
Learning Distributions of Transformations from Small Datasets for Applied Image Synthesis

by

Amy (Xiaoyu) Zhao

B.A.Sc., Engineering Science, University of Toronto, 2010
S.M., Electrical Engineering and Computer Science, M.I.T., 2015

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology



February 2020

© 2019 Massachusetts Institute of Technology
All Rights Reserved.



Signature of Author: Signature redacted

Department of Electrical Engineering and Computer Science
October 11, 2019

Certified by: Signature redacted

  John V. Guttag
Dugald C. Jackson Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: Signature redacted

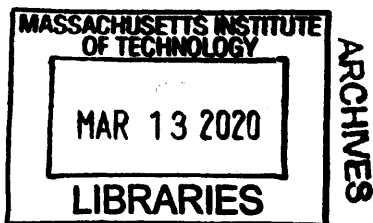
  Frédo Durand
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: Signature redacted

Adrian V. Dalca
Instructor of Radiology, Harvard Medical School
Thesis Supervisor

Accepted by: Signature redacted

   Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students



Learning Distributions of Transformations from Small Datasets for Applied Image Synthesis

by Amy (Xiaoyu) Zhao

Submitted to the Department of Electrical Engineering and Computer Science
on October 11, 2019, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Much of the recent research in machine learning and computer vision focuses on applications with large labeled datasets. However, in realistic settings, it is much more common to work with limited data. In this thesis, we investigate two applications of image synthesis using small datasets.

First, we demonstrate how to use image synthesis to perform data augmentation, enabling the use of supervised learning methods with limited labeled data. Data augmentation – typically the application of simple, hand-designed transformations such as rotation and scaling – is often used to expand small datasets. We present a method for *learning* complex data augmentation transformations, producing examples that are more diverse, realistic, and useful for training supervised systems than hand-engineered augmentation. We demonstrate our proposed augmentation method for improving few-shot object classification performance, using a new dataset of collectible cards with fine-grained differences. We also apply our method to medical image segmentation, enabling the training of a supervised segmentation system using just a single labeled example.

In our second application, we present a novel image synthesis task: synthesizing time lapse videos of the creation of digital and watercolor paintings. Using a recurrent model of paint strokes and a novel training scheme, we create videos that tell a plausible visual story of the painting process.

Thesis Supervisor: John V. Guttag

Title: Dugald C. Jackson Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Frédo Durand

Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Adrian V. Dalca

Title: Instructor of Radiology, Harvard Medical School

Acknowledgments

I would like to start by thanking my thesis committee: John, Frédo, Adrian and Bill. Their feedback throughout my work has been invaluable. I want to thank my advisor, John, for his unwavering support and positivity. Over the past several years, John has never failed to encourage and celebrate my wins. Perhaps more importantly, he has also never failed to empathize briefly with my failures and promptly start looking for other plans and solutions. Not only has John's guidance helped me to improve as a researcher, he has also taught me much about grit, compassion and integrity in all aspects of life. My co-advisor, Frédo, has been a great source of guidance over the years as well. I found myself repeatedly impressed by Frédo's breadth *and* depth of knowledge in computer graphics and computer vision. My ideas were better formulated and my papers were more clearly written thanks to Frédo's input. I want to thank Adrian for dedicating so much of his time towards helping me grow as a researcher. Adrian's guidance was valuable not only because of his formidable technical ability, but for his outlook on coming up with research ideas and finding creative solutions. I want to thank Bill for joining my committee and lending his insights to my research. Bill's unique and thoughtful approach to research has opened my eyes, on multiple occasions, to new ways of describing and approaching problems.

I am grateful to my labmates: Jose, Ani, Katie, Harini, Maggie, Divya, Davis, Marianne, Tiam, Jen, Guha, Joel, Yun, Anima, Jenna, and Garthee. They shared the struggles of a PhD with me, and made my time at MIT memorable and fun.

I would like to thank my then-boyfriend, now-(since-my-thesis-defense)-fiancé, Rob, whom I specifically requested to not propose at my defense but has always insisted on delighting and slightly aggravating me. Rob has been a continual source of encouragement and the occasionally much-needed source of perspective. Rob was there to celebrate my successes, but also to remind me that failures are temporary and can be overcome.

I want to thank my parents, Qihua and Rongqiong. They have always pushed me to reach higher and be better, but more importantly, they have sacrificed much to offer me the opportunities to set such lofty goals. All of my successes (but not my failures) are a reflection of their dedication. I also want to thank my sister, Linda, who inspires me to be a better sister, coding mentor, and friend.

Last but not least, I want to thank the other animals in my life: Mochi, Mocha, James and Michel, whose paws have warmed my shoulder and my heart through many a paper deadline.

Introduction

“Data! Data! Data! I can’t make bricks without clay.”

The Adventure in the Copper Beaches
Sir Arthur Conan Doyle

As machine learning techniques are developed and incorporated into systems around the world, our reliance on data continues to grow. The most powerful learning-based methods – integrated into products for speech recognition [176], facial recognition [156], and content recommendation [21, 57], to name a few – are trained on thousands to millions of labeled examples.

Since the renaissance of deep learning, much of the research in computer vision has focused on applications where large datasets are available. These applications include image classification [42], object segmentation [98], or human pose estimation [7]. In many areas, the largest advancements have been presented by companies with large repositories of application-specific data, such as Facebook [156]. Outside of these few application areas, the need for large amounts of labeled training data is a major barrier to the use of learning-based methods.

Researchers have devised a variety of ways of hurdling these barriers and learning from small datasets. These approaches fall into several major categories: crowd-sourced data annotation, novel learning methods, novel network architectures, and data augmentation.

Crowd-sourced data annotation has become a popular avenue for collecting labeled data at large scales. Amazon Mechanical Turk [5], which provides a platform for outsourcing digital tasks, is often leveraged for obtaining common knowledge labels such as scene [192] or action classes [83]. However, there are many real-world applications where expertise, time, or cost requirements preclude the use of crowd-sourcing. In this work, we discuss two such areas: specialized object classification, and medical image segmentation.

A rich area of novel learning methods that deal with small datasets is transfer learning. In transfer learning, knowledge learned from larger datasets is used to help with learning on more specialized tasks with limited data [25, 65, 124, 184]. Another area of a novel training schemes is meta-learning [50, 150, 164], where the model optimization procedure is designed to facilitate learning from related tasks. Aside from using in-



Figure 1.1: We present a learning-based data augmentation method that transforms existing examples (top) into new examples (bottom) differing in spatial configuration (*e.g.*, shape, rotation) and appearance (*e.g.* specular effects).

novative training schemes, researchers have designed specialized network architectures such as multiple modules for foreground and background synthesis [15], or boundary prediction modules for one-shot object segmentation in videos [24]. These modules can be seen as a way of incorporating prior knowledge about the problem into the model, helping it to learn from fewer training examples.

In learning-based tasks, data augmentation is widely used to increase the amount of training data and to reduce overfitting. Most methods utilize simple, parameterized transformation functions such as rotation and scaling [67, 92]. Recently, there has been some interest in learning how to perform data augmentation more effectively by creating combinations of transformations [37, 135]. Several works explore the synthesis of more examples by learning from existing data [60], or using image synthesis techniques [125].

In this thesis, we explore ways to utilize the latter two approaches to learn from small datasets. We design image synthesis models that incorporate prior knowledge in their design, facilitating learning on small datasets. We apply some of these approaches to data augmentation. We demonstrate the utility of our methods in several areas where the scarcity of labeled data presents a barrier to the use of modern machine learning and computer vision techniques.

■ 1.1 Few-shot object classification

One such area is the classification of specialized objects. While there are many large-scale image classification datasets focusing on animals and common objects [42, 91, 98], obtaining labeled examples for more specialized objects (*e.g.*, automobile or aircraft models [89, 103, 181]) can be challenging, often resulting in just a few examples of each class.

In Chapter 2, we consider the problem of few-shot object classification, in which only a few labeled examples of each class are available as training data. We focus on identifying collectible cards with fine-grained differences, under various spatial orientations and lighting conditions. When working with small labeled datasets, it is common to rely on hand-engineered data augmentation functions to synthesize more labeled examples. However, these functions are often too simple to produce realistic and diverse new examples. We present a novel data augmentation method that *learns* to synthesize varied

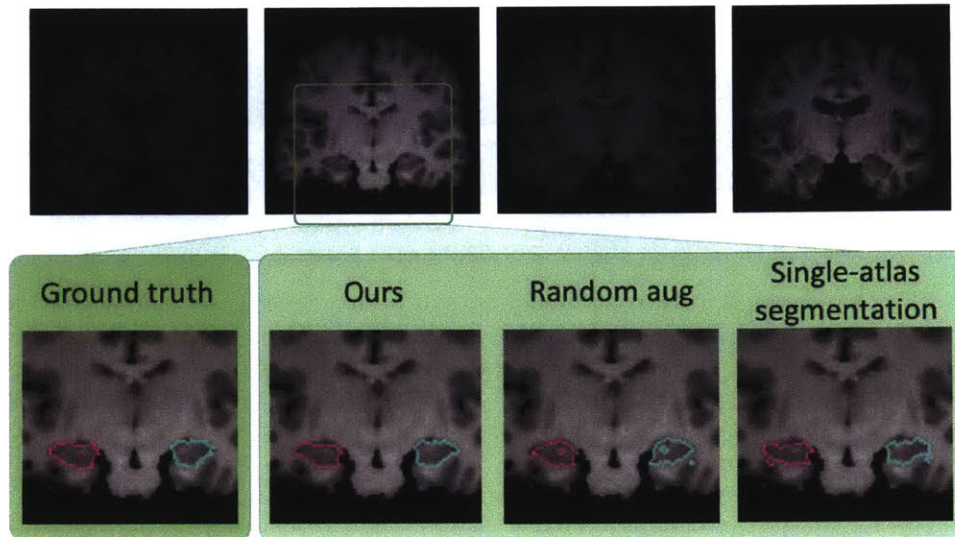


Figure 1.2: We design a learning-based data augmentation method that leverages unlabeled examples to synthesize realistic and varied new examples. This method enables supervised segmentation using just a single labeled example.

and realistic examples from a small labeled dataset. We show some sample results from our method in Figure 1.1. We then demonstrate that training a supervised classifier with these examples produces more accurate results than traditional hand-tuned data augmentation.

■ 1.2 Medical image segmentation

Another area that suffers from the scarcity of labeled data is biomedical imaging. In many biomedical imaging tasks such as diagnoses or treatment planning, a core task is to quickly and accurately delineate anatomical structures in images. This task is referred to as segmentation. When enough labeled data is available for a particular modality and anatomical region, supervised deep learning-based segmentation methods produce state-of-the-art results. However, obtaining manual segmentation labels for medical images requires considerable expertise and time. In most clinical image datasets, very few manually labeled images are available.

In Chapter 3, we use a variant of our learned data augmentation approach to synthesize training data for segmenting magnetic resonance images (MRIs) of human brains. Our method requires only a single labeled example, and leverages unlabeled examples to synthesize examples with realistic anatomical and intensity variations. We show that our synthesized examples enable the training of a convolutional neural network for MRI segmentation, even when only a single labeled example is available. The trained segmentation model outperforms existing state-of-the-art methods for one-shot segmen-



Figure 1.3: We design a recurrent model to synthesize time lapse videos that tell a visual story of how a painting might have been created. The model receives only the completed painting (the rightmost patch) as input. Here, we show two examples of videos synthesized by our method.

tation [190]. We show some examples of segmentation results in Figure 1.2.

■ 1.3 Stochastic video synthesis

Chapters 2 and 3 illustrate how to use image synthesis techniques to perform data augmentation more effectively, with applications in classification and segmentation. In Chapter 4, we introduce a new image synthesis task: creating time lapse videos depicting the creation of paintings. There are no large-scale datasets of such videos. Furthermore, there is significant variability in how people paint; different artists might complete parts of a scene in different orders, using different brush sizes and strokes.

We present a method for synthesizing videos that mimic the painting process of human artists. We decompose time lapse videos into spatial patches and short temporal segments, facilitating training with a small video dataset. We show that our model captures a distribution of plausible time lapse videos; some examples are shown in Figure 1.3. To our knowledge, ours is the first work that attempts to predict distributions of videos of the past, given a single current image as input.

■ 1.4 Contributions

In this thesis, we present the following contributions:

Image synthesis through transformations. We approach image synthesis through the successive application of transformations. Many existing image synthesis techniques take an image as input and create a new image as a direct output of a neural network [55, 73, 169, 194]. Applying these *direct synthesis* approaches to downstream tasks such as data augmentation can be challenging, since these methods do not necessarily preserve the semantic content of the original input image [97, 146]. In contrast, we use neural networks to output functions that we use to transform existing images in label-preserving ways. We show that using transformations can help to leverage the original

image content in the synthesis process, which can be useful for preserving important information such as anatomical structures in medical images, or class-related information in objects.

Domain knowledge-constrained transformations. We show how to use domain knowledge to design useful constraints for transformations. We use separate spatial and appearance transformations to capture viewing angle and lighting variations in photographs of planar objects, and to capture anatomical and intensity variations in brain MRI scans. We use a recurrent model of appearance transformations to capture the repetitive and additive nature of paint strokes. By restricting the space of transformations in a meaningful way, we facilitate the learning of transformation distributions from small datasets.

Leveraging distributions of transformations. We demonstrate that sampling transformations from our learned distributions can be used to synthesize diverse examples, which are useful for the downstream tasks of augmenting training datasets for classification and segmentation, and for creating visually interesting time lapse videos.

Data Augmentation for Object Classification

■ 2.1 Introduction

Humans are adept at learning new visual concepts. Even young children can identify novel objects after seeing just a few examples [93, 177]. Contrast this with the most powerful modern machine learning systems, which are often trained on millions of examples. What makes humans excel at this task?

Studies in human object perception indicate that people tend to encode object information from consistent viewpoints, often called canonical views [23, 126, 174]. One model of human visual perception suggests that humans recognize shapes from different angles by imagining transformations being applied to a canonical, upright representation – essentially performing “mental rotations” [32, 110, 157].

Data augmentation can be seen as an analog to this process. When designing recognition and prediction systems for natural images, researchers often use simple, hand-engineered transformations such as rotation, flipping and scaling to create more training examples. In a sense, these new examples mimic what an existing training example might look like from different viewing positions. However, these hand-engineered transformations have limited ability to simulate more complex realistic effects.

We propose a data augmentation method that is more inspired by the human perceptual process. We learn a distribution of realistic label-preserving transformations from a small number of similarly-shaped examples. We then synthesize new training images by applying transformations to canonical examples in the training set. We describe transformations as a sequential application of a spatial warp field and an pixel-wise color change. This modular approach enables us to model a complex space of realistic effects including non-linear deformations, 3D rotations, and varied lighting effects.

We target a common but challenging application: classifying objects using limited training examples. We demonstrate our method on improving object classification performance compared to standard hand-engineered augmentation in an extreme case of limited data – during training, our classifiers see multiple examples of each training class, but only one exemplar image of each test class. We show that the concept of transforming exemplars is an effective tool for augmenting training sets. We use

two datasets: the MNIST dataset of handwritten digits [95], and a new dataset of collectible cards called Augmented Magic Images (AMI). The AMI dataset contains computer-generated images of the collectible cards, as well as realistic labeled examples such as photos of cards under specular lighting. We demonstrate that our method learns to synthesize non-linear warps representing variations in writing style in MNIST digits. Our method also learns 3D transformations and complex lighting effects from the AMI dataset. For both datasets, we show that augmenting training using our synthesized examples can significantly improve classification performance compared to baseline approaches.

■ 2.2 Related work

■ 2.2.1 Data augmentation

In image-based supervised learning tasks, it is common to perform data augmentation using simple parameterized transforms such as rotation and scaling. These transforms can reduce overfitting and improve test performance [67, 92]. However, the performance gains can vary greatly with the selection of transformation functions and parameter settings [47].

Recent works have proposed learning data augmentation transformations from data. Hauberg *et al.* [60] focus on data augmentation for classifying MNIST digits. They learn digit-specific spatial transformations, and sample training images and transformations to create new examples aimed at improving MNIST classification performance. In contrast, we learn class-independent transformations, and we do not require multiple examples for each class. Other recent works focus on learning combinations of simple transformation functions (*e.g.*, rotation and contrast enhancement) to perform data augmentation for natural images [37, 135]. Cubuk *et al.* [37] use a search algorithm to find augmentation policies that maximize classification accuracy. Ratner *et al.* [135] learn to create combinations of basic transformation functions by training a generative adversarial network on user input. The simple transformations explored by these works are insufficient for capturing many of the variations in realistic photographs of objects.

Other data augmentation approaches use domain-specific knowledge and implementations. For example, several works on pose-invariant face recognition make use of 3D models or domain-specific loss terms, which can be time-consuming to engineer [36, 68, 108]. Our system leverages more general domain knowledge in its decomposition of transformations into spatial and appearance components. It learns to use these components to implicitly represent relevant information (*e.g.*, handwriting stroke width, 3D structure).

■ 2.2.2 Domain adaptation

In image-to-image domain adaptation, the goal is to learn transformations from one image domain to another. This approach can be useful for data augmentation when translating from a source domain in which it is easy to obtain labels (*e.g.*, computer-generated images) to a target domain where it is hard (*e.g.*, realistic photos) [54, 147].

For supervised learning tasks, it is imperative that the applied transformation is label-preserving. Pairwise similarity metrics have been used with generative adversarial networks (GANs) to help preserve labels during unsupervised domain adaptation [183]. In contrast to these unsupervised domain adaptation approaches, our approach learns *transformations* that are label-preserving from a small number of examples. Class-conditional or categorical GANs have been used to synthesize new examples of specific classes, but must typically be trained on large datasets with many examples per class [114, 152]. In contrast, we tackle tasks that include a small set of labeled images, and does not require pairs for every class.

Several recent works focus on transforming computer-generated images to realistic images [97, 146]. These approaches directly synthesize the output images and rely on additional loss terms to encourage label preservation. We build upon this concept of synthesizing realistic images from canonical representations, using a learned distribution of label-preserving transformations.

■ 2.2.3 Few-shot learning

Several approaches have been proposed to improve few-shot and one-shot classification performance on datasets such as MNIST, Omniglot and ImageNet [118, 164, 173]. Our work does not assume a typical few-shot learning setup – rather than learning from a few examples of each class, we learn a distribution of transformations from a subset of classes, and use these transformations to improve one-shot classification of held-out classes. Furthermore, we focus on creating more data, which is an orthogonal task to designing systems that can learn from limited data. Our data augmentation approach could likely be combined with few-shot learning methods for further improvements in classification performance.

■ 2.3 Method

We tackle the task of assigning class labels to images of objects. Let $\{x^{(i)}, l^{(i)}\}$ be a collection of labeled images, where each image $x^{(i)}$ is a sample from the distribution $p(x|l; l = l^{(i)})$. Let $\tau^{(k)}$ be a transformation sampled from a distribution of label-preserving transforms $p(\tau)$. Data augmentation aims to apply transformations $\tau^{(k)}$ to examples in the training set, yielding transformed images $\tau^{(k)}(x^{(i)})$ that also have a high probability under the distribution $p(x|l; l = l^{(i)})$. We introduce a generative model for $p(\tau)$. We learn this model from pairs of same-class examples in the dataset, and then use the model to generate new labeled examples.

■ 2.3.1 Conditional generative model

We describe transformations of objects in images as a combination of label-preserving spatial and appearance changes. More precisely, we model a transformation τ as a composition of two functions: a spatial transformation τ_s and an appearance transformation τ_a : $\tau(\cdot) = \tau_a(\tau_s(\cdot))$. Intuitively, the spatial transform describes differences in

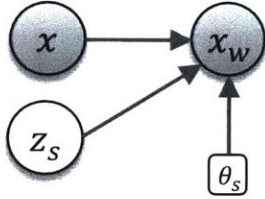


Figure 2.1: In a dataset with no color variations, we model spatial differences between examples as a transformation function τ_s that is generated from a random latent variable z_s . We use circles to represent random variables and boxes to represent parameters. Shaded circles represent observed variables.

shape (caused, *e.g.*, by varied viewing angles), while the appearance transform describes differences in color (caused, *e.g.*, by different lighting). We designed this decomposition to be expressive enough to describe many of the transformations one might observe from realistic objects. Furthermore, using a modular representation leads to explainable transformations.

Spatial model

We first describe the spatial transformation τ_s , which is applied to an input image x to produce a warped image $\tau_s(x)$. Suppose for now that we have a dataset without color variations. For example, in MNIST, examples of the same digit class only differ from one another by spatial changes. We let x represent a canonical example of a digit, and let x_w represent non-canonical examples of the same digit class. We let τ_s be a dense flow field generated from the latent variable z_s , that depends on the input image x . We design this dependence on x to capture a variety of complex transformations; for instance, transformations that vary a digit’s thickness might depend on properties of the original image. We assume z_s is generated from the multivariate standard normal distribution. We model x_w as noisy observations of the warped image $\tau_s(x)$:

$$p_{\theta_s}(x_w|z_s; x) = \mathcal{N}(x_w; \tau_s(x), \sigma_s^2 \mathbb{I}) \quad (2.1)$$

where σ_s represents fixed noise in the image space. We use $\tau_s(x) = x \circ f$ to denote the application of a dense flow field f , and compute $f = g_{\theta_s}(z_s, x)$ where $g(\cdot)$ is a function parameterized by θ_s . This model is summarized in Figure 2.1.

Learning

We want to find the parameters θ_s that maximize the likelihood of each same-class pair (x, x_w) :

$$\begin{aligned} \arg \max_{\theta_s} p_{\theta_s}(x_w, x) &= \arg \max_{\theta_s} p_{\theta_s}(x_w; x) \\ &= \arg \max_{\theta_s} \int_{z_s} p_{\theta_s}(x_w|z_s; x) p(z_s) dz_s. \end{aligned} \quad (2.2)$$

This integral is intractable, and the posterior $p(z_s|x_w; x)$ is also intractable, preventing the use of the EM algorithm. We instead use variational inference and introduce a

distribution $q_{\phi_s}(z_s|x_w; x)$ that approximates the posterior $p(z_s|x_w; x)$ [86, 179, 180]. We derive:

$$\begin{aligned}
& \arg \max_{\theta_s} \int_{z_s} p_{\theta_s}(x_w|z_s; x) p(z_s) dz_s \\
&= \arg \max_{\theta_s, \phi_s} \log \int_{z_s} p_{\theta_s}(x_w|z_s; x) p(z_s) \frac{q_{\phi_s}(z_s|x_w; x)}{q_{\phi_s}(z_s|x_w; x)} dz_s \\
&= \arg \max_{\theta_s, \phi_s} \log E_{z_s \sim q_{\phi_s}(z_s|x_w; x)} \left[p_{\theta_s}(x_w|z_s; x) \frac{p(z_s)}{q_{\phi_s}(z_s|x_w; x)} \right] \\
&= \arg \max_{\theta_s, \phi_s} E_{z_s \sim q_{\phi_s}(z_s|x_w; x)} \left[\log p_{\theta_s}(x_w|z_s; x) + \log p(z_s) - \log q_{\phi_s}(z_s|x_w; x) \right] \\
&= \arg \max_{\theta_s, \phi_s} E_{z_s \sim q_{\phi_s}(z_s|x_w; x)} \left[\log p_{\theta_s}(x_w|z_s; x) \right] - KL[q_{\phi_s}(z_s|x_w; x) || p(z_s)], \quad (2.3)
\end{aligned}$$

where $KL[\cdot || \cdot]$ denotes the Kullback-Liebler divergence. This derived expression is the negative evidence lower bound [86, 179, 180]. We choose the approximate posterior distribution to be a multivariate normal with a diagonal covariance matrix:

$$q_{\phi_s}(z_s|x, x_w) = \mathcal{N}(z_s; \mu_{\phi_s}(x, x_w), \sigma_{\phi_s}(x, x_w)\mathbb{I}),$$

where $\mu_{\phi_s}(x, x_w), \sigma_{\phi_s}(x, x_w)$ are functions parametrized by ϕ_s .

We maximize the variational lower bound in Equation (2.3) using stochastic gradient descent. For each image pair $(x^{(i)}, x_w^{(j)})$, we obtain $z_s^{(k)}, \tau_s^{(k)}$ as follows:

$$\begin{aligned}
z_s^{(k)} &\sim q_{\phi_s}(z_s|x_w; x) \\
\tau_s^{(k)} &= g_{\theta_s}(z_s^{(k)}, x)
\end{aligned} \quad (2.4)$$

and combine Equation (2.3) with Equation (2.1) to obtain the loss function:

$$L(\theta_s, \phi_s; x^{(i)}, x_w^{(j)}, z_s^{(k)}) = KL[q_{\phi_s}(z_s|x_w^{(j)}; x^{(i)}) || p(z_s)] - \lambda_s || \tau_s^{(k)}(x^{(i)}) - x_w^{(j)} ||^2, \quad (2.5)$$

where $\lambda_s = \frac{1}{2\sigma_s^2}$.

Spatial and appearance model

So far, we described a model for spatial differences in a dataset. We next model a dataset that contains images of objects under different spatial orientations and different lighting conditions. We describe the appearance transform model that we use to capture these lighting differences.

