Modeling Visual Minutiae: Gestures, Styles, and Temporal Patterns

Shiry Ginosar

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-148
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-148.html

August 13, 2020

Modeling Visual Minutiae: Gestures, Styles, and Temporal Patterns

By

Shiry Sara Ginosar

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexei A. Efros, Chair
Professor Jitendra Malik
Professor Alison Gopnik

Summer 2020

Modeling Visual Minutiae:
Gestures, Styles, and Temporal Patterns

Shiry Sara Ginosar

Abstract

Modeling Visual Minutiae: Gestures, Styles, and Temporal Patterns

by

Shiry Sara Ginosar

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Alexei A. Efros, Chair

The human visual system is highly adept at making use of the rich subtleties of the visual world such as non-verbal communication signals, style, emotion, and the fine-grained details of individuals. Computer vision systems, by contrast, excel in categorical tasks, such as classification and detection, where training often relies on single-word or simple bounding-box annotations. These simple annotations do not capture the richness of the visual world which is often hard to describe in words or localize in an image. Our current systems are thus left to only make use of the obvious, easily describable parts of the visual input. This dissertation investigates several initial directions toward modeling visual minutiae and endowing computer vision systems with rich perception.

Part I describes methods for learning directly from video data without the need for human-provided annotations. The section begins by discussing the use of multi-modal correlations between audio and motion for modeling conversational gestures—an essential part of human communication that is currently ignored by machine perception. The section then proposes a simple method for capturing the appearance details of individual people in motion, which can be used to implement a "do-as-I-do" motion-transfer application.

Part II explores ways to discover temporal visual patterns in historical data. The section begins by discussing data-mining methods in a dataset of historical high school yearbook portraits where fashion and behavioral styles change over time. The rest of the section proposes an unsupervised method to learn to disentangle the time-varying visual factors from the permanent ones in a large dataset of urban scenes.

Part III discusses one possible avenue for testing whether our man-made systems have achieved human-like rich perception by comparing their performance to that of humans on a unique dataset of abstract art.

To my beautiful children, Yonatan and Alma,
for enabling and inspiring my work
and to my fabulous husband, Michael,
for supporting me in every possible way.

'And what is the use of a book,' thought Alice, 'without pictures or conversations?'

— Lewis Carroll, *Alice in Wonderland*

Contents

List of Figures

List of Tables

Acknowledgments

I would like to start by thanking the Computer Science Division at UC Berkeley for the unbelievably amazing experience it was to spend my Ph.D. years here. It was all I ever dreamt it would be and so much more. I could not have picked a better place to be.

I would like to thank my dear advisor, Alyosha, who first took me on because he thought I was free with a long horizon NSF fellowship but ended up keeping me and supporting me through a whole Ph.D. and two children. Alyosha is a fluttering butterfly of high-level ideas, an amazingly strict and detail-oriented perceptual discriminator of pixels, and at times an incarnation of a Jewish mother. Most of all, I thank him for always being my best advocator and supporter. I thank Jitendra Malik who declined, repeatedly, to be my Ph.D. advisor, but ended up becoming so much more: a de-facto advisor, a mentor, and a friend (and recently my official postdoctoral advisor after all). The best advice he ever gave me was that I should go work for Alyosha because surely we will get along "quite well"! I thank Alison Gopnik - a never-ending source of inspiration. I thank Noah Snavely for always being a cheerful fountain of bright and concrete ideas through hours and hours of remote collaboration.

Luis von Ahn and Manuel Blum from Carnegie Mellon have made a clear contribution to my path as a researcher. They took a chance on me when I was a lowly EE undergrad, showed me that Computer Science could be cool, and taught me how to do absolutely crazy research. Most importantly, they completely changed the course of my path which eventually lead me to where I am today, finishing a Ph.D. at Berkeley. I thank Bjoern Hartmann for helping me start off my Ph.D. path and for being a wonderful manager of people and ideas. I thank Marti Hearst for her guidance and support.

A special mention goes to all the undergraduate and masters students with whom I have worked. I feel honored to have worked with them and that they allowed me to take a small part in setting their path. I could not have done this without them. In particular, I would like to thank Kate Rakelly, Sara Sachs, Caroline Chan, Amir Bar, and Andrew Liu (a full-fledged research engineer by the time we started

collaborating!). They really did make all the difference.

I would like to thank all my peers and colleagues in the Efros, Malik, and Darrell groups. Coming into the computer vision group three years into my Ph.D. would not have been possible without them. Thank you to all the people who always wanted to look at my pixels or explain the latest machine learning algorithm at length. In particular, the "original" Berkeley Efros group who became my family: Carl Doersch, Richard Zhang, Jun-Yan Zhu, and especially Tinghui Zhou with whom I collaborated closely and from whom I learned the valuable lesson of focusing on the actionable advice from one's advisor and ignoring the rest. The new kids on the block, members of the best-pod-ever: Jasmine Collins, Allan Jabri, Sasha Sax, and Ashish Kumar, who I am certain one day will draw me a cognitive map. Most importantly, I would like to thank the Malik "Indian" crew, who are still the brightest and sweetest people I have ever met in research: Bharath Hariharan, Saurabh Gupta, Abhishek Kar, Pulkit Agrawal, and Shubham Tulsiani. Much of what I now know, I learned from them. I was honored to end up in the same boat with Shubham (a.k.a. number 2—thorny on the outside and sweet on the inside) during a world pandemic while the seas boiled and dried up around us. I would not have wanted to share this experience with anyone else.

I thank all the postdoctoral students who let me apprentice with them and spent countless hours working with me closely or brainstorming in the lab: Ross Girshick, Philipp Krahenbuhl, Mathieu Aubry, Katerina Fragkiadaki, Phillip Isola, David Fouhey, Andrew Owens and Angjoo Kanazawa.

Additionally, I would like to thank my cohort of Ph.D. friends who made the long and winding road seem shorter: Valkyrie Savage, Orianna DeMasi, Mitar Milutinovic, Georgia Gkioxari, Evan Shelhamer, Dev Akhawe, Daniel Haas and many more.

Chapter 1

Introduction

Imagine your partner trying to subtly hint to you from across a crowded room that it's time to go home. How do you know exactly what they mean? Is it the pitch of their voice? A minute gesture? Perhaps a slight raise of an eyebrow? Even when talking to a stranger you can intuitively tell whether she is engaged in the conversation or evading your questions, and you can actively change your responses accordingly. Humans are subconsciously attuned to the rich subtleties of the visual world such as non-verbal social signals, style, emotion, individual particularities *etc.*, and know how to put them to good use.

Computers, in contrast, cannot yet fully perceive such rich signals. They see the world in terms of generic categories, often described by a single word. Looking at your loved one from across the room, a computer vision system could detect and segment a "person", a "table", perhaps some "chairs"—tasks where we ask simple questions for which the answer is simple to describe. But this system would fail miserably to pick up on the indescribable visual cues that humans seem to process effortlessly. This dissertation aims to move toward enabling computers to perceive the world richly, as humans do.

In this dissertation, I present an eclectic set of tasks that require rich perception. To address these tasks, I will describe models that learn from large swaths of raw data without relying on detailed human annotations, taking advantage of visual details that are overlooked by current methods.

This data-driven approach toward rich perception requires several technical innovations. In this dissertation, I will focus on a couple of techniques to get at these rich details directly from data. In Part I, I will focus on learning from video by using cross-modal supervision from audio-visual data, and using the temporal information in video data to model fine-grained details of a person's appearance in motion. In Part II, I will discuss using the internal structure of image collections

to learn disentangled representations of permanent versus time-varying things. In Part III, I will discuss one possible way to test whether our systems have achieved rich perception.

Part I: Learning Individual Styles of Motion and Appearance from Video Data Humans are active and vocal creatures. To learn a representation of people that is not merely categorical or textual but captures all their subtleties, we must model motion and audio as well as visual details. This is especially important in social interactions where speech does not occur in a void but is accompanied by conversational gestures. Researchers in psychology and linguistics have long been interested in modeling the relationship between speech and gesture. However, they lack the tools to process large amounts of observational data and rely instead on manual coding of small-scale footage of in-lab participants. Instead, in Chapter 2, I describe a data-driven approach to the study of the indescribable relationship between an individual's speech and motion. I propose the first method for learning the motion signature and style of an individual directly from a large dataset of in-the-wild videotaped monologues using *temporal cross-modal translation* from speech to gesture. The learned model captures the multi-modal essence of a speaker by taking as input only a raw audio recording of their speech and synthesizing synchronized gestures that are true to their typical motion style.

Human motion and articulation in space enable choreography and performance. While other creative pursuits such as storytelling and musical composition lend themselves to textual descriptions, a choreography does not. In Chapter 3, I set to capture the indescribable details of human performance in such a way that would also enable others to follow the same moves. There are two fundamental challenges in performing such a *"Do as I Do"* video retargeting – automatically transferring the motion from a source to a target subject. First, learning the *visual subtleties* of the target person: their lifelike appearance distribution while performing multiple different poses. Second, mapping the body structure of the source subject to that of the target. Computer graphics-based methods attempt to explicitly model these mathematically and thus may struggle to capture the high frequencies of a realistic appearance. Instead, I propose a simple and effective approach to motion retargeting by learning a video-to-video translation. This implicitly learns the appearance distribution of the target subject from recorded footage by using the quality of synthesis as a training signal.

Part II: Discovering Temporal Visual Patterns The texts of history books do not communicate the volume and richness of information offered by imagery. While an ever-increasing number of photographs capture much of this otherwise lost

historical information, presently historians are only able to analyze images manually and at a small scale. This process is tedious and subjective, preventing historians from forming hypotheses about temporal changes. Could we automatically discover visual changes over time? The standard approach of supervised classification (naming some property and then training systems to identify it) would be difficult to apply here. This is because much of what changes are indescribable and hard to annotate details. In Chapter 4, I describe an approach based on using a large historical dataset of American high school yearbook portraits spanning a century. Here, facial portraits provide a single anchor point to connect different historical times. The learning process then only has to tease out the temporally varying details from the permanent identity of the content. Using weakly-supervised methods on this data, I discover, for example, the typical fashions of each era and the increase over time in people's tendency to smile when being photographed.

Working with data where only one factor changes over time is not always possible. To generalize to broader settings, we must *learn* to disentangle intrinsic factors in an *unsupervised* fashion. In Chapter 5, I consider the *plenoptic function* [1], a classical way to capture the entire visual world across space and time. This theoretical construct represents all images taken from every possible location on Earth at every possible time through history. Unfortunately, the plenoptic function is purely a thought experiment as such a dataset would be intractable to capture and store. However, this dataset would be extremely redundant and highly compressible! Many images would depict the same view with varying illumination or the same lighting conditions at different locations. Rather than store all of the pixels, we could instead store a small number of intrinsic, disentangled factors representing scene aspects—if only we knew what those parameters were and how to decode an image from them. None of these factors lend themselves to the annotation required for supervised learning as they are indescribable and ill-defined. Instead, I propose a method that learns to disentangle two latent factors in an *unsupervised* way: time-varying illumination, and permanent geometric scene properties. The main insight is to use a large-scale outdoor dataset which varies both spatially and temporally and is extremely redundant and highly compressible: historical Google Street View Time Machine images. Here, the same locations were captured repeatedly through time across a major city. Using the learned factorization, I can generate images of the same scene under new lighting or weather conditions and realistically change the geometry of the scene by simply swapping or modifying the underlying factors.

Part III: How Should We Test For Rich Perception? Several plausible methods could be used to test whether our systems have achieved human-like rich perception. A straight forward approach could simply be to ask people to describe

what they see [2, 3] and later compare their descriptions to those produced by computer vision systems. Another approach could be to decode the internal visual representation of humans, for example from fMRI activations.

In Chapter 6, I explore a different direction to probe at one aspect of this question. Consider the many possible representations of the same object made by different visual artists. Many visually abstract depictions are still recognizable to us. However, if we compared them to a real image, we would realize how little corresponds to reality. The visual system is invariant to some things about the world—a fact that allows us to often treat a line drawing interchangeably with a real photograph [4]. In essence, the human visual system is robust to some types of deviations from the manifold of natural images. Artists often play with these deviations in their abstract-yet-recognizable depictions of objects.

In computer vision, we create models that attempt to model the natural visual world. These models are also robust to some types of deviations from the manifold of natural images. However, since we usually compare human vision and machine vision on natural data where the two intersect, we may never find out that the off-the-manifold deviations of the two models do not necessarily align. If the goal of computer vision is to mimic the *human* rich perceptual performance, then we should strive to create models that align with the human view of the world. To see whether we succeeded, we must make sure to test our models on imagery which is off the natural image manifold but is still recognizable by humans. A great way to get at such imagery is to use art. In particular, Cubist paintings depict objects from many viewpoints at once, reordering parts of the depicted objects. In this chapter, I test several popular object detection algorithms that were designed to work by detecting rearrangements of mid-level parts on this type of paintings.

Finally, I conclude in Chapter 7, and discuss possible directions for future work.

Part I

Learning Individual Styles of Motion and Appearance from Video Data

Chapter 2

Learning to Model Conversational Gestures from Audio-Visual Data

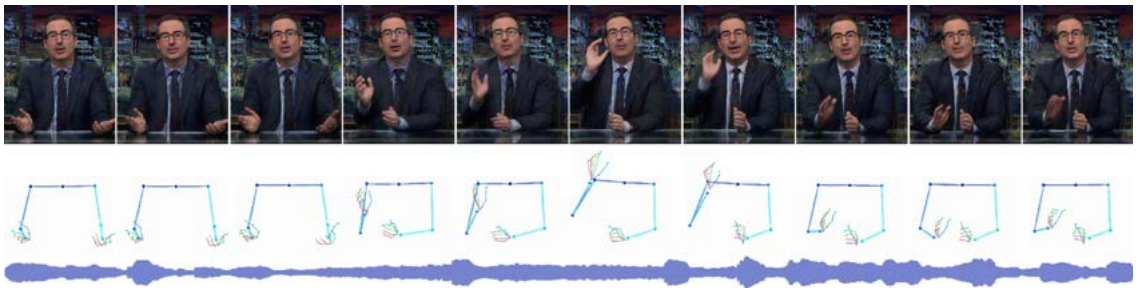


Figure 2.1: **Speech-to-gesture translation example**. In this chapter, we study the connection between conversational gesture and speech. Here, we show the result of our model that predicts gesture from audio. From the bottom upward: the input audio, arm and hand pose predicted by our model, and video frames synthesized from pose predictions using [5]. (See http://shiry.eecs.berkeley.edu/speech2gesture/ for video results.)

Human speech is often accompanied by hand and arm gestures. Given audio speech input, we generate plausible gestures to go along with the sound. Specifically, we perform cross-modal translation from "in-the-wild" monologue speech of a single speaker to their hand and arm motion (Figure 2.1). We train on unlabeled videos for which we only have noisy pseudo ground truth from an automatic pose detection system. Our proposed model significantly outperforms baseline methods in a quantitative comparison. To support research toward obtaining a computational understanding of the relationship between gesture and speech, we release a large video dataset of person-specific gestures.[1]

[1]This work was first published as *Learning Individual Styles of Conversational Gestures* in CVPR, 2019 [6].

2.1 Motivation

When we talk, we convey ideas via two parallel channels of communication—speech and gesture. These conversational, or co-speech, gestures are the hand and arm motions we spontaneously emit when we speak [7]. They complement speech and add non-verbal information that help our listeners comprehend what we say [8]. Kendon [9] places conversational gestures at one end of a continuum, with sign language, a true language, at the other end. In between the two extremes are pantomime and emblems like "Italianate", with an agreed-upon vocabulary and culture-specific meanings. A gesture can be subdivided into phases describing its progression from the speaker's rest position, through the gesture preparation, stroke, hold and retraction back to rest.

Is the information conveyed in speech and gesture correlated? This is a topic of ongoing debate. The *hand-in-hand* hypothesis claims that gesture is redundant to speech when speakers refer to subjects and objects in scenes [10]. In contrast, according to the *trade-off hypothesis*, speech and gesture are complementary since people use gesture when speaking would require more effort and vice versa [11]. We approach the question from a data-driven learning perspective and ask to what extent can we predict gesture motion from the raw audio signal of speech.

