the face example before, the RGB reconstruction is much redder than it ought to be compared to the ground truth night images.

Figure 3.6 shows a comparison between RGB reconstruction and alpha reconstruction on our example daytime image. The alpha reconstruction again avoids the significant color shift. The alpha reconstruction is also crisper than the RGB reconstruction, largely due to using the higher-resolution input images during the reconstruction.

The text along the bottom is less blurry for the alpha reconstruction (e.g. in Figure 3.7) because it can mostly copy it wholesale from the input images. An interesting aside, though, all of the text along the bottom is more or less identical to in the original image, except for the hour on the timestamp. The manifold traversal can tell that most of the text is uncorrelated (or unchanged) with the change from day to night, but it knows that the ten's digit on the

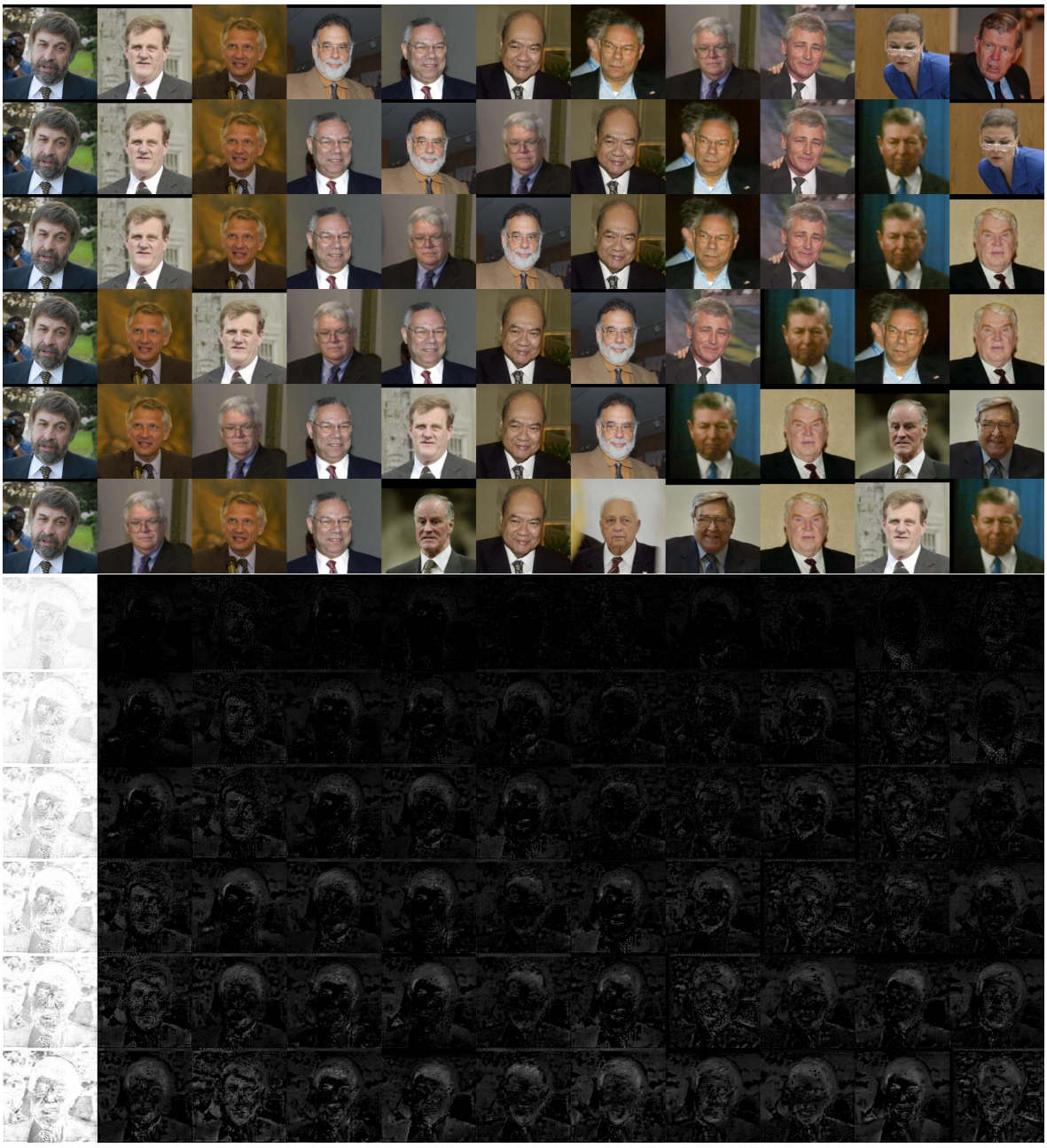


Figure 3.3: Example images (top) and alpha masks (bottom) used for LFW alpha reconstruction. The example images consist of the input image (left), and the ten source and target images closest to the output features from the manifold traversal. When these example images are multiplied by their alpha masks and summed up, we get the alpha reconstruction results in Figure 3.2.