Similarly to the spatial model described above, we let x represent a canonical example of an object, and let y represent other examples of the same object class. We let τ_a represent an appearance transformation, which is applied to an input image x to produce a transformed image $\tau_a(x)$. This transformation is generated from a latent

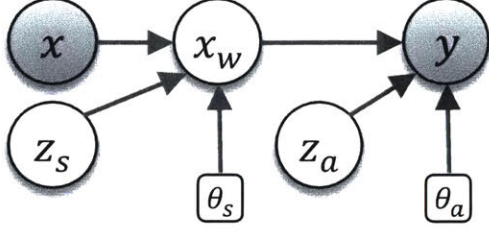


Figure 2.2: A graphical representation of our joint model for data augmentation. Circles indicate random variables while rectangles represent parameters. Shaded circles depict observed quantities.

variable z_a and is dependent on the input image x . For instance, the shadows in an object depend on the shape of the object. Similarly to Section 2.3.1, we assume z_a is generated from the multivariate standard normal distribution.

In most realistic images, spatial and appearance transformations occur together. Compared to a canonical representation of an object, a realistic photo might have a different viewing angle as well as different lighting. One way to learn separate models of spatial and appearance transformations is to use some method of correcting for spatial transformations while we learn appearance transformations, and vice versa. We explore this approach in Chapter 3, and show that it is effective for learning transformations in cases with limited training data. In this work, we design a joint model for spatial and appearance transformations, outlined in Figure 2.2.

The joint model is an extension of the spatial model we presented earlier. We let x_w represent a noisy observation of a spatially warped image as in Equation (2.1), and let y represent a noisy observation of a transformed image:

$$p_{\theta_s, \theta_a}(y|z_a, z_s; x) = \mathcal{N}(y; \tau_a(\tau_s(x)), \sigma_a^2 \mathbb{I}), \quad (2.6)$$

σ_a represents fixed noise in the image space, and τ_s is as defined earlier. We similarly define $\tau_a(\tau_s(x)) = \tau_s(x) + c$, where $c = h_{\theta_a}(z_a, \tau_s(x))$ is a per-pixel color change, computed using the function $h(\cdot)$ that is parameterized by θ_a . We extend Equation (2.2), and learn the parameters θ_s, θ_a that maximize the likelihood:

$$\begin{aligned} \arg \max_{\theta_s, \theta_a} p_{\theta_s, \theta_a}(y, x) &= \arg \max_{\theta_s, \theta_a} p_{\theta_s, \theta_a}(y; x) \\ &= \arg \max_{\theta_s, \theta_a} \int_{z_s, z_a, x_w} p_{\theta_s}(x_w|z_s; x) p(z_s) p_{\theta_a}(y|z_a, x_w) p(z_a) dz_s dz_a dx_w \\ &= \arg \max_{\theta_s, \theta_a} \int_{z_s, z_a} E_{x_w \sim p_{\theta_s}(x_w|z_s; x)} [p_{\theta_a}(y|z_a, x_w) p(z_s) p(z_a)] dz_s dz_a \end{aligned} \quad (2.7)$$

For simplicity, we approximate the expectation over $p_{\theta_s}(x_w|z_s; x)$ using a point estimate of the mean. We assume that $p_{\theta_s}(x_w|z_s; x)$ is as defined in Equation (2.1). This gives:

$$\arg \max_{\theta_s, \theta_a} p_{\theta_s, \theta_a}(y; x) \approx \arg \max_{\theta_s, \theta_a} \int_{z_s, z_a} p_{\theta_s, \theta_a}(y|z_a, \tau_s(x)) p(z_s) p(z_a) dz_s dz_a \quad (2.8)$$

As before, this integral is intractable, and the posterior $p(z_s, z_a|y; x)$ is also intractable. We use variational inference and introduce the approximate posterior distribution $q_{\phi_s, \phi_a}(z_s, z_a|y; x)$ [86, 179, 180]. We assume that this distribution decomposes as $q_{\phi_s, \phi_a}(z_s, z_a|y; x) = q_{\phi_s}(z_s|y; x)q_{\phi_a}(z_a|y, \tau_s(x))$. Similarly to Equation (2.3), we derive:

$$\begin{aligned}
& \arg \max_{\theta_s, \theta_a} \int_{z_s, z_a} p_{\theta_s, \theta_a}(y|z_a, \tau_s(x)) p(z_s) p(z_a) dz_s dz_a \\
&= \arg \max_{\theta_s, \theta_a} \int_{z_s, z_a} p_{\theta_s, \theta_a}(y|z_a, \tau_s(x)) p(z_s) p(z_a) \frac{q_{\phi_s}(z_s|y; x) q_{\phi_a}(z_a|y, \tau_s(x))}{q_{\phi_s}(z_s|y; x) q_{\phi_a}(z_a|y, \tau_s(x))} dz_s dz_a \\
&= \arg \max_{\theta_s, \theta_a, \phi_s, \phi_a} \log E_{z_s \sim q_{\phi_s}(z_s|y; x)} \left[E_{z_a \sim q_{\phi_a}(z_a|y, \tau_s(x))} \left[p_{\theta_s, \theta_a}(y|z_a, \tau_s(x)) \frac{p(z_s)}{q_{\phi_s}(z_s|y; x)} \frac{p(z_a)}{q_{\phi_a}(z_a|y, \tau_s(x))} \right] \right] \\
&= \arg \max_{\theta_s, \theta_a, \phi_s, \phi_a} E_{z_s \sim q_{\phi_s}(z_s|y; x)} E_{z_a \sim q_{\phi_a}(z_a|y, \tau_s(x))} \left[\log p_{\theta_s, \theta_a}(y|z_a, \tau_s(x)) \right] \\
&\quad - KL[q_{\phi_s}(z_s|y; x) || p(z_s)] - KL[q_{\phi_a}(z_a|y, \tau_s(x)) || p(z_a)]. \tag{2.9}
\end{aligned}$$

We maximize this variational lower bound using stochastic gradient descent. For each image pair $(x^{(i)}, y^{(j)})$, we obtain samples:

$$z_s^{(k)} \sim q_{\phi_s}(z_s|y; x), \quad q_{\phi_s}(z_s|y; x) = \mathcal{N}(z_s; \mu_{\phi_s}(x, y), \sigma_{\phi_s}(x, y)\mathbb{I}), \tag{2.10}$$

$$z_a^{(l)} \sim q_{\phi_a}(z_a|y, \tau_s(x)), \quad q_{\phi_a}(z_a|\tau_s(x), y) = \mathcal{N}(z_a; \mu_{\phi_a}(\tau_s(x), y), \sigma_{\phi_a}(\tau_s(x), y)\mathbb{I}), \tag{2.11}$$

where $\mu_{\phi_s}(x, y), \sigma_{\phi_s}(x, y)$ are functions parameterized by ϕ_s , and $\mu_{\phi_a}(\tau_s(x), y), \sigma_{\phi_a}(\tau_s(x), y)$ are functions parameterized by ϕ_a . We apply spatial and appearance transformations as follows:

$$\begin{aligned}
\tau_s^{(k)}(x) &= x \circ f^{(k)}, & f^{(k)} &= g_{\theta_s}(z_s^{(k)}, x), \\
\tau_a^{(l)}(x) &= x + c^{(l)}, & c^{(l)} &= h_{\theta_a}(z_a^{(l)}, \tau_s(x)).
\end{aligned}$$

Rearranging Equation (2.9) and combining it with Equation (2.6), we obtain the joint loss terms:

$$\begin{aligned}
L(\theta_s, \theta_a, \phi_s, \phi_a; x^{(i)}, y^{(j)}, z_s^{(k)}, z_a^{(l)}) &= KL[q_{\phi_s}(z_s|y^{(j)}, x^{(i)}) || p(z_s)] \\
&\quad + KL[q_{\phi_a}(z_a|y^{(j)}, \tau_s^{(k)}(x^{(i)})) || p(z_a)] \\
&\quad - \lambda_a || \tau_a^{(l)}(\tau_s^{(k)}(x^{(i)})) - y^{(j)} ||^2, \tag{2.12}
\end{aligned}$$

where $\lambda_a = \frac{1}{2\sigma_a^2}$ is a reconstruction loss weight. Intuitively, the KL terms encourage the approximate posterior distributions to be close to the priors, while the L2 term encourages accurate reconstruction of the transformed example.

One challenge of optimizing the presented joint model is balancing the relative contributions of the spatial and appearance transformations. In our early experiments, we found that it was easy for the joint model to represent all of the differences between x and y through pixel-wise changes in τ_a , while applying no change through τ_s . Such a

solution does not represent a meaningful decomposition of the transformation from x and y , and is likely to produce unrealistic outputs when we sample new transformations. As a way to regularize τ_a and encourage a meaningful spatial transformation τ_s , we introduce a normalized local cross-correlation loss $L_{cc}(\tau_s(x), y)$. This loss encourages $\tau_s(x)$ to be spatially aligned to y , while being robust to lighting and color changes.

Let $y(p)$ represent the pixel intensity in image y at location p , and let $\tilde{y}(p)$ represent the pixel intensity in image y at location p with the local mean subtracted out. We compute the local mean over a region of n^2 pixels centered at p , and we choose $n = 5$ in our experiments. Then, using t in place of $\tau_s(x)$ in the interest of readability:

$$L_{cc}(t, y) = \sum_{p \in \Omega} \frac{\left(\sum_{p_i} (t(p) - \tilde{t}(p)) (y(p) - \tilde{y}(p)) \right)^2}{\left(\sum_{p_i} (t(p) - \tilde{t}(p))^2 \right) \left(\sum_{p_i} (y(p) - \tilde{y}(p))^2 \right)}, \quad (2.13)$$

where Ω is the set of all pixel locations in the image.

In addition, to encourage spatial consistency, we encourage spatial smoothness of the spatial and appearance transformation volumes that are produced by our model, f and c . We use the losses $L_{sm,s}(f) = \|\nabla^2 f\|^2$ and $L_{sm,a}(c) = \|\nabla c\|^2$. Our final loss is therefore:

$$\begin{aligned} L(\theta_s, \theta_a, \phi_s, \phi_a; x^{(i)}, y^{(j)}, z_s^{(k)}, z_a^{(l)}) = & KL[q_{\phi_s}(z_s | y^{(j)}, x^{(i)}) | p(z_s)] \\ & + KL[q_{\phi_a}(z_a | y^{(j)}, \tau_s^{(k)}(x^{(i)})) | p(z_a)] \\ & - \lambda_a \|\tau_a^{(l)}(\tau_s^{(k)}(x^{(i)})) - y^{(j)}\|^2 \\ & + \lambda_s L_{cc}(\tau_s^{(k)}(x^{(i)}), y^{(j)}) \\ & + \lambda_{sm,s} \mathcal{L}_{sm}(f^{(k)}) + \lambda_{sm,a} \mathcal{L}_{sm}(c^{(l)}), \end{aligned} \quad (2.14)$$

where $\lambda_s, \lambda_a, \lambda_{sm,s}, \lambda_{sm,a}$ are loss hyper-parameters.

■ 2.3.2 Network architecture

We implement the joint spatial and appearance transform model described above using two connected conditional variational autoencoders (CVAEs) [86, 166, 180]. We model $q_{\phi_s}(z_s | y; x)$ and $q_{\phi_a}(z_a | \tau_s(x), y)$ as encoders with network parameters ϕ_s, ϕ_a , and model $p_{\theta_s}(x_w | x, z_s), p_{\theta_a}(y | x_w, z_a)$ as decoders with network parameters θ_s, θ_a . The two CVAEs can be seen as capturing the spatial and appearance transformations in a modular way. Given two input images (either x, y for the spatial transformation or $\tau_s(x), y$ for the appearance transformation), each encoder outputs the mean μ_ϕ and variance Σ_ϕ of the distribution of the latent variable z , and then samples from the distribution [87]. We use a separate encoder to capture the dependence of the transformation on the input image x . The encoded image is concatenated with the sampled latent vector, and then decoded into the final transformation volume. We illustrate the architecture of a single transform CVAE in Figure 2.3. The spatial transform CVAE can be used alone on datasets containing only spatial variations. In the full model, the spatial and appearance

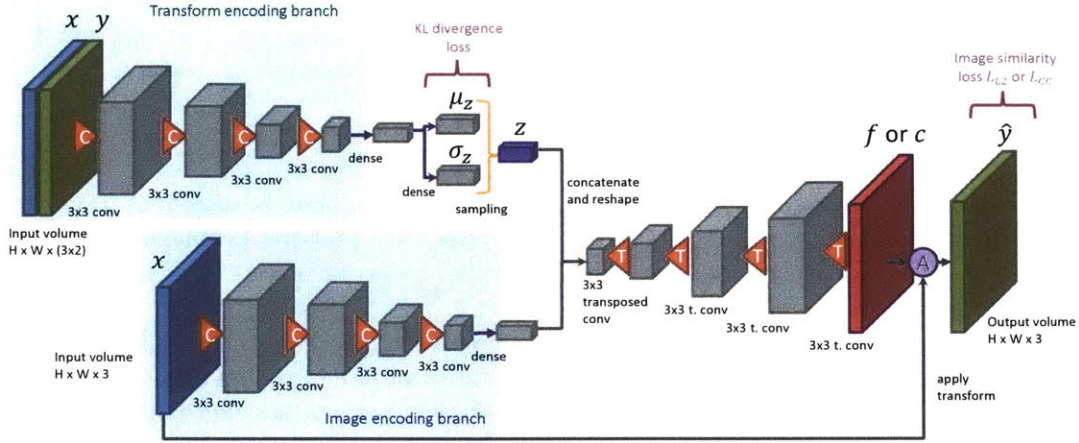


Figure 2.3: Architecture of a transform conditional variational autoencoder. Each encoding branch is implemented using a series of 3×3 strided convolutions, each followed by a leaky ReLU activation. The decoder is similar.

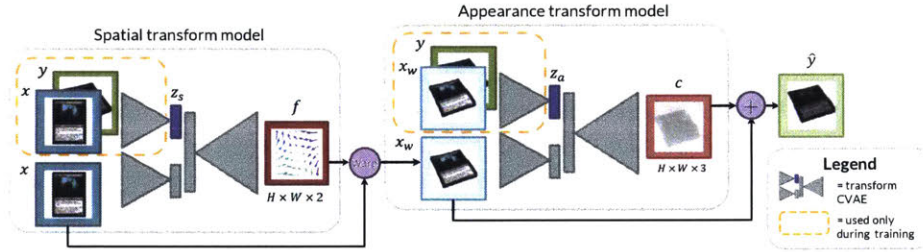


Figure 2.4: In the sequential transform CVAE network, the spatial transform model applies the flow field f to the input x using a spatial transformer layer [16, 74]. The appearance model applies c to the warped image $\tau(x)$ through an elementwise addition. At test time, the upper encoding branch of each transform CVAE is discarded and z_s , z_a are each sampled from the multivariate standard normal.

transformations are each captured using a CVAE, and are joined sequentially as shown in Figure 2.4.

For an input RGB image of size $H \times W \times 3$, the spatial transform model captures τ_s using a displacement field f of size $H \times W \times 2$. For a warped image $\tau_s(x)$, the value at pixel location i, j is $\tau_s(x)(i, j) = x(i + f_u(i, j), j + f_v(i, j))$, where f_u, f_v represent the horizontal and vertical displacement components respectively. The appearance transform CVAE receives as input the resulting $\tau_s(x)$ along with the target image y . The appearance transformation τ_a is implemented as a volume c of size $H \times W \times 3$. The application of τ_a is implemented as a per-pixel additive transform for each color channel: $\tau_a(\tau_s(x)(i, j)) = \tau_s(x)(i, j) + c(i, j)$.

We train these neural networks on same-class pairs from a dataset to capture distri-

butions of spatial and appearance transformations. Then, we sample transformations from our learned distributions and apply them to existing examples to create new labeled training examples. We describe our experiments in the following section.

■ 2.4 Experiments

We use our data augmentation method to improve few-shot classification performance. We present two datasets that pose challenging few-shot learning problems. In each dataset, only a single canonical example is available for some classes, while several examples are available for other classes.

We first demonstrate our spatial transform model on a subset of the MNIST dataset, a simple dataset of handwritten digits. We present examples of the images synthesized by our method, and show that our method is effective for synthesizing new examples even in the face of limited training data. We also introduce a new dataset of colorful collectible cards, the Augmented Magic Images (AMI).

■ 2.4.1 Datasets

MNIST To emulate realistic scenarios with limited labeled examples, we first reduce the standard MNIST training set to include 200 random examples images of each digit. We hand-select one image of each digit as canonical. We construct 5 dataset splits: in each split, we designate 2 classes as one-shot classes, where only the canonical is available at train time. We make 200 examples available for each of the other 8 classes. We design this experiment to use a similar number of training examples to our AMI dataset, which we describe below.

We use our spatial transform model to synthesize new examples of each digit, and then train a classifier on the augmented training set. We tune hyperparameters on a validation set of 100 held-out examples of each class.

Augmented Magic Images (AMI) We introduce a new dataset consisting of images of collectible cards from the card game Magic: The Gathering. Each unique card appearance in the game is represented by a unique card ID. We downloaded a dataset of canonical card images from the Gatherer database [122], which contains computer-generated representations of what is printed on each card. We show several examples in Figure 2.5.

Our dataset consists of 315 card IDs, with one canonical example (downloaded from the Gatherer database) per ID. In addition, we took 2062 photographs of the 315 card IDs in realistic foreground conditions, with approximately 7 photos per card. These photographs show real Magic cards from various angles, and under different lighting conditions caused by flash, varied exposure, and varied lighting colors. We used a semi-automated method to segment the card in each photo from the background. We manually segmented the cards in 350 photos, and trained a foreground/background segmentation network using the VGG19 architecture [148] to segment the rest. Some segmented photos are shown in Figure 2.5. While synthesizing realistic backgrounds is



Figure 2.5: Our dataset contains images of computer-rendered Magic cards (column 1), as well as actual cards photographed in realistic conditions (columns 2-4) including 2D and 3D rotations and specular lighting.

Table 2.1: Distribution of card appearances in the AMI dataset, representing 315 distinct card IDs. A sample train-test split is shown. The photographs feature cards with 2D rotations, 3D rotations, and varied lighting.

Appearance	Total	Transform CVAE training set	Classifier training set	Classifier test set
Computer-generated	315	210	315	0
Real photos	2062	1384	1384	678

an important problem in data augmentation, we focus on the complex transforms of the objects themselves in this work.

To evaluate how our method improves few-shot classification, we design an experiment where for some one-shot classes, only the canonical card is available at training time. For the remaining classes, the canonical example and the segmented photographs are included in the training set. We evaluate our dataset with a 3-fold split, where we designate approximately a third of the classes as one-shot classes in each split. The distribution of appearances in the training and test sets are shown in Table 2.1.

■ 2.4.2 Synthesis results

In this section, we show that our method can be used to synthesize realistic and varied new examples. We train our transform models on same-class image pairs from each training set, where each pair consists of a canonical example and a randomly chosen example of the same class. We then sample transformations from our learned models and use them to synthesize new examples.

We first demonstrate our method on the MNIST dataset. Since MNIST exhibits no color variations, we train only a spatial transform CVAE using L2 as the reconstruction loss, with a reconstruction weight of $\frac{1}{2\sigma_s^2}$ with $\sigma_s = 0.01$. For AMI, we use a full transform CVAE to capture spatial and appearance variations. For the spatial transform hyperparameters, we use reconstruction and smoothness regularization weights $\lambda_s = 2000$, $\lambda_{sm,s} = \lambda_s$, and a cross-correlation neighborhood size of 5×5 . We pre-train the spatial transform model to facilitate convergence, using normalized local

cross-correlation as the reconstruction loss. During joint training, we use the same spatial transform hyperparameters. For the appearance transform hyperparameters, we use a reconstruction weight of $\lambda_a = \frac{1}{2\sigma_a^2}$, $\sigma_a = 0.01$ and smoothness regularization weight $\lambda_{sm,a} = 5\lambda_a$.



Figure 2.6: Our method learns to transform existing examples (top) into new examples (bottom) differing in spatial configuration (*e.g.*, shape, rotation) and appearance (*e.g.* specular effects).

Figure 2.6 and Figure 2.7 show examples of the class-independent transforms learned by our model. We show two different techniques for obtaining the spatial and appearance transform encodings z_s and z_a . Figure 2.6 shows results from sampling the encodings from the standard normal distribution. Figure 2.7 presents results when obtaining the encodings from a random training image pair. The transform CVAEs learn transforms from training pairs that are generalizable to new test examples. The synthesized transforms are realistic and label-preserving, making them appropriate for data augmentation.

Finally, Figure 2.8 shows that in the AMI dataset, sampling from the full transform CVAE produces new views of input images rather than simply memorizing existing transformations in the training set.

■ 2.4.3 Classification performance

We now evaluate the utility of our synthesized examples for improving classification performance. For MNIST, we construct a classifier using a smaller version of the VGG19 architecture [148]. For AMI, we use a VGG19 network pre-trained on ImageNet [43]. Each classifier is trained on only the canonical example of each one-shot class, and all examples of the remaining classes.

We evaluate several methods of training the classifier: no augmentation (*no aug*), hand-tuned augmentation (*hand-aug*), our automated augmentation (*flow-synth* and *full-synth*), and a mix of hand-tuned and our augmentation (*flow-synth + hand-aug* and *full-synth + hand-aug*).

The *hand-aug* baseline for both MNIST and AMI is implemented using hand-engineered transformation functions: random rotations, translations, shearing, and spatial scaling – a common process for image-based learning systems. For AMI, we include additional random blurring and global color intensity scaling. The parameters of the transformations were tuned to match the range of appearances in each dataset as

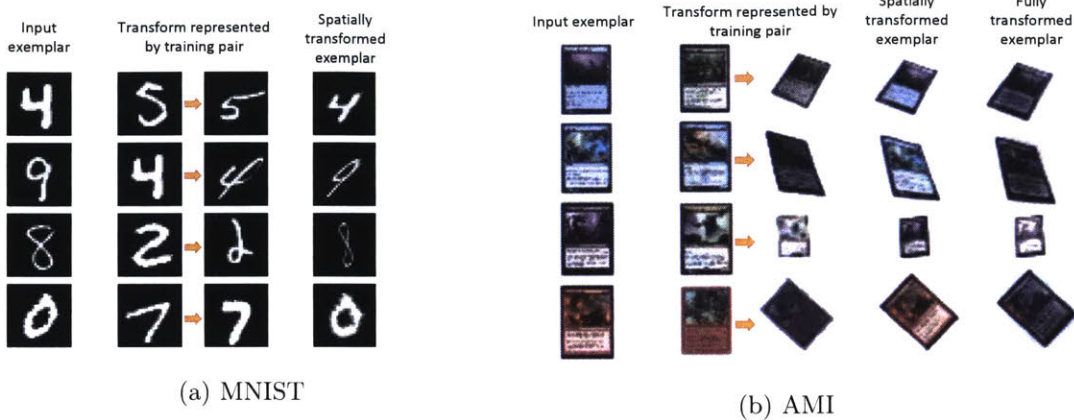


Figure 2.7: From each same-label training pair, our network learns a transform embedding that is generalizable even to held-out classes. On MNIST, these generalized transforms include slanting to the left or right, and decreasing the stroke width. Some digits appear as both test inputs and training inputs in different dataset folds. From AMI, our flow transform CVAE learns to apply 3D rotations, and the color transform CVAE learns to simulate varied lighting. The last AMI example shows a potential failure case where the synthesized cool-toned lighting appears to change the color content of the card.

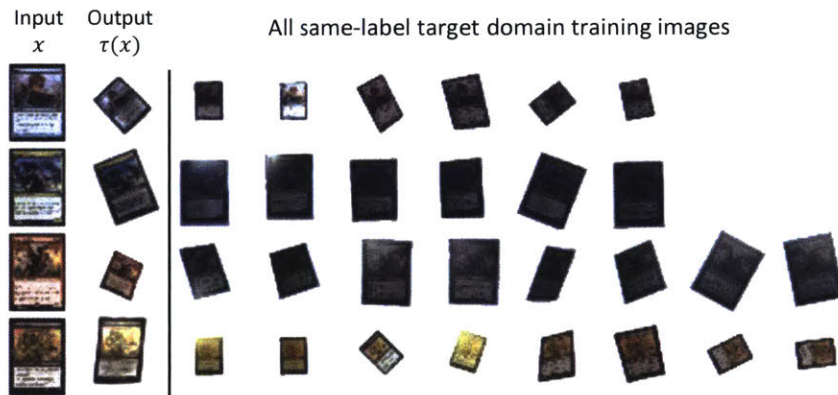


Figure 2.8: Given an input image from the training set and random z_s, z_a , the full transform model can produce transformed cards that are different from all existing images of that label in the AMI dataset.

accurately as possible. For each dataset, we evaluated several parameter settings and selected the best one. This is representative of how data augmentation parameters are tuned in practice. During training, we create a random hand-augmented example for each real example.