We present a method for *temporal cross-modal translation*. Given an input audio clip of a spoken statement (Figure 2.1 bottom), we generate a corresponding motion of the speaker's arms and hands which matches the style of the speaker, despite the fact that we have never seen or heard this person say this utterance in training (Figure 2.1 middle). We then use an existing video synthesis method to visualize what the speaker might have looked like when saying these words (Figure 2.1 top).

To generate motion from speech, we must learn a mapping between audio and pose. While this can be formulated as translation, in practice there are two inherent challenges to using the natural pairing of audio-visual data in this setting. First, gesture and speech are *asynchronous*, as gesture can appear before, after or during the corresponding utterance [12]. Second, this is a *multimodal* prediction task as speakers may perform different gestures while saying the same thing on different occasions. Moreover, acquiring human annotations for large amounts of video is infeasible. We therefore need to get a training signal from *pseudo ground truth* of $2D$ human pose detections on unlabeled video.

Nevertheless, we are able to translate speech to gesture in an end-to-end fashion from the raw audio to a sequence of poses. To overcome the asynchronicity issue we use a large temporal context (both past and future) for prediction. Temporal context also allows for smooth gesture prediction despite the noisy automatically-annotated pseudo ground truth. Due to multimodality, we do not expect our predicted motion

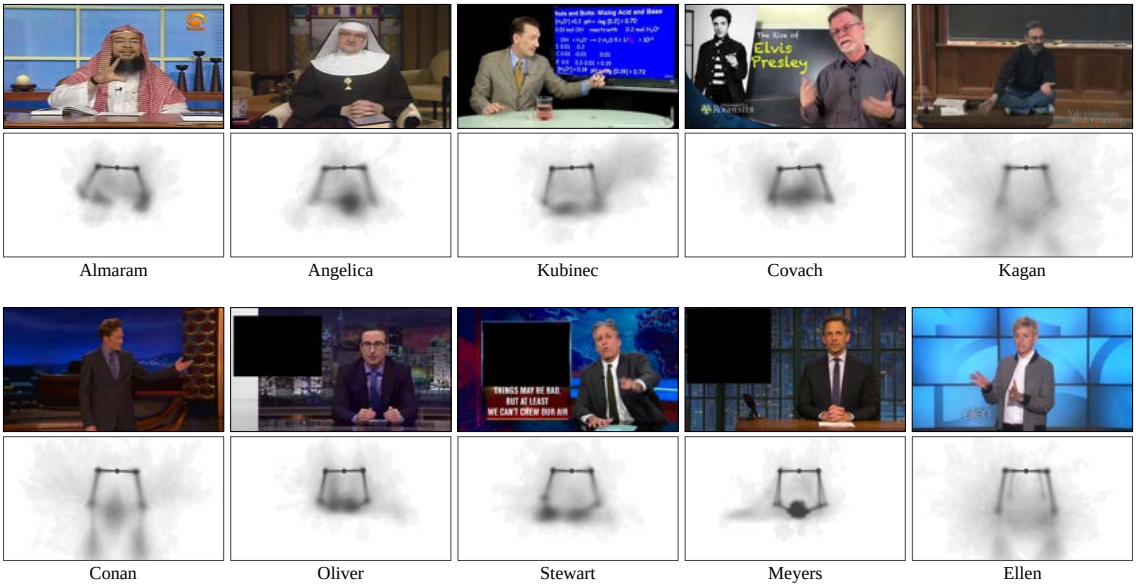


Figure 2.2: **Speaker-specific gesture dataset.** We show a representative video frame for each speaker in our dataset. Below each one is a heatmap depicting the frequency that their arms and hands appear in different spatial locations (using the skeletal representation of gestures shown in Figure 2.1). This visualization reveals the speaker's resting pose, and how they tend to move—for example, *Angelica* tends to keep her hands folded, whereas *Kubinec* frequently points towards the screen with his left hand. Note that some speakers, like *Kagan*, *Conan* and *Ellen*, alternate between sitting and standing and thus the distribution of their arm positions is bimodal.

to be the same as the ground truth. However, as this is the only training signal we have, we still use automatic pose detections for learning through regression. To avoid regressing to the mean of all modes, we apply an adversarial discriminator [13] to our predicted motion. This ensures that we produce motion that is "real" with respect to the current speaker.

Gesture is idiosyncratic [7], as different speakers tend to use different styles of motion (see Figure 2.2). It is therefore important to learn a personalized gesture model for each speaker. To address this, we present a large, 144-hour *person-specific* video dataset of 10 speakers that we make publicly available[2]. We deliberately pick a set of speakers for which we can find hours of clean single-speaker footage. Our speakers come from a diverse set of backgrounds: television show hosts, university

[2] http://shiry.eecs.berkeley.edu/speech2gesture/

lecturers and televangelists. They span at least three religions and discuss a large range of topics from commentary on current affairs through the philosophy of death, chemistry and the history of rock music, to readings in the Bible and the Qur'an.

2.2 Background

Conversational Gestures McNeill [7] divides gestures into several classes [7]: *emblematics* have specific conventional meanings (*e.g.* "thumbs up!"); *iconics* convey physical shapes or direction of movements; *metaphorics* describe abstract content using concrete motion; *deictics* are pointing gestures, and *beats* are repetitive, fast hand motions that provide a temporal framing to speech.

Many psychologists have studied questions related to co-speech gestures [7, 9] (See [14] for a review). This vast body of research has mostly relied on studying a small number of individual subjects using recorded choreographed story retelling in lab settings. Analysis in these studies was a manual process. Our goal, instead, is to study conversational gestures in the wild using a data-driven approach.

Conditioning gesture prediction on speech is arguably an ambiguous task, since gesture and speech may not be synchronous. While McNeill [7] suggests that gesture and speech originate from a common source and thus should co-occur in time according to well-defined rules, Kendon [9] suggests that gesture starts before the corresponding utterance. Others even argue that the temporal relationships between speech and gesture are not yet clear and that gesture can appear before, after or during an utterance [12].

Sign language and emblematic gesture recognition There has been a great deal of computer vision work geared towards recognizing sign language gestures from video. This includes methods that use video transcripts as a weak source of supervision [15], as well as recent methods based on CNNs [16, 17] and RNNs [18]. There has also been work that recognizes emblematic hand and face gestures [19, 20], head gestures [21], and co-speech gestures [22]. By contrast, our goal is to predict co-speech gestures from audio.

Conversational agents Researchers have proposed a number of methods for generating plausible gestures, particularly for applications with conversational agents [23]. In early work, Cassell *et al.* [24] proposed a system that guided arm/hand motions based on manually defined rules. Subsequent rule-based systems [25] proposed new ways of expressing gestures via annotations.

More closely related to our approach are methods that learn gestures from speech and text, without requiring an author to hand-specify rules. Notably, [26] synthesized gestures using natural language processing of spoken text, and Neff [27] proposed a system for making person-specific gestures. Levine *et al.* [28] learned to map acoustic prosody features to motion using a HMM. Later work [29] extended this approach to use reinforcement learning and speech recognition, combined acoustic analysis with text [30], created hybrid rule-based systems [31], and used restricted Boltzmann machines for inference [32]. Since the goal of these methods is to generate motions for virtual agents, they use lab-recorded audio, text, and motion capture. This allows them to use simplifying assumptions that present challenges for in-the-wild video analysis like ours: *e.g.*, [28] requires precise 3D pose and assumes that motions occur on syllable boundaries, and [32] assumes that gestures are initiated by an upward motion of the wrist. In contrast with these methods, our approach does not explicitly use any text or language information during training—it learns gestures from raw audio-visual correspondences—nor does it use hand-defined gesture categories: arm/hand pose are predicted directly from audio.

Visualizing predicted gestures One of the most common ways of visualizing gestures is to use them to animate a 3D avatar [29, 33, 34]. Since our work studies personalized gestures for in-the-wild videos, where 3D data is not available, we use a data-driven synthesis approach inspired by Bregler *et al.* [35]. To do this, we employ the pose-to-video method of Chan *et al.* [5], which uses a conditional generative adversarial network (GAN) to synthesize videos of human bodies from pose.

Sound and vision Aytar *et al.* [36] use the synchronization of visual and audio signals in natural phenomena to learn sound representations from unlabeled in-the-wild videos. To do this, they transfer knowledge from trained discriminative models in the visual domain, to the audio domain.

Synchronization of audio and visual features can also be used for synthesis. Langlois *et al.* [37] try to optimize for synchronous events by generating rigid-body animations of objects falling or tumbling that temporally match an input sound wave of the desired sequence of contact events with the ground plane. More recently, Shlizerman *et al.* [38] animated the hands of a 3D avatar according to input music. However, their focus was on music performance, rather than gestures, and consequently the space of possible motions was limited (*e.g.*, the zig-zag motion of a violin bow). Moreover, while music is uniquely defined by the motion that generates it (and is synchronous with it), gestures are neither unique to, nor synchronous with speech utterances.

Several works have focused on the specific task of synthesizing videos of faces speaking, given audio input. Chung *et al.* [39] generate an image of a talking face from a still image of the speaker and an input speech segment by learning a joint embedding of the face and audio. Similarly, [40] synthesizes videos of Obama saying novel words by using a recurrent neural network to map speech audio to mouth shapes and then embedding the synthesized lips in ground truth facial video. While both methods enable the creation of fake content by generating faces saying words taken from a different person, we focus on single-person models that are optimized for animating same-speaker utterances. Most importantly, generating gesture, rather than lip motion, from speech is more involved as gestures are asynchronous with speech, multimodal and person-specific.

2.3 A Speaker-Specific Gesture Dataset

We introduce a large 144-hour video dataset specifically tailored to studying speech and gesture of individual speakers in a data-driven fashion. As shown in Figure 2.2, our dataset contains in-the-wild videos of 10 gesturing speakers that were originally recorded for television shows or university lectures. We collect several hours of video per speaker, so that we can individually model each one. We chose speakers that cover a wide range of topics and gesturing styles. Our dataset contains: 5 talk show hosts, 3 lecturers and 2 televangelists.

Gesture representation and annotation We represent the speakers' pose over time using a temporal stack of 2D skeletal keypoints, which we obtain using OpenPose [41]. From the complete set of keypoints detected by OpenPose, we use the 49 points corresponding to the neck, shoulders, elbows, wrists and hands to represent gestures. Together with the video footage, we provide the skeletal keypoints for each frame of the data at a 15fps. Note, however, that these are not ground truth annotations, but a proxy for the ground truth from a state-of-the-art pose detection system.

Quality of dataset annotations All ground truth, whether from human observers or otherwise, has associated error. The pseudo ground truth we collect using automatic pose detection may have much larger error than human annotations, but it enables us to train on much larger amounts of data. Still, we must estimate whether the accuracy of the pseudo ground truth is good enough to support our quantitative conclusions. We compare the automatic pose detections to labels obtained from human observers on a subset of our training data and find that the pseudo ground

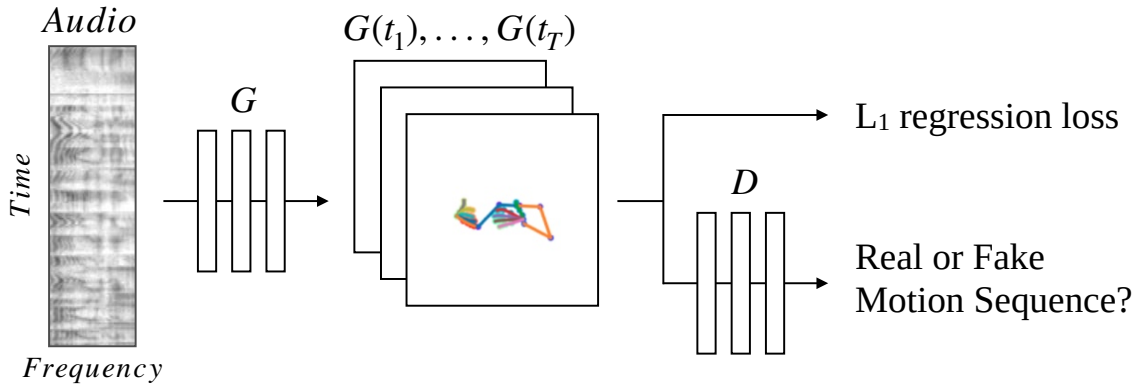


Figure 2.3: **Speech to gesture translation model.** A convolutional audio encoder downsamples the $2D$ spectrogram and transforms it to a $1D$ signal. The translation model, G, then predicts a corresponding temporal stack of $2D$ poses. L_1 regression to the ground truth poses provides a training signal, while an adversarial discriminator, D, ensures that the predicted motion is both temporally coherent and in the style of the speaker.

truth is close to human labels and that the error in the pseudo ground truth is small enough for our task.

2.4 Method

Given raw audio of speech, our goal is to generate the speaker's corresponding arm and hand gesture motion. We approach this task in two stages—first, since the only signal we have for training are corresponding audio and pose detection sequences, we learn a mapping from speech to gesture using L_1 regression to temporal stacks of $2D$ keypoints. Second, to avoid regressing to the mean of all possible modes of gesture, we employ an adversarial discriminator that ensures that the motion we produce is plausible with respect to the typical motion of the speaker.

2.4.1 Speech-to-Gesture Translation

Any realistic gesture motion must be temporally coherent and smooth. We accomplish smoothness by learning an audio encoding which is a representation of the whole utterance, taking into account the full temporal extent of the input speech, $\mathbf{s}$, and predicting the whole temporal sequence of corresponding poses, $\mathbf{p}$, at once (rather than recurrently).

Our fully convolutional network consists of an audio encoder followed by a $1D$ UNet [42, 43] translation architecture, as shown in Figure 2.3. The audio encoder takes a $2D$ log-mel spectrogram as input, and downsamples it through a series of convolutions, resulting in a $1D$ signal with the same sampling rate as our video (15 Hz). The UNet translation architecture then learns to map this signal to a temporal stack of pose vectors (see Section 2.3 for details of our gesture representation) via an L_1 regression loss:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{\mathbf{s},\mathbf{p}}[||\mathbf{p} - G(\mathbf{s})||_1]. \tag{2.1}$$

We use a UNet architecture for translation since its bottleneck provides the network with past and future temporal context, while the skip connections allow for high frequency temporal information to flow through, enabling prediction of fast motion.

2.4.2 Predicting Plausible Motion

While L_1 regression to keypoints is the only way we can extract a training signal from our data, it suffers from the known issue of regression to the mean which produces overly smooth motion. To combat the issue and ensure that we produce realistic motion, we add an adversarial discriminator [5, 43] D, conditioned on the difference of the predicted sequence of poses. i.e. the input to the discriminator is the vector $\mathbf{m} = [p_2 - p_1, \ldots, p_T - p_{T-1}]$ where p_i are $2D$ pose keypoints and T is the temporal extent of the input audio and predicted pose sequence. The discriminator D tries to maximize the following objective while the generator G (translation architecture, Section 2.4.1) tries to minimize it:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{\mathbf{m}}[\log D(\mathbf{m})] + \mathbb{E}_{\mathbf{s}}[\log(1 - G(\mathbf{s}))], \tag{2.2}$$

where s is the input audio speech segment and m is the motion derivative of the predicted stack of poses. Thus, the generator learns to produce real-seeming speaker motion while the discriminator learns to classify whether a given motion sequence is real. Our full objective is therefore:

$$\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{L_1}(G). \tag{2.3}$$

2.4.3 Implementation Details

We obtain translation invariance by subtracting (per frame) the neck keypoint location from all other keypoints in our pseudo ground truth gesture representation (section 2.3). We then normalize each keypoint (*e.g.* left wrist) across all frames by subtracting the per-speaker mean and dividing by the standard deviation. During

training, we take as input spectrograms corresponding to about 4 seconds of audio and predict 64 pose vectors, which correspond to about 4 seconds at a 15Hz frame-rate. At test time we can run our network on arbitrary audio durations. We optimize using kingmaadam [?] with a batch size of 32 and a learning rate of 10^{-4}. We train for 300K/90K iterations with and without an adversarial loss, respectively, and select the best performing model on the validation set.

2.5 Experiments

We show that our method produces motion that quantitatively outperforms several baselines, as well as a previous method that we adapt to the problem.

2.5.1 Setup

We describe our experimental setup including our baselines for comparison and evaluation metric.

Baselines

We compare our method to several other models.