Figure 3.4: Three example marina images, during the day (left), sunset (middle), and night (right).

Figure 3.5: Marina results with different traversal distances using RGB reconstruction. As we increase the traversal distance (left to right), the lamps turn on and the sky darkens, but the pedestrians and boat trail remain fully vivid.

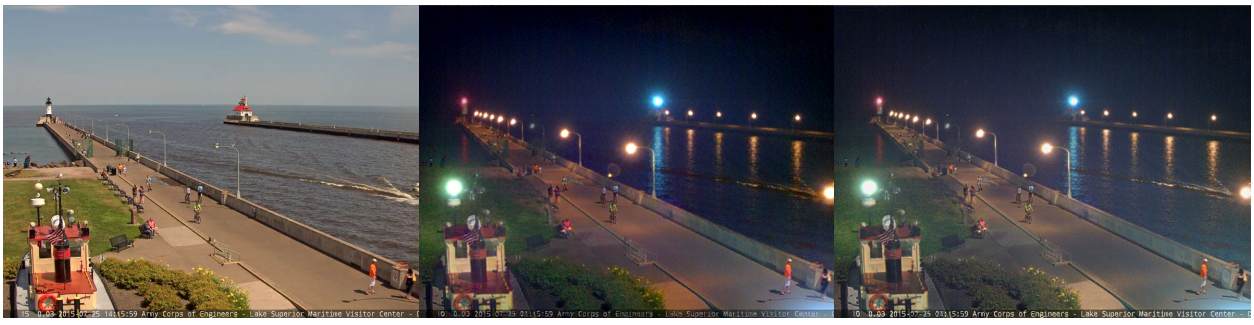


Figure 3.6: Marina day-to-night results, with the downsampled original image (left), the RGB reconstruction (middle), and the alpha reconstruction (right).

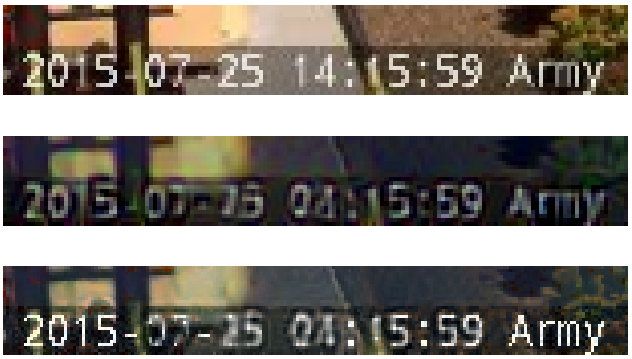


Figure 3.7: A closeup on the marina images' timestamps in the original image (top), the RGB reconstruction (middle), and the alpha reconstruction (bottom). The alpha reconstruction text is cleaner than the RGB reconstruction, especially in places where the text is uncorrelated with time of day.

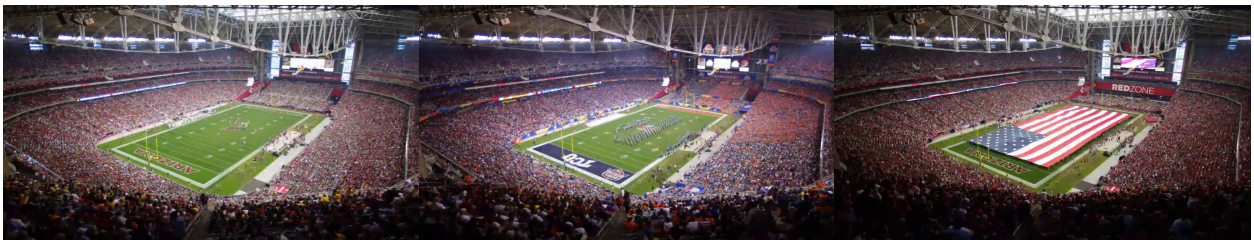


Figure 3.8: Three example Fiesta Bowl image, of the game before (left), the Fiesta Bowl itself (middle), and the game after (right).

hour *is* correlated. The manifold traversal changes this digit from 1 to 0, corresponding to a change from 2pm to 4am, consistent with the change from day to night.