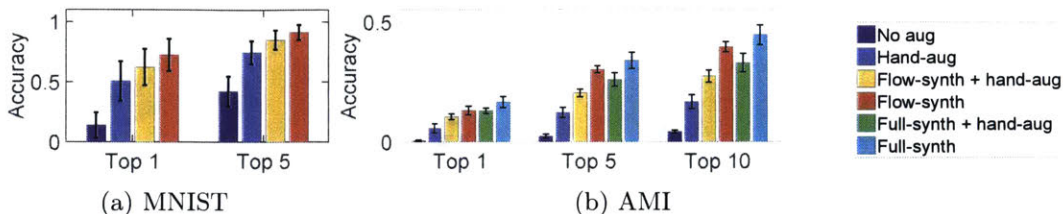


Figure 2.9: Our transform CVAE models learn to produce meaningful examples that help classification accuracy. Flow-synth indicates training with synthetic examples produced only by our flow transform CVAE with no color transforms, while full-synth indicates synthesis by our full transform CVAE.

In the *synth* experiments, each canonical example $x^{(i)}$ with label $l^{(i)}$ is passed as input to our transform model with randomly sampled $z_s^{(k)}, z_a^{(l)}$; the transformed image $\tau_a^{(l)}(\tau_s^{(k)}(x^{(i)}))$ is then assigned the label $l^{(i)}$ and used to train the classifier. We train the classifier on a ratio of one synthesized example to each real example.

Results For MNIST, we train a classifier on each of the 5-fold dataset splits, and report the average top-1 and top-5 classification accuracy on the non-canonical examples of the one-shot digits; none of these examples were seen during training. To reduce noise due to fluctuating model accuracy over the course of training, we compute the average classification accuracy over 10 sets of model weights from the last 50 epochs of training. For the AMI dataset, we train a classifier on each of the 3-fold dataset splits. We evaluate these classifiers in terms of top 1, 5, and 10 accuracy on the photographs of the one-shot card IDs; none of the photographs were seen during training. For both datasets, we train each classifier until convergence.

Figure 2.9 shows that a baseline classifier with no augmentation performs barely better than chance on this difficult task. Hand-tuned augmentation performs better, attaining top-1 accuracies of $50.6\% \pm 16.5$ and $5.74\% \pm 1.93$ respectively on MNIST and AMI. However, the hand-tuned functions cannot capture the full range of variations in our datasets. Training the classifier using only augmented examples synthesized by our model results in the best performance, with top-1 accuracies of $72.5\% \pm 13.4$ and $16.6\% \pm 2.4$ respectively. This improvement over the *synth + hand-aug* model (with accuracies of $62.1\% \pm 15.2$ and $13.0\% \pm 1.12$ respectively) is explained by the fact these classifiers are trained on the same number of augmented examples, so the *synth + hand-aug* is trained on a lower proportion of synthetic examples. These results indicate that our model learns label-preserving transforms that are relevant to the dataset, and that synthesizing new training examples using these transformations produces the highest quality of data augmentation.

Table 2.2: We compare the use of our approach after training on 500 examples of each digit for performing data augmentation. We report the error rate on the full MNIST test set.

Method	ConvNet Test Error in % (std)
<i>InfiMNIST500</i> [60, 100]	1.04 (0.07)
<i>AlignMNIST500</i> [60]	0.84 (0.05)
Ours (flow transform CVAE)	1.0 (0.06)

■ 2.5 Comparison with previous work

Hauberg *et al.* present a data augmentation approach on the MNIST dataset that is similar to ours [60]. They learn a distribution of transformations on a *per-class basis* from 500 examples of each MNIST digit, and demonstrate the use of these transformations for creating new examples of each class for data augmentation.

We analogously train a spatial transform CVAE to learn the distribution of transformations between *all* same-class pairs, using 500 examples of each digit. We then perform data augmentation as described in Section 2.4.3 to train a convolutional classifier with the architecture described in [60]. In Table 2.2, we compare our augmentation approach to that of Hauberg *et al.* (*AlignMNIST500*) as well as the baseline Infinite MNIST algorithm (*InfiMNIST500*) [100]. Our approach attains a comparable level of test error, despite not being designed to learn class-specific transformation distributions. As shown in the previous section, our method has the additional advantage of not requiring multiple examples of every class.

■ 2.6 Discussion

We presented a novel learning-based approach to image synthesis for data augmentation. Our model learns a distribution of label-preserving transformations from labeled pairs of images, and is capable of producing new images of an object that are unlike any other examples in the training set. For instance, our model learns non-linear deformations to vary writing style and stroke thickness for handwritten digits. It also learns to apply 3D rotations and specular lighting effects to collectible cards; these effects are likely to be useful for other planar objects such as books, CDs, *etc.* Finally, we demonstrated that our approach is useful for training object classifiers with limited data, particularly in few-shot scenarios where only one example is available for some classes, and only a few examples were available for the other classes. There are some limitations of this work that could be addressed to expand its utility in other areas:

Limited datasets. Our method makes assumptions about the input images that do not apply to many classification datasets. For instance, the 2D flow field is effective at warping images of objects without backgrounds or occlusions. This limits us to simplified datasets of natural images that do not contain realistic backgrounds. However,

this approach could be applied to 3D volumes, which we show in the next chapter. Another interesting extension to this work might incorporate a mechanic for applying transformations only to foreground objects in images containing backgrounds.

Hyperparameter tuning. Using joint CVAEs requires us to tune 4 hyperparameters that control the weights on the reconstruction and smoothing loss of each transformation. Selecting appropriate values for these hyperparameters can require a significant amount of time and computational resources.

Other transformations. In this work, we used smooth flow fields and per-pixel color changes to capture spatial and appearance variations within a dataset. It might be interesting to explore other forms of transformations such as piece-wise flow fields or physically-motivated lighting transformations.

In the following chapter, we discuss a related project in which we address the first two limitations.

Data Augmentation for Medical Image Segmentation

■ 3.1 Introduction

In this chapter, we explore how to use data augmentation to improve medical image segmentation. This chapter builds on our work presented in [190]. Medical image segmentation is the task of identifying anatomically relevant regions or structures in images. This task is crucial to many biomedical imaging applications, such as performing population analyses, diagnosing disease, and planning treatments. When enough labeled data is available, supervised deep learning-based segmentation methods produce state-of-the-art results. However, obtaining manual segmentation labels for medical images requires considerable expertise and time. In most clinical image datasets, there are very few manually labeled images. The problem of limited labeled data is exacerbated by differences in image acquisition procedures across machines and institutions, which can produce wide variations in resolution, image noise, and tissue appearance [96].

To overcome these challenges, many supervised biomedical segmentation methods focus on hand-engineered preprocessing steps and architectures [116, 129]. It is also common to use hand-tuned data augmentation to increase the number of training examples [4, 123, 129, 137, 139]. Data augmentation functions such as random image rotations or random nonlinear deformations are easy to implement, and are effective at improving segmentation accuracy in some settings [123, 129, 137, 139]. However, these functions have limited ability to emulate real variations [48], and can be highly sensitive to the choice of parameters [47].

We address the challenges of limited labeled data by learning to synthesize diverse and realistic labeled examples. Our automated approach to data augmentation leverages unlabeled images. Using learning-based registration methods, we model the set of spatial and appearance transformations between images in the dataset. These models capture the anatomical and imaging diversity in the unlabeled images. We synthesize new examples by sampling transformations and applying them to a single labeled example.

We demonstrate the utility of our method on the task of one-shot segmentation of brain magnetic resonance imaging (MRI) scans. We use our method to synthe-

size new labeled training examples, enabling the training of a supervised segmentation network. This strategy outperforms state-of-the art one-shot biomedical segmentation approaches, including single-atlas segmentation and supervised segmentation with hand-tuned data augmentation.

■ 3.2 Related work

■ 3.2.1 Medical image segmentation

We focus on the segmentation of brain MR images, which is challenging for several reasons. Firstly, human brains exhibit substantial anatomical variations [52, 131, 161]. Secondly, MR image intensity can vary as a result of subject-specific noise, scanner protocol and quality, and other imaging parameters [96]. This means that a tissue class can appear with different intensities across images – even images of the same MRI modality.

Many existing segmentation methods rely on scan pre-processing to mitigate these intensity-related challenges. Pre-processing methods can be costly to run, and developing techniques for realistic datasets is an active area of research [31, 153]. Our augmentation method tackles these intensity-related challenges from another angle: rather than removing intensity variations, it enables a segmentation method to be robust to the natural variations in MRI scans.

A large body of classical segmentation methods use *atlas-based* or *atlas-guided* segmentation, in which a labeled reference volume, or *atlas*, is aligned to a target volume using a deformation model, and the labels are propagated using the same deformation [11, 30, 41, 61]. When multiple atlases are available, they are each aligned to a target volume, and the warped atlas labels are fused [70, 88, 142, 167]. In atlas-based approaches, anatomical variations between subjects are captured by a deformation model, and the challenges of intensity variations are mitigated using pre-processed scans, or intensity-robust metrics such as normalized cross-correlation. However, ambiguities in tissue appearances (*e.g.*, indistinct tissue boundaries, image noise) can still lead to inaccurate registration and segmentations. We address this limitation by training a segmentation model on diverse realistic examples, making the model more robust to such ambiguities. We focus on having a single atlas, and demonstrate that our strategy outperforms atlas-based segmentation. If more than one segmented example is available, our method can leverage them.

Supervised learning approaches to biomedical segmentation have gained popularity in recent years. To mitigate the need for large labeled training datasets, these methods often use hand-engineered pre-processing steps and architectures. Data augmentation has also been shown to be essential in some tasks. Similarly to data augmentation in classification tasks, which we discussed in the last chapter, researchers often use random rotations, flipping and translations [4, 81, 116, 129, 137, 139, 188]. Random smooth warp fields can also help to simulate deformations in tissues [137].

Semi-supervised and unsupervised approaches have also been proposed to combat

the challenges of small training datasets. These methods do not require paired image and segmentation data. Rather, they leverage collections of segmentations to build anatomical priors [40], to train an adversarial network [80], or to train a novel semantic constraint [53]. In practice, collections of images are more readily available than segmentations. Rather than rely on segmentations, our method leverages a set of unlabeled images.

■ 3.2.2 Spatial and appearance transform models

Models of shape and appearance have been used in a variety of image analyses. We discussed some models that were designed for natural images in the previous chapter. Here, we focus on spatial and appearance transform models in medical applications.

In medical image registration, a spatial deformation model is used to establish semantic correspondences between images. This mature field spans optimization-based methods [8, 12, 141, 145], and recent learning-based methods [16, 17, 39, 90, 136, 151, 182]. We build upon VoxelMorph, an unsupervised registration method that we presented in [16, 17], to learn spatial transformations.

Many medical image registration methods focus on intensity-normalized images or intensity-independent objective functions, and do not explicitly account for variations in image intensity. For unnormalized images, models of intensity transformations have been used to remove bias field effects from MRI [94, 171]. Spatial and appearance transform models have been used together to register objects that differ in shape as well as texture. Many works build upon the framework of Morphable Models [79] or Active Appearance Models (AAMs) [34, 35], in which statistical models of shape and texture are constructed. AAMs have been used to localize anatomical landmarks [33, 130] and perform segmentation [115, 128, 163]. We build upon these concepts by using convolutional neural networks to learn models of spatial and intensity transformations. Rather than learning transform models for the end goal of registration or segmentation, we sample from these models to synthesize new training examples. As we show in our experiments, augmenting a segmenter’s training set in this way can produce more robust segmentations than performing segmentation using the transform models directly.

■ 3.2.3 Few-shot segmentation of natural images

Few-shot segmentation is a challenging task in semantic segmentation and video object segmentation. Existing approaches focus mainly on natural images. Methods for few-shot semantic segmentation incorporate information from prototypical examples of the classes to be segmented [45, 144]. Few-shot video segmentation is frequently implemented by aligning objects in each frame to a labeled reference frame [75, 159]. Other approaches leverage large labeled datasets of supplementary information such as object appearances [24], or incorporate additional information such as human input [133]. Medical images present different challenges from natural images; for instance, the visual differences between tissue classes are very subtle compared to the typical differences between objects in natural images.

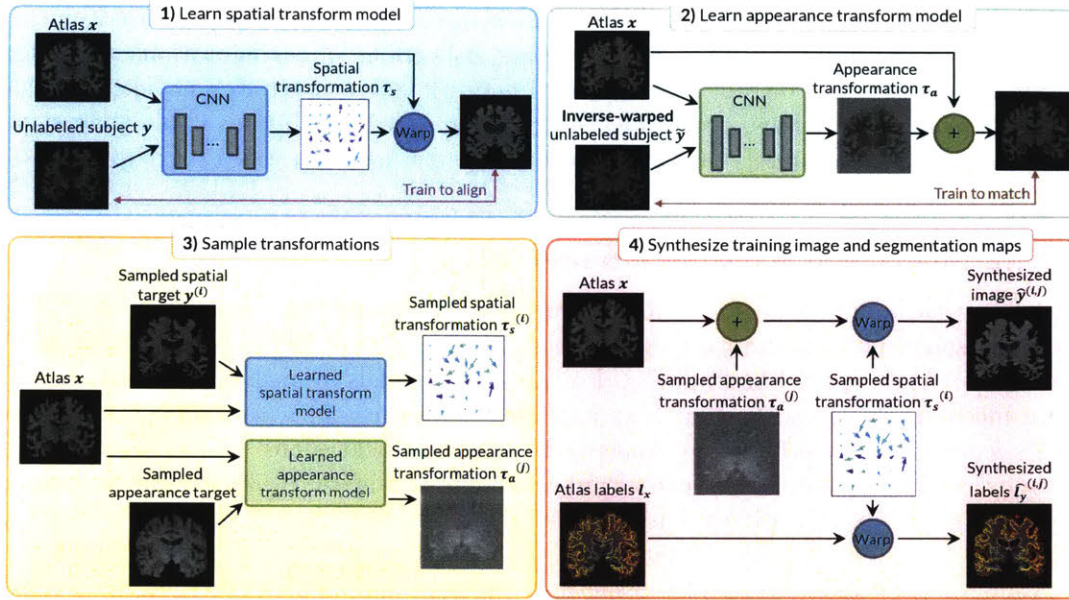


Figure 3.1: An overview of the proposed method. We learn independent spatial and appearance transform models to capture the variations in our image dataset. We then use these models to synthesize a dataset of labeled examples. This synthesized dataset is used to train a supervised segmentation network.

■ 3.2.4 Data augmentation

As discussed in Section 2.2, data augmentation is commonly performed for natural images using simple parameterized transformations such as rotation and scaling. For medical images, some works have similarly used rotations [129], translations and flipping [3]. Other works have used random smooth flow fields to simulate anatomical variations [113, 137, 138]. Like data augmentation transformations for natural images, these random flow fields can improve test performance, but have limited ability to simulate the complex anatomical differences in medical images.

In Section 2.2, we also reviewed recent works that proposed to learn data augmentation transformations [37, 60, 135]. These existing methods focus on natural images, and are ill-suited to capture the subtle differences in MRI scans.

■ 3.3 Method

■ 3.3.1 Transform models for augmentation

We propose to improve one-shot biomedical image segmentation by combining our transformation learning system from Chapter 2 with the registration capabilities of VoxelMorph in a semi-supervised framework.

Let $\{y^{(i)}\}$ be a set of biomedical image volumes, and let the pair (x, l_x) represent a labeled reference volume, or *atlas*, and its corresponding segmentation map. In brain MRI segmentation, each x and y is a grayscale 3D volume. We focus on the challenging case where only one labeled atlas is available, since it is often difficult in practice to obtain many segmented volumes. Our method can be easily extended to leverage additional segmented volumes.

To perform data augmentation, we apply transformations $\tau^{(k)}$ to the labeled atlas x . As in Chapter 2, we decompose transformations into spatial and appearance changes. Here, we learn separate spatial and appearance transform models to capture the distribution of anatomical and appearance differences between the labeled atlas and each unlabeled volume. Using the two learned models, we synthesize labeled volumes $\{\{\hat{y}^{(k)}, \hat{l}_y^{(k)}\}\}$ by applying a spatial transformation and an appearance transformation to the atlas volume, and by warping the atlas label maps using the spatial transformation. Compared to traditional single-atlas segmentation [11, 30, 41, 61], which suffers from uncertainty or errors in the spatial transform model, we use the same spatial transformation to synthesize the volume *and* label map. This ensures that the warped label map matches the newly synthesized volume. These synthetic examples form a labeled dataset that characterizes the anatomical and appearance variations in the unlabeled dataset. Along with the atlas, this new training set enables us to train a supervised segmentation network. This process is outlined in Fig. 3.1.

■ 3.3.2 Spatial and appearance transform models

We describe the differences between scans using a combination of spatial and intensity transformations. Specifically, we define a transformation $\tau(\cdot)$ from one volume to another as a composition of a spatial transformation $\tau_s(\cdot)$ and an intensity or *appearance* transformation $\tau_a(\cdot)$, *i.e.*, $\tau(\cdot) = \tau_s(\tau_a(\cdot))$. In contrast to Chapter 2, we apply the appearance transformation rather than the spatial transformation first in this decomposition. We discuss this change in Section 3.3.3.

We assume a spatial transformation takes the form of a smooth voxel-wise displacement field u . Following the medical registration literature, we define the *deformation* function $\phi = \text{id} + u$, where id is the identity function. We use $x \circ \phi$ to denote the application of the deformation ϕ to x . To model the distribution of spatial transformations in our dataset, we compute the deformation that warps atlas x to each volume $y^{(i)}$ using $\phi^{(i)} = g_{\theta_s}(x, y^{(i)})$, where $g_{\theta_s}(\cdot, \cdot)$ is a parametric function that we describe later. We write the approximate inverse deformation of $y^{(i)}$ to x as $\phi^{-1(i)} = g_{\theta_s}(y^{(i)}, x)$.

We model the appearance transformation $\tau_a(\cdot)$ as per-voxel addition in the spatial frame of the atlas. We compute this per-voxel volume using the function

$$\psi^{(i)} = h_{\theta_a}(x, y^{(i)} \circ \phi^{-1(i)}),$$

where $y^{(i)} \circ \phi^{-1(i)}$ is a volume that has been registered to the atlas space using our learned spatial model. In practice, we found the registration accuracy to be better

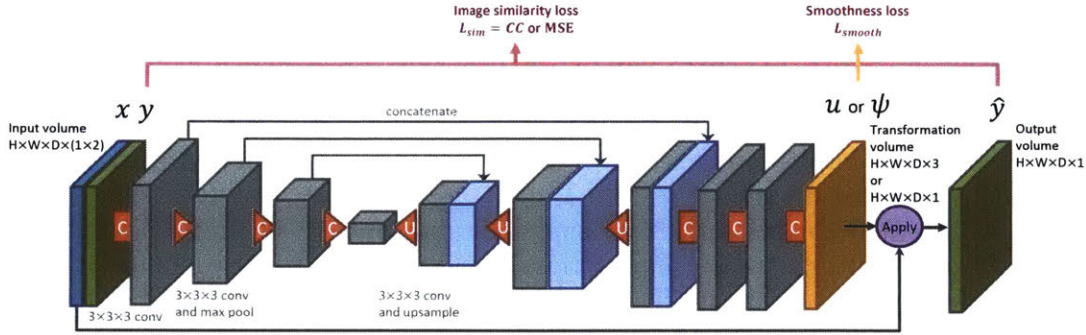


Figure 3.2: We use a convolutional neural network based on the U-Net architecture [137] to learn each transform model. The application of the transformation is a spatial warp for the spatial model, and a voxel-wise addition for the appearance model. Each convolution uses $3 \times 3 \times 3$ kernels, and is followed by a LeakyReLU activation layer. The encoder uses max pooling layers to reduce spatial resolution, while the decoder uses upsampling layers.

when we trained a separate subject-to-atlas spatial transform model. In summary, our spatial and appearance transformations are:

$$\tau_s^{(i)}(x) = x \circ \phi^{(i)}, \quad \phi = g_{\theta_s}(x, y^{(i)}) \quad (3.1)$$

$$\tau_a^{(i)}(x) = x + \psi^{(i)}, \quad \psi^{(i)} = h_{\theta_a}(x, y^{(i)} \circ \phi^{-1(i)}). \quad (3.2)$$

■ 3.3.3 Learning

We aim to capture the distributions of the transformations τ_s and τ_a between the atlas and the unlabeled volumes. We estimate the functions $g_{\theta_s}(\cdot, \cdot)$ and $h_{\theta_a}(\cdot, \cdot)$ in Eqs. (3.1) and (3.2) using separate convolutional neural networks, with each network using the general architecture outlined in Fig. 3.2. Inspired by the success of recent unsupervised registration methods, we optimize the spatial and appearance models independently. Independent training has also been demonstrated with Morphable Models [79] and Active Appearance Models [33, 35]. This contrasts with our approach in Chapter 2, where we train the spatial and appearance models jointly.

For our spatial model, we leverage our previous work, VoxelMorph [16, 17, 39]. VoxelMorph learns to output a smooth displacement vector field that registers one image to another by jointly optimizing an image similarity loss and a displacement field smoothness term. We use a variant of VoxelMorph with normalized cross-correlation as the image similarity loss, enabling the estimation of $g_{\theta_s}(\cdot, \cdot)$ with unnormalized input volumes.

We use a similar approach to learn the appearance model. Naively, one might define $h_{\theta_a}(\cdot, \cdot)$ from Eq. (3.2) as a simple per-voxel subtraction of the volumes in the atlas space. While this transformation would perfectly reconstruct the target image,

it would include extraneous details when the registration function ϕ^{-1} is imperfect, resulting in image details in $x + \psi$ that do not match the anatomical labels. We instead design $h_{\theta_a}(\cdot, \cdot)$ as a neural network that produces a per-voxel intensity change in an anatomically consistent manner. Specifically, during training, we use an image similarity loss as well as a semantically-aware smoothness regularization. Given the network output $\psi^{(i)} = h_{\theta_a}(x, y^{(i)} \circ \phi^{-1})$, we define a smoothness regularization function based on the atlas segmentation map:

$$L_{smooth}(c_x, \psi) = (1 - c_x)\nabla\psi, \quad (3.3)$$

where c_x is a binary image of anatomical boundaries computed from the atlas segmentation labels l_x , and ∇ is the spatial gradient operator. Intuitively, this term discourages dramatic intensity changes within the same anatomical region. Although the appearance model never receives segmentation labels as input, this regularization term helps to train the model to make anatomically-consistent predictions.

Since this semantically-aware regularization term leverages the atlas labels, it motivates the application of the appearance transformation in the spatial frame of the atlas. Unlike our previously presented decomposition in Chapter 2, in this project we apply the appearance transformation first, before warping the example using a spatial transformation.

In the total appearance transform model loss L_a , we use mean squared error for the image similarity loss $L_{sim}(\hat{y}, y) = \|\hat{y} - y\|^2$. We experimented with applying this image similarity loss in the spatial frame of the atlas, and in the spatial frame of the subject. Even though we compute the appearance transformation using inputs in the spatial frame of the atlas, we found that applying the final image similarity loss in the spatial frame of the subject was helpful for improving the final segmentation model’s accuracy.

We balance the similarity loss with the regularization term \mathcal{L}_{smooth} :

$$\begin{aligned} L_a(x, y^{(i)}, \phi^{(i)}, \phi^{-1^{(i)}}, \psi^{(i)}, c_x) \\ = L_{sim}((x + \psi^{(i)}) \circ \phi^{(i)}, y^{(i)}) + \lambda_a L_{smooth}(c_x, \psi^{(i)}), \end{aligned}$$

where λ_a is a hyperparameter.

■ 3.3.4 Synthesizing new examples

The models described in Eqs. (3.1) and (3.2) enable us to sample spatial and appearance transformations $\tau_s^{(i)}, \tau_a^{(j)}$ by sampling target volumes $y^{(i)}, y^{(j)}$ from an unlabeled dataset. Since the spatial and appearance targets can be different subjects, our method can combine the spatial variations of one subject with the intensities of another into a single synthetic volume \hat{y} . We create a labeled synthetic example by applying the transformations computed from the target volumes to the labeled atlas:

$$\begin{aligned} \hat{y}^{(i,j)} &= \tau_s^{(i)}(\tau_a^{(j)}(x)), \\ \hat{l}_y^{(i,j)} &= \tau_s^{(i)}(l_x). \end{aligned}$$

This process is visualized in steps 3 and 4 in Fig. 3.1. These new labeled training examples are then included in the labeled training set for a supervised segmentation network. While we modeled continuous distributions of transformations in Chapter 2, here, we show that sampling from discrete distributions of transformations is also effective for data augmentation.

■ 3.3.5 Segmentation network

The newly synthesized examples are useful for improving the performance of a supervised segmentation network. We demonstrate this using a network based on the state-of-the-art segmentation architecture described in [140]. To account for GPU memory constraints, the network is designed to segment one slice at a time. We train the network on random slices from the augmented training set. We select the number of training epochs using early stopping on a validation set. We emphasize that the exact segmentation network architecture is not the focus of this work, since our method can be used in conjunction with any supervised segmentation network.