Always predict the median pose Speakers spend most of their time in rest position [9], so predicting the speaker's median pose can be a high-quality baseline. For a visualization of each speaker's rest position, see Figure 2.2.

Predict a randomly chosen gesture In this baseline, we randomly select a different gesture sequence (which does not correspond to the input utterance) from the training set of the same speaker, and use this as our prediction. While we would not expect this method to perform well quantitatively, there is reason to think it would generate qualitatively appealing motion: these are real speaker gestures—the only way to tell they are fake is to evaluate how well they corresponds to the audio.

Nearest neighbors Instead of selecting a completely random gesture sequence from the same speaker, we can use audio as a similarity cue. For an input audio track, we find its nearest neighbor for the speaker using pretrained audio features, and transfer its corresponding motion. To represent the audio, we use the state-of-the-art VGGish feature embedding [44] pretrained on AudioSet [45], and use cosine distance on normalized features.

RNN-based model [38] We further compare our motion prediction to an RNN architecture proposed by Shlizerman *et al.*. Similar to us, Shlizerman *et al.* predict arm and hand motion from audio in a $2D$ skeletal keypoint space. However, while our model is a convolutional neural network with log-mel spectrogram input, theirs uses a 1-layer LSTM model that takes MFCC features (a low-dimensional, hand-crafted audio feature representation) as input. We evaluated both feature types and found that for [38], MFCC features outperform the log-mel spectrogram features on all speakers. We therefore use their original MFCC features in our experiments. For consistency with our own model, instead of measuring L_2 distance on PCA features, as they do, we add an extra hidden layer and use L_1 distance.

Ours, no GAN Finally, as an ablation, we compare our full model to the prediction of the translation architecture alone, without the adversarial discriminator.

Evaluation Metrics

Our main quantitative evaluation metric is the L_1 regression loss of the different models in comparison. We additionally report results according to the percent of correct keypoints (PCK) [46], a widely accepted metric for pose detection. Here, a predicted keypoint is defined as correct if it falls within $\alpha \max(h, w)$ pixels of the ground truth keypoint, where h and w are the height and width of the person bounding box, respectively.

We note that PCK was designed for localizing object parts, whereas we use it here for a cross-modal prediction task (predicting pose from audio). First, unlike L_1, PCK is not linear and correctness scores fall to zero outside a hard threshold. Since our goal is not to predict the ground truth motion but rather to use it as a training signal, L_1 is more suited to measuring how we perform on average. Second, PCK is sensitive to large gesture motion as the correctness radius depends on the width of the span of the speaker's arms. While [46] suggest $\alpha = 0.1$ for data with full people and $\alpha = 0.2$ for data where only half the person is visible, we take an average over $\alpha = 0.1, 0.2$.

2.5.2 Quantitative Evaluation

We compare the results of our method to the baselines using our quantitative metrics. To assess whether our results are perceptually convincing, we conduct a user study. Finally, we ask whether the gestures we predict are person-specific and whether the input speech is indeed a better predictor of motion than the initial pose of the gesture.

Model	Mey.	Oli.	Con.	Ste.	Ell.	Kag.	Kub.	Cov.	Ang.	Alm.	Avg. L1	Avg. PCK
Median	0.66	0.69	0.79	0.63	0.75	0.80	0.80	0.70	0.74	0.76	0.73	38.11
Random	0.93	1.00	1.10	0.94	1.07	1.11	1.12	1.00	1.04	1.08	1.04	26.55
NN [44]	0.88	0.96	1.05	0.93	1.02	1.11	1.10	0.99	1.01	1.06	1.01	27.92
RNN [38]	0.61	0.66	0.76	0.62	**0.71**	0.74	0.73	0.72	**0.72**	**0.75**	0.70	39.69
Ours, no GAN	**0.57**	**0.60**	**0.63**	**0.61**	**0.71**	**0.72**	**0.68**	**0.69**	0.75	0.76	**0.67**	**44.62**
Ours, GAN	0.77	0.63	0.64	0.68	0.81	0.74	0.70	0.72	0.78	0.83	0.73	41.95

Table 2.1: Quantitative results for the speech to gesture translation task using L_1 loss (lower is better) on the test set. The rightmost column is the average PCK value (higher is better) over all speakers and $\alpha = 0.1, 0.2$.

Numerical Comparison

We compare to all baselines on 2,048 randomly chosen test set intervals per speaker and display the results in Table 2.1. We see that on most speakers, our model outperforms all others, where our no-GAN condition is slightly better than the GAN one. This is expected, as the adversarial discriminator pushes the generator to snap to a single mode of the data, which is often further away from the actual ground truth than the mean predicted by optimizing L_1 loss alone. Our model outperforms the RNN-based model on most speakers. Qualitatively, we find that this baseline predicts relatively small motions on our data, which may be due to the fact that it has relatively low capacity compared to our UNet model.

Human Study

To gain insight into how synthesized gestures perceptually compare to real motion, we conducted a small-scale real vs. fake perceptual study on Amazon Mechanical Turk. We used two speakers who are always shot from the same camera viewpoint: Oliver, whose gestures are relatively dynamic and Meyers, who is relatively stationary. We visualized gesture motion using videos of skeletal wire frames. To provide participants with additional context, we included the ground truth mouth and facial keypoints of the speaker in the videos.

Participants watched a series of video pairs. In each pair, one video was produced from a real pose sequence; the other was generated by an algorithm—our model or a baseline. Participants were then asked to identify the video containing the motion that corresponds to the speech sound (we did not verify that they in fact listened to the speech while answering the question). Videos of 4 seconds or 12

seconds each of resolution 400×226 (downsampled from 910×512 in order to fit two videos side-by-side on different screen sizes) were shown, and after each pair, participants were given unlimited time to respond. We sampled 100 input audio intervals at random and predicted from them a $2D$-keypoint motion sequence using each method. Each task consisted of 20 pairs of videos and was performed by 300 different participants. Each participant was given a short training set of 10 video pairs before the start of the task, and was given feedback indicating whether they had correctly identified the ground-truth motion.

We compared all the gesture-prediction models (Section 2.5.1) and assessed the quality of each method using the rate at which its output fooled the participants. Interestingly, we found that for the dynamic speaker all methods that generate realistic motion fooled humans at similar rates. As shown in Table 2.2, our results for this speaker were comparable to real motion sequences, whether selected by an audio-based nearest neighbor approach or randomly. For the stationary speaker who spends most of the time in rest position, real motion was more often selected as there is no prediction error associated with it. While the nearest neighbor and random motion models are significantly less accurate quantitatively (Table 2.1), they are perceptually convincing because their components are realistic.

The Predicted Gestures are Person-Specific

For every speaker's speech input (Figure 2.4 rows), we predict gestures using all *other* speakers' trained models (Figure 2.4 columns). We find that on average, predicting using our model trained on a different speaker performs better numerically than predicting random motion, but significantly worse than always predicting the median pose of the input speaker (and far worse than the predictions from the model trained on the input speaker). The diagonal structure of the confusion matrix in Figure 2.4 exemplifies this.

Speech is a Good Predictor for Gesture

Seeing the success of our translation model, we ask how much does the audio signal help *when the initial pose of the gesture sequence is known.* In other words, how much can sound tell us beyond what can be predicted from motion dynamics. To study this, we augment our model by providing it the pose of the speaker directly preceding their speech, which we incorporate into the bottleneck of the UNet (Figure 2.3). We consider the following conditions: *Predict median pose*, as in the baselines above. *Predict the input initial pose*, a model that simply repeats the input initial ground-truth pose as its prediction. *Speech input*, our model. *Initial pose input*, a variation of our model in which the audio input is ablated and the network

	Oliver		Meyers	
Model	4 seconds	12 seconds	4 seconds	12 seconds
Median	12.1 ± 2.8	6.7 ± 2.0	34.0 ± 4.2	25.8 ± 3.9
Random	$\mathbf{34.2 \pm 4.0}$	$\mathbf{29.1 \pm 3.7}$	$\mathbf{40.9 \pm 4.6}$	$\mathbf{34.3 \pm 4.4}$
NN [44]	$\mathbf{36.9 \pm 3.9}$	$\mathbf{26.4 \pm 3.8}$	$\mathbf{43.5 \pm 4.5}$	$\mathbf{33.3 \pm 4.4}$
RNN [38]	18.2 ± 3.2	10.0 ± 2.5	37.5 ± 4.6	19.4 ± 3.6
Ours, no GAN	25.0 ± 3.8	19.8 ± 3.4	36.1 ± 4.3	$\mathbf{33.1 \pm 4.2}$
Ours, GAN	$\mathbf{35.4 \pm 4.0}$	$\mathbf{27.8 \pm 3.9}$	33.2 ± 4.4	22.0 ± 4.0

Table 2.2: Human study results for the speech to gesture translation task on 4 and 12-second video clips of two speakers—one dynamic (Oliver) and one relatively stationary (Meyers). As a metric for comparison, we use the percentage of times participants were fooled by the generated motions and picked them as real over the ground truth motion in a two-alternative forced choice. We found that humans were not sensitive to the alignment of speech and gesture. For the dynamic speaker, gestures with realistic motion—whether randomly selected from another video of the same speaker or generated by our GAN-based model—fooled humans at equal rates (no statistically significant difference between the bolded numbers). Since the stationary speaker is usually at rest position, real unaligned motion sequences look more realistic as they do not suffer from prediction noise like the generated ones.

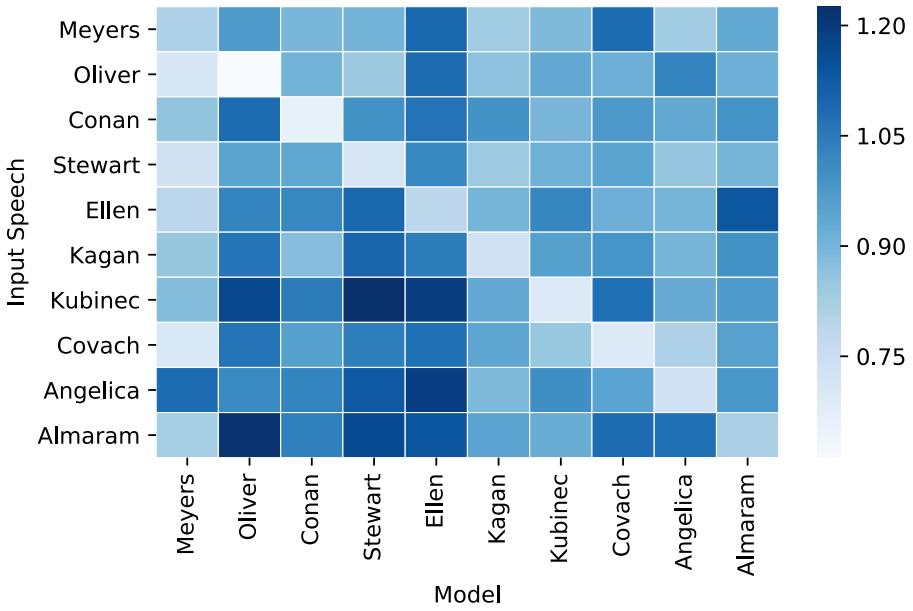


Figure 2.4: **Our trained models are person-specific.** For every speaker audio input (row) we apply all other individually trained speaker models (columns). Color saturation corresponds to L_1 loss values on a held out test set (lower is better). For each row, the entry on the diagonal is lightest as models work best using the input speech of the person they were trained on.

	Model	Avg. L_1	Avg. PCK
Pred.	Predict the median pose	0.73	38.11
	Predict the input initial pose	0.53	60.50
Input	Speech input	0.67	44.62
	Initial pose input	0.49	61.24
	Speech & initial pose input	**0.47**	**62.39**

Table 2.3: How much information does sound provide once we know the initial pose of the speaker? We see that the initial pose of the gesture sequence is a good predictor for the rest of the 4-second motion sequence (second to last row), but that adding audio improves the prediction (last row). We use both average L_1 loss (lower is better) and average PCK over all speakers and $\alpha = 0.1, 0.2$ (higher is better) as metrics of comparison. We compare two baselines and three conditions of inputs.

predicts the future pose from only an initial ground-truth pose input, and *Speech & initial pose input*, where we condition the prediction on both the speech and the initial pose.

Table 2.3 displays the results of the comparison for our model trained without the adversarial discriminator (no GAN). When comparing the *Initial pose input* and *Speech & initial pose input* conditions, we find that the addition of speech significantly improves accuracy when we average the loss across all speakers ($p < 10^{-3}$ using a two sided t-test). Interestingly, we find that most of the gains come from a small number of speakers (*e.g.* Oliver) who make large motions during speech.

2.5.3 Qualitative Results

We qualitatively compare our speech to gesture translation results to the baselines and the ground truth gesture sequences in Figure 2.5.

2.6 Discussion

Humans communicate through both sight and sound, yet the connection between these modalities remains unclear [9]. In this chapter, we proposed the task of predicting person-specific gestures from "in-the-wild" speech as a computational means of studying the connections between these communication channels. We created a large person-specific video dataset and used it to train a model for predicting

Figure 2.5: **Speech to gesture translation qualitative results.** We show the input audio spectrogram and the predicted poses overlaid on the ground-truth video for Dr. Kubinec (lecturer) and Conan O'Brien (show host).

gestures from speech. Our model outperforms other methods in an experimental evaluation.

Despite its strong performance on these tasks, our model has limitations that can be addressed by incorporating insights from other work. For instance, using audio as input has its benefits compared to using textual transcriptions as audio is a rich representation that contains information about prosody, intonation, rhythm, tone and more. However, audio does not directly encode high-level language semantics that may allow us to predict certain types of gesture (*e.g.* metaphorics), nor does it separate the speaker's speech from other sounds (*e.g.* audience laughter). Additionally, we treat pose estimations as though they were ground truth, which introduces significant amount of noise—particularly on the speakers' fingers.

We see our work as a step toward a computational analysis of conversational gesture, and opening three possible directions for further research. The first is in using gestures as a representation for video analysis: co-speech hand and arm motion make a natural target for video prediction tasks. The second is using in-the-wild gestures as a way of training conversational agents: we presented one way of visualizing gesture predictions, based on GANs [5], but, following classic work [23], these predictions could also be used to drive the motions of virtual agents. Finally, our method is one of only a handful of initial attempts to predict motion from audio. This cross-modal translation task is fertile ground for further research.

Chapter 3

Human Appearance in Motion

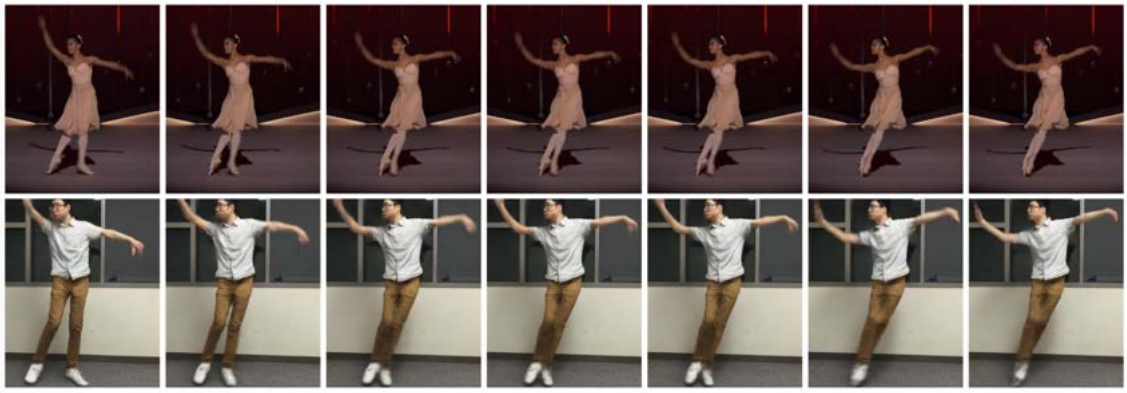


Figure 3.1: **"Do as I Do" motion transfer:** given a YouTube clip of a ballerina (top), and a video of a graduate student performing various motions, our method transfers the ballerina's performance onto the student (bottom). Video: https://youtu.be/mSaIrz8lM1U

This chapter presents a simple method for "do as I do" motion transfer: given a source video of a person dancing, we can transfer that performance to a novel (amateur) target after only a few minutes of the target subject performing standard moves. We approach this problem as video-to-video translation using pose as an intermediate representation. To transfer the motion, we extract poses from the source subject and apply the learned pose-to-appearance mapping to generate the target subject. We predict two consecutive frames for temporally coherent video results and introduce a separate pipeline for realistic face synthesis. Although our method is quite simple, it produces surprisingly compelling results (see video). This motivates us to also provide a forensics tool for reliable synthetic content detection, which is able to distinguish videos synthesized by our system from real data. In addition, we release a first-of-its-kind open-source dataset of videos that can be legally used for

training and motion transfer.[1]

3.1 Motivation

Consider the two video sequences on Figure 3.1. The top row is the input – it is a YouTube clip of a ballerina (the *source* subject) performing a sequence of motions. The bottom row is the output of our algorithm. It corresponds to frames of a different person (the *target* subject) apparently performing the same motions. The twist is that the target person never performed the same exact sequence of motions as the source, and, indeed, knows nothing about ballet. He was instead filmed performing a set of standard moves, without specific reference to the precise actions of the source. And, as is obvious from the figure, the source and the target are of different genders, have different builds, and wear different clothing.