3.2 Video

3.2.1 Fiesta Bowl

The first video we use is a timelapse of the 2010 Fiesta Bowl, in Phoenix, Arizona. We use the 320 frames of the Fiesta Bowl game as the source group, and the 488 total frames of the matches before and after the Fiesta Bowl as the target group. Our test images are then some

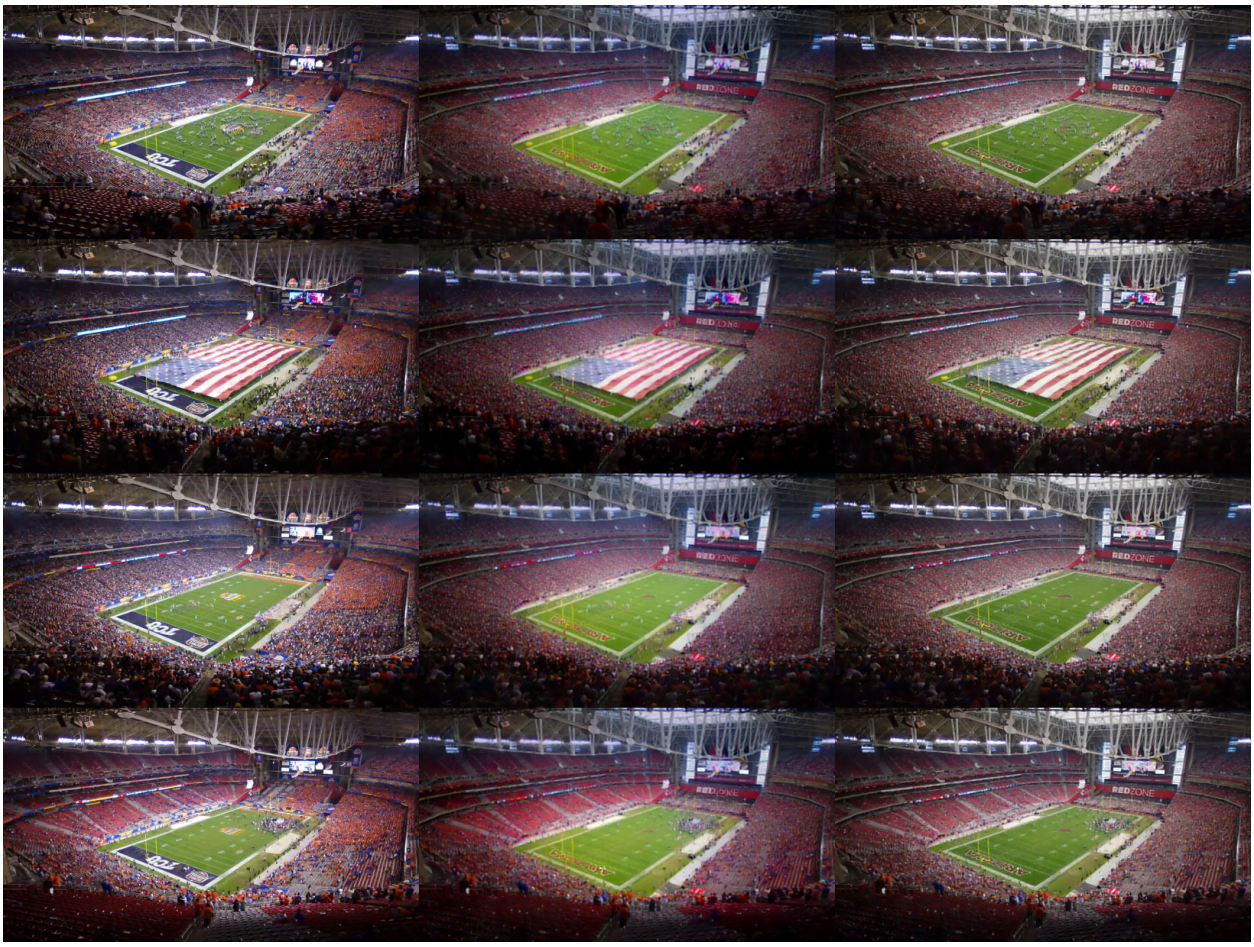


Figure 3.9: Fiesta Bowl results for four different frames (top to bottom), with the original images (left), the RGB reconstructions (middle), and the alpha reconstructions (right).

subset of the 320 Fiesta Bowl frames. (Overlapping our "training" and "testing" image sets like this doesn't matter in this unsupervised context.) We performed the manifold traversal on 480x270 pixel images and upsampled the reconstructed images by 1.5x to 720x404 pixels during reconstruction.