■ 3.3.6 Implementation

We implemented all models using Keras [28] and Tensorflow [1]. The application of a spatial transformation to an image is implemented using a differentiable 3D spatial transformer layer [16]; a similar layer that uses nearest neighbor interpolation is used to transform segmentation maps. For simplicity, we capture the forward and inverse spatial transformations described in Section 3.3.2 using two identical neural networks. For the appearance transform model, we use the hyperparameter setting $\lambda_a = 0.02$. We train our transform models with a single pair of volumes in each batch, and train the segmentation model with a batch size of 16 slices. All models are trained with a learning rate of $5e^{-4}$. Our code is available at <https://github.com/xamyzhao/brainstorm>.

■ 3.4 Experiments

We demonstrate that our automatic augmentation method can be used to improve brain MRI segmentation. We focus on one-shot segmentation of unnormalized scans – a challenging but practical scenario. Intensity normalization methods such as bias field correction [51, 149, 154] can work poorly in realistic situations (*e.g.*, clinical-quality scans, or scans with stroke [153] or traumatic brain injury).

■ 3.4.1 Data

We use the publicly available dataset of T1-weighted MRI brain scans described in [16]. The scans are compiled from eight databases: ADNI [119], OASIS [105], ABIDE [107], ADHD200 [112], MCIC [56], PPMI [106], HABS [38], and Harvard GSP [64]; the segmentation labels are computed using FreeSurfer [51]. As in [16], we resample the brains to $256 \times 256 \times 256$ with 1mm isotropic voxels, and affinely align and crop the im-

ages to $160 \times 192 \times 224$. We do not apply any intensity corrections, and we perform skull-stripping by zeroing out voxels with no anatomical label. For evaluation, we use segmentation maps of the 30 anatomical labels described in [16].

We focus on the task of segmentation using a single labeled example. We randomly select 101 brain scans to be available at training time. In practice, the atlas is usually selected to be close to the anatomical average of the population. We select the most similar training example to the anatomical average computed in [16], and designate it as the atlas. This atlas is the single labeled example that is used to train our transform models; the segmentation labels of the other 100 training brains are not used. We use an additional 50 scans as a validation set for hyperparameter tuning, and an additional 100 scans as a held-out test set.

■ 3.4.2 Segmentation baselines

Single-atlas segmentation (*SAS*): We use the same state-of-the-art registration model [16] that we trained for our method’s spatial transform model in a single-atlas segmentation framework. We register the atlas to each test volume, and warp the atlas labels using the computed deformation field [11, 30, 41, 61, 88]. That is, for each test image $y^{(i)}$, we compute $\phi^{(i)} = g_{\theta_s}(x, y^{(i)})$ and predict labels $\hat{l}_y^{(i)} = l_x \circ \phi^{(i)}$.

Data augmentation using single-atlas segmentation (*SAS-aug*): We use SAS results as labels for the unannotated training brains, which we then include as training examples for supervised segmentation. This adds 100 new training examples to the segmenter training set.

Hand-tuned random data augmentation (*rand-aug*): Similarly to [113, 137, 138], we create random smooth deformation fields by sampling random vectors on a sparse grid, and then applying bilinear interpolation and spatial blurring. We evaluated several settings for the amplitude and smoothness of the deformation field, including the ones described in [137], and selected the settings that resulted in the best segmentation performance on the validation set. We synthesize variations in imaging intensity using a global intensity multiplicative factor sampled uniformly from the range $[0.5, 1.5]$, similarly to [69, 81]. We selected the range to match the intensity variations in the dataset. This is representative of how augmentation parameters are tuned in practice. This augmentation method synthesizes a new randomly transformed brain in each training iteration.

Supervised: We train a fully-supervised segmentation network that uses ground truth labels for all 101 examples (the training examples plus the atlas) in our training dataset. Apart from the atlas labels, these labels are not available for any of the other methods. This method serves as an upper bound.

Table 3.1: Segmentation performance in terms of Dice score [44], evaluated on a held-out test set of 100 scans. We report the mean Dice score (and standard deviation in parentheses) across all 30 anatomical labels and 100 test subjects. We also report the mean pairwise improvement of each method over the SAS baseline.

Method	Dice score	Pairwise Dice improvement
SAS	0.759 (0.137)	-
SAS-aug	0.775 (0.147)	0.016 (0.041)
Rand-aug	0.765 (0.143)	0.006 (0.088)
Ours-coupled	0.795 (0.133)	0.036 (0.036)
Ours-indep	0.804 (0.130)	0.045 (0.038)
Ours-indep + rand-aug	0.815 (0.123)	0.056 (0.044)
Supervised (upper bound)	0.849 (0.092)	0.089 (0.072)

■ 3.4.3 Variants of our method

Independent sampling (*ours-indep*): As described in Section 3.3.4, we sample spatial and appearance target images independently to compute $\tau_s^{(i)}, \tau_a^{(j)}$. With 100 unlabeled targets, we obtain 100 spatial and 100 appearance transformations, enabling the synthesis of 10,000 different labeled examples. Because of memory constraints, we synthesize a random labeled example in each training iteration, rather than adding all 10,000 new examples to the training set.

Coupled sampling (*ours-coupled*): To highlight the efficacy of our independent transform models, we compare *ours-indep* to a variant of our method where we sample each of the spatial *and* appearance transformations from the same target image. This results in 100 possible synthetic examples. As in *ours-indep*, we synthesize a random example in each training iteration.

***Ours-indep* + *rand-aug*:** When training the segmenter, we alternate between examples synthesized using *ours-indep*, and examples synthesized using *rand-aug*. The addition of hand-tuned augmentation to our synthetic augmentation could introduce additional variance that is unseen even in the unlabeled set, improving the robustness of the segmenter.

■ 3.4.4 Evaluation metrics

We evaluate the accuracy of each segmentation method in terms of Dice score [44], which quantifies the overlap between two anatomical regions. A Dice score of 1 indicates perfectly overlapping regions, while 0 indicates no overlap. The predicted segmentation labels are evaluated relative to anatomical labels generated using FreeSurfer [51].

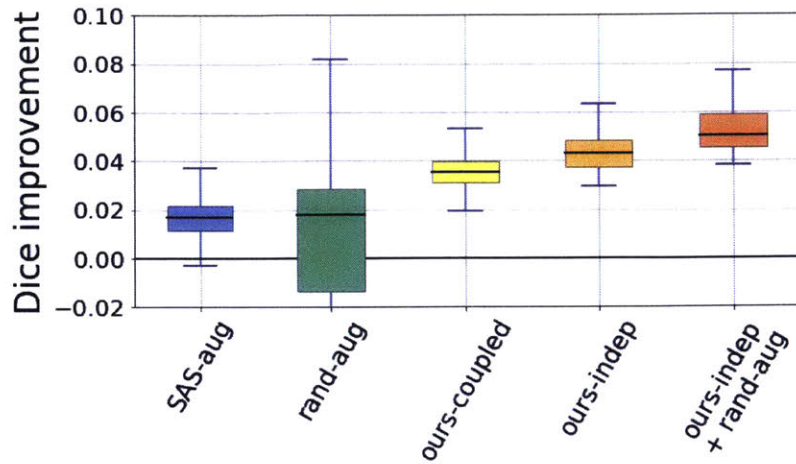


Figure 3.3: Pairwise improvement in mean Dice score (with the mean computed across all 30 anatomical labels) compared to the SAS baseline, shown across all test subjects.

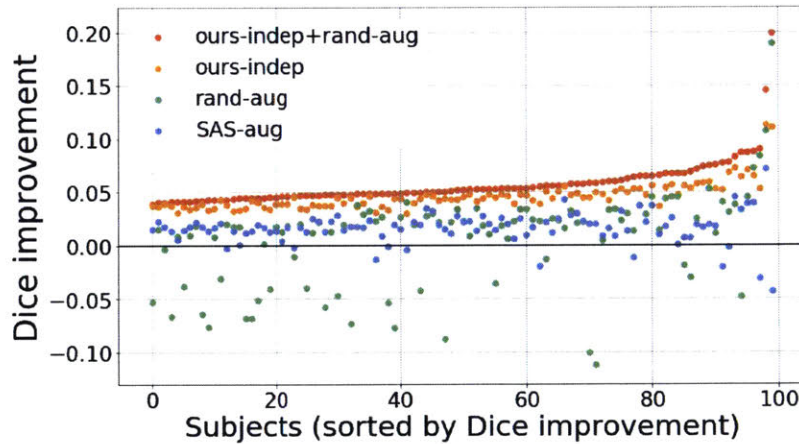


Figure 3.4: Pairwise improvement in mean Dice score (with the mean computed across all 30 anatomical labels) compared to the SAS baseline, shown for each test subject. Subjects are sorted by the Dice improvement of *ours-indep+rand-aug* over SAS.

■ 3.4.5 Results

Segmentation performance

Table 3.1 shows the segmentation accuracy attained by each method. Our methods outperform all baselines in mean Dice score across all 30 evaluation labels, showing significant improvements over the next best baselines *rand-aug* ($p < 1e-15$ using a paired t-test) and *SAS-aug* ($p < 1e-20$).

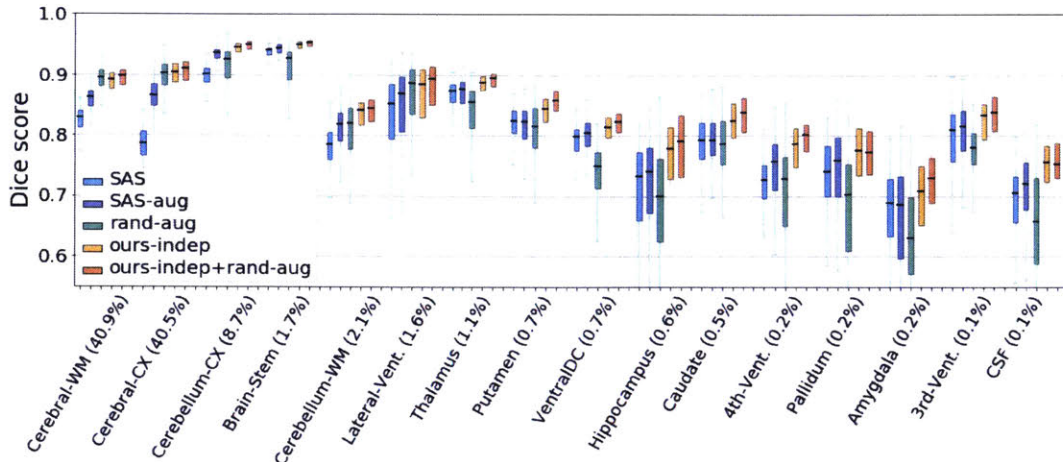


Figure 3.5: Segmentation accuracy of each method across various brain structures. Labels are sorted by the volume occupied by each structure in the atlas (shown in parentheses), and labels consisting of left and right structures (*e.g.*, Hippocampus) are combined. We abbreviate the labels: white matter (WM), cortex (CX), ventricle (vent), and cerebrospinal fluid (CSF).

In Figs. 3.3 and 3.4, we compare each method to the single-atlas segmentation baseline. Fig. 3.3 shows that our methods attain the most improvement on average, and are more consistent than hand-tuned random augmentation. Fig. 3.4 shows that *ours-indep + rand-aug* is consistently better than each baseline on every test subject. *Ours-indep* alone is always better than *SAS-aug* and *SAS*, and is better than *rand-aug* on 95 of the 100 test scans.

Fig. 3.5 shows that *rand-aug* improves Dice over *SAS* on large anatomical structures, but is detrimental for smaller ones. In contrast, our methods produce consistent improvements over *SAS* and *SAS-aug* across all structures. We show several examples of segmented hippocampi in Fig. 3.6.

Synthesized images

Our independent spatial and appearance models enable the synthesis of a wide variety of brain appearances. Fig. 3.7 shows some examples where combining transformations produces realistic results with accurate labels.

■ 3.5 Discussion

Why do we outperform single-atlas segmentation? Our methods rely on the same spatial registration model that is used for *SAS* and *SAS-aug*. Both *ours-coupled* and *SAS-aug* augment the segmenter training set with 100 new images.

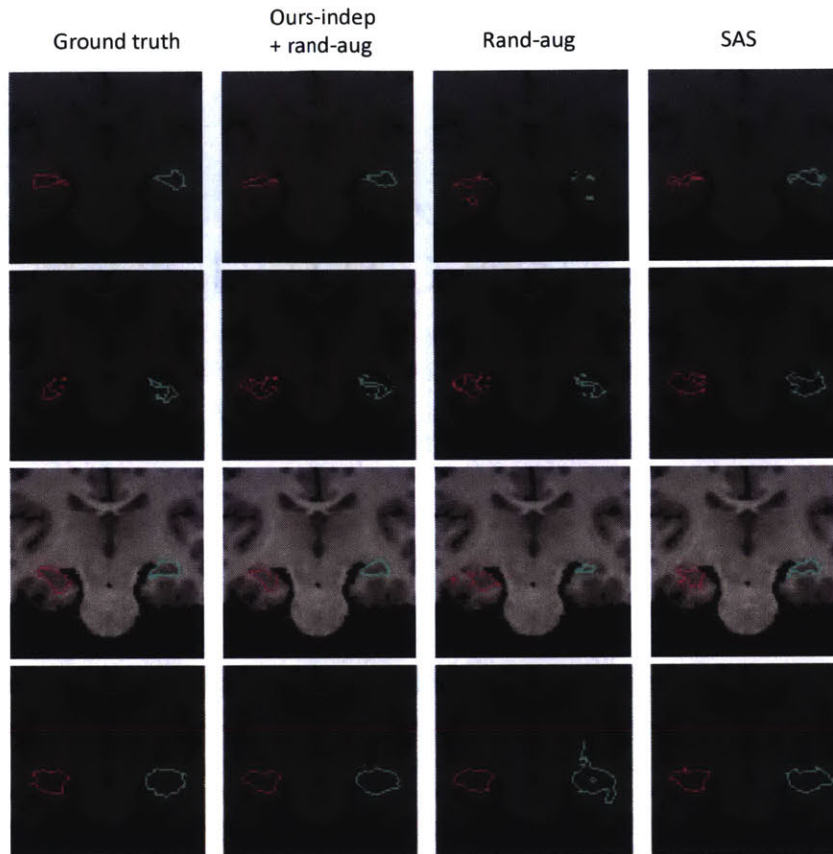


Figure 3.6: Hippocampus segmentation predictions for two test subjects (rows). Our method (column 2) produces more accurate segmentations than the baselines (columns 3 and 4).

To understand why our method produces better segmentations, we examine the augmented images. Our method warps the image in the same way as the labels, ensuring that the warped labels match the transformed image. On the other hand, *SAS-aug* applies the warped labels to the original image, so any errors or noise in the registration results in a mis-labeled new training example for the segmenter. Fig. 3.8 highlights examples where our method synthesizes image texture within the hippocampus label that is more consistent with the texture of the ground truth hippocampus, resulting in a more useful synthetic training example.

Extensions Our framework lends itself to several plausible future extensions. In Chapter 2, we modeled a continuous distributions of transformations using latent variable models, which we implemented using CVAEs. In this chapter, we showed that sampling discrete transformations is effective for data augmentation for segmentation. An

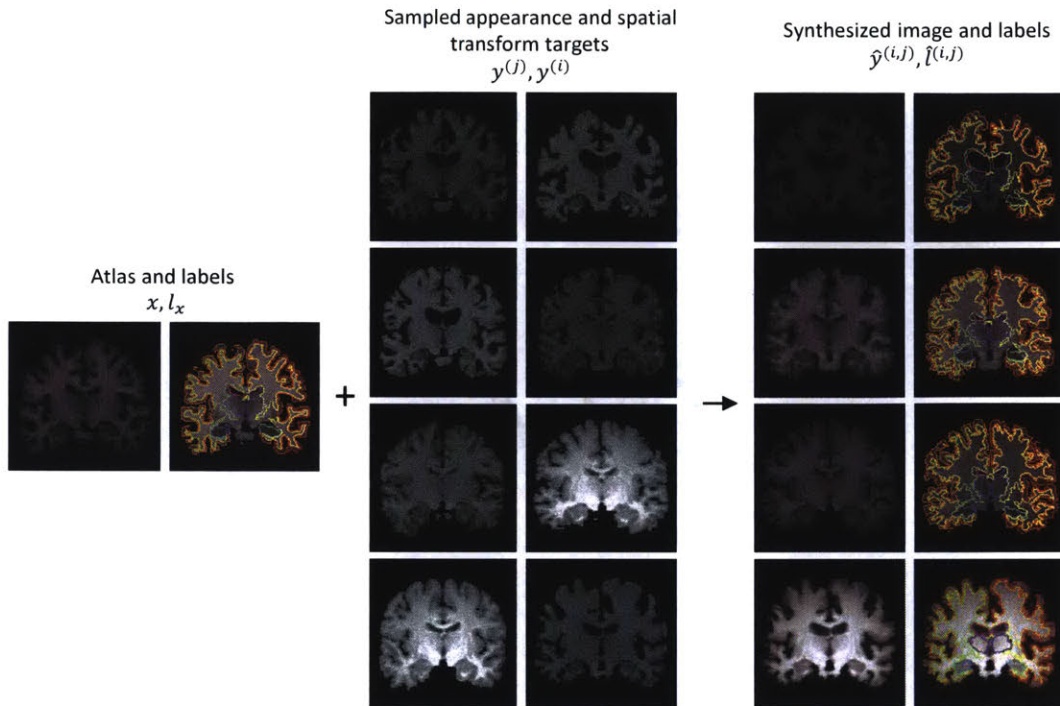


Figure 3.7: Since we model spatial and appearance transformations independently, we are able to synthesize a variety of combined effects. We show some examples synthesized using transformations learned from the unlabeled set; these transformations form the bases of our augmentation model. Notably, the top row shows a synthetic image where the appearance transformation produced a darkening effect, and the spatial transformation enlarged the ventricles. In the second row, the atlas is brightened slightly and the ventricles have shrunk.

interesting extension to this work could investigate continuous distributions of transformations, which could enable interpolation between transformations in the training set. In our early experiments, we found that balancing the hyperparameters of CVAEs was more challenging for MRI data than for Magic cards or handwritten digits. This difference is likely related to the more complex variations between MRI scans. Selecting useful hyperparameters for CVAEs is a known challenge [49, 104], and it would be interesting to explore this avenue further.

In Section 3.3.2, we discussed the use of an approximate inverse deformation function for learning the appearance transformation in the reference frame of the atlas. Rather than learning a separate inverse spatial transform model, future extensions should leverage existing work in diffeomorphic registration [9, 10, 20, 39, 186].

In this chapter, we demonstrated our data augmentation approach on brain MRIs. Since the method uses no brain- or MRI-specific information, it is feasible to extend it

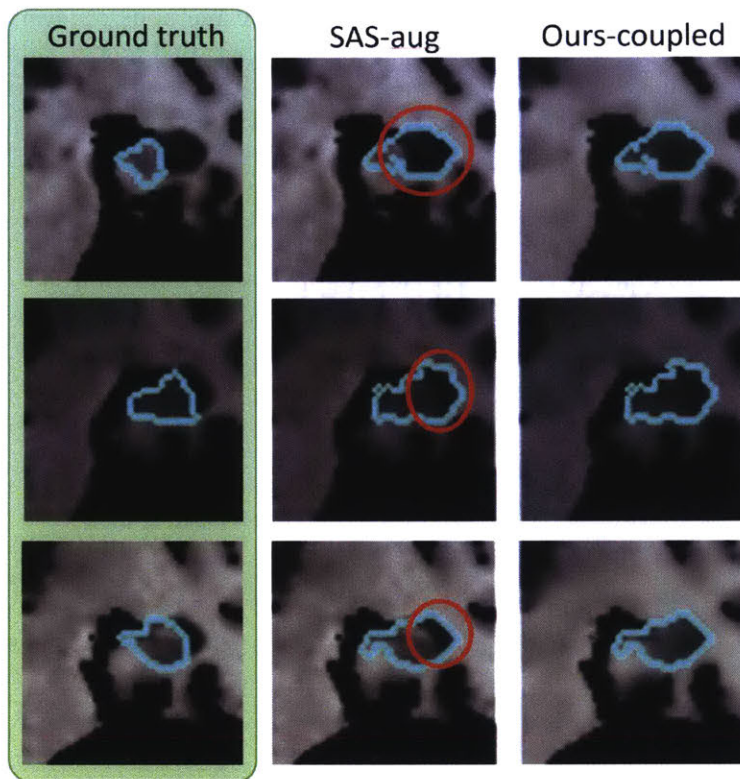


Figure 3.8: Synthetic training examples produced by *SAS-aug* (column 2) and *ours-coupled* (column 3). When the spatial model (used by both methods) produces imperfect warped labels, *SAS-aug* pairs the warped label with incorrect image textures. Our method still produces a useful training example by matching the synthesized image texture to the label.

to other anatomy or imaging modalities, such as CT.

■ 3.6 Conclusion

We presented a learning-based method for data augmentation, and demonstrated it on one-shot medical image segmentation.

We start with one labeled image and a set of unlabeled examples. Using learning-based registration methods, we model the set of spatial and appearance transformations between the labeled and unlabeled examples. These transformations capture effects such as non-linear deformations and variations in imaging intensity. We synthesize new labeled examples by sampling transformations and applying them to the labeled example, producing a wide variety of realistic new images.

We use these synthesized examples to train a supervised segmentation model. The

segmenter out-performs existing one-shot segmentation methods on every example in our test set, approaching the performance of a fully supervised model. This framework enables segmentation in many applications, such as clinical settings where time constraints permit the manual annotation of only a few scans.

In summary, this work shows that:

- learning independent models of spatial and appearance transformations from unlabeled images enables the synthesis of diverse and realistic labeled examples, and
- these synthesized examples can be used to train a segmentation model that outperforms existing methods in a one-shot scenario.

Stochastic Video Synthesis

■ 4.1 Introduction

In the previous two chapters, we showed how to design machine learning systems to model distributions of shapes and appearances within a dataset. Here we explore a related but different question: can we design a machine learning system that captures stochastic decisions made by humans?

We are interested in modeling how human artists create paintings. Many of us have perhaps imagined how Johannes Vermeer might have laid down paint, stroke by stroke, to achieve the striking contrasts of *Girl with a Pearl Earring*. While we cannot know the exact process that Vermeer followed, we can learn much from the work of modern artists. We present a learning-based model for how human artists create digital and watercolor paintings. We use the model to synthesize time lapse videos that tell a visual story of how a painting may have been created.

This problem presents two main challenges for learning-based approaches. Firstly, there is a great deal of variation in how people create art. Suppose two artists are asked to paint the same scene. One artist might start with the sky, while another might start with the mountains in the distance. Some artists might prefer to finish each object before moving onto the next, while others might lay down the base color of each object before moving onto finer details. Secondly, there is stochasticity in the decision made at each time step. As artists work on paintings, many of them will alternate between working on various objects in the scene. In a scene full of partially-completed objects, there are often no visual cues indicating what an artist will spend the next stroke on.

In this chapter, we present an exploration of the time lapse video synthesis problem. We identify the challenges of modeling time lapses of paintings, and address individual challenges through experiments on synthetic datasets. We present a recurrent model that synthesizes human painter-like strokes on small patches from images. We then discuss several next steps for extending the method to full images.

■ 4.2 Related work

To the best of our knowledge, ours is the first work on modeling and synthesizing distributions of videos representing the past, given a single current frame. Our task of

synthesizing time lapse videos of art is also novel. Here, we review the literature in two main areas: video synthesis and prediction, and art synthesis.

■ 4.2.1 Future frame prediction

Future video frame prediction is the problem of synthesizing the next frame or next few frames of a video, given a sequence of past frames. Most existing approaches train convolutional neural networks on large collections of natural videos, to predict a single frame [99], predict a future sequence as a volume [109, 165], or predict a future sequence recurrently [134, 162]. Zhou *et al.* focus on synthesizing short time lapse videos of physical processes such as melting, rotting and flowers blooming, given an initial frame [193]. These existing methods focus on capturing the single most likely future sequence, while we aim to capture a *distribution* of plausible past sequences. In [179], a conditional variational autoencoder is used to model a distribution of possible future frames. In contrast, we model a distribution of past videos.

Many of the existing methods for future frame prediction focus on natural videos, where the primary challenge is accurately extrapolating the motions of objects. The inputs to these methods often contain visual cues about the direction of the motion and the action that is being performed. On our work, we model the addition of color in paintings, which often has many more plausible outcomes.

■ 4.2.2 Frame interpolation

A classical problem in video processing is frame interpolation, in which the goal is to temporally interpolate between two frames. Classical approaches often estimate dense flow fields [13, 172, 185] or phase [111] to guide the interpolation process. More recent methods use convolutional neural networks to directly synthesize the interpolated frame [121], or combine flow fields with estimates of scene information [76, 120]. In our problem of time lapse synthesis, only the end frame is provided as input; we rely on the model to capture the distribution of initial frames. Our problem also differs from frame interpolation in that we focus on long-term predictions, while most frame interpolation methods predict a single or a few intermediate frames.

■ 4.2.3 Art synthesis

The graphics community has long been interested in simulating physically realistic paint strokes in digital media. Many existing methods focus on physics-based models of fluids or brush bristles [18, 19, 26, 29, 170, 178]. More recent learning-based methods leverage datasets of real paint strokes [85, 101, 191], often posing the artistic stroke synthesis problem as a texture transfer or style transfer problem [6, 102], which we discuss in further detail below. Several works focus on simulating watercolor-specific effects such as edge darkening [117, 168]. In our synthesis problem, we do not focus on representing individual brush strokes in a physically realistic way; rather, we focus on capturing the progression of paint strokes over the course of creating a painting.