In this chapter, we propose a simple but surprisingly effective approach for "Do as I Do" video retargeting – automatically transferring the motion from a source to a target subject. Given two videos – one of a *target* person whose appearance we wish to synthesize, and the other of a *source* subject whose motion we wish to impose onto our target person – we transfer motion between these subjects by learning a simple video-to-video translation. With our framework, we create a variety of videos, enabling untrained amateurs to spin and twirl like ballerinas, perform martial arts kicks, or dance as vibrantly as pop stars.

To transfer motion between two video subjects in a frame-by-frame manner, we must learn a mapping between images of the two individuals. Our goal is, therefore, to discover an image-to-image translation [43] between the source and target sets. However, we do not have corresponding pairs of images of the two subjects performing the same motions to supervise learning this translation. Even if both subjects perform the same routine, it is still unlikely to have an exact frame to frame pose correspondence due to body shape and motion style unique to each subject.

We observe that keypoint-based pose preserves motion signatures over time while abstracting away as much subject identity as possible and can serve as an intermediate representation between any two subjects. We therefore use pose stick figures obtained from off-the-shelf human pose detectors, such as OpenPose [41,47,48], as an intermediate representation for frame-to-frame transfer, as shown in Figure 3.2. We then learn an image-to-image translation model between pose stick figures and images of our target person. To transfer motion from source to target, we input the

[1]This work was first published as *Everybody Dance Now* in ICCV, 2019 [5].

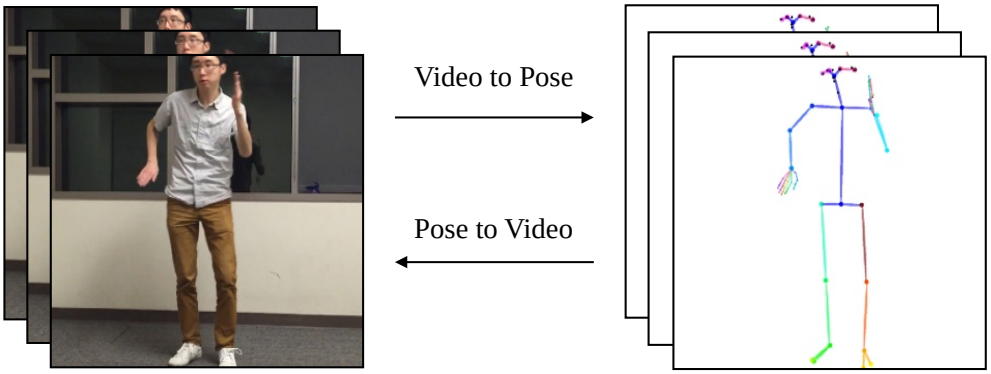


Figure 3.2: **Video to pose correspondences.** Our method creates correspondences by detecting poses in video frames (Video to Pose) and then learns to generate images of the target subject from the estimated pose (Pose to Video).

pose stick figures from the source into the trained model to obtain images of the target subject in the same pose as the source.

The central contribution of our work is a surprisingly simple method for generating compelling results on human motion transfer. We demonstrate complex motion transfer from realistic in-the-wild input videos and synthesize high-quality and detailed outputs (see Section 3.4.3 for examples). Motivated by the high quality of our results, we introduce an application for detecting if a video is real or synthesized by our method. We strongly believe that it is important for work in image synthesis to explicitly address the issue of fake detection (Section 3.5).

Furthermore, we release a two-part dataset: First, five long single-dancer videos which we filmed ourselves that can be used to train and evaluate our model, and second, a large collection of short YouTube videos that can be used for transfer and fake detection. We specifically designate the single-dancer data to be high-resolution open-source data for training motion transfer and video generation methods. The subjects whose data we release have all consented to allowing the data to be used for research purposes. For more details, see our project website https://carolineec.github.io/everybody_dance_now.

3.2 Background

Over the last two decades there has been extensive work dedicated to motion transfer. Early methods focused on creating new content by manipulating existing

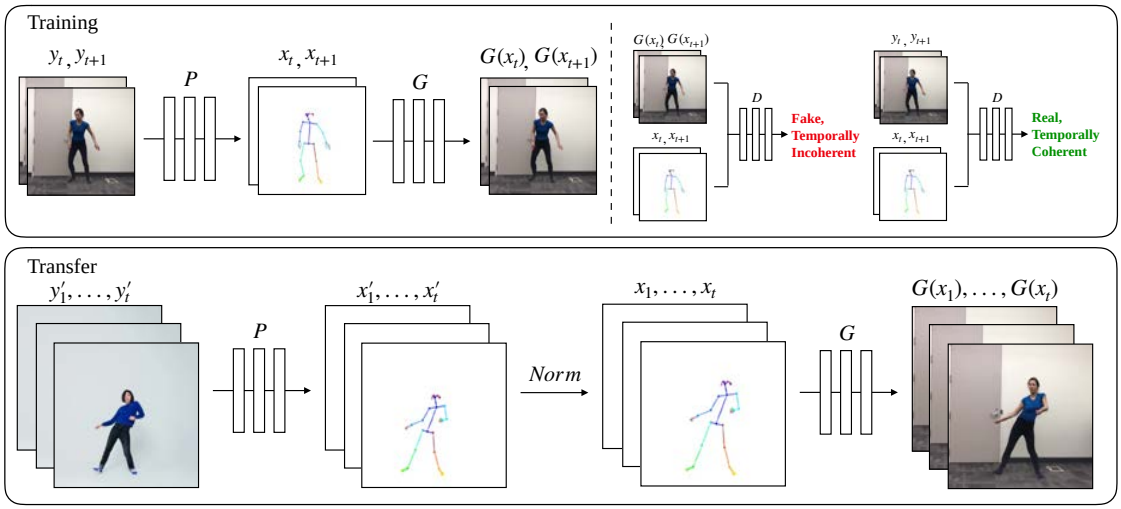


Figure 3.3: **System.**(Top) **Training**: Our model uses a pose detector P to create pose stick figures from video frames of the target subject. We learn the mapping G alongside an adversarial discriminator D which attempts to distinguish between the "real" correspondences $(x_t, x_{t+1}), (y_t, y_{t+1})$ and the "fake" sequence $(x_t, x_{t+1}), (G(x_t), G(x_{t+1}))$. (Bottom) **Transfer**: We use a pose detector P to obtain pose joints for the source person that are transformed by our normalization process $Norm$ into joints for the target person for which pose stick figures are created. Then we apply the trained mapping G.

video footage [35, 49, 50]. For example, Video Rewrite [35] creates videos of a subject saying a phrase they did not originally utter by finding frames where the mouth position matches the desired speech. Efros et al. [49] use optical flow as a descriptor to match different subjects performing similar actions allowing "Do as I do" and "Do as I say" retargeting. Classic computer graphics approaches to motion transfer attempt to perform this in 3D. Ever since the retargeting problem was proposed between animated characters [51], solutions have included the use of inverse kinematic solvers [52] and retargeting between significantly different 3D skeletons [53]. Our approach is similarly designed for in-the-wild video subjects, although we learn to synthesize novel motions rather than manipulating existing frames and we use 2D representations.

Several approaches rely on calibrated multi-camera setups to 'scan' a target actor and manipulate their motions in a new video through a fitted 3D model of the target. To obtain 3D information, Cheung et al. [54] propose an elaborate multi-view system to calibrate a personalized kinematic model, obtain 3D joint estimations, and render images of a human subject performing new motions.

Xu et al. [55] use multi-view captures of a target subject performing simple motions to create a database of images and transfer motion through a fitted 3D skeleton and corresponding surface mesh for the target. Work by Casas et al. use 4D Video Textures [56] to compactly store a layered texture representation of a scanned target person and use their temporally coherent mesh and data representation to render video of the target subject performing novel motions. In contrast, our approach explores motion transfer between 2D video subjects and avoid data calibration and lifting into 3D space.

Similarly to our method, recent works have applied deep learning for reanimation in different applications and rely on more detailed input representations. Given synthetic renderings, an interior face model, and a gaze map as input, Kim et al. [57] transfer head position and facial expressions between human subjects and render their results in detailed portrait videos. Our problem is analogous to this work except we retarget full body motion, and the inputs to our model as 2D pose stick figures as opposed to more detailed 3D representations. Similarly, Martin-Brualla et al. [58] apply neural re-rendering to enhance rendering of human motion capture for VR/AR purposes. The primary focus of this work is to render realistic humans in real time and similarly uses a deep network to synthesize their final result, but unlike our work does not address motion transfer between subjects. Villegas et al. [59] focus on retargeting motion between rigged skeletons and demonstrate reanimation in 3D characters without supervised data. Similarly, we learn to retarget motion using a skeleton-like intermediate representation, however we transfer full body motion between human subjects who are not rigged to the skeleton unlike animated

characters.

Recent methods focus on disentangling motion from appearance and synthesizing videos with novel motion [60, 61]. MoCoGAN [60] employs unsupervised adversarial training to learn this separation and generates videos of subjects performing novel motions or facial expressions. This theme is continued in Dynamics Transfer GAN [61] which transfers facial expressions from a source subject in a video onto a target person given in a static image. Similarly, we apply our representation of motion to different target subjects to generate new motions. However, in contrast to these methods we specialize on synthesizing detailed dance videos.

Modern approaches have shown success in generating detailed single images of human subjects in new poses [62–72]. Works including Ma et al. [67, 68] and Siarohin et al. [69] have introduced novel architectures and losses for this purpose. Furthermore, [70, 73] have shown pose is an effective supervisory signal for future prediction and video generation. However these works are not designed specifically for motion transfer. Rather than generating possible views of a previously unseen person from a single input image, we are interested in learning the style of a single, known person from large amounts of personalized video data and synthesizing them dancing in a detailed high-resolution video.

Concurrent with our work, [74–77] learn mappings between videos and demonstrate motion transfer between faces and from poses to body. Wang et al. [77] achieves results of similar quality to ours with a more complex method and significantly more computational resources.

Our work is made possible by recent rapid advances along two separate directions: robust pose estimation, and realistic image-to-image translation. Modern pose detection systems including OpenPose [41, 47, 48] and DensePose [78] allow for surprisingly reliable and fast pose extraction in a variety of scenarios. At the same time, the recent emergence of image-to-image translation models, pix2pix [43], CoGAN [79], UNIT [80], CycleGAN [81], DiscoGAN [82], Cascaded Refinement Networks [83], and pix2pixHD [84], have enabled high-quality single-image generation. We build upon these two building blocks by using pose detection as an intermediate representation and extending upon single-image generation to synthesize temporally-coherent, surprisingly realistic videos.

3.3 Method

Given a video of a source person and another of a target person, our goal is to generate a new video of the target enacting the same motions as the source. To accomplish this task, we divide our pipeline into three stages – pose detection, global

pose normalization, and mapping from normalized pose stick figures to the target subject. See Figure 3.3 for an overview of our pipeline. In the pose detection stage we use a pre-trained state-of-the-art pose detector to create pose stick figures given frames from the source video. The global pose normalization stage accounts for differences between the source and target body shapes and locations within the frame. Finally, we design a system to learn the mapping from the pose stick figures to images of the target person using adversarial training. Next we describe each stage of our system.

3.3.1 Pose Encoding and Normalization

Encoding body poses To encode the body pose of a subject image, we use a pre-trained pose detector P (OpenPose [41, 47, 48]) which accurately estimates 2D x, y joint coordinates. We then create a colored pose stick figure by plotting the keypoints and drawing lines between connected joints as shown in Figure 3.2.

Global pose normalization In different videos, subjects may have different limb proportions or stand closer or farther to the camera than one another. Therefore when retargeting motion between two subjects, it may be necessary to transform the pose keypoints of the source person so that they appear in accordance with the target person's body shape and location as in the **Transfer** section of Figure 3.3. We find this transformation by analyzing the heights and ankle positions for the poses of each subject and use a linear mapping between the closest and farthest ankle positions in both videos. After gathering these positions, we calculate the scale and translation for each frame based on its corresponding pose detection.

3.3.2 Pose to Video Translation

Our video synthesis method is based off of an adversarial single frame generation process presented by Wang et al. [84]. In the original conditional GAN setup, the generator network G engages in a minimax game against multi-scale discriminator $D = (D_1, D_2, D_3)$. The generator must synthesize images in order to fool the discriminator which must discern between "real" (ground truth) images and "fake" images produced by the generator. The two networks are trained simultaneously and drive each other to improve - Gr learns to synthesize more detailed images to deceive D which in turn learns differences between generated outputs and ground truth data. For our purposes, G synthesizes images of a person given a pose stick figure.

Such single-frame image-to-image translation methods are not suitable for video synthesis as they produce temporal artifacts and cannot generate the fine details

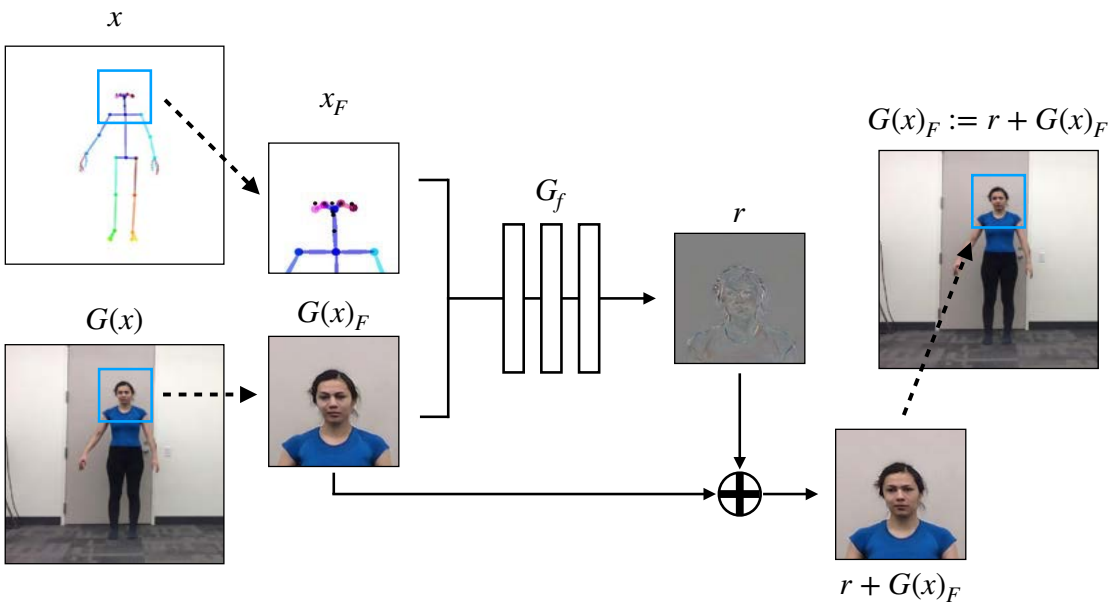


Figure 3.4: **Face GAN setup.** Residual is predicted by generator G_f and added to the original face prediction from the main generator.

important in perceiving humans in motion. We therefore add a learned model of temporal coherence as well as a module for high resolution face generation.