Figure 3.9 shows four example image frames from the video and the RGB and alpha reconstructions for those frames. The manifold traversal successfully changes the in-zone decorations and the center logo on the field, while leaving the the people on the field visible. Both reconstruction methods add in the "Red Zone" sign at the back, properly change the skylight to always be day, and get rid of the blue and orange color scheme amongst the audience. On the downside, they both have trouble maintaining the people along the sidelines.

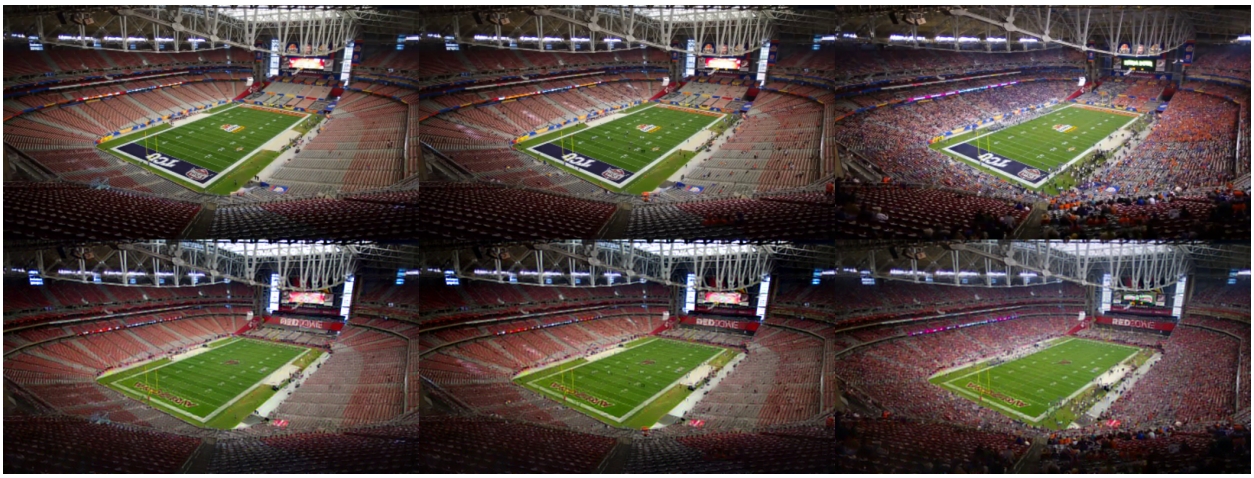


Figure 3.10: The text in the incremental RGB reconstruction slowly improves as the video progresses (left to right).

The people on the sideline don't stick around as much likely because there are many more people on the sideline during the Fiesta Bowl, and the manifold traversal picked that up as a meaningful difference and tried to erase them.

Alpha reconstruction does much better at reconstructing the "Arizona" and "Red Zone" text in the image, as well as the smaller details in the field. While RGB reconstruction had to make the text and symbols from scratch using just the convolutional features as guides, the alpha reconstruction could just copy them wholesale from the source images. The RGB reconstruction has color artifacts on the far half of the flag, which the regularizer fails to remove due to the sharp gradients in the image. The RGB reconstruction also exhibits a blueing effect across the image, notably making the field look yellower, in contrast to the reddening effect the LFW and marina examples exhibit. The alpha reconstruction largely avoids these color artifacts and the color shift, as before.

When applying the incremental approach to RGB reconstruction, the "Red Zone" text in the RGB reconstruction does improve over time, as shown in Figure 3.10. The CNN features for a single image seem to have too much ambiguity for an accurate RGB reconstruction, but the frame-to-frame noise in the CNN features appears to slowly resolve that ambiguity.

Figure 3.11: Example Raking images.