Another relevant research area is style transfer, where images are transformed to simulate a painting-like style [62, 66] or a cartoon-like style [189]. More recently, neural networks have been used for generalized artistic style transfer [55, 194]. Style transfer methods present some useful techniques for using neural networks to synthesize artistic images; however, they do not deal with the challenges of synthesizing the temporal aspect of the art creation process.

Some works model parts of the art creation process, including simple tasks such as hatching or other repetitive strokes [77, 175]. Recently, there has been some interest in designing robots to mimic the art creation process through sketching portraits [158] or drawing graffiti [22]. We present a more general model of digital and watercolor paint strokes that is learned rather than engineered.

■ 4.3 Overview

This chapter presents a preliminary exploration of the time lapse video synthesis problem. We start in Section 4.4 by introducing the time lapse datasets we collected, and the challenges that they present. We design several synthetic time lapse datasets to illustrate some specific challenges.

With the dataset and task in mind, we formalize the video synthesis problem in Section 4.5, and then derive a recurrent probabilistic model for capturing changes at each time step. We evaluate our model in Section 4.6. We first present experiments on our synthetic datasets, and use them to illustrate limitations and design decisions in our model. Then, we present qualitative and quantitative experiments on our digital and watercolor painting datasets. Finally, we conclude the chapter in Section 4.7, where we discuss a road map for future work that could contribute to a more complete solution to the time lapse synthesis problem.

■ 4.4 Datasets

We first introduce our real datasets, and then describe how we design synthetic datasets to help examine the challenges of the real data.

■ 4.4.1 Real datasets

Digital paintings

We collected 115 recordings of digital painting time lapse videos from YouTube. We selected videos that showed digital *painting* processes (using digital brushes that mimic physical brush textures, as opposed to, say, photo editing) where the content mainly depicted landscapes or still lifes. We chose videos with minimal zooming and panning. Many of the videos were produced by the time lapse recording feature in some digital drawing applications, which automatically excludes zooming and panning. Each video was downloaded at 360×640 resolution and cropped spatially around the painting area. We visually inspected videos and manually removed segments that contained

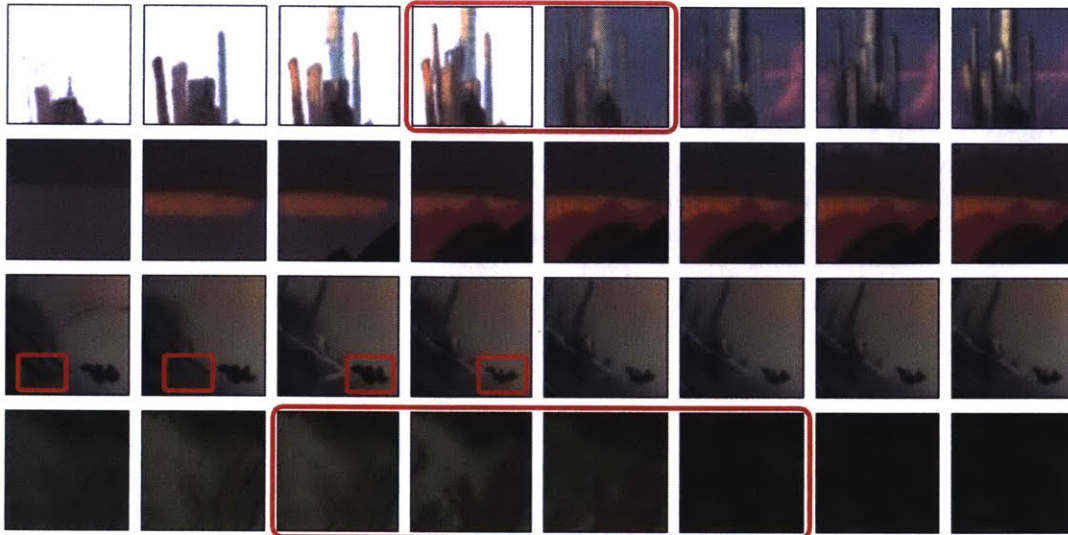


Figure 4.1: We show several digital painting sequences, where each row is a temporal sequence from a 50×50 patch. Each sequence spans 21 seconds, with frames spaced 3 seconds apart. These sequences show a variety of ways to add paint, including fine strokes (row 1), broad strokes (rows 2 and 4), and filling (row 1). Some challenges of this dataset include erasing (row 3), and drastic changes in color and composition (row 4).

movements such as translations, flipping and zooming. We also cropped each video in time to show only the painting process (without any introductions or other animations). Figure 4.1 shows some example video sequences. We split this dataset in an approximate 70:15:15 ratio into training, validation and test sets by video. Since there were only approximately 20 artists in this dataset, with a high imbalance (some artists contributed close to 20 paintings each, while other artists contributed only one) to we chose to allow some artists to appear across the training, validation and test sets.

Watercolor paintings

Similarly to the digital paintings, we collected 109 recordings of watercolor painting time lapse videos from YouTube. We chose paintings of landscapes, still lifes and portraits. We selected only videos containing very little movement of the watercolor paper. Again, we downloaded the videos in 360×640 resolution, cropped each frame to the painting area, and cropped each video in time to only show the painting process (excluding any introductions, animations or sketching that might have preceded the painting). We show some examples in Figure 4.2. As in the digital paintings dataset, we also split this dataset in a 70:15:15 ratio by video into training, validation and test sets.

A challenge with videos of physical paintings is the presence of the hand, paintbrush

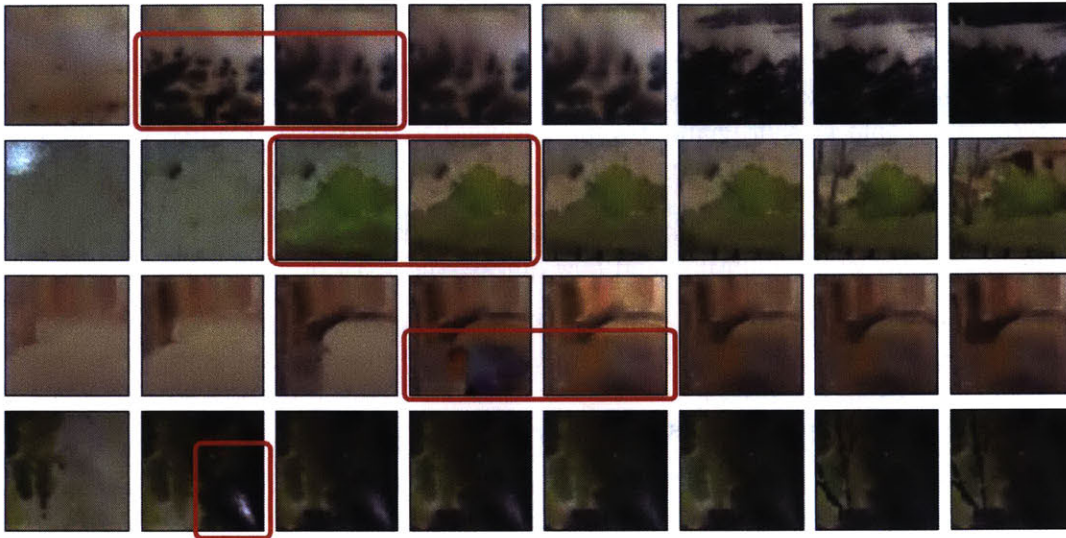


Figure 4.2: We show several watercolor painting sequences, with each row representing a temporal sequence in a 50×50 patch. These sequences span approximately 5 – 10 minutes each, with frames spaced roughly 1 minute apart. These sequences illustrate some watercolor-specific challenges, including diffusion of paint (row 1), changes in the lighting on the paper (row 2), dramatic fading effects as paint dries (row 3), and specular effects on wet paint (row 4).

and shadows in many video frames. Since these things do not belong to the content of the painting, we decided to remove any frames that contained these intrusions. We manually annotated approximately 1000 images to train a binary classifier. We designed the classifier as a simple convolutional neural network that predicted 1 for the presence of a hand, paintbrush or large shadow, and 0 otherwise. We used this classifier to remove these undesired frames from each video, and then manually removed any frames that the classifier missed.

Patch extraction

Both the digital and watercolor painting datasets contain a relatively small number of examples for machine learning purposes. To facilitate learning with small numbers of videos, we use 50×50 image patches from each video. We first scaled each video spatially by a factor of 0.7, to a maximum size (if the background was not cropped out) of 126×224 . We then extracted as many 50×50 patches as could be tiled across the frame with minimal overlap, resulting in a maximum of 15 patches per video. We selected this scaling factor such that most patches contained visually interesting content and some spatial context (*e.g.*, half of a tree instead of a single leaf). This process produced 1183 digital patch videos and 815 watercolor patch videos.

■ 4.4.2 Challenges of real painting datasets

Although there is much variability in how individual artists complete paintings, there are also some high-level patterns in our painting datasets. Below, we discuss some patterns that could present challenges for learning-based systems. Figures 4.1 and 4.2 also illustrate some of these challenges.

Variable video lengths: Artists complete paintings at different rates. This is affected by the complexity of the painting, and how quickly the artist applies paint.

Stochastic trajectories: Different artists might paint similar scenes in different orders. For instance, one artist might paint the sky, a house, and then the ground, while another artist might work in the reverse order. This variance might also occur with the same artist during different painting sessions.

Variable scales and shapes Over the course of a painting, an artist often makes strokes that vary dramatically in size and shape. Early on, an artist might use broad strokes that cover the entire sky, while later, they might use a small brush to add fine details.

Erasing and undoing in digital paintings: In digital art programs, artists have the option to undo past actions. The amount of undoing varies from artist to artist – some artists might completely change the composition of a piece several times during the course of a time lapse video.

Non-paint effects in digital paintings In digital paintings, the artist has access to many effects beyond simply filling in color. Tools that apply local blurring, smudging, or specialized paintbrush shapes are common in digital art applications such as Adobe Photoshop [155] or Procreate [71]. Artists can also apply global effects simulating varied lighting or tones. Many artists use a variety of such tools to complete each painting.

Physical effects in watercolor paintings: Watercolor paintings exhibit several distinctive effects resulting from the physical interaction of paint, water, and paper. These effects include specular lighting on wet paint, pigments fading as they dry, and water spreading outwards on the paper from the point of contact with the brush.

■ 4.4.3 Synthetic datasets

We design several synthetic datasets to simulate these challenges mentioned above. We summarize these datasets in Table 4.1.

Checkerboards-basic

We create a simple simulated dataset by filling in tiles in a colorful checkerboard according to a fixed pattern in time. Each video frame is 24×24 , with 3 randomly colored squares along each edge. We fill in 1 or 2 squares in each time step in a snake-like

Dataset	Shape variability	“Painting” rate variability	Trajectory variability
Checkerboards-basic	-	1-2 blocks filled per step	-
Checkerboards-warped	Low	1-2 blocks filled per step	-
Superpixels	High	Variable superpixel sizes	Random superpixel filled per step

Table 4.1: We synthesize video datasets to mimic some of the challenges of working with real paintings. Here, we present how each dataset was designed to mimic certain challenges.

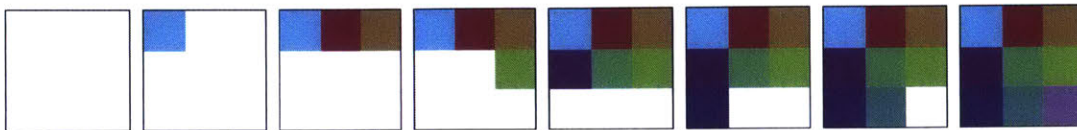


Figure 4.3: In the *checkerboards* dataset, we synthesize videos in which a colorful checkerboard is filled in over several time steps.

pattern, moving left to right in the first row, right to left in the second row, and left to right in the next row. We show the frames of an example video in Figure 4.3. This dataset simulates the challenges of variable painting rates, and variable shapes.

Checkerboards-warped

We warp each video in *checkerboards-basic* with a random smooth flow field, creating a video of a distorted checkerboard that is filled in section-by-section in the same trajectory. This effect helps to simulate the challenges of variable painting rates, with more variable shapes.

Superpixels

To create a more challenging dataset, we synthesize videos based on filling in irregular shapes in a semi-randomized pattern. We compute superpixels on images from the CIFAR10 dataset of natural images[91]. We use the SLIC superpixel segmentation function in scikit-image [2, 160], which essentially performs k -means clustering in RGB- (x,y) space, dividing each image into irregularly-shaped regions of similar colors. To turn this information into a video, we scan the image from top-to-bottom, filling in a

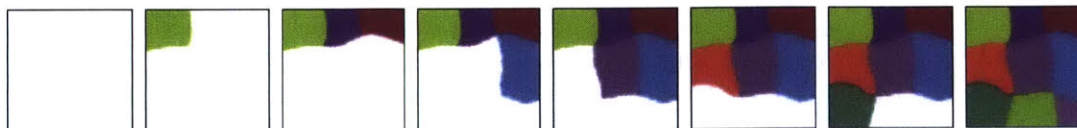


Figure 4.4: In the *warped checkerboards* dataset, we synthesize videos in which a randomly warped colorful checkerboard is filled in over several time steps.

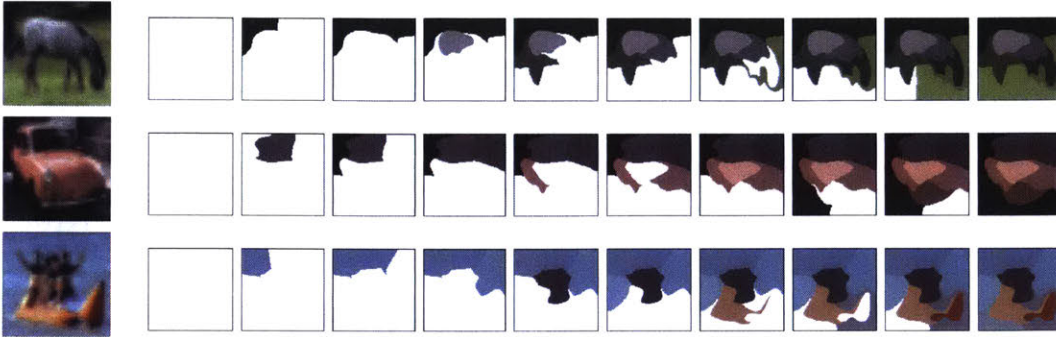


Figure 4.5: In the *superpixels* dataset, we compute coarse superpixels from images that we randomly selected from CIFAR10. We fill in superpixels from top to bottom, randomly selecting superpixels that are at the same height.

random empty superpixels in the current row of pixels. Each superpixel is filled with the average color in the region. This simulates the challenge of stochastic trajectories and variable shapes. We show examples in Figure 4.5.

In the following sections, we design a model for synthesizing time lapses of digital and watercolor paintings. We then demonstrate our model first on our synthetic datasets, and then on our real painting datasets.

■ 4.5 Method

■ 4.5.1 Problem definition

Suppose we have a collection of time lapse videos $\{\mathbf{x}^{(i)}\}$, where each video consists of a sequence of frames $x_1^{(i)}, \dots, x_{T^{(i)}}^{(i)}$ originally collected at 30 frames per second, or $f_s = 30$. Our goal is to synthesize new time lapse videos that:

- (1) show realistic painting dynamics,
- (2) are computationally feasible to synthesize, and
- (3) are visually interesting.

To address goals (1) and (2), we choose to synthesize time lapse videos at an approximate frequency of f frames per second. We select this *synthesis frame rate* f for each dataset to be sufficiently high to capture interesting changes such as individual strokes or several strokes at once, but sufficiently low to complete each painting in a reasonable number of time steps. To address goal (3), we enforce that a noticeable, non-zero change is made at each time step. We elaborate on these criteria in Section 4.5.3, where we discuss our training procedure.

To synthesize new time lapse videos, we first model the distribution of true time lapse videos, and then sample from that model. Our modeled distribution should capture each

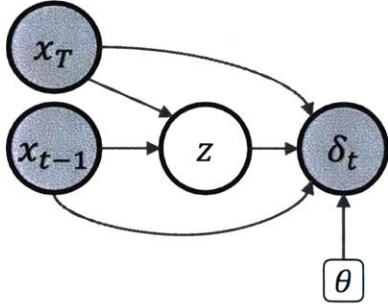


Figure 4.6: We model each paint stroke δ_t as being generated from the latent variable z . Circles represent random variables, with shaded circles denoting observed variables. Rounded rectangles represent model parameters.

true video, sampled approximately at our defined rate f : $x_0^{(i)}, x_{\Delta+\epsilon}^{(i)}, x_{2\Delta+\epsilon}^{(i)}, \dots, x_{T(i)}^{(i)}$, where $\Delta = \frac{f_s}{f}$ is the synthesis period in number of frames, and ϵ represents some allowable variance in time.

■ 4.5.2 Model

We propose a temporally recurrent *paint stroke model* that can learn from videos of variable lengths. Later, in our evaluations, we demonstrate how this design produces superior results compared to baseline methods that produce fixed-length videos.

Our recurrent paint stroke model predicts a *stroke* δ_t at each time instance t . δ_t represents the pixel-wise change that should be added to the previous frame x_{t-1} to produce the current frame x_t ; that is, in the training data, $x_t = x_{t-1} + \delta_t$. We model δ_t as being generated from a random latent variable z . The random latent variable is conditioned on the completed piece x_T and the image content at the previous time step, x_{t-1} . We show a diagram of this model in Figure 4.6. Note that each stroke by our definition does not necessarily correspond to a single paint stroke – a stroke could represent one or multiple physical or digital paint strokes.

We wish to maximize the likelihood:

$$\arg \max_{\theta} p_{\theta}(\delta_t, x_{t-1}, x_T) = \arg \max_{\theta} p_{\theta}(\delta_t | x_{t-1}; x_T) = \arg \max_{\theta} \int_z p_{\theta}(\delta_t | z, x_{t-1}; x_T) dz.$$

Similarly to the model presented in Chapter 2, the integral is intractable, so we use variational inference [87]. We introduce the distribution $q_{\phi}(z | x_{t-1}, \delta_t; x_T)$. We obtain the following expression:

$$\begin{aligned} & \arg \max_{\theta} \int_z p_{\theta}(\delta_t | z, x_{t-1}; x_T) dz \\ &= \arg \max_{\theta, \phi} \mathbb{E}_{z \sim q_{\phi}(z | x_{t-1}, \delta_t; x_T)} [\log p_{\theta}(\delta_t | z, x_{t-1}; x_T)] - KL[q_{\phi}(z | x_{t-1}, \delta_t; x_T) || p(z)], \end{aligned}$$

where $KL[\cdot || \cdot]$ denotes the Kullback-Liebler divergence. We propose that the distribu-

tions $p_\theta(\delta_t|z, x_{t-1}; x_T)$, $q_\phi(z|x_{t-1}, \delta_t; x_T)$, $p(z)$ take the form of multivariate normals:

$$\begin{aligned} p_\theta(\delta_t|z, x_{t-1}, x_T) &= \mathcal{N}(\delta_t; x_t - x_{t-1}, \sigma_\delta^2 \mathbb{1}), \\ q_\phi(z|x_{t-1}, x_T, \delta_t) &= \mathcal{N}(z; \mu_\phi(x_{t-1}, \delta_t, x_T), \Sigma_\phi(x_{t-1}, \delta_t, x_T) \mathbb{1}), \\ p(z|x_{t-1}, x_T) &= \mathcal{N}(z; 0, \mathbb{1}), \end{aligned}$$

where $\mu_\phi(\cdot)$, $\Sigma_\phi(\cdot)$ are learned functions parameterized by ϕ , and σ_δ is a hyperparameter. Then, we can write the loss optimization for our model as:

$$\arg \min_{\theta, \phi} \lambda L_{L2}(x_{t-1}, x_T, \delta_t) + L_{KL}(x_{t-1}, x_T, \delta_t),$$

where $L_{L2}(x_{t-1}, x_T, \delta_t) = \frac{1}{2\sigma_\delta^2} \|\delta_t - (x_t - x_{t-1})\|^2$ and $L_{KL}(x_{t-1}, x_T, \delta_t) = \frac{1}{2} (-\log \Sigma_\phi + \Sigma_\phi + \mu_\phi^2)$, and λ is a hyperparameter.

In image synthesis tasks, using L2 as an image similarity loss often produces blurry results [72]. We instead optimize the L1 distance in pixel space as well as the L2 distance in a perceptual feature space, which we denote as L_{L1} and L_{VGG} respectively. Perceptual losses are commonly used in image synthesis and processing tasks to produce sharper and more visually pleasing results [46, 72, 78, 121, 187]. We use the L2 distance between normalized VGG features as described in [187].

■ 4.5.3 Training

We use two stages of training to facilitate convergence: pre-training, and sequential training. In each stage, we train on short sequences of frames from our video dataset. We discuss our sequence selection criteria below.

Sequence selection

As discussed in Section 4.5.1, we wish to produce time lapse videos at a synthesis frame rate f , such that the synthetic videos capture painting-like dynamics, are sufficiently short to be feasible to synthesize, and contain a noticeable stroke at each time step. We choose f based on each dataset’s stroke statistics.

As defined in Section 4.5.1, a stroke is the pixel-wise change that occurs in a patch over some time step Δ . We compute strokes from true video patches taking the pixel-wise difference in time. We define a “noticeable stroke” to occupy at least 10% of each patch by area. To compute true stroke areas, we binarize each true stroke using an experimentally determined pixel intensity threshold, and then apply erosion and dilation operators to reduce the effect of noise. We compute the stroke area by summing the resulting binary stroke mask. Figure 4.7 shows the distribution of the smallest time step Δ that produces a noticeable stroke, for each painting dataset.

Based on these distributions, we select the synthesis period for digital paintings to be $\Delta_d = 45$ frames = 1.5s with an allowable variance $\epsilon_d \in [-40, 40]$. For watercolors, we choose a synthesis period of $\Delta_w = 600$ frames = 20s with an allowable variance $\epsilon_w \in [-400, 400]$. The difference in these periods is largely because the digital paintings dataset contains videos that have already been sped up significantly from real time.

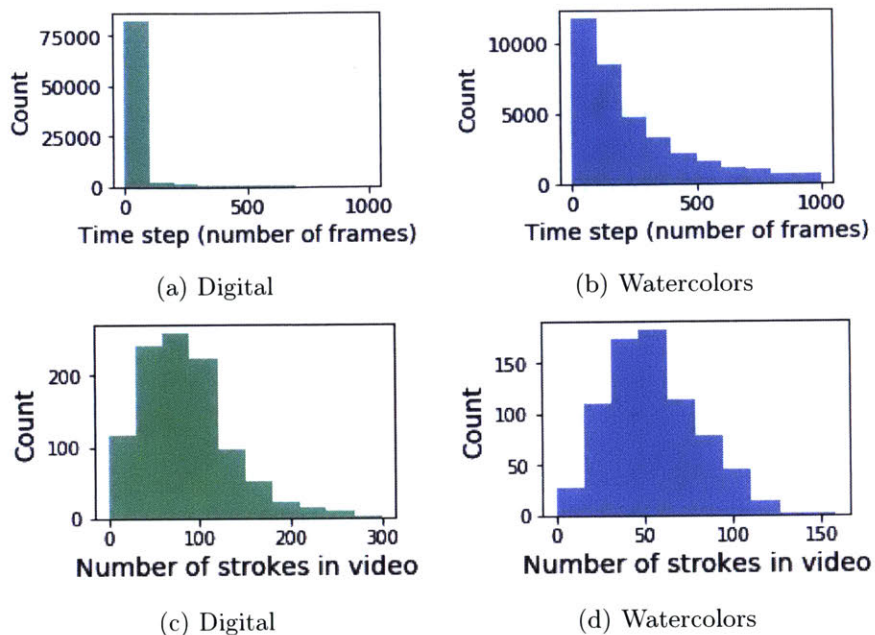


Figure 4.7: Statistics of “noticeable” strokes for each training dataset. In the first row, we show the distribution of time differences that result in noticeable strokes. In the digital paintings dataset, the median and mean time steps are 1 frame and 56 frames respectively. In the watercolors dataset, the median and mean time step are 210 and 668 respectively. In both datasets, the mean time step is much higher than the median due to long tails in the distributions. We also count the number of noticeable strokes in each dataset (bottom row). The mean and median number of strokes in the digital paintings dataset are 78 and 87 respectively. In the watercolor paintings dataset, they are 56 and 53 respectively.

Pre-training

We first train the recurrent paint stroke model on short sequences from each video. Since the model only requires information from the previous time step as input, we sample all sequences of length 2 from the video dataset that satisfy the above criteria. We denote training examples as (input, target) tuples. The model is pre-trained on $(x_{t-1}, x_T; x_t), t = 1, \dots, T$. This is illustrated in Figure 4.8. We also include sequences $(x_{blank}, x_T; x_0)$ from videos that start with a blank or approximately uniformly-colored frame, where x_{blank} is a completely white patch. These *starter sequences* are important for teaching our model how to start a painting at test time.