Temporal smoothing To create video sequences, we modify the single image generation setup to enforce temporal coherence between adjacent frames as shown in Figure 3.3 (top right). Instead of generating individual frames, we predict two consecutive frames where the first output $G(x_{t-1})$ is conditioned on its corresponding pose stick figure x_{t-1} and a zero image z (a placeholder since there is no previously generated frame at time $t-2$). The second output $G(x_t)$ is conditioned on its corresponding pose stick figure x_t and the first output $G(x_{t-1})$. Consequently, the discriminator is now tasked with determining both the difference in realism and temporal coherence between the "fake" sequence $(x_{t-1}, x_t, G(x_{t-1}), G(x_t))$ and "real" sequence $(x_{t-1}, x_t, y_{t-1}, y_t)$. The temporal smoothing changes are now reflected in the updated GAN objective

$$\begin{aligned}\mathcal{L}_{\text{smooth}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x_t, x_{t+1}, y_t, y_{t+1})] \\ +\mathbb{E}_x[\log(1 - D(x_t, x_{t+1}, G(x_t), G(x_{t+1}))]\end{aligned} \tag{3.1}$$

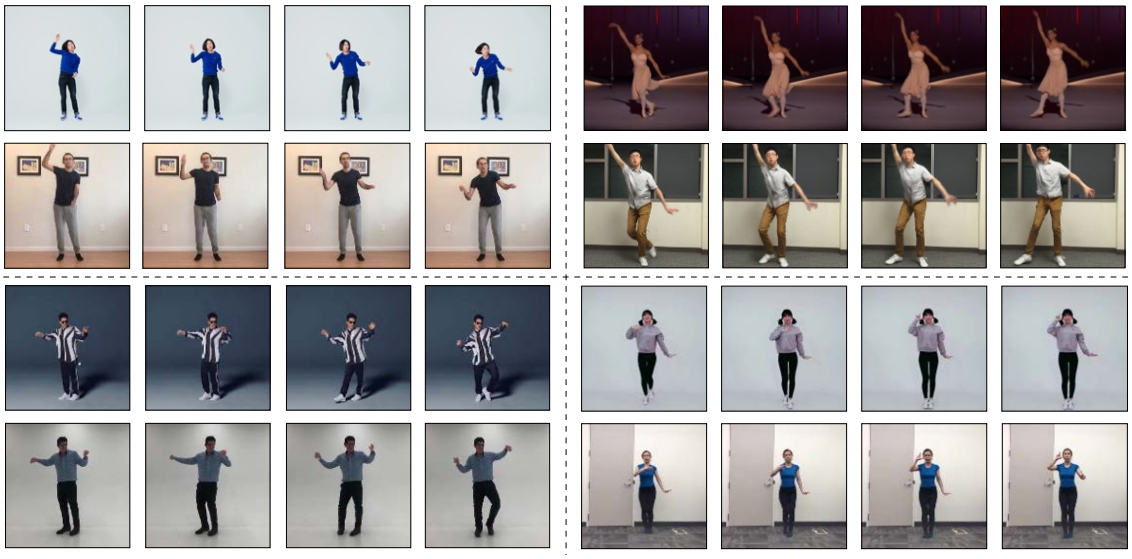


Figure 3.5: **Transfer results.** In each section we show four consecutive frames. The top row shows the source subject and the bottom row shows the synthesized outputs of the target person.

Face GAN We add a specialized GAN setup to add more detail and realism to the face region as shown in Figure 3.4. After generating the full image of the scene with the main generator G, we input a smaller section of the image centered around the face (i.e. 128×128 patch centered around the nose keypoint), $G(x)_F$, and the input pose stick figure sectioned in the same fashion, x_F, to another generator G_f which outputs a residual $r = G_f(x_F, G(x)_F)$. The final synthesized face region is the addition of the residual with the face region of the main generator $r + G(x)_F$. A discriminator D_f then attempts to discern the "real" face pairs (x_F, y_F) from the "fake" face pairs $(x_F, r + G(x)_F)$, similarly to the original pix2pix [43] objective:

$$\begin{aligned}\mathcal{L}_{\text{face}}(G_f, D_f) = \mathbb{E}_{(x_F, y_F)}[\log D_f(x_F, y_F)] \\ +\mathbb{E}_{x_F}[\log\left(1 - D_f(x_F, G(x)_F + r)\right)].\end{aligned} \tag{3.2}$$

Here x_F is the face region of the original pose stick figure x and y_F is the face region of ground truth target person image y. Similarly to the full image, we add a perceptual reconstruction loss on comparing the final face $r + G(x)_F$ to the ground truth target person's face y_F.

3.3.3 Full Objective

We employ training in stages where the full image GAN is optimized separately from the specialized face GAN. First we train the main generator and discriminator (G, D) during which the full objective is -

$$\min_G((\max_{D_i}\sum_{k_i}\mathcal{L}_{\text{smooth}}(G, D_k)) + \lambda_{FM}\sum_{k_i}\mathcal{L}_{\text{FM}}(G, D_k) \\ +\lambda_P(\mathcal{L}_P(G(x_{t-1}), y_{t-1}) + \mathcal{L}_P(G(x_t), y_t))) \tag{3.3}$$

Where $i = 1, 2, 3$. Here, $\mathcal{L}_{\text{GAN}}(G, D)$ is the single image adversarial loss presented in the original pix2pix paper [43]:

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x))] \tag{3.4}$$

$\mathcal{L}_{\text{FM}}(G, D)$ is the discriminator feature-matching loss presented in pix2pixHD, and $\mathcal{L}_P(G(x), y)$ is the perceptual reconstruction loss [85] which compares pretrained VGGNet [86] features at different layers of the network.

After this stage, the full image GAN weights are frozen and we optimize the face GAN with objective

$$\min_{G_f}\left(\left(\max_{D_f}\mathcal{L}_{\text{face}}(G_f, D_f)\right) + \lambda_P\mathcal{L}_P(r + G(x)_F, y_F)\right) \tag{3.5}$$

where $\mathcal{L}_{\text{FM}}(G, D)$ is the discriminator feature-matching loss presented in pix2pixHD, and $\mathcal{L}_P$ is a perceptual reconstruction loss [85] which compares pretrained VGGNet [86] features at different layers of the network.

3.4 Experiments

We compare our performance to baseline methods on multiple target subjects and source motions.

3.4.1 Setup

We collect two types of data long, open-source, single-dancer *target* videos which we film ourselves to train our model on and make publicly available, and in-the-wild *source* videos collected online for motion transfer.

Baseline methods 1) **Nearest Neighbors**. For each source video frame, we retrieve the closest match in the training target sequence using the following pose distance metric: For two poses p, p' each with n joints $p_1, ..., p_n$ and $p'_1, ..., p'_n$, we define the distance between them as the normalized sum of the L2 distances between the corresponding joints $p_k = (x_k, y_k)$ and $p'_k = (x'_k, y'_k)$:

$$d(p, p') = \frac{1}{n} \sum_{k=1}^{n} \|p_k - p'_k\|_2 \tag{3.6}$$

The adjacent target matches frames are then concatenated into a frame-by-frame nearest neighbors sequence.

2) Balakrishnan *et al.* (PoseWarp) [62] generate images of a given target subject in a new pose. While, unlike ours, this method is designed for single image synthesis, we use it to synthesize a video frame-by-frame for comparison.

Ablation conditions 1) **Frame-by-frame synthesis** (FBF). In this condition we ablate our temporal smoothing setup and apply pix2pixHD [84] on a per-frame basis. 2) **Temporal smoothing** (FBF+TS). In this condition we ablate the Face GAN module to study the difference it makes on the final result. 3) **Our model** (FBF+TS+FG). uses both temporal smoothing and a Face GAN.

Evaluation metrics We use perceptual studies on Mechanical Turk for evaluating the video results of our final method in comparison to ablated conditions and baselines. For the ablation study, we further measure the quality of each synthesized frame using two metrics: 1) **SSIM**. Structural Similarity [87] and 2) **LPIPS** Learned Perceptual Image Patch Similarity [88]. We examined the pose distance seen in Equation 3.6 to measure the similarity between input and synthesized pose. However, we found this 'distance' to be not very informative due to noisy detections.

3.4.2 Quantitative Evaluation

We quantitatively compare our approach against the baselines, and then against ablated versions of our method.

Comparison to Baselines

We compare our method to baselines on the same transfer task for all subjects for which we filmed longer videos. From a single out-of-sample source video, we synthesize a transfer video for every baseline-subject pair. We then crop the same

Method	1	2	3	4	5	Total
NN	95.9%	96.4%	94.6%	95.8%	94.7%	95.1%
PoseWarp [62]	83.1%	69.9%	88.7%	84.6%	74.4%	83.3%

Table 3.1: Comparison to baselines using perceptual studies for subjects 1 through 5 and in total average. We report the percentage of time participants chose **our** method as more realistic than the baseline.

Method	1	2	3	4	5	Total
NN	85%	93%	94%	90%	91%	91.2%
PoseWarp [3]	77.5%	70%	80%	90%	78.7%	79.1%

Table 3.2: Comparison of our method without Face GAN (FBF+TS variant) to baselines for subjects 1 through 5 and in total average. We report the percentage of time participants chose the FBF+TS ablation as more realistic than the baseline.

10-second snippets of video for each baseline and subject pair and use these for our perceptual studies.

Participants on MTurk watched a series of video pairs. In each pair, one video was synthesized using our method; the other by a baseline. They were then asked to pick the more realistic one. Videos of resolution 144×256 (as this is the highest resolution that PoseWarp baseline can produce) were shown, and after each pair, participants were given unlimited time to respond. Each task consisted of 18 pairs of videos and was performed by 100 distinct participants. Table 3.1 displays the results of this study and shows that participants indicated our method is more realistic 95.1% and 83.3% of the time on average in comparison to the Nearest Neighbors and PoseWarp [62] baselines respectively.

We include an additional perceptual study to verify our method is not preferred over the others simply due to more emphasis on face synthesis. We compare the FBF+TS variant (without the Face GAN module) to both baselines in Table 3.2. We find that the FBF+TS ablation is consistently preferred, albeit slightly less than our full model, over the Nearest Neighbors and PoseWarp baselines 91.2% and 79.1% of the time on average respectively.

Ablation Study

We perform an ablation study on held-out test data of the target subject (the source and target are the same) since we do not have paired same-pose frames across

Region	Metric	FBF	FBF+TS	FBF+TS+FG
Face	SSIM	0.784	0.811	**0.816**
	LPIPS	0.045	0.039	**0.036**
Body	SSIM	0.828	**0.838**	**0.838**
	LPIPS	0.057	0.051	**0.050**

(a) Metric comparison for synthesized face (top) and full-body (bottom) regions. Metrics are averaged over the 5 subjects. For SSIM higher is better. For LPIPS lower is better.

Condition	1	2	3	4	5	Total
FBF	54.1%	69.7%	62.4%	53.8%	60.0%	58.8%
FBF+TS	59.6%	56.4%	50.3%	53.0%	53.1%	53.9%

(b) Perceptual study results for subjects 1 through 5 and in total average. We report the percentage of time participants chose **our** method as more realistic than the ablated conditions.

Table 3.3: Ablation studies. We compare frame-by-frame synthesis (FBF), adding temporal smoothing (FBF+TS) and our final model with temporal smoothing and Face GAN modules (FBF+TS+FG).

Condition	1	2	3	4	5	Total
Prefer FBF+TS	60.5%	62%	57.5%	50%	62.5%	58.5%

Table 3.4: Comparison of our method without Face GAN (FBF+TS) to the FBF ablation for subjects 1 through 5 and in total average. We report the percentage of time participants chose the FBF+TS ablation over the FBF ablation.

subjects.

As shown in Table 3.3a(bottom), both SSIM and LPIPS scores are similar for all model variations on the body regions. Scores on full images are even more similar, as the ablated models have no difficulty generating the static background. However, Table 3.3a(top) demonstrates the effectiveness of our face residual generator by showing the improvement of our full model over the the FBF+TS condition.

As these comparisons are in a frame-by-frame fashion they do not emphasize the usefulness of our temporal smoothing setup. The effect of this module can be seen in the qualitative video results and in the perceptual studies results in Table 3.3b. Here we see that our method is preferred 58.8% and 53.3% of the time over frame-by-frame synthesis and the No Face GAN (FBF+TS) setup respectively. In general, this shows that incorporating temporal information at training time positively influences video results. Although the effect of the Face GAN can be be somewhat subtle, overall this addition benefits our results, especially in the case of subject 1 whose training video is very sharp where facial details are easily visible.

We further compare our method without the Face GAN (FBF+TS) to the frame-by-frame (FBF) ablation to verify our temporal smoothing setup alone improves result quality. Table 3.4 reports that the FBF+TS ablation is preferred on average over the FBF alone. Note that for subject 4 FBF produced noticeable flickering, but FBF+TS introduced texture artifacts on his loose shirt (see Figure 3.9).

3.4.3 Qualitative Results

Transfer results for multiple source and target subjects can be seen in Figure 3.5. The advantage of using the Face GAN module can be seen in a single frame comparison in Figure 3.6. As mentioned, [62] is designed for single image synthesis. Nonetheless, even for a single frame transfer, we outperform [62] as we show in Figure 3.7.

While the above single-image and quantitative results (Section 3.4.2) suggest the superiority of our approach, more significant differencse can be observed in our video. There we find the temporal modeling produces more frame to frame coherence than the frame-by-frame ablation, and that adding a specialized facial generator and discriminator adds considerable detail and realism.

3.5 Detecting Fake Videos

Recent progress on image synthesis and generative models has narrowed the gap between synthesized and real images and videos, which has raised legal and ethical questions on video authenticity (among many other social implications). Given the

Figure 3.6: **Face image comparison on held-out data.** We compare frame-by-frame synthesis (FBF), adding temporal smoothing (FBF+TS) and our full model (FBF+TS+FG).

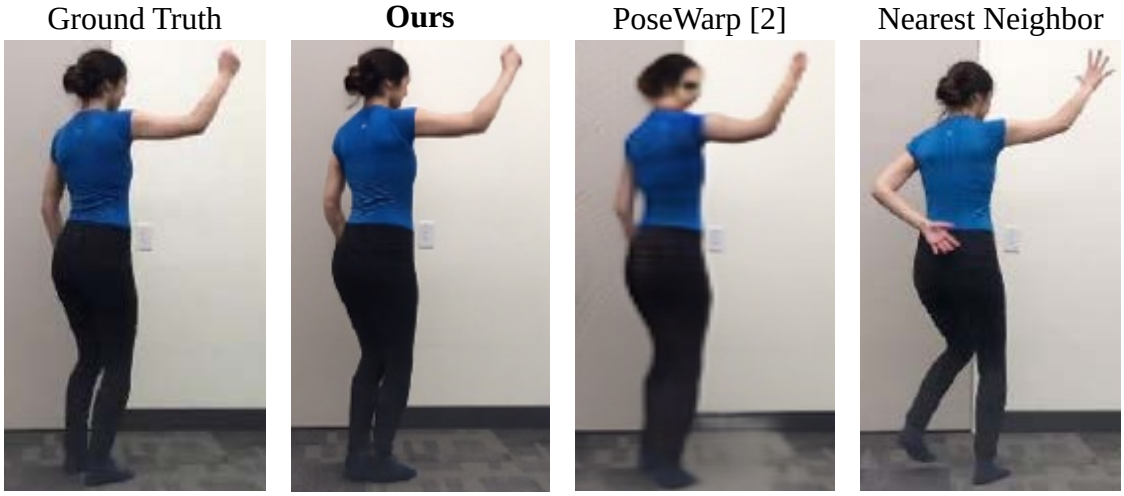


Figure 3.7: **Comparison on single-frame synthesis.** Comparison between our model, [62], and nearest neighbors on single-frame synthesis on held-out data.

high quality of our results, it is important to investigate mechanisms for detecting computer-generated videos including ones generated by our model.

We train a fake-detector to identify fake videos created by our system — given a video, the fake-detector flags it as real or fake. We train the fake-detector in a parallel fashion to our synthesis process, to classify whether a sequence of 2 consecutive frames is real (from ground-truth frames) or fake (from our generation). This allows the fake-detector to exploit cues based on the fidelity of individual frames as well as consistency across time. To make a decision for the whole video in question, we multiply the decision probabilities for all consecutive frame pairs. For the purpose of training the fake-detector, we collect a 62-subject set of short 1920×1080 resolution dancing videos. This larger dataset is collected from public YouTube videos where a subject dances in front of a static camera for an average of 3 minutes. We split this set into 48 subjects for training and 14 held-out subjects for testing.