3.2.2 Raking Futility

The second video we use is a timelapse of someone raking leaves up in their backyard. We use the 250 frames at the end of the video as our source group, and the 250 frames at the beginning as our target group, with the intent of erasing any progress the person makes in raking throughout the video. No matter how much he rakes there should still be the same number of leaves on the ground. We performed the manifold traversal on 360x270 pixel images and upsampled to 544x408 pixels during reconstruction.

Figure 3.12 shows three example image frames from the video and the alpha reconstructions of those frames. The manifold traversal successfully determines that the leaves on the ground are the salient feature, and it adds leaves as the person is raking, erasing most of that person's progress. There is still a noticeable decrease in leaves as he is raking, though. Increasing the distanced traveled along the manifold may help maintain peak leafiness, but it will also worsen a second problem with the video. In many of the frames, parts of the person have already been erased and replaced with the background. Increasing the traversal distance would make the problem worse, erasing even more of the person, as shown in Figure 3.13. The problem seems to be the training set is very small (250 source and 250 target images) and does not have many examples of the person walking around. The manifold then does not have a very robust representation of this person or what to do with them when performing the traversal. Even with this small of a training set, the traversal and reconstruction still perform reasonably for most frames.

Figure 3.12: Raking results for three different image frames (left to right), with the original images (top) and the alpha reconstructions (bottom).

(a) Original (b) Short Traversal (c) Long Traversal (d) Target

Figure 3.13: Raking results as we increase our traversal distance. We have (a) the original input image, (b & c) results using short and long manifold traversals, and (d) an example of the target images we're approaching. When we perform the manifold traversal, we slowly erase the person and their lawnmower from the image.

Chapter 4

Conclusions and Future Work

4.1 Conclusion

The baseline method and RGB reconstruction perform reasonably on the Labeled Faces in the Wild (LFW), marina, and Fiesta Bowl datasets. The contributions in this thesis offer improvements, however:

- The novel alpha reconstruction method performs better in places where manifold traversal seeks the change small image features. It also corrects a problem where the image regularizer tends to slightly change image colors.
- The alpha regularization also supports higher resolution image traversal by providing a bridge between high-resolution image reconstruction and the low resolution imagery from which CNN features are computed.
- The incremental reconstruction increases the consistency of the reconstruction as images change in a video. It also gives a significant improvement to the running time when the algorithm is applied to video.

Altogether, these improvements lead to a significantly cleaner output image that more closely matches the target image group, without losing more details from the original image. This supports the broader use of the manifold traversal approach for more general semantic image editing.

4.2 Future Work

We continue to be constrained by large memory requirements, both in main memory and GPU memory. If we can reduce the algorithms memory usage, or devise a clever method for performing the algorithm piecemeal over each image, we could run the manifold traversal on the original high-resolution images instead of downsampling. In addition to producing higher-resolution outputs, the per-pixel noise introduced by the reconstruction process would be less harmful.

The incremental alpha reconstruction method for video currently requires using one set of images for alpha compositing every output frame. For video with significant changes in POV or scene, we may require a very large number of images to be able to reconstruct every frame. It may be possible to use a much smaller set of images and swap out only a few at a time to maintain the speedup and consistency benefits of incremental reconstruction.

Tuning how far the manifold traversal goes is currently done by manual trial-and-error. Finding a good total variation regularizer coefficient is not straightforward, and the ideal values change depending on the size and content of the input images. It should be possible to automate this search for reasonable coefficients by comparing the output images to our input images and example images. When the regularization strength is in the sweet spot, the output image varies smoothly with the regularizer coefficient. Too strong regularization leads to output images identical to the input images, while too weak regularization results in a sudden loss of unique features in the input image, replaced with target image features.

References

[1] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. *arXiv preprint arXiv:1207.4404*, 2012.

[2] Gareth J Edwards, Christopher J Taylor, and Timothy F Cootes. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 300–305. IEEE, 1998.

[3] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, Nov 2010.

[4] Jacob R. Gardner, Matt J. Kusner, Yixuan Li, Paul Upchurch, Kilian Q. Weinberger, and John E. Hopcroft. Deep manifold traversal: Changing labels with convolutional features. *CoRR*, abs/1511.06421, 2015.

[5] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[6] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[7] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[8] Srinivasa G. Narasimhan, Chi Wang, and Shree K. Nayar. *Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III*, chapter All the Images of an Outdoor Scene, pages 148–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[9] Robert Pless and Richard Souvenir. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*, 1:83–94, 2009.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C.

Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Deep Semantic Image Interpolation, Little, M.S. 2016