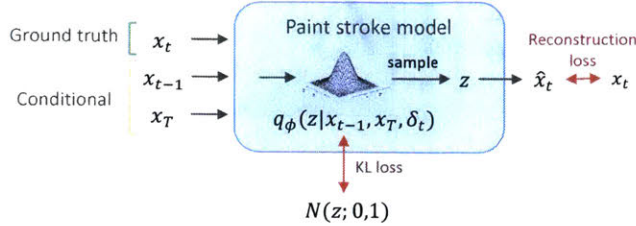


Figure 4.8: We define a recurrent paint stroke model that predicts the stroke at each time step, which is added to the previous frame to predict the current frame. This model is trained on adjacent frames from our video dataset.

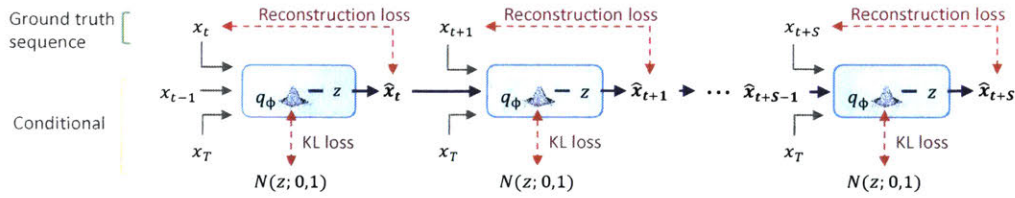


Figure 4.9: In sequential CVAE training, the paint stroke model is trained to reconstruct the correct frame by building on its previous predictions, for a given number of time steps S .

Sequential CVAE and sampling training

To synthesize a video, we must run our recurrent model for multiple time steps, building upon its own predictions. It is common when making sequential predictions to observe compounding errors or artifacts over time [162]. We use a sequential training scheme to enforce that the outputs of the model are accurate over multiple time steps. We alternate between two sequential training modes: *sequential CVAE training* and *sequential sampling training*.

In the *sequential CVAE trainer*, the paint stroke model is trained to produce the true frames for several time steps S , while building upon its own predictions in previous time steps. This helps to control the divergence of the model’s outputs due to compounding errors. We illustrate this training scheme in Figure 4.10.

In the *sequential sampling trainer*, the paint stroke model is trained to produce reasonable strokes by decoding latent samples from the prior distribution. This training mode is designed to account for a limitation of CVAEs and VAEs. CVAEs and VAEs typically offer a trade-off between sharp training reconstructions with unreasonable samples, and blurry training reconstructions with plausible samples [49]. A proposed explanation is that the decoder portion of the network is trained only on samples from the distribution $q_\phi(z|x_{t-1}, \delta_t; x_T)$, which might have a high KL divergence from the prior distribution $p(z)$ that we sample from at test time [49, 104]. In other words, the typical CVAE training scheme does not ensure that all samples from the prior distribution decode to frames that are likely under the training distribution.

Our system relies on sampling latent vectors from the prior distribution to pro-

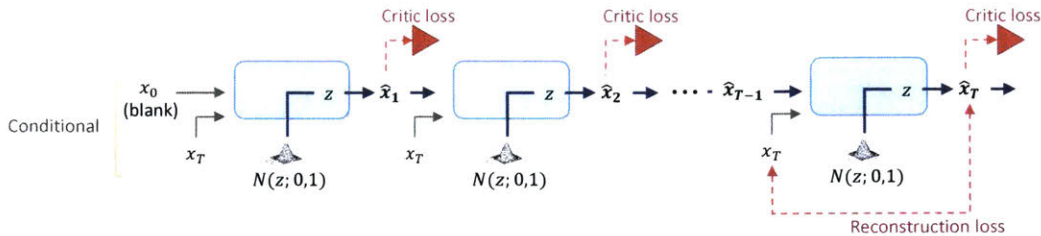


Figure 4.10: In sequential sampling training, we use a conditional frame critic to encourage all frame samples from our model to look reasonable. We use a reconstruction loss on the final frame to ensure that our model completes the painting within the specified number of time steps T .

duce realistic frames at test time. We use a conditional critic [59] to encourage the distribution of sampled strokes to match the distribution of true strokes. The critic receives samples of (x_{t-1}, x_t, x_T) as “real” examples, and $(x_{t-1}, \hat{x}_t, x_T)$ as “fake” examples, where \hat{x}_T denotes frames predicted by our model. Although critics and discriminators are most commonly used in generative adversarial networks [58, 59, 132], they have also been used to map an imposed prior to a data distribution [104], or as an additional image reconstruction loss in a VAE [84]. Our sequential sampling trainer can be thought of as a sequential application of [104], which uses an image discriminator in place of a KL divergence loss to encourage sampled images to be likely under the training image distribution. In addition to the critic loss, we also introduce a frame similarity loss after a specified number of time steps τ , to encourage the sequential model to eventually produce the completed painting. Based on the dataset statistics presented in Figure 4.7, we select $\tau = 40$.

■ 4.5.4 Network architecture

We use a CVAE similar to the appearance transform model discussed in Chapter 2. As shown in Figure 4.11, we use implement conditional information through concatenation operations. This reduces the number of parameters required in the network, and ensures that information at various spatial scales is incorporated to the network’s prediction of δ_t .

■ 4.5.5 Network training

We implement our model using Keras [28] and Tensorflow [1]. We pre-train our recurrent model on sequences of length 2 until the validation loss plateaus. We select the hyperparameter controlling the reconstruction loss weight to be $\lambda = 50$. We then alternate between sequential CVAE training and sequential sampling training, on odd and even iterations respectively, for approximately 200 epochs, which we determined through visual inspection of the synthesized validation videos. When we perform sequential CVAE training, we use sequence lengths of $S = 3, 5$. For our sequential sampling trainer, we

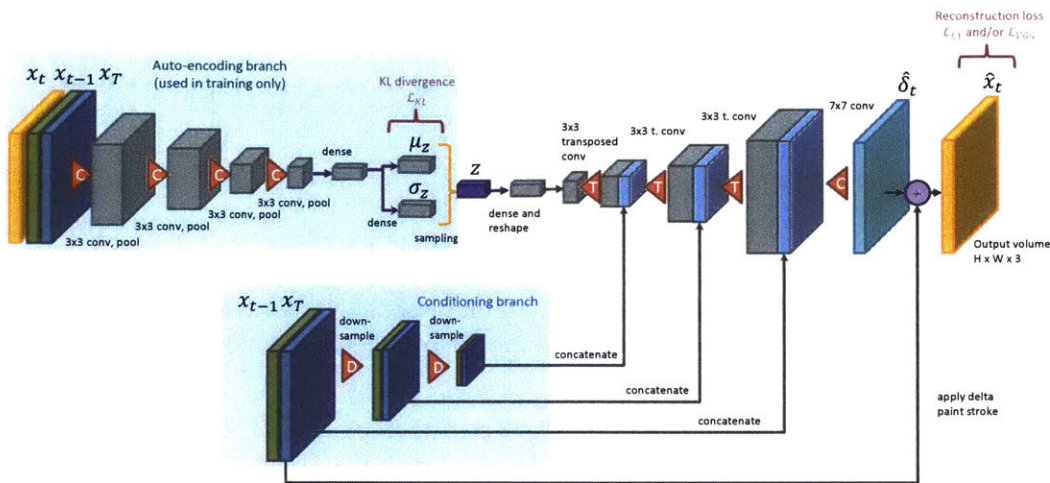


Figure 4.11: We implement our recurrent painter model using a conditional variational autoencoder (CVAE). At training time, the network is encouraged to reconstruct the current frame x_t , while sampling the latent z from a distribution that is close to the standard normal. At test time, the auto-encoding branch is removed, and z is sampled from the standard normal.

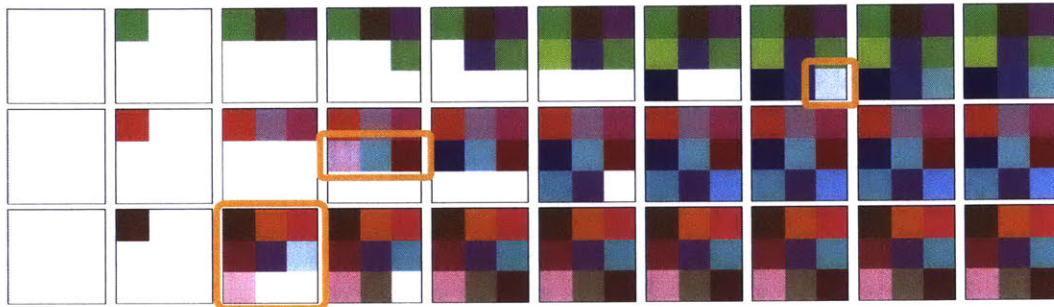
use WGAN-GP [59] with the default gradient penalty weight of 10, and a critic model based on [27], with 5 critic training iterations for every generator training iteration. In all stages of training, we use a batch size of 16 and a learning rate of $1e^{-4}$.

■ 4.6 Experiments

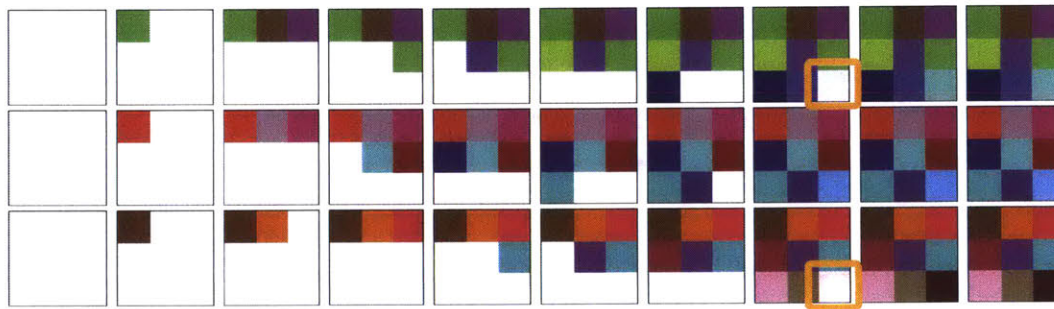
■ 4.6.1 Synthetic results

We first demonstrate the utility of the different training schemes presented in Section 4.5.3 on our synthetic datasets. In image and video synthesis tasks, artifacts and compounding errors are common challenges [27, 162, 194]. We show in Figure 4.12 that even on a simple dataset such as *checkerboards-basic*, our pre-trained recurrent model produces artifacts and errors that compound over time (*e.g.*, squares being filled with faded colors). Sequential CVAE training reduces some of these errors.

We show predicted videos from more complex synthetic datasets (*checkerboards-warped*, *superpixels*) in Figures 4.13 and 4.14. These figures show that even sequential CVAE training is insufficient for consistently producing artifact-free strokes, or for completing each image in the expected number of steps. Alternating between a sequential CVAE trainer and a sequential sampling trainer helps with both of these issues.

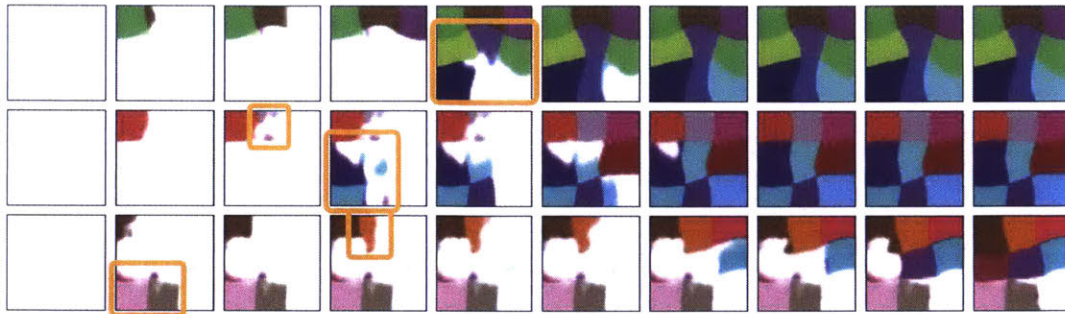


(a) Pre-trained

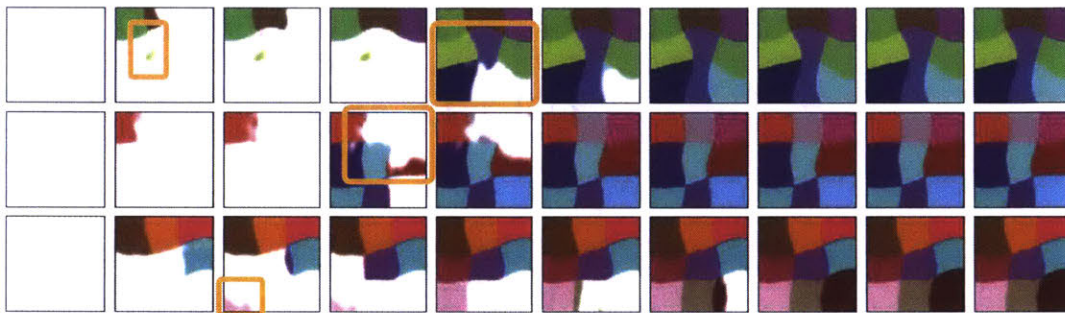


(b) After sequential CVAE training

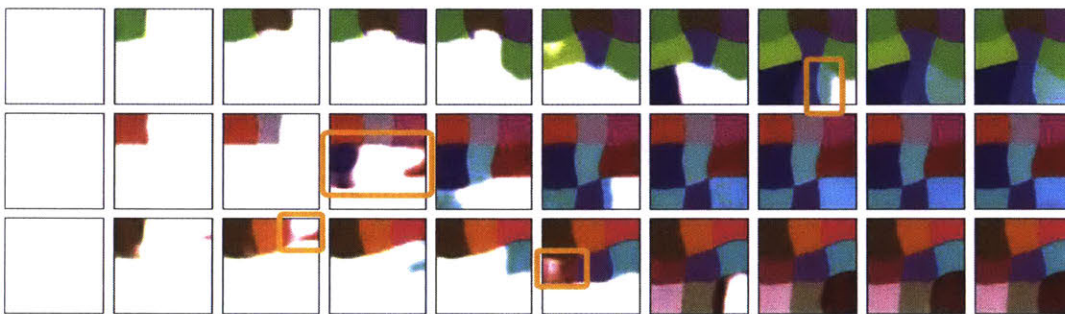
Figure 4.12: Predicted videos from the *checkerboards-basic* dataset, after pre-training, and after sequential CVAE training. a) Pre-training alone produce errors such as squares filled with the wrong color (all rows), or too many squares being filled in one time step (rows 2-3). b) Sequential CVAE training helps to fill in the correct number of squares, although some squares are still partially filled.



(a) Pre-trained

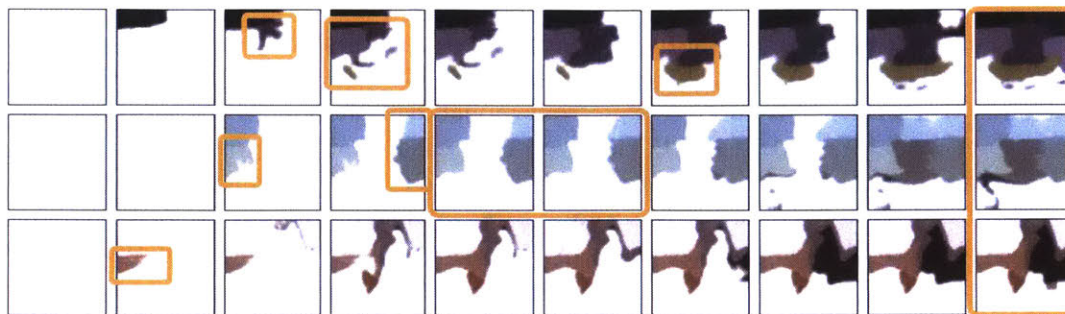


(b) After sequential CVAE training

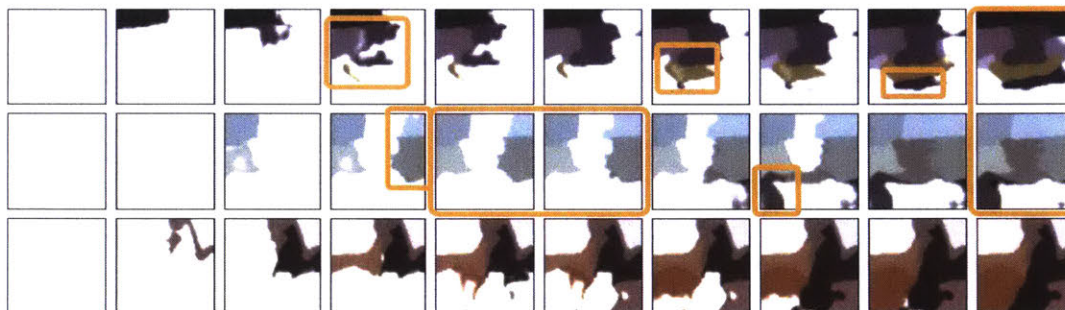


(c) After alternating (sequential CVAE and sequential sampling) training

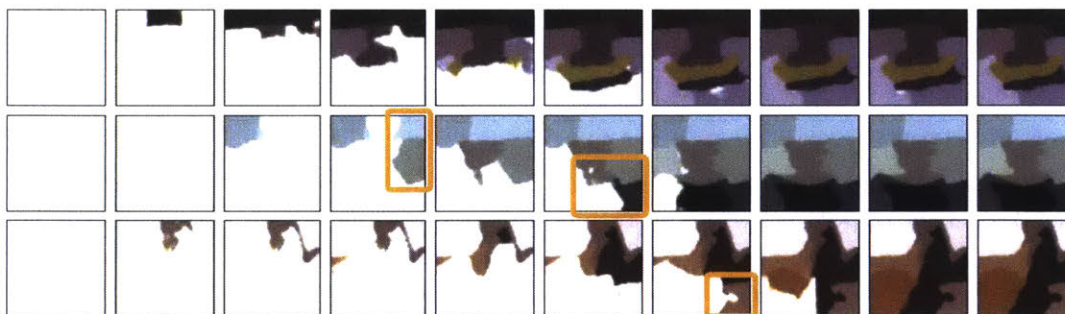
Figure 4.13: Predicted videos from the *checkerboards-warped* dataset, after pre-training, sequential CVAE training, and alternating sequential training. a) The pre-trained model produces many artifacts such as partially-filled regions, and regions filled in an incorrect order. b) Sequential CVAE training helps to alleviate some of these issues. c) Alternating between sequential CVAE training and sequential sampling training produces the most accurate region filling trajectory, with only a few errors in the form of partially-filled regions.



(a) Pre-trained



(b) After sequential CVAE training



(c) After alternating (sequential CVAE and sequential sampling) training

Figure 4.14: Predicted videos from the *superpixels* dataset, which contains complex shapes and a more stochastic filling pattern. a) The pre-trained model produces many artifacts, and often does not fill superpixels in order from top to bottom (highlighted). The model also does not complete the “painting”. b) Sequential CVAE training also produces artifacts, incorrect filling orders (highlighted), and does not complete the frame. c) Our alternating sequential training scheme reduces the number of artifacts, and also reduces instances of superpixels being filled in the wrong order.

■ 4.6.2 Digital and watercolor painting results

We evaluate our predicted painting time lapses both qualitatively and quantitatively. The stochastic nature of our task makes it challenging to compare our predicted videos to ground truth videos. We do not expect our model to always create paintings the way that the original human artist did. However, our model should capture a distribution of plausible painting time lapses – that is, we expect our model to consistently produce time lapse videos that look like they were created by human artists. We elaborate on these qualities in the following sections. We also expect the ground truth time lapse videos to be captured as a part of our model’s learned video distribution.

We investigate some related baseline methods:

- **Linear interpolation (*interp*):** As a simple baseline, we simply linearly interpolate each pixel value in the frame from white (a blank canvas) to its value in the completed piece of art.
- **Deterministic video synthesis (*unet*):** In image synthesis tasks, it is common to use a simple encoder-decoder architecture with skip connections, similar to U-Net [137], to synthesize each image [72]. We adapt this technique to synthesize the entire video at once.
- **Stochastic video synthesis (*vdp*):** In our visual deprojection work [14], we demonstrate how to synthesize distributions of videos of a fixed length. We apply this technique to the synthesis of time lapse videos.

We experiment with training each method on digital or watercolor paintings only (*-digital* and *-watercolors* respectively), as well as on the combined paintings dataset (*-paintings*).

Qualitative results

In Figures 4.15 and 4.16, we examine time lapse videos created for digital and watercolor paintings respectively. These results are sampled coarsely in time to show the progression of the painting over 30 steps. In the interest of space, we only compare our method to the *unet* and *vdp* baselines, omitting the simpler *interp* baseline in this qualitative evaluation. We present only models trained on the combined *paintings* dataset, since these models produce the most visually consistent results across the digital and watercolor validation sets. The ground truth videos in these figures show how human painters tend to work in a coarse-to-fine manner, using broad strokes near the start of a painting, and finer strokes near the end. Our method produces more plausible paint strokes than the baselines. These figures also show the most similar predictions made by each baseline to the ground truth video. Our prediction often shows a similar trajectory to the ground truth painting, indicating that our model captures realistic trajectories in its distribution of videos.

In Figure 4.18, we show several short sequences at a finer time scale. In addition to working in a coarse-to-fine manner of time, the human artist tends to work within a

single object or a few object boundaries in each time step, making strokes that spatially and semantically coherent. Our method also makes localized changes compared to the baseline *vdv*, which changes the entire patch in each time step. In Figure 4.17, we demonstrate that our model can be used to synthesize a variety of plausible painting trajectories.

Quantitative results

We also quantitatively evaluate how well each method captures the distribution of realistic time lapse videos. For our evaluation videos, we extract sequences from the patches in each test set, using the temporal sampling rates defined in Section 4.5.3. We filter these sequences using the same criteria that we use for the training sets of each method, removing frames that exhibit no change from the previous frame. We then crop or pad (using the last frame) each video in time to be 40 frames long. We evaluate several video similarity metrics:

- **Best (across k samples) overall video similarity:** As in [14], for each test patch, we draw k sample videos from our model and evaluate the most similar predictions to the true video. We show how this loss changes for varying values of k . This metric measures whether a time lapse video is likely under the distribution modeled by our system.

Accurately evaluating image similarity is still an open challenge [73, 143]. Apart from common per-pixel measures such as mean absolute error (L1), the literature in image and video synthesis uses human evaluations [73, 127, 165, 194] and deep perceptual similarity metrics such as LPIPS [187]. Works that use GANs for image synthesis often use Inception Score [143], or more recently, the Fréchet Inception Distance (FID) [63], to evaluate the similarity of the synthesized and real image distributions [82, 127]. We use L1 and LPIPS [187] measures to evaluate frame similarity. We do not report distribution similarity metrics such as FID, as it is unclear how these metrics can be adapted to videos.

- **Best (across k samples) stroke shape similarity:** We compare the shapes of strokes made in the ground truth video to those made in our predicted videos, to show that our model paints in similar semantic regions. As discussed in Section 4.5.3, we define *stroke area* as a binary map of the changes made in each time step. We show some examples of stroke areas in Figure 4.19. For each test video, we compare each true stroke to the most similar predicted stroke, as measured by intersection-over-union (IOU). We report the average IOU over all non-empty ground truth strokes. Intuitively, this metric describes whether the collection of ground truth stroke shapes exists in the predicted video, regardless of the order in which they were performed.