We train a separate synthesis model for each of the 48 train subjects to produce fake content for detection. By training our fake-detector on multiple fake videos depicting a large set of subjects we ensure that it generalizes to detecting fakes of different people and does not over-fit to one or two individuals. We note that since each person dancing performs a rich set of motions we require less training data than for detecting fakes in still images.

Figure 3.8: **Multi-subject synchronized dancing.** By applying the same source motion to multiple subjects, we can create the effect of them performing synchronized dance moves.

We evaluate our fake-detector on synthesized videos for 14 held-out test subjects. We use both motion taken from the same subject (where the source and target are the same person) and motion driven by a different source subject (Bruno Mars and

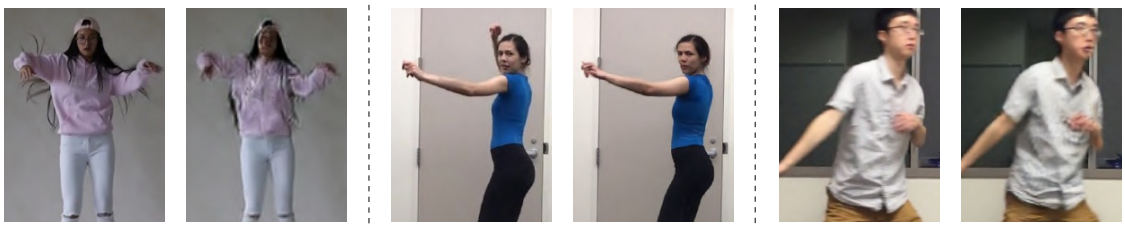


Figure 3.9: **Failure cases.** Ground truth appearance reference (left) followed by our results (right).

Source Motion	Same subject	Mars	Copeland
Accuracy	95.68%	96.70%	97.00%

Table 3.5: Fake detection average accuracy for held-out target subjects. As seen in the rows, fake videos were created for each target subject using same-subject and different-subject source motions.

Misty Copeland) to synthesize fake videos for each held out subject. Our results are shown in Table 3.5. Overall, the fake-detector successfully distinguishes real and fake sequences regardless of where the source motion is from. As expected, our fake detection accuracy is lowest for same-person motion transfer, and is highest for transfer of motion from a prima ballerina (Misty Copeland).

3.6 Potential Applications

One fun application of our system is to create a motion-synchronized dancing video with multiple subjects (say, for making a family reunion video). Given trained synthesis models for multiple subjects, we use the same source video to drive the motion of all target subjects — creating an effect of them performing the same dance moves in a synchronized manner. See Figure 3.8 and the video.

Several systems based on our prototype description were recently successfully employed commercially. One example is an augmented reality stage performance art piece where a 3D-rendered dancer appears to float next to a real dancer [89]. Another is an in-game entertainment application making NBA players dance [90].

3.7 Discussion

Our relatively simple model is usually able to create arbitrarily long, good-quality videos of a target person dancing given the movements of a source dancer to follow. However, it suffers from several limitations.

We have included examples of visual artifacts in Figure 3.9. On the left, our model struggles with loose clothing or hair which is not conveyed well through pose. The middle columns show a missing right arm which was not detected by OpenPose. On the right we observe some texture artifacts in shirt creases. Further work could focus on improving results by combining target videos with different clothing or scene lighting, improving pose detection systems, and mitigating the artifacts caused by high frequency textures in loose/wrinkled clothing or hair.

Our pose normalization solution does not account for different limb lengths or camera positions. These discrepancies additionally widen the gap between the motion seen in training and testing. However, our model is able to generalize to new motions fairly well from the training data. When filming a target sequence, we have no specific source motion in mind and do not require the target subject performing similar motions to any source. We instead learn a single model that generalizes to a wide range of source motion. However our model sometimes struggles to extrapolate to radically different poses. For example, artifacts can occur if the source motion contains extreme poses such as handstands if the target training data did not contain such upside-down poses. Future work could focus on the training data, i.e. what poses and how many are needed to learn a effective model. This area relates to work on understanding which training examples are most influential [91].

Part II

Discovering Temporal Visual Patterns

Chapter 4

A Visual Historical Record of High School Portraits

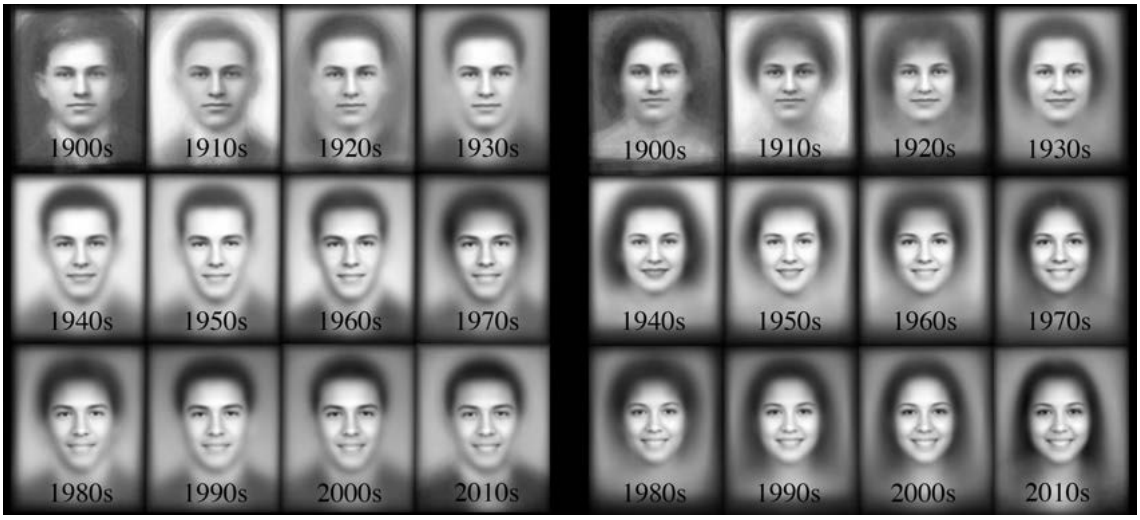


Figure 4.1: **Average images of students by decade.** The evolving fashions and facial expression throughout the 20th century are evident in this simple aggregation. For example, notice the increasing extent of smiles over the years and the recent tendency for women to wear their hair long. In contrast, note that the suit is the default dress code for men throughout.

Imagery offers a rich description of our world and communicates a volume and type of information that cannot be captured by text alone. Fortunately, since the invention of the camera, an ever-increasing number of photographs capture much of this otherwise lost information. This plethora of artifacts documenting our "visual

culture" is a treasure trove of knowledge barely tapped by historians who thus far could only analyze it manually and at small scale. In this chapter we perform scalable historical analysis on a large-scale historical image dataset. Our main contributions are: 1) A publicly-available dataset of 168,055 (37,921 frontal-facing) American high school yearbook portraits. 2) Weakly-supervised data-driven techniques to discover historical visual trends in fashion and identify date-specific visual patterns. 3) A classifier to predict when a portrait was taken, with median error of 5 years. 4) A new method for discovering and displaying the visual elements used by the CNN-based date-prediction model to date portraits, finding that they correspond to the tell-tale fashions of each era.[1]

4.1 Motivation

In their quest to understand the past, historians—from Herodotus to the present day—primarily rely on textual records. However, some details are perceived as too mundane to put down in writing or too difficult to accurately describe. For example, it would be hard for a future historian to understand what the term "hipster glasses" refers to, just as it is difficult for us to imagine what "flapper galoshes" might look like from a written description alone [93]. The invention of the Daguerreotype in 1839 as a means of relatively cheap, automatic image capture heralded a new age of massive visual data creation with potentially profound implications for historians. This new format was complementary to historical texts, as it could both capture those nuances and transmit non-verbal information that would otherwise be lost.

The study of history is often an exercise in finding patterns in large amounts of data. For written accounts, historians have begun to use digital humanities techniques to automatically mine large text corpora. For example, using Google Books it is possible to study a diverse set of topics such as word usage over time, the histories of events like the Civil War or the spread of Influenza, and even advances in disciplines like Biology and Gender Studies [94]. In contrast, despite the abundance of historical visual data over the last century and a half, historians are still limited by the speed of manual curation. There are perhaps many unseen visual connections that are missed because tools for large-scale visual data mining have yet to be introduced into the field.

We take a new approach to the analysis of visual historical data by introducing data-driven methods suited to mining large image collections. Moreover, we present a large visual historical dataset that can support such methods. By treating large

[1]This work was first published as *A Century of Portraits: A Visual Historical Record of American High School Yearbooks* in IEEE Transactions on Computational Imaging, September 2017 [92].

historical photo collections as a whole, we expect to learn things that cannot be inferred from the inspection of a small number of artifacts in isolation.

One of the most interesting historical trends is the evolution in the appearance of people over time. To study these trends we present a collection of one type of widely available yet little used historical visual data—a century's worth of United States high school yearbooks (Fig 4.1). Yearbooks, an iconic American high school staple, have been published since the wide adoption of film (the first Kodak camera was released in 1888) and contain standardized portrait photos of the graduating class. As such, yearbook portraits provide a consistent visual format through which one can examine changes in content from personal style choices to developing social norms. In this chapter, we present a large-scale dataset of yearbook portraits spanning the entire 20th century, and report on a number of experiments to analyze it.

First, we mine the portrait data to discover trends over time and date-specific visual patterns. We examine changes in social norms by studying the practice of smiling to the camera and men's changing hair styles during the social changes of the 1960s. Additionally, we discover that fluctuations in the popularity of eyewear is correlated with advances in contact lens technology. Finally, we mine for the quintessential "look" of each decade by employing a technique of discriminative clustering. Our data-driven results are consistent with existing historical records of the fashion trends in hair, makeup and eyewear from the 20th century.

Second, we use the time-correlated visual variability in the portraits to predict, from an image of a face alone, when the photograph was taken. We call this the *portrait dating* problem. Using a convolutional neural network (CNN) classifier trained on our dataset we achieve impressive performance, dating yearbook portraits within an average of five years of their true date. We further demonstrate some generalization to an unseen dataset of historical celebrity portraits despite the large differences in appearance between high school students and adult actresses and models.

Finally, while CNN classifiers have proven time and again to be the leading tool for many image domains, it remains challenging to tell *why* a specific classification decision has been made. This is particularly important for tasks like dating where the labels are weak, the visual space is huge, and much of the visual data might be irrelevant to the task. We propose a method to peak inside this "black box" to discover which parts of the image were most useful for the classification decision. At the core of our approach lies the insight that we can disable parts of the network without altering the dating decision. Our visualization algorithm traverses the network top to bottom and iteratively removes any spatial units whose absence will not affect the final classification. At the end of this process we are left with the spatial locations at each layer that carry sufficient information for dating the input

portrait.

The main contributions of this chapter are: 1) A publicly-available historical image dataset that comprises a large scale collection of yearbook portraiture from the last 120 years in the United States[2]. 2) Data-driven methods to discover historical visual patterns in fashion and social norms. 3) A CNN classifier to predict the date in which a portrait was taken, with median error of 5 years. 4) A new method for visualizing the time-specific elements used by the CNN classifier to date the portraits.

4.2 Background

4.2.1 Historical Data Analysis

Researchers in the humanities are able to tease out historical information from ever larger text corpora thanks to advances in natural language processing and information retrieval. For example, these advances (together with the availability of large-scale storage and OCR technology) enabled Michel et al. [94] to conduct a thorough study of about 4% of all books ever printed resulting in a quantitative analysis of cultural and linguistic trends. Large historical image collections will enable researchers to conduct similar analyses of visual trends.

To date, the study of historical images has been relatively limited. Some examples include modeling the evolution of automobile design [95] and architecture [96] as well as *image dating*–determining the date when historical color photographs were taken [97, 98]. Here we extend upon these works by presenting a dataset that we use to answer a broader set of questions. We note that, concurrent and independent of our work, [99] also proposed using yearbook data for image dating.

4.2.2 Modeling Style

Recently several researchers focused on modeling fashion items. In HipsterWars, Kiapour et al. [100] take a supervised approach and use an online game to crowd-source human annotations of five current clothing-style categories that are then used to train models for style classification. Hidayati et al. [101] take a weakly-supervised approach to discover the recent (2010-2014) trends in the New York City fashion week catwalk shows. They extract color and texture features and use these to discover the representative visual style elements of each season via discriminative clustering [102]. While we also deal with fashion and style in this chapter, our focus is on changes in style through a much longer period of history. Because our dataset includes scanned

[2] http://shiry.eecs.berkeley.edu/yearbooks/

images from earlier time periods, much of it consists of lower resolution and quality images than the recent datasets described above. This makes some of the above approaches unsuitable for our data.

4.2.3 Deep Neural Networks

Of the many CNN architectures designed in recent years, the VGG [86] network is one of the best-performing and most versatile. It is designed as a deep network of 16 convolutional layers with spatially-grouped feature maps and two fully connected layers on top. The VGG model trained on ILSVRC 2012 [103] has proven to generalize well to various computer vision tasks with proper fine-tuning (or further training) on the target data and task. In this chapter, we use VGG for the task of portrait dating. We further develop a method to visualize what a CNN uses to make inference decisions and use VGG as our example network on which we run our experiments.

4.2.4 Deep Neural Network Visualization

Several attempts have been made to visually understand the inner-workings of deep networks. One of the first aimed to find input images that maximize the activation of specific units in the network [104]. Following a similar approach, Zhou et al. [105] ask which segments of an image are most responsible for a particular classification decision. By analyzing these image segments, they discover that object detectors emerge in CNNs trained on a scene detection task. While our approach shares some similarities with [105], we do not force our visual elements to be enclosed in image regions, allowing us to discover not just objects but more ephemeral visual structures.

Zeiler et al. [106] also examine which parts of the image result in the highest response of single spatial units by systematically obstructing parts the image. They use deconvolutional networks to invert the effect of pooling layers and reconstruct an approximation of the input pixels from the activations of intermediate layers of the network.

Most similar to our approach, Simonyan et al. [107] use the network gradient propagated back to pixel space for a single input image as an approximation of which spatial locations would maximize the classification score if changed. This method discovers the spatial locations that affect the class score for a canonical image from this class and only reveals the general location of the object in the image. In contrast, our approach takes into account the unique path which the input image takes through the network and therefore discovers which visual elements were used

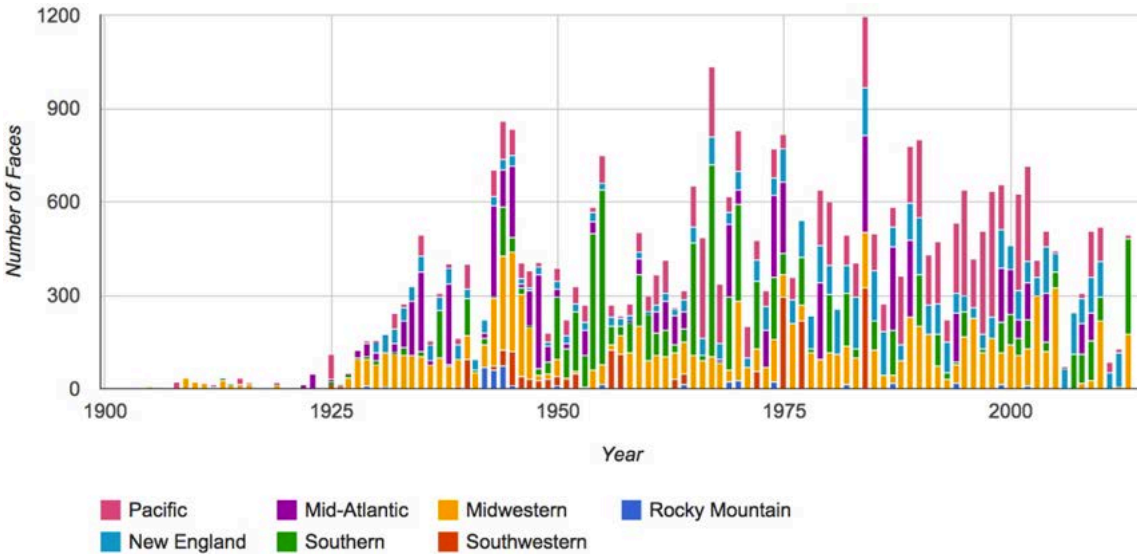


Figure 4.2: The distribution of portraits per year and region.

by the CNN to classify *this* image. As a result, our method focuses on localized areas within objects that correspond to discriminative visual features.

Mahendran et al. [108] and Dosovitskiy et al. [109] visualize images that produce a specific code in CNNs. While those visualizations show the information retained in the network throughout different layers, they do not highlight the different spatial locations a certain classification was built on. Furthermore, they have shown only limited success in visualizing the upper layers, including the classification layer of a network.

Finally, since the goal of our visualization approach is to find the visual elements that CNNs use for inference, we evaluate the elements we find based on their visual consistency and class-discriminativeness. A similar goal was explored by [110] who used modified input images in order to localize the discriminative parts of the image. In contrast, we allow the important regions to emerge from applying the classifier to the original images. The search for discriminative visual elements has preceded the current vogue of deep learning. Singh et al. [111] and Doersch et al. [102] both used HOG features in order to automatically detect discriminative clusters of visually-consistent image patches.

4.3 The Yearbook Dataset

We are at an auspicious moment for collecting historical yearbooks as it has become standard in recent years for local libraries to digitally scan their yearbook archives. This trend enabled us to download publicly available yearbooks from various online resources such as the Internet Archive and numerous local library websites. We collected 949 scanned yearbooks from American high schools ranging from 1905-2013 across 128 schools in 27 states. These contain 168,055 individual senior-class portrait photographs in total along with many more underclassmen portraits that were not used in this project. After removing all non-frontal facing we were left with a dataset of 37,921 photographs that depict individuals from 814 yearbooks across 115 high schools in 26 states.

On average, 28.8 faces are included in the dataset from each yearbook with an average of 329 faces per school across all years. The distribution of photographs over year and region is depicted in Figure 4.2. Overall, 46.4% of the photos come from the 100 largest cities according to US census [112].

Let us consider the potential biases in our data sample as compared to the high school age population of the United States. Since 1902 America's high schools have followed a standard format in terms of the population they served [113]. Yet, this does not mean that the population of high school students has always been an unbiased sample of the US youth population. In the early 1900s, less than 10% of all American 18-year-olds graduated from high school, but by end of the 1960s graduation rates increased to almost 50% [113]. Moreover, the standardization of high schools in the United States left out most of the African American population, especially in the South, until the middle of the 20th century [114].

In our dataset 53.4% of the photos are of women, and 46.6% are of men. As the true gender proportion in the population is only available in a census year we are unsure if this is a bias in our data. However, the gender imbalance may be due to the fact that historically girls are disproportionately more likely than boys to attend high school through graduation [113].

4.3.1 Data Preprocessing

In order to turn raw yearbooks into an image dataset we performed several pre-processing operations. First, we manually identified the scanned pages of senior-class portraits. After converting these to grayscale for consistency across years, we automatically detected and cropped to faces. We then extracted facial landmarks from each face and estimated its pose with respect to the camera using the IntraFace system [115]. This allowed us to filter out images of students who were not facing

forward. Next, we aligned all faces to the mean shape using an affine transform based on the computed facial landmarks. Finally, we divided the photos into those depicting males and females using an SVM in the whitened HOG feature space [116, 117] and resolved difficult cases by crowdsourcing a gender classification task on Mechanical Turk.

4.4 Mining the Visual Historical Record

We demonstrate the use of our historical dataset in answering questions of historical and social relevance.

4.4.1 Getting a Sense of Each Decade

The simplest visual-data summarization technique of facial composites dates back to the 1870s and is attributed to Sir Francis Galton [118]. Here we use this technique to organize the portraits chronologically. Figure 4.1 (first page) displays the pixel-mean of images of male and female students for each decade in our data. These average images showcase the main modes of the popular fashions in each time period.

4.4.2 Capturing Trends Over Time

We capture changes in attributes that always occur in a portrait (degrees of smiling) as well as in accessories or styles that are present in only some of the population at a given time.

Smiling in Portraiture

A close observation of the decade average images in Figure 4.1 reveal a change over time in the facial expression of portrait subjects. In particular, these days we take for granted that we should smile when our picture is being taken; however, smiling at the camera was not always the norm and in this section we try and quantify this change over time numerically.

In her paper, Kotchemidova studied the appearance of smiles in photographic portraits using the traditional historical methods of analyzing sample images manually [119]. She reports that in the late 19th century people posing for photographs still followed the habits of painted portraiture subjects. These included keeping a serious expression since a smile was hard to maintain for as long as it took to paint a portrait. Also, etiquette and beauty standards dictated that the mouth be

kept small – resulting in an instruction to “say prunes" (rather than cheese) when a photograph was being taken [119]. All of this changed during the 20th century when amateur photography became widespread. In fact, Kotchemidova suggests that it was the attempt to associate photography with happy occasions like holidays and travel that led the photographic monopoly, Kodak, to educate the public through visual advertisements that the obvious expression one should have in a snapshot is a smile. This century-long advertisement campaign was a great success. By World War II, smiles were so widespread in portraiture that no one questioned whether photographs of the GIs sent to war should depict them with a smile [119].

To verify the apparent trend in our average images and Kotchemidova’s claims regarding the presence and extent of smiles in portrait photographs in a data-driven way, we devised a simple lip-curvature metric and applied it to our dataset. We compute the lip curvature by taking the average of the two angles indicated in Figure 4.3 (Left) where the point that forms the hypotenuse of the triangle is the midpoint between the bottom of the top lip and the top of the bottom lip of the student. The same facial keypoints were used here as in image alignment (see section 4.3.1). Figure 4.3 (Right) is a montage of students ordered in ascending order of lip curvature value from left to right. It demonstrates that the lip-curvature metric quantifies the smile intensities in our data in a meaningful way.

We verify that our metric generalizes beyond yearbook portraits by testing it on the BP4D-Spontaneous dataset that contains images of participants showing various degrees of facial expressions with ground truth labels of expression intensity [120]. BP4D uses labels drawn from the Facial Action Coding System, which is commonly used in facial expression analysis. This system consists of Action Units (AU) that correspond to the intensity of contraction of various facial muscles. Following previous work done on smile intensity estimation [121], we compared our smile intensity metric with the activation of AU12 (Lip corner puller) as it corresponds to the contraction of muscles that raise the corners of the mouth into a smile. A higher AU12 value represents a higher contraction of muscles around the corner of the mouth, resulting in a larger smile. Figure 4.4 displays the average lip curvature for each value of AU12 for 3 male and 3 female subjects, corresponding to 2,500-3,000 samples for each AU12 value (0-5). As the simple lip-curvature metric we used correlates with increasing AU12 values on BP4D images, it is a decent indicator for smile intensities beyond our yearbook dataset.

Using our verified lip-curvature metric we plot the trend of average smile intensities in our data over the past century in Figure 4.5. Corresponding montages of smile intensities over the years are included in Figure 4.6, where we picked the student with the smile intensity closest to the average for each 10-year bucket from 1905 to 2005. These figures corroborate Kotchemidova’s theory and demonstrate the

Figure 4.3: **Smile intensity metric.** Left: the lip curvature metric is the average of the two marked angles. Right: women and men portraits sorted by increasing lip curvature.

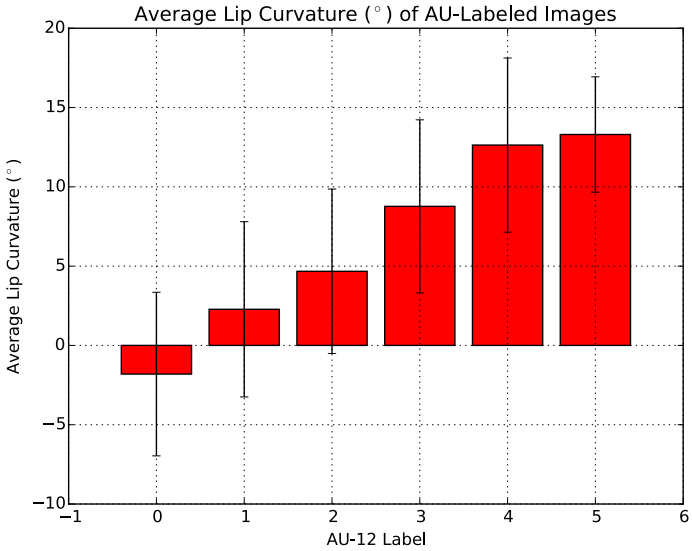


Figure 4.4: Average lip curvature correlates with AU-12 labels on BP4D data (error bars denote standard deviation).

rapid increase in the popularity and intensity of smiles in portraiture from the 1900s to the 1950s, a trend that still continues today; however, they also reveal another trend—women significantly and consistently smile more than men. This phenomenon has been discussed extensively in the literature (see the meta-review in [122]), but until now required intensive manual annotation in order to discover and analyze. For example, in her 1982 article Ragan manually analyzed 1,296 high school and university yearbooks and media files in order to reveal a similar result [123]. By use of a large historical data collection and a simple smile-detector we arrived at the same conclusion with a minimal amount of annotation and no manual effort.

Glasses

Measuring the degree of smiles is easy to apply to each portrait in the collection since every subject exhibits *some* degree of mouth curvature, albeit sometimes a negative one. We now extend our study of trends to accessories and fashions that are only worn by a fraction of the population and that require a classification decision per portrait to determine if the specific style or accessory is exhibited. We first study the usage of glasses by taking advantage of a kernel of annotated celebrity portraits from the PubFig dataset [124]. We fine tune VGG [86], a deep classification system pre-trained on ILSVRC [103], on the celebrity portraits that are marked as wearing glasses. We then apply the trained classifier to our Yearbook dataset to find persons wearing glasses in our data. In Figure 4.7 we graph the fraction of the student population that is wearing glasses for males and females over time. It is interesting to note that glasses are more popular among male students, and to observe that the dips in glasses popularity correlate with the introduction of contact lenses.

Men's Hairstyles post 1960

The final trend we study is changes in men's hairstyles since the social movements of the 1960s which brought about long hair styles and Afros. Here we could not find an existing annotated dataset with appropriate annotations. We therefore segmented out the hair in each portrait following [125] and determined whether the depicted person had long hair or an Afro by checking whether the segmentation map consists of hair under the depicted person's chin or high above his face, respectively. Unfortunately, due to the low resolution of some of the portraits in our dataset, the resulting classifiers were not accurate enough. Figure 4.8 shows the fraction of the population with these hairstyles after a manual process of removing false positives and adding some false negatives to our classifications. We note that our findings corroborate other sources [126, 127] which claim that the Afro hairstyle was

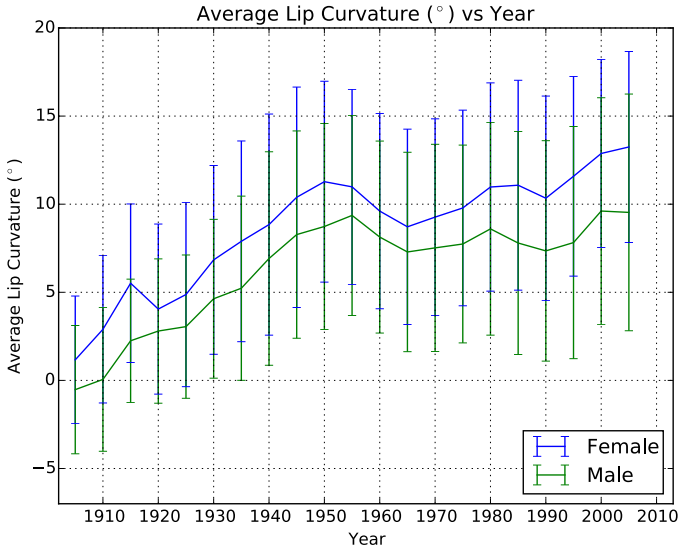


Figure 4.5: **Smiles increasing over time, but women always smile more than men.** Male and female Average lip curvature by year with one standard deviation error bars. Note the fall in smile extent from the 50s to the 60s, for which we did not find prior mention.

Figure 4.6: Images with the closest smile to the mean of that period (10-year bins from 1905 (left) to 2005 (right)). Note the increasing extent of smiles.

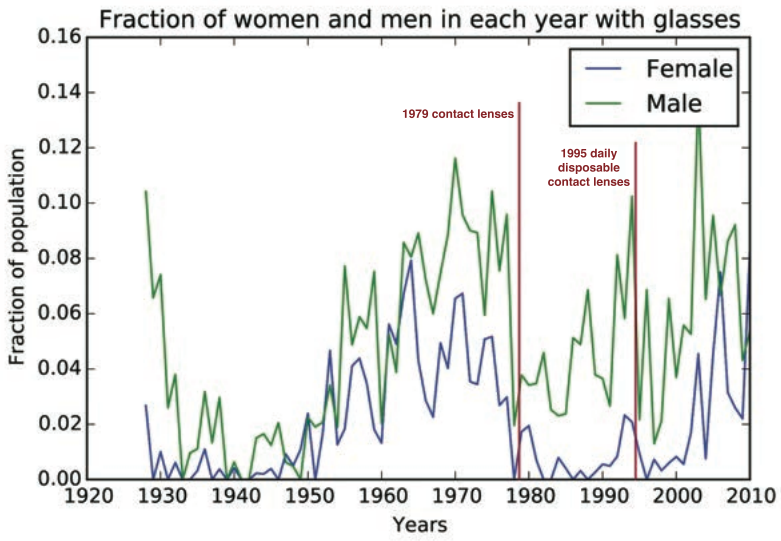


Figure 4.7: The use of glasses over time dips in correlation with advances in contact lenses, but glasses are consistently more popular among men.

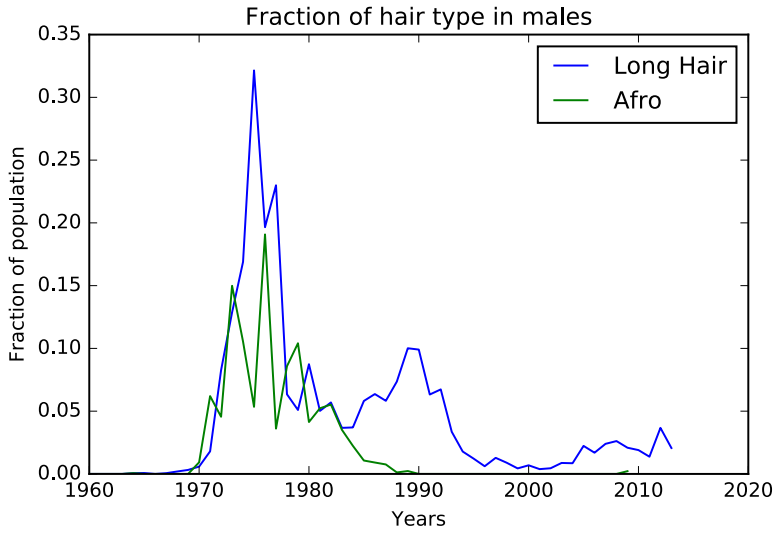


Figure 4.8: The fraction of male students with an Afro or long hair.

predominantly popular from the late 1960s through the late 1970s after which many individuals switched to a more styled version of the natural hairdo.

4.4.3 Mining for Date-Specific Patterns

The average images of each decade from Figure 4.1 show us the main modes of the styles of each decade. However, in each time period or even classroom not every one shares the same style. In fact, we would expect to find several representative and visually discriminative features for every decade. These are the things that make us immediately recognize a particular style as “20s" or “60s", for example, and allow humans to effortlessly guess the decade in which a portrait was taken. They are also the things that are usually hard to put into writing and require a visual aid when describing; this makes them excellent candidates for data-driven methods.

We find the most representative women's styles in hair and facial accessories for each decade using a discriminative mode seeking algorithm [128] on yearbook portraits cropped to contain only the face and hair. Since our portraits are aligned, we can treat them as a whole rather than look for mid-level representative patches as has been done in previous work [102, 128]. The output of the discriminative mode seeking algorithm is a set of detectors and their detected portraits that make up the visual clusters for each decade. We sort these clusters according to how discriminative they are, specifically, how many portraits they contain in the top 20 detections from the target decade versus other decades. In order to ensure a good visual coverage of the target decade, we remove clusters that include in their top 60 detections more than 6 portraits (10%) that were already represented by a higher ranking cluster.