In Figures 4.20 and 4.21, we show the effect of the number of sampled videos on several video similarity metrics. We summarize some of these results in Tables 4.2

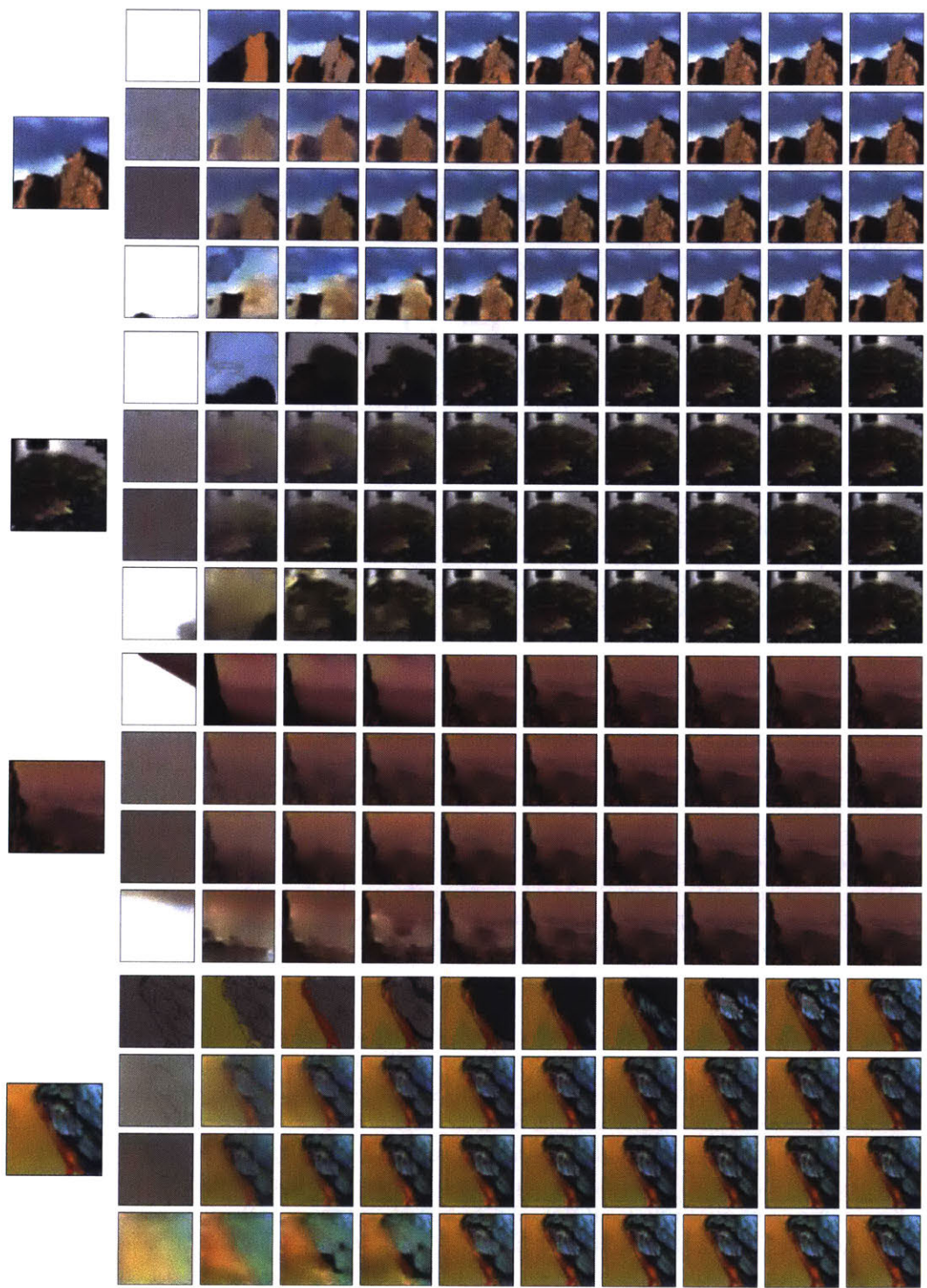


Figure 4.15: Videos predicted using input patches from the digital paintings test set. Each group of 4 rows shows the input patch on the left, the ground truth video in the first row, and predictions made by *unet-paintings*, *vdp-paintings* and *ours-paintings* in the subsequent rows. For the stochastic methods *vdp-paintings* and *ours-paintings*, we show the best video (with the lowest L1 error compared to the ground truth video) out of 2000 samples. The baselines (*unet-paintings*, *vdp-paintings*) appear to gradually interpolate between a grey frame and the completed patch. In contrast, our method uses strokes that vary from coarse to fine spatial scales over time. In the first three examples, our method fills in similar regions in each time step compared to the ground truth video.

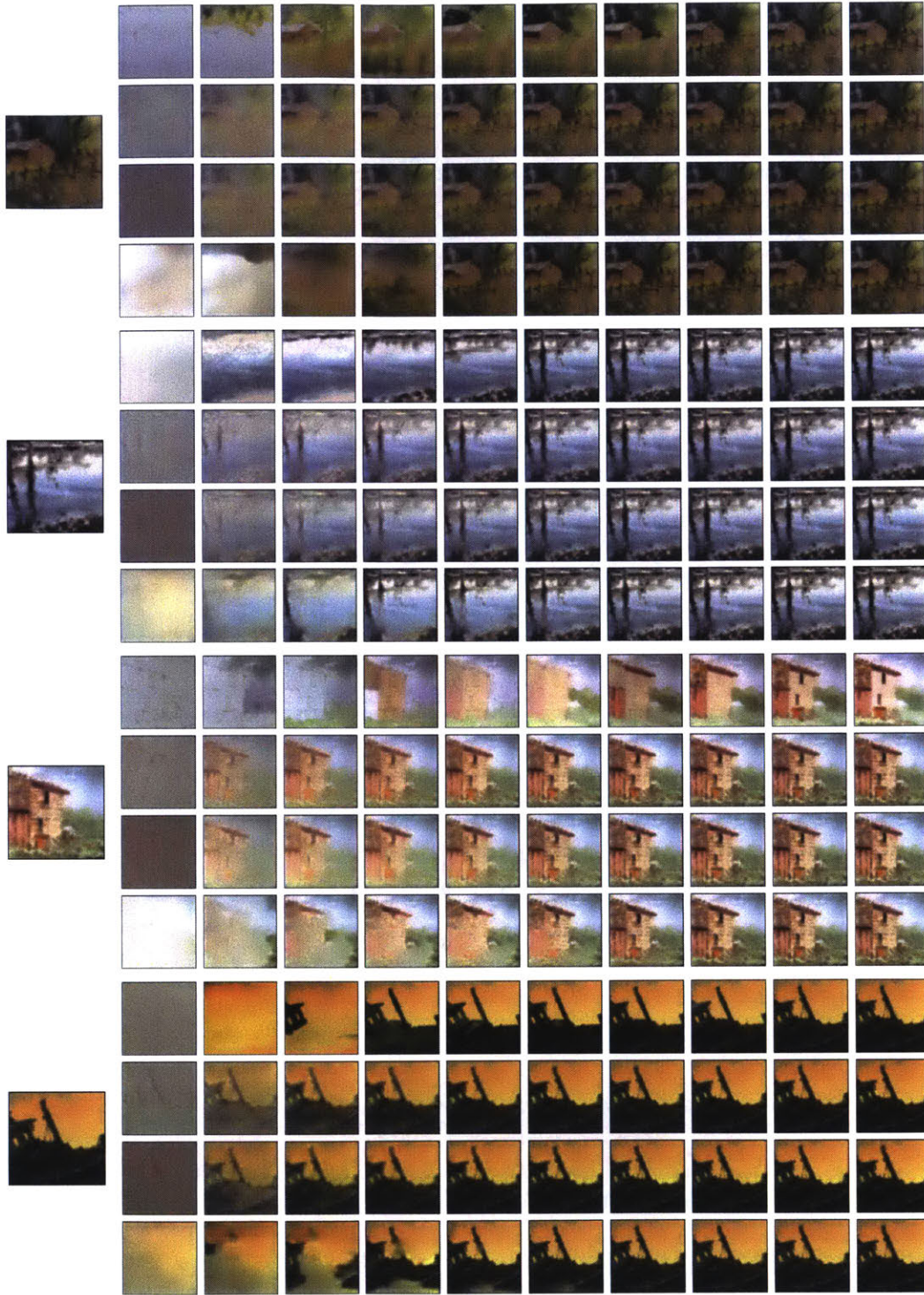
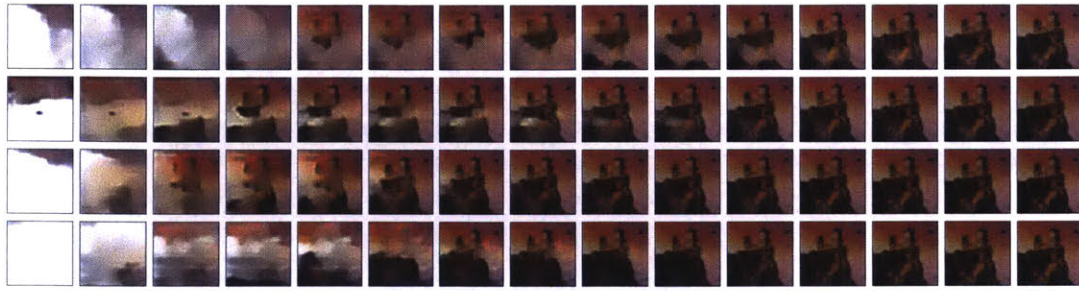
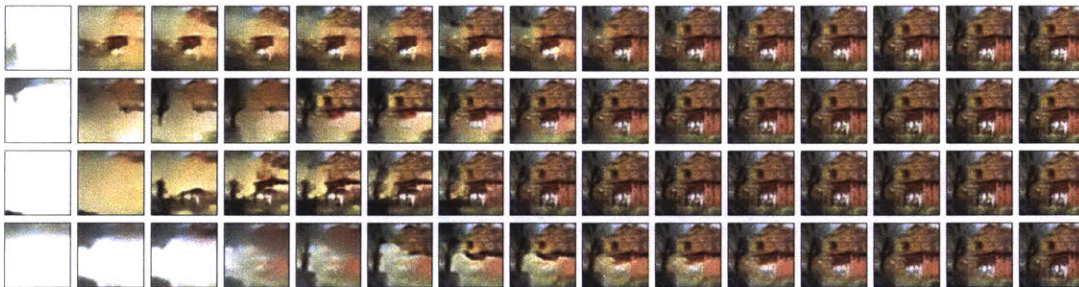
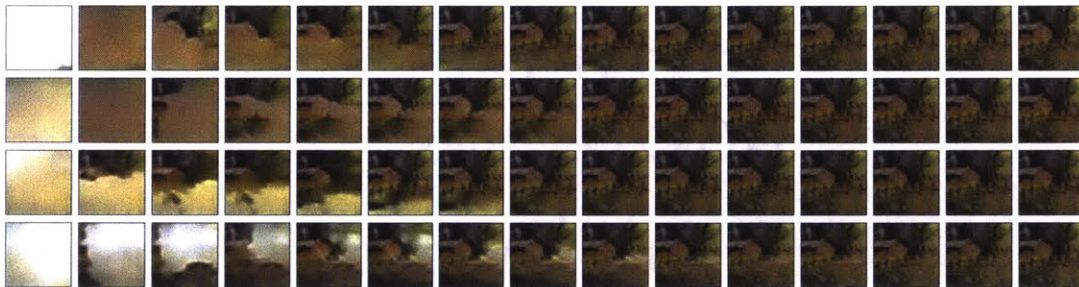


Figure 4.16: Videos predicted using input patches from the watercolor paintings test set. Each group of 4 rows shows the input patch on the left, the ground truth video in the first row, and predictions made by *UNET-paintings*, *VDP-paintings* and *OURS-paintings* in the subsequent rows. For the stochastic methods *VDP-paintings* and *OURS-paintings*, we show the best video (with lowest L1 error compared to the ground truth video) out of 2000 samples. Our method uses fewer interpolation-like effects than the baseline methods, painting in regions rather than everywhere at once. In all four examples, our method uses a similar painting trajectory to the ground truth video.

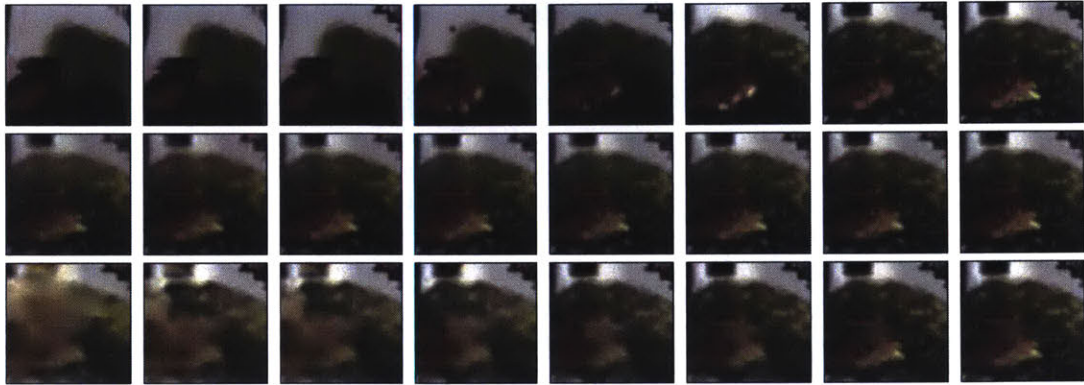


(a) Digital paintings

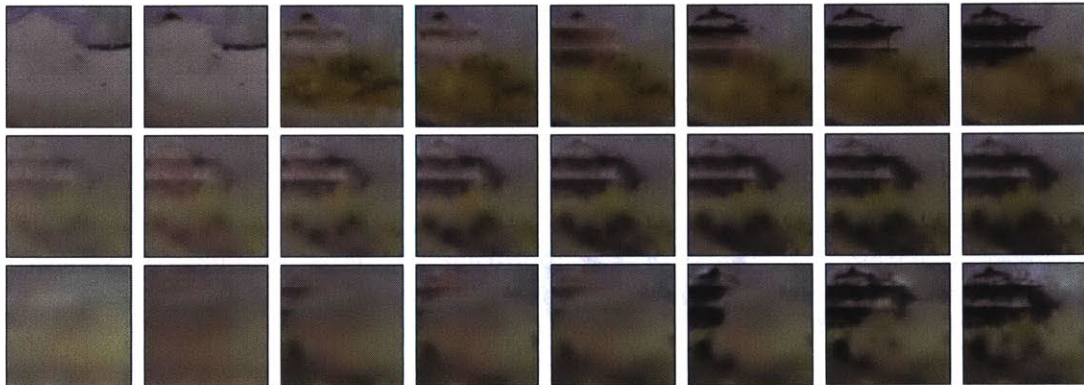


(b) Watercolor paintings

Figure 4.17: Several videos created by sampling from our model. Each group of 4 rows shows 4 different time lapse video samples created from the same input patch. Our method produces diverse and plausible painting trajectories.



(a) Digital painting



(b) Watercolor painting

Figure 4.18: A close-up of short sequences from the digital and watercolor test sets. In each group of 3 sequences, we show the ground truth (top), predictions made by *vdp* (middle), and predictions made by our model (bottom). Our model makes coarser changes near the start of the sequence, and adds layers of finer details near the end. Our model also makes changes that are spatially localized, working mostly on the jar in the first video, and on the building in the second. This produces an effect that is more similar to the ground truth process than the baseline.



Figure 4.19: Examples of stroke areas computed from a video, with the true video frames (top) and computed stroke areas between each pair of frames (bottom).

Method	Best video similarity (top 10)	
	L1	LPIPS
interp	0.52 (0.12)	0.25 (0.06)
UNET-digital	0.15 (0.08)	0.10 (0.05)
UNET-paintings	0.15 (0.07)	0.10 (0.05)
vdp-digital	0.15 (0.07)	0.10 (0.05)
vdp-paintings	0.14 (0.07)	0.10 (0.05)
ours-digital	0.13 (0.06)	0.10 (0.04)
ours-paintings	0.13 (0.06)	0.09 (0.04)

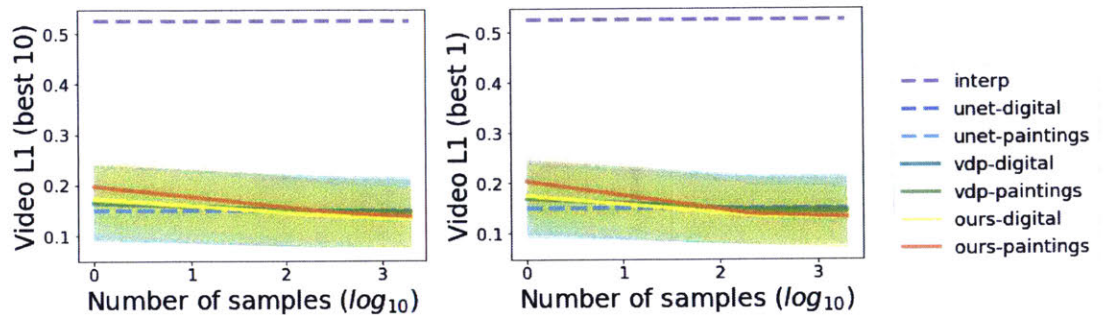
Table 4.2: We compare 2000 samples from our predicted video distributions to the videos in our digital paintings test set.

and 4.3. As the results indicate, the baselines are much more capable of accurately representing watercolor paintings compared to digital paintings. This is likely caused by differences in dataset size. While we have approximately the same number of videos in the digital and watercolor painting datasets, there are fewer sequences in the watercolors dataset that satisfy our sequence criteria compared to the digital paintings dataset (by about a factor of 5); thus, the watercolor dataset is relatively easier for video-based methods compared to our recurrent method.

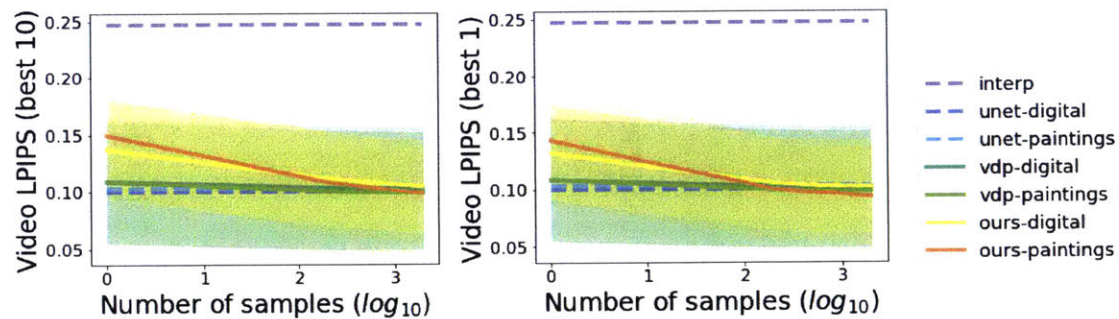
4.7 Summary and discussion

In this chapter, we explored a novel video synthesis problem: creating time lapse videos that depict the creation of paintings. We designed a recurrent probabilistic model to capture the seemingly random decisions of human painters. We demonstrated limitations and design decisions of our model on several synthetic datasets. Finally, we applied our model to digital and watercolor paintings, and sampled from it to synthesize realistic and varied paint strokes. Our results indicate that a recurrent probabilistic model is a powerful tool for capturing stochastic effects from small video datasets.

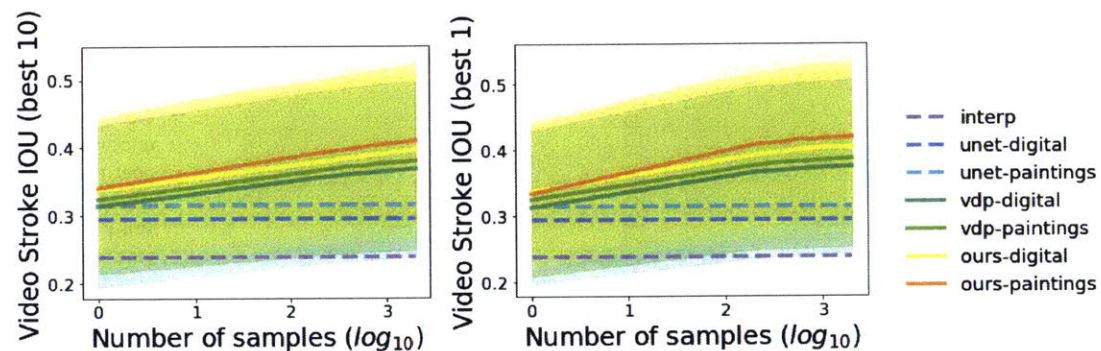
Our work in this chapter presents a first look at how to tackle this video synthesis problem. Here, we present a road map for addressing limitations of our approach, with the goal of formulate a more complete solution.



(a) Video similarity by L1 distance (lower is better)

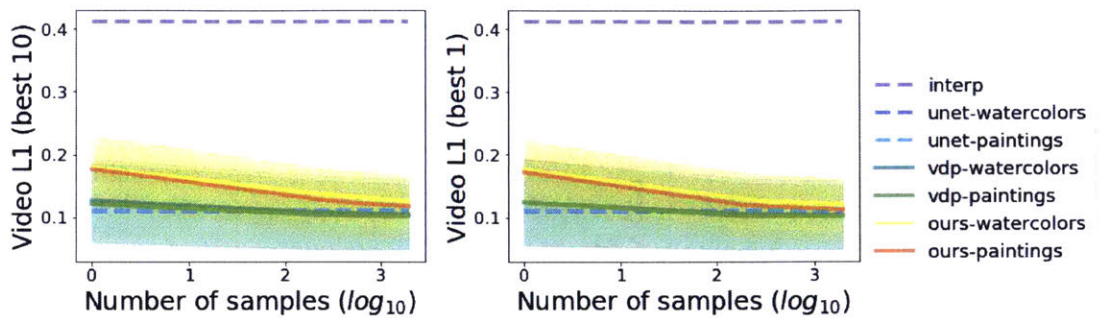


(b) Video similarity by LPIPS [187] (lower is better)

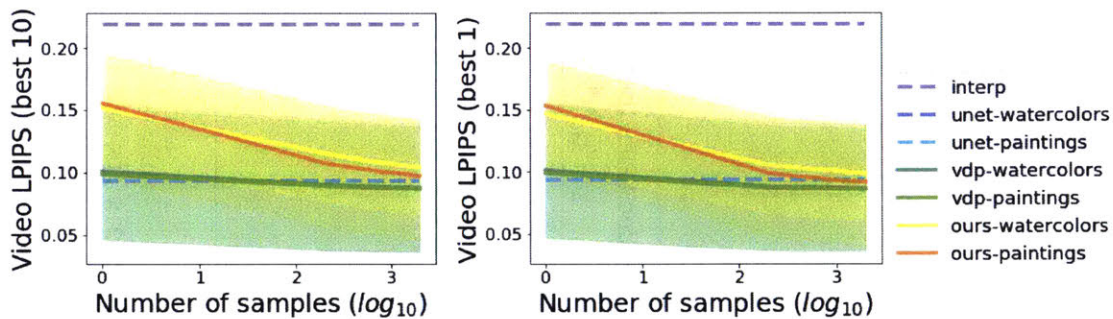


(c) Video similarity by stroke IOU (higher is better)

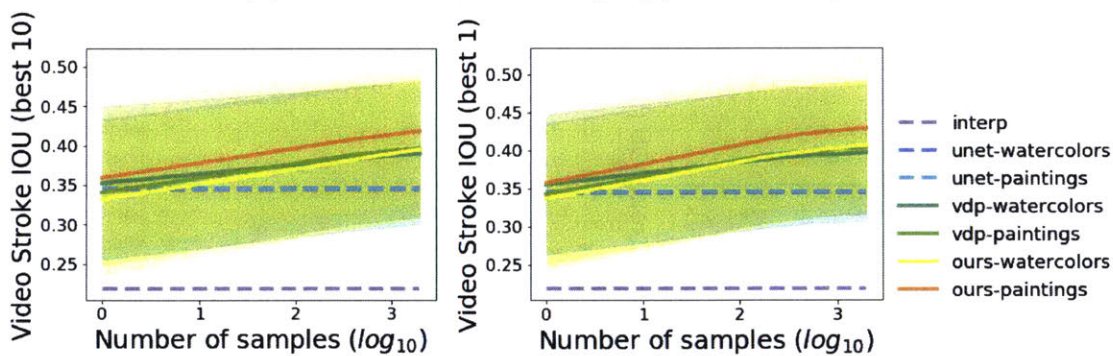
Figure 4.20: Best video similarity (across multiple samples) compared to the digital paintings test set. We show the average similarity across the top 10 best sample videos (left) and the single best sample video (right) compared to each test video. Dotted lines indicate deterministic methods. We show standard deviations as shaded regions for our method, as well as the stochastic baseline *vdp*. Our methods make predictions that are closer to the true time lapse video as we increase the number of samples.



(a) Video similarity by L1 distance (lower is better)



(b) Video similarity by LPIPS [187] (lower is better)



(c) Video similarity by stroke IOU (higher is better)

Figure 4.21: Best video similarity (across multiple samples) compared to the watercolors paintings test set. We show the average similarity across the top 10 best sample videos (left) and the single best sample video (right) compared to each test video. Dotted lines indicate deterministic methods. We show standard deviations as shaded regions for our method, as well as the stochastic baseline *vdp*. Our methods make predictions that are closer to the true time lapse video as we increase the number of samples.

Method	Best video similarity (top 10)	
	L1	LPIPS
interp	0.41 (0.11)	0.22 (0.04)
unet-watercolors	0.11 (0.07)	0.09 (0.05)
unet-paintings	0.11 (0.07)	0.09 (0.05)
vdp-watercolors	0.10 (0.06)	0.09 (0.05)
vdp-paintings	0.10 (0.05)	0.09 (0.05)
ours-watercolors	0.13 (0.04)	0.10 (0.04)
ours-paintings	0.12 (0.05)	0.10 (0.05)

Table 4.3: We compare 2000 samples from our predicted video distributions to the videos in our watercolor paintings test set.

Patch-based approach. We demonstrated our method on 50×50 patches from each piece of art. A clear next step would be to adapt this method to synthesize time lapse videos of full paintings. However, our method is designed to always synthesize noticeable strokes within each patch. Naively applying it to every patch in a painting would likely produce unrealistic progressions where every patch changes in each time step. Capturing the movement of the painter’s attention throughout the scene will likely be crucial to the realism of a time lapse of a painting. This might be implemented using an explicit attention mechanism, such as predicting a probability map of which object the artist will focus on next, and only synthesizing strokes within that object. A multi-scale critic [127, 169] might also be useful for encouraging realistic changes across coarse and fine spatial scales.

Limited evaluation of model design. We presented qualitative evaluations of our training schemes in Section 4.6.1. A more complete analysis should include ablation studies of these training schemes on the real painting datasets as well. It would also be interesting to examine other design decisions such as modeling paint strokes δ_t instead of frames x_t , or the choice of the number of latent dimensions.

Limited evaluation of distributions. Many of the evaluations presented in this chapter focus on select samples from our learned distribution of videos (*e.g.*, Figures 4.15 and 4.16), or the closest sample to the ground truth video (*e.g.*, Figures 4.20 and 4.21). It is unclear how to quantitatively evaluate the quality of our learned video distributions, since the ground truth videos represent sparse samples from a difficult-to-define space of realistic time lapse videos. Human evaluations have been used to quantify the realism of one synthetic video distribution compared to another [165]. We should also use a human study to compare the realism of videos *randomly* sampled from our method to those generated by baseline methods.

Limited evaluation metrics. Although our qualitative results in Figures 4.15 and 4.16 show that our method produces strokes with a distinctly different visual quality from the

baselines, the difference is not as pronounced in our quantitative metrics. Developing representative metrics is an active area of research for image synthesis [72]. A human study (as described above) might be useful for quantifying these differences.

Discussion and Conclusion

In this thesis, we tackled the challenge of learning from small image and video datasets. In Chapters 2 and 3, we focused on learning to synthesize images for the downstream tasks of improving few-shot object classification and one-shot medical image segmentation. In contrast to hand-engineered data augmentation transformations that synthesize examples with limited realism, we *learned* a distribution of transformations from existing data. We applied sampled transformations to existing examples, creating realistic and varied new labeled examples. We showed that training supervised systems with our synthesized examples improves classification and segmentation performance compared to existing data augmentation techniques.

In Chapter 4, we presented a novel image synthesis task: synthesizing time lapse videos depicting the creation of paintings. We collected a small dataset of a few hundred time lapse videos of digital and watercolor paintings being created, with each video representing a single example of the myriad ways to paint a scene. We presented a recurrent model along with a novel training strategy, and showed that our model could be used to synthesize plausible visual stories of how a painting might have been created.

A major contribution of this thesis is highlighting the usefulness of learned transformations for applied image synthesis tasks, particularly when training examples are scarce. We showed how to learn transformations from existing data to avoid the costly process of hand-engineering data augmentation transformations. At first glance, this proposal seems circular: how can we train a model to produce transformations when we don't have enough data to train a supervised classification or segmentation system? The answer, as we describe below, is that transformation models can be easier to train.

There are several key advantages to learning transformations that enable learning from limited data. The first is that some transformations are generalizable across examples. In Chapter 2, we learned rotations and lighting changes. In the context of classifying Magic: The Gathering cards, these transformations are applicable to all examples, regardless of their class label. Secondly, the space of transformations might require less application- and dataset-specific expertise, and can be easier to define. For example, while we might have a hard time describing the space of all Magic: The Gathering cards we would like to classify, we know that photographs of the same card can differ from each other by spatial translations and rotations, as well as lighting intensity and color. We incorporate this outside knowledge of the classification task

into our model using constraints on the space of transformations: we encourage our model to learn smooth flow fields to represent rotations, and smooth color changes to represent changes in lighting. The third advantage of transformations is that they do not always need to be learned from labeled examples. In Chapter 2, we used training pairs that belonged to the same class, which requires class labels. However, in Chapter 3, we demonstrated how to learn dense flow fields and intensity changes from un-segmented examples.

A theme throughout this work is the synthesis of images using transformations. Rather than directly outputting synthetic images, our models output transformation functions that are then applied to existing images. This approach has two major advantages compared to direct synthesis. Firstly, using transformations reduces the burden on the model to accurately capture details of the input image, helping to preserve semantic content (which is crucial in data augmentation), and image sharpness (which is helpful for creating visually pleasing art). Secondly, transformations provide an avenue for imposing interpretable constraints on the space of synthesized images. For instance, in Chapter 3, by encouraging our model to learn smooth flow fields, we synthesized brain MRI scans and label maps that are anatomically consistent.

In each project we presented, it was important to synthesize examples that were realistic and diverse. We synthesized realistic examples using transformations that we learned from real examples. This is in contrast to most existing data augmentation methods that use simple, hand-engineered transformations. These transformation functions can be insufficient for simulating realistic effects such as anatomical differences between MRI scans, or realistic lighting on objects. We demonstrated several techniques for synthesizing diverse examples, such as decomposing transformations into independent spatial and appearance components that we could mix-and-match (Chapters 2 and 3) and probabilistic modeling of distributions of transformations (Chapters 2 and 4). We hope this thesis inspires future work that combines human knowledge with data-driven models to extend the power of machine learning to less-explored applications.

Bibliography

- [1] Martín Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süssstrunk. Slic superpixels compared to state-of-the-art super-pixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.
- [3] Zeynettin Akkus, Issa Ali, Jiri Sedlar, Timothy L Kline, Jay P Agrawal, Ian F Parney, Caterina Giannini, and Bradley J Erickson. Predicting 1p19q chromosomal deletion of low-grade gliomas from mr images using deep learning. *arXiv preprint arXiv:1611.06939*, 2016.
- [4] Zeynettin Akkus, Alfia Galimzianova, Assaf Hoogi, Daniel L Rubin, and Bradley J Erickson. Deep learning for brain mri segmentation: state of the art and future directions. *Journal of digital imaging*, 30(4):449–459, 2017.
- [5] Inc. Amazon Mechanical Turk. Amazon mechanical turk: Overview, 2005.
- [6] Ryoichi Ando and Reiji Tsuruno. Segmental brush synthesis with stroke images. 2010.
- [7] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [8] J. Ashburner and K. Friston. Voxel-based morphometry-the methods. *Neuroimage*, 11:805–821, 2000.
- [9] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.
- [10] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008.

- [11] Caroline Baillard, Pierre Hellier, and Christian Barillot. Segmentation of brain 3d mr images using level sets and dense registration. *Medical image analysis*, 5(3):185–194, 2001.
- [12] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46:1–21, 1989.
- [13] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- [14] Guha Balakrishnan, Adrian V. Dalca, Amy Zhao, John V. Gutttag, Fredo Durand, and William T Freeman. Visual deprojection: Probabilistic recovery of collapsed dimensions. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [15] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Gutttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8340–8348, 2018.
- [16] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Gutttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9252–9260, 2018.
- [17] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Gutttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 2019.
- [18] William Baxter, Yuanxin Liu, and Ming C Lin. A viscous paint model for interactive applications. *Computer Animation and Virtual Worlds*, 15(3-4):433–441, 2004.
- [19] William V Baxter and Ming C Lin. A versatile interactive 3d brush model. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 319–328. IEEE, 2004.
- [20] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.
- [21] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.
- [22] Daniel Berio, Sylvain Calinon, and Frederic Fol Leymarie. Learning dynamic graffiti strokes with a compliant robot. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3981–3986. IEEE, 2016.

- [23] Volker Blanz, Michael J Tarr, and Heinrich H Bülthoff. What object attributes determine canonical views? *Perception*, 28(5):575–599, 1999.
- [24] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.
- [25] Xudong Cao, David Wipf, Fang Wen, Genquan Duan, and Jian Sun. A practical transfer learning algorithm for face verification. In *Proceedings of the IEEE international conference on computer vision*, pages 3208–3215, 2013.
- [26] Zhili Chen, Byungmoon Kim, Daichi Ito, and Huamin Wang. Wetbrush: Gpu-based 3d painting simulation at the bristle level. *ACM Transactions on Graphics (TOG)*, 34(6):200, 2015.
- [27] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.
- [28] François Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015.
- [29] Nelson S-H Chu and Chiew-Lan Tai. Moxi: real-time ink dispersion in absorbent paper. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 504–511. ACM, 2005.
- [30] Cybèle Ciofolo and Christian Barillot. Atlas-based segmentation of 3d cerebral structures with competitive level sets and fuzzy control. *Medical image analysis*, 13(3):456–470, 2009.
- [31] Daniel Coelho de Castro and Ben Glocker. Nonparametric density flows for mri intensity normalisation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 206–214, 09 2018.
- [32] Lynn A Cooper and Roger N Shepard. Chronometric studies of the rotation of mental images. In *Visual information processing*, pages 75–176. Elsevier, 1973.
- [33] Tim F Cootes and Christopher J Taylor. Statistical models of appearance for medical image analysis and computer vision. In *Medical Imaging 2001: Image Processing*, volume 4322, pages 236–249. International Society for Optics and Photonics, 2001.
- [34] Timothy F Cootes, C Beeston, Gareth J Edwards, and Christopher J Taylor. A unified framework for atlas matching using active appearance models. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 322–333. Springer, 1999.

- [35] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):681–685, 2001.
- [36] Daniel Crispell, Octavian Biris, Nate Crosswhite, Jeffrey Byrne, and Joseph L Mundy. Dataset augmentation for pose and lighting invariant face recognition. *arXiv preprint arXiv:1704.04326*, 2017.
- [37] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.
- [38] Alexander Dagley, Molly LaPoint, Willem Huijbers, Trey Hedden, Donald G McLaren, Jasmeer P Chatwal, Kathryn V Papp, Rebecca E Amariglio, Deborah Blacker, Dorene M Rentz, et al. Harvard aging brain study: dataset and accessibility. *NeuroImage*, 144:255–258, 2017.
- [39] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 729–738. Springer, 2018.
- [40] Adrian V Dalca, John Guttag, and Mert R Sabuncu. Anatomical priors in convolutional networks for unsupervised biomedical segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9290–9299, 2018.
- [41] Benoit M Dawant, Steven L Hartmann, J-P Thirion, Frederik Maes, Dirk Vandermeulen, and Philippe Demaerel. Automatic 3-d segmentation of internal structures of the head in mr images using a combination of similarity and free-form transformations. i. methodology and validation on normal subjects. *IEEE transactions on medical imaging*, 18(10):909–916, 1999.
- [42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [43] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [44] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [45] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018.

- [46] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666, 2016.
- [47] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2016.
- [48] Zach Eaton-Rosen, Felix Bragman, Sebastien Ourselin, and M Jorge Cardoso. Improving data augmentation for medical image segmentation. In *International Conference on Medical Imaging with Deep Learning*, 2018.
- [49] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018.
- [50] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- [51] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.
- [52] Martin A Frost and Rainer Goebel. Measuring structural–functional correspondence: spatial variability of specialised brain regions after macro-anatomical alignment. *Neuroimage*, 59(2):1369–1381, 2012.
- [53] Pierre-Antoine Ganaye, Michaël Sdika, and Hugues Benoit-Cattin. Semi-supervised learning for segmentation under semantic constraint. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 595–602. Springer, 2018.
- [54] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by back-propagation. *arXiv preprint arXiv:1409.7495*, 2014.
- [55] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.
- [56] Randy L Gollub et al. The mcic collection: a shared repository of multi-modal, multi-site brain image data from a clinical investigation of schizophrenia. *Neuroinformatics*, 11(3):367–388, 2013.
- [57] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.

- [58] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [59] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.
- [60] Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pages 342–350, 2016.
- [61] Pierre Hellier and Christian Barillot. A hierarchical parametric algorithm for deformable multimodal image registration. *Computer Methods and Programs in Biomedicine*, 75(2):107–115, 2004.
- [62] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460. ACM, 1998.
- [63] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.
- [64] Avram J Holmes et al. Brain genomics superstruct project initial data release with structural, functional, and behavioral measures. *Scientific data*, 2, 2015.
- [65] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Deep transfer metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 325–333, 2015.
- [66] Fay Huang, Bo-Hui Wu, and Bo-Ru Huang. Synthesis of oil-style paintings. In *Pacific-Rim Symposium on Image and Video Technology*, pages 15–26. Springer, 2015.
- [67] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.
- [68] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv preprint arXiv:1704.04086*, 2017.

- [69] Zeshan Hussain, Francisco Gimenez, Darvin Yi, and Daniel Rubin. Differential data augmentation techniques for medical imaging classification tasks. In *AMIA Annual Symposium Proceedings*, volume 2017, page 979. American Medical Informatics Association, 2017.
- [70] Juan Eugenio Iglesias and Mert R Sabuncu. Multi-atlas segmentation of biomedical images: a survey. *Medical image analysis*, 24(1):205–219, 2015.
- [71] Savage Interactive. *Procreate Artists’ Handbook*. Savage, 2016.
- [72] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.
- [73] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [74] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.
- [75] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proc. CVPR*, volume 1, 2017.
- [76] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.
- [77] Pierre-Marc Jodoin, Emric Epstein, Martin Granger-Piché, and Victor Ostromoukhov. Hatching by example: a statistical approach. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 29–36. ACM, 2002.
- [78] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [79] Michael J Jones and Tomaso Poggio. Multidimensional morphable models: A framework for representing and matching object classes. *International Journal of Computer Vision*, 29(2):107–131, 1998.
- [80] Thomas Joyce, Agisilaos Chartsias, and Sotirios A Tsafaris. Deep multi-class segmentation without ground-truth labels. 2018.

- [81] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.
- [82] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
- [83] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [84] Salman H Khan, Munawar Hayat, and Nick Barnes. Adversarial training of variational auto-encoders for high fidelity image generation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1312–1320. IEEE, 2018.
- [85] Mikyung Kim and Hyun Joon Shin. An example-based approach to synthesize artistic strokes using graphs. In *Computer Graphics Forum*, volume 29, pages 2145–2152. Wiley Online Library, 2010.
- [86] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
- [87] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [88] Arno Klein and Joy Hirsch. Mindboggle: a scatterbrained approach to automate brain labeling. *NeuroImage*, 24(2):261–280, 2005.
- [89] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.
- [90] Julian Krebs et al. Robust non-rigid registration through agent-based action learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 344–352. Springer, 2017.
- [91] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [92] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

- [93] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [94] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.
- [95] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [96] Kelvin K Leung, Matthew J Clarkson, Jonathan W Bartlett, Shona Clegg, Clifford R Jack Jr, Michael W Weiner, Nick C Fox, Sébastien Ourselin, Alzheimer’s Disease Neuroimaging Initiative, et al. Robust atrophy rate measurement in alzheimer’s disease using multi-site serial mri: tissue-specific intensity normalization and parameter selection. *Neuroimage*, 50(2):516–523, 2010.
- [97] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. Semantic-aware gradgan for virtual-to-real urban scene adaption. *arXiv preprint arXiv:1801.01726*, 2018.
- [98] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [99] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017.
- [100] Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, pages 301–320, 2007.
- [101] Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam Finkelstein. Real-brush: painting with examples of physical media. *ACM Transactions on Graphics (TOG)*, 32(4):117, 2013.
- [102] Michal Lukáč, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Šỳkora. Brushables: Example-based edge-aware directional texture painting. In *Computer Graphics Forum*, volume 34, pages 257–267. Wiley Online Library, 2015.
- [103] Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

- [104] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- [105] Daniel S Marcus et al. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.
- [106] Kenneth Marek et al. The parkinson progression marker initiative. *Progress in neurobiology*, 95(4):629–635, 2011.
- [107] Adriana Di Martino et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.
- [108] Iacopo Masi, Anh Tuan Tran, Tal Hassner, Jatuporn Toy Leksut, and Gérard Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision*, pages 579–596. Springer, 2016.
- [109] Michael Mathieu, Camille Couprie, and Yann Lecun. Deep multi-scale video prediction beyond mean square error. 11 2016.
- [110] Jacqueline Metzler and Roger N Shepard. Transformational studies of the internal representation of three-dimensional objects. 1974.
- [111] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1418, 2015.
- [112] Michael P Milham et al. The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Frontiers in systems neuroscience*, 6:62, 2012.
- [113] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.
- [114] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [115] Steven C Mitchell, Johan G Bosch, Boudewijn PF Lelieveldt, Rob J Van der Geest, Johan HC Reiber, and Milan Sonka. 3-d active appearance models: segmentation of cardiac mr and ultrasound images. *IEEE transactions on medical imaging*, 21(9):1167–1178, 2002.

- [116] Pim Moeskops, Max A Viergever, Adriëne M Mendrik, Linda S de Vries, Manon JNL Benders, and Ivana Išgum. Automatic segmentation of mr brain images with a convolutional neural network. *IEEE transactions on medical imaging*, 35(5):1252–1261, 2016.
- [117] Santiago E Montesdeoca, Hock Soon Seah, Pierre Bénard, Romain Vergne, Joëlle Thollot, Hans-Martin Rall, and Davide Benvenuti. Edge-and substrate-based effects for watercolor stylization. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 2. ACM, 2017.
- [118] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6673–6683, 2017.
- [119] Susanne G Mueller et al. Ways toward an early diagnosis in alzheimer’s disease: the alzheimer’s disease neuroimaging initiative (adni). *Alzheimer’s & Dementia*, 1(1):55–66, 2005.
- [120] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018.
- [121] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.
- [122] Wizards of the Coast LLC. Gatherer: The magic card database, 2017.
- [123] Americo Oliveira, Sérgio Pereira, and Carlos A Silva. Augmenting data when training a cnn for retinal vessel segmentation: How to warp? In *Bioengineering (ENBENG), 2017 IEEE 5th Portuguese Meeting on*, pages 1–4. IEEE, 2017.
- [124] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.
- [125] Frederik Pahde, Patrick Jähnichen, Tassilo Klein, and Moin Nabi. Cross-modal hallucination for few-shot fine-grained recognition. *arXiv preprint arXiv:1806.05147*, 2018.
- [126] Stephen Palmer. Canonical perspective and the perception of objects. *Attention and performance*, pages 135–151, 1981.
- [127] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

- [128] Brian Patenaude, Stephen M Smith, David N Kennedy, and Mark Jenkinson. A bayesian model of shape and appearance for subcortical brain segmentation. *Neuroimage*, 56(3):907–922, 2011.
- [129] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.
- [130] Vaclav Potesil, Timor Kadir, Günther Platsch, and Michael Brady. Personalized graphical models for anatomical landmark localization in whole-body medical images. *International Journal of Computer Vision*, 111(1):29–49, 2015.
- [131] J Rademacher, U Bürgel, Stefan Geyer, T Schormann, A Schleicher, H-J Freund, and Karl Zilles. Variability and asymmetry in the human precentral motor system: a cytoarchitectonic and myeloarchitectonic brain mapping study. *Brain*, 124(11):2232–2258, 2001.
- [132] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [133] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alexei A Efros, and Sergey Levine. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373*, 2018.
- [134] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.
- [135] Alexander J Ratner, Henry R Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. *arXiv preprint arXiv:1709.01643*, 2017.
- [136] Marc-Michel Rohé et al. Svf-net: Learning deformable image registration using shape matching. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 266–274. Springer, 2017.
- [137] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [138] Holger R Roth, Christopher T Lee, Hoo-Chang Shin, Ari Seff, Lauren Kim, Jianhua Yao, Le Lu, and Ronald M Summers. Anatomy-specific classification of medical images using deep convolutional nets. *arXiv preprint arXiv:1504.04003*, 2015.

- [139] Holger R Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim B Turkbey, and Ronald M Summers. Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 556–564. Springer, 2015.
- [140] Abhijit Guha Roy, Sailesh Conjeti, Debdoot Sheet, Amin Katouzian, Nassir Navab, and Christian Wachinger. Error corrective boosting for learning fully convolutional networks with limited data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 231–239. Springer, 2017.
- [141] D. Rueckert et al. Nonrigid registration using free-form deformation: Application to breast mr images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999.
- [142] Mert R Sabuncu, BT Thomas Yeo, Koen Van Leemput, Bruce Fischl, and Polina Golland. A generative model for image segmentation based on label fusion. *IEEE transactions on medical imaging*, 29(10):1714–1729, 2010.
- [143] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.
- [144] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.
- [145] D. Shen and C. Davatzikos. Hammer: Hierarchical attribute matching mechanism for elastic registration. *IEEE Transactions on Medical Imaging*, 21(11):1421–1439, 2002.
- [146] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 6, 2017.
- [147] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.
- [148] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [149] John G Sled, Alex P Zijdenbos, and Alan C Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE transactions on medical imaging*, 17(1):87–97, 1998.

- [150] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.
- [151] Hessam Sokooti et al. Nonrigid image registration using multi-scale 3d convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 232–239. Springer, 2017.
- [152] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.
- [153] Ramesh Sridharan, Adrian V Dalca, Kaitlin M Fitzpatrick, Lisa Cloonan, Allison Kanakis, Ona Wu, Karen L Furie, Jonathan Rosand, Natalia S Rost, and Polina Golland. Quantification and analysis of large multimodal clinical image studies: Application to stroke. In *International Workshop on Multimodal Brain Image Analysis*, pages 18–30. Springer, 2013.
- [154] Martin Styner, Christian Brechbuhler, G Szckely, and Guido Gerig. Parametric estimate of intensity inhomogeneities applied to mri. *IEEE Trans. Med. Imaging*, 19(3):153–165, 2000.
- [155] Adobe Systems. *Adobe Photoshop CC manual*. Adobe Systems, 2019.
- [156] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [157] Michael J Tarr and Steven Pinker. Mental rotation and orientation-dependence in shape recognition. *Cognitive psychology*, 21(2):233–282, 1989.
- [158] Patrick Tresset and Frederic Fol Leymarie. Portrait drawing by paul the robot. *Computers & Graphics*, 37(5):348–363, 2013.
- [159] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3899–3908, 2016.
- [160] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.
- [161] David C Van Essen and Donna L Dierker. Surface-based and probabilistic atlases of primate cerebral cortex. *Neuron*, 56(2):209–225, 2007.

- [162] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017.
- [163] Graham Vincent, Gwenael Guillard, and Mike Bowes. Fully automatic segmentation of the prostate using active appearance models. *MICCAI Grand Challenge: Prostate MR Image Segmentation*, 2012, 2012.
- [164] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.
- [165] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.
- [166] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.
- [167] Hongzhi Wang, Jung W Suh, Sandhitsu R Das, John B Pluta, Caryne Craige, and Paul A Yushkevich. Multi-atlas segmentation with joint label fusion. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):611–623, 2013.
- [168] Miaoyi Wang, Bin Wang, Yun Fei, Kanglai Qian, Wenping Wang, Jiating Chen, and Jun-Hai Yong. Towards photo watercolorization with artistic verisimilitude. *IEEE transactions on visualization and computer graphics*, 20(10):1451–1460, 2014.
- [169] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.
- [170] Der-Lor Way and Zen-Chung Shih. The Synthesis of Rock Textures in Chinese Landscape Painting. *Computer Graphics Forum*, 2001.
- [171] William M Wells, W Eric L Grimson, Ron Kikinis, and Ferenc A Jolesz. Adaptive segmentation of mri data. *IEEE transactions on medical imaging*, 15(4):429–442, 1996.
- [172] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. Optical flow guided tv-l 1 video interpolation and restoration. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 273–286. Springer, 2011.

- [173] Alex Wong and Alan L Yuille. One shot learning via compositions of meaningful patches. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1197–1205, 2015.
- [174] Andrew T Woods, Allison Moore, and Fiona N Newell. Canonical views in haptic object perception. *Perception*, 37(12):1867–1878, 2008.
- [175] Jun Xing, Hsiang-Ting Chen, and Li-Yi Wei. Autocomplete painting repetitions. *ACM Transactions on Graphics (TOG)*, 33(6):172, 2014.
- [176] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE, 2018.
- [177] Fei Xu and Joshua B Tenenbaum. Word learning as bayesian inference. *Psychological review*, 114(2):245, 2007.
- [178] Songhua Xu, Min Tang, Francis Lau, and Yunhe Pan. A solid model based virtual hairy brush. In *Computer Graphics Forum*, volume 21, pages 299–308. Wiley Online Library, 2002.
- [179] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems*, pages 91–99, 2016.
- [180] Xinchun Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.
- [181] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.
- [182] Xiao Yang et al. Quicksilver: Fast predictive image registration—a deep learning approach. *NeuroImage*, 158:378–396, 2017.
- [183] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. Pixel-level domain transfer. In *European Conference on Computer Vision*, pages 517–532. Springer, 2016.
- [184] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *European conference on computer vision*, pages 127–140. Springer, 2010.

- [185] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1235–1248, 2013.
- [186] Miaomiao Zhang, Ruizhi Liao, Adrian V Dalca, Esra A Turk, Jie Luo, P Ellen Grant, and Polina Golland. Frequency diffeomorphisms for efficient image registration. In *International conference on information processing in medical imaging*, pages 559–570. Springer, 2017.
- [187] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [188] Wenlu Zhang, Rongjian Li, Houtao Deng, Li Wang, Weili Lin, Shuiwang Ji, and Dinggang Shen. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, 108:214–224, 2015.
- [189] Yong Zhang, Weiming Dong, Chongyang Ma, Xing Mei, Ke Li, Feiyue Huang, Bao-Gang Hu, and Oliver Deussen. Data-driven synthesis of cartoon faces using different styles. *IEEE Transactions on image processing*, 26(1):464–478, 2017.
- [190] Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8543–8553, 2019.
- [191] Ming Zheng, Antoine Milliez, Markus Gross, and Robert W Sumner. Example-based brushes for coherent stylized renderings. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 3. ACM, 2017.
- [192] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.
- [193] Yipin Zhou and Tamara L. Berg. Learning temporal transformations from time-lapse videos. volume 9912, pages 262–277, 10 2016.
- [194] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.