Figure 4.9 displays the four most representative women's hair and eyeglass styles of each decade from the 1930s until the 2000s. Each row corresponds to a visual cluster in that decade. The left-most entry in the row is the cluster average, and to its right we display the top 6 portrait detections of the discriminative detector that created the cluster. We only display a single woman from each graduating class in order to ensure that the affinity within each cluster is not due to biases in the data that result from the photographic or scanning artifacts of each physical yearbook. Looking at Figure 4.9, we get an immediate sense of the attributes that make each decade's style distinctive. Some of the emergent attributes are especially interesting since they would be hard to describe in words. For example, the particular style of curly bangs of the 40s or the “winged" flip hairstyle of the 60s [127]. Finding and categorizing these manually would be painstaking work. With our large dataset these attributes emerge from the data by using only the year-label supervision.

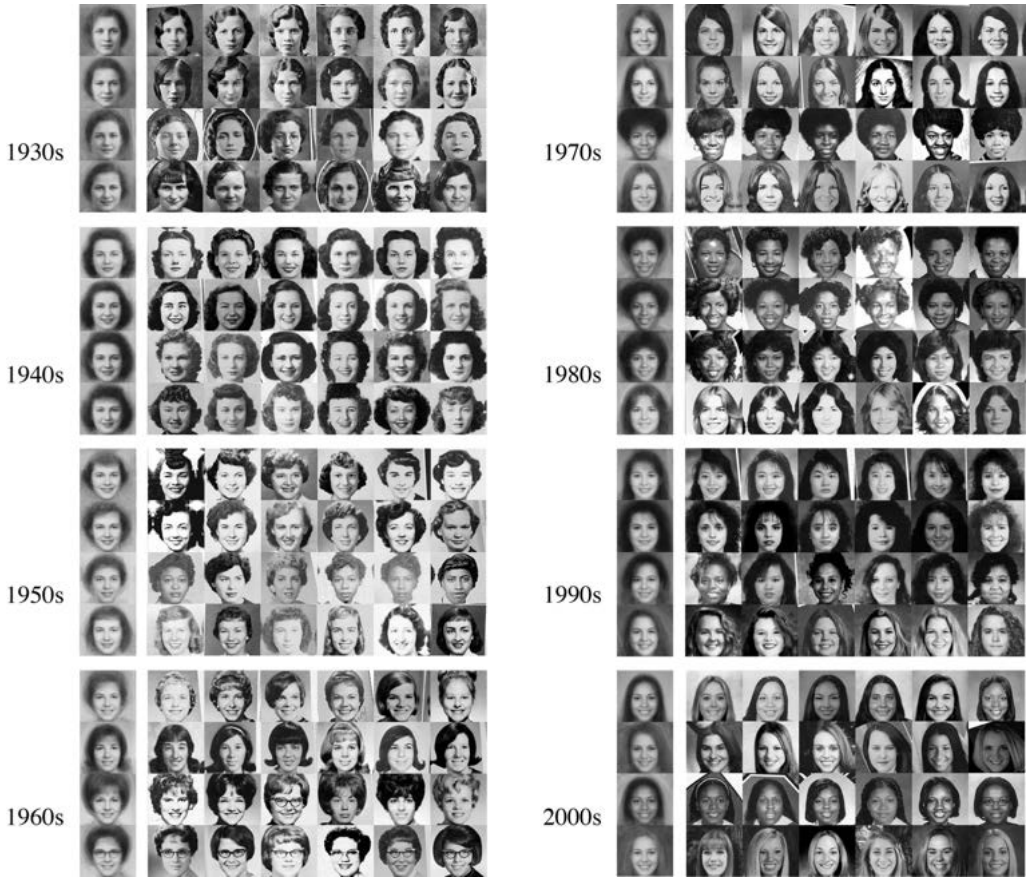


Figure 4.9: **Discriminative clusters of high school girls' styles from each decade of the 20th century.** Each row corresponds to a single detector and the cluster of its top 6 detections over the entire dataset. Only one girl per graduating class is shown in the top detections. The left-most entry in each row displays the cluster average. Note that the clusters correspond to the quintessential hair and accessory styles of each decade. Notable examples according to the Encyclopedia of Hair [127] are: The finger waves of the 30s. The pin curls of the 40s and 50s. The bob, "winged" flip, bubble cut and signature glasses of the 60s. The long hair, Afros and bouffants of the 70s. The perms and bangs of the 80s and 90s and the straight long hair fashionable in the 2000s. These decade-specific fashions emerge from the data in a weakly-supervised, data-driven process.

4.5 Dating Historical Images

In Section 4.4.3 we found distinctive visual patterns that occur in different decades. Here we ask whether there are enough decade-specific visual patterns to be able to predict when in the 20th century a portrait of a face was taken. We call this the *portrait dating* problem.

We extend the work of Palermo et al. [97] in dating color photographs to the realm of black and white portraiture photography where we cannot rely on the changes in image color profiles over time. We choose to train a deep neural network classification model for dating photographs based on the recent success of such models for other visual classification tasks [86]. Here we pose the task of dating the portraits of female students as an 83-way year-classification task between the years 1928 and 2010, for which we have more than 50 female images per year. We evaluate the performance of our model on a subset of images drawn from the yearbook dataset, the *yearbook test set*. To evaluate the generalization capability of our dating model to non-yearbook portraits, we evaluate the model on the *celebrity test set* – a small set of 56 gray-scale head shots of female celebrities, annotated with year labels, that we cropped and aligned to the yearbook images.

Our date-prediction model is based on the VGG-16 model [86] that was pre-trained on the ILSVRC dataset [103]. We fine-tune **all** layers with the task of predicting the date of a photo. As a baseline, we freeze the weights of all other layers and fine-tune only the last classification layer (fc_8) of the network. The training procedure and network implementation are detailed at the end of this section. We further compare our classification performance to *chance*, which we define as the inverse of the number of classes. Results for both the yearbook test set and celebrity test set are shown in Table 4.1. Fine-tuning the full network on the yearbook data improves the classification accuracy on the yearbook test set by a large margin. Furthermore, the confusion matrix for the fully fine-tuned network on the yearbook test set, Figure 4.10, reveals that the predictions are rarely far off the mark. The diagonal structure indicates that most of the confusion occurs between neighboring years, matching our intuition that visual trends such as hairstyle transcend the single-year boundary.

The success in dating yearbook portraits may be misleading since there are biases in the Yearbook dataset that the network can exploit, such as similar backgrounds and low-level image statistics. To test how well the dating classifier generalizes beyond yearbooks we apply it to images in the celebrity dataset which has its own, albeit different, biases. While for this dataset the prediction accuracy is lower, and full fine-tuning does not improve accuracy, we find that fine-tuning all layers reduces the median L1 distance between the predicted and ground truth year for the celebrity

Model	Yearbook Accuracy	Yearbook L1 Med	Celebrity Accuracy	Celebrity L1 Med
Chance	1.20%	-	1.20%	-
Baseline	6.18%	6 [yr]	1.79%	14.50 [yr]
FT on YB	11.31%	4 [yr]	1.79%	9 [yr]

Table 4.1: Classification accuracy and L1 median distance from the ground truth year for the yearbook and celebrity test sets.

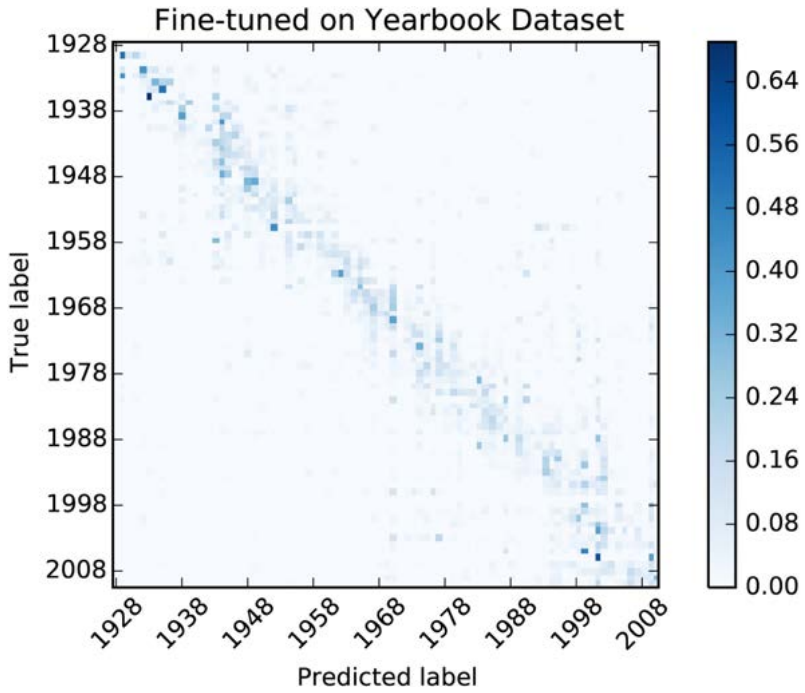


Figure 4.10: Normalized soft confusion matrix for the fully fine-tuned model on the *yearbook test set*. The diagonal structure demonstrates that confusion mostly occurs between neighboring years.

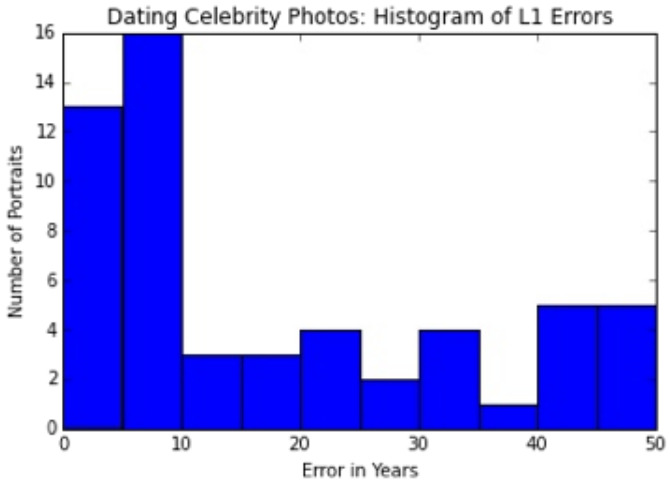


Figure 4.11: Many of the L1 errors in dating celebrities are close to zero.

portraits. (Table 4.1). The reduced performance may be due to the domain shift between portraits of high school students and celebrity glamour shots; celebrity hairstyles can be quite different than those of the general public. Additionally, our celebrity test set may simply be too small to serve as an informative test set. However, for many of the celebrities the results are surprisingly good as many of the L1 errors are close to zero (Figure 4.11). While dating does not generalize well for all celebrities, we can look at individual good predictions as in Figure 4.12.

Implementation Details: For the dating task, we use portraits that were cropped to the face and hair alone. The *yearbook test set* consists of 20% of the portraits taken between 1982 and 2010, with the remaining 80% of images for training and validation. To minimize training biases due to photographic and scanning artifacts, we separate test and training images drawn from the same school by at least a decade. To further minimize these biases, we use the built-in Photoshop noise reduction filter on all the yearbook images and resize them to 96 by 96 pixels. In all of our experiments, we use the Caffe [129] implementation of the VGG network architecture (modified to allow for $96px$ inputs) [86] that was pre-trained on the ILSVRC dataset [103]. Both networks are trained for $100K$ iterations using SGD and a softmax loss with image-mirroring data augmentation, learning rate of 0.001 ($\gamma = 0.1$, stepsize $= 20K$) and momentum of 0.9.

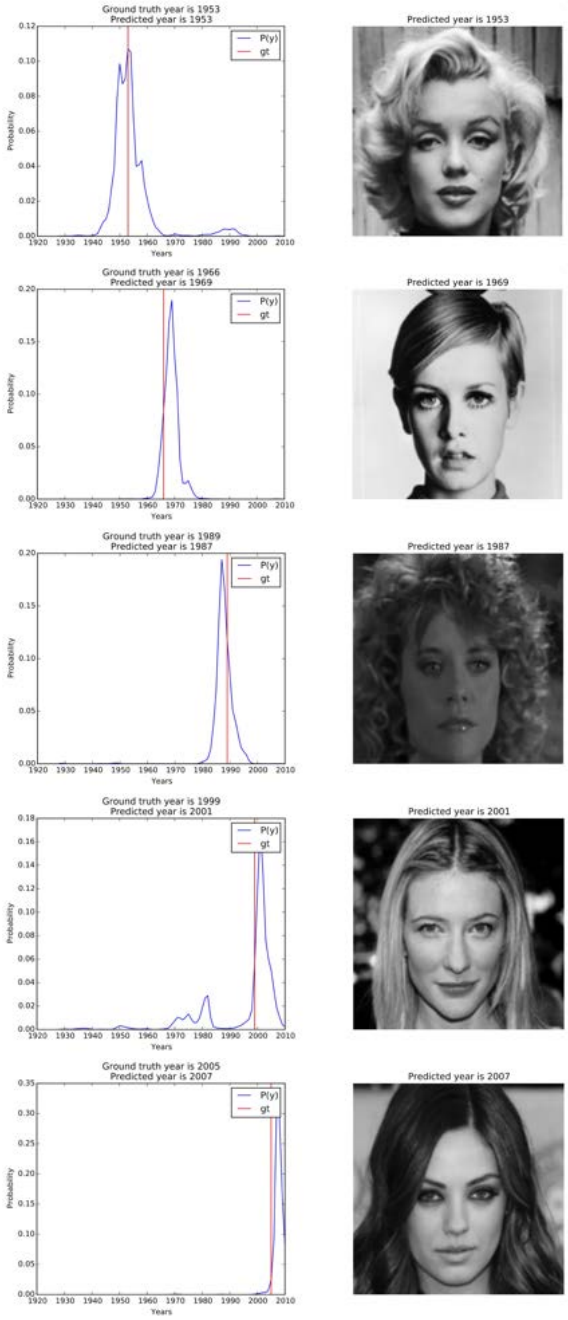


Figure 4.12: **Good celebrity dating predictions.** Red indicates the ground truth year. Blue indicates the prediction distribution.

4.6 What time specific patterns is the classifier using for dating?

In section 4.5 we demonstrated that it is possible to train a classifier to guess the date in which a portrait was taken. But what is the classifier doing? What time-specific visual features is it picking up on? In this section we proposed a method able to find the different explanations and visualize which pixels are most responsible for a given dating decision.

4.6.1 Preliminaries

A convolutional neural network f is a feed-forward network consisting of multiple layers l of units that are connected across layers with non-linearities. Given an input x, the network computes hidden activations z_l at every layer, and combines them into an output $y = f(x)$. These latent representations at the intermediate layers are grouped into spatial locations, such that several features are activated at each spatial location in different feature channels. As CNNs are often trained with a logistic regression loss, the output y is a probability distribution.

While the ensemble of hidden activations learns a large, distributed code for the training data, it is never used in its entirety to represent a single input – different inputs take different paths through the network during inference. Therefore, for a single input we can safely disable the spatial locations throughout the network that are not part of the path for this specific input and keep the same output. This process of removing unused locations that do not participate in the computation of a particular $y = f(x)$ allows us to visualize the parts of the input (here, an image) that do. In the next section we present an algorithm that implements this process.

4.6.2 Top-Down Selection of Spatial Units

For the selection of spatial units, the objective function seeks to maintain the same output distribution while removing unnecessary units, thus asking 'What parts of the image were used to make *this* decision?'. Given an input x we compute its resulting probabilistic output $y = f(x)$ by running a forward pass over the network. We then run a single top-down optimization pass where we disable units in spatial locations that are not needed to produce the probability distribution y. Since our goal is to maintain the same output distribution, we use the KL divergence, a distance measure between two probability distributions, as our objective function. Specifically, we define the objective to be the KL divergence $D_{KL}(y||\hat{y}_l)$ of the predicted output $\hat{y}_l$ after spatial unit removal at layer l from the true final output distribution of the

network y:

$$D_{KL}(y||\hat{y}_l) = \sum_c (y_c \log \frac{y_c}{\hat{y}_{lc}}), \tag{4.1}$$

where c refers to a single entry in the probabilistic output of the CNN (or a single class).

We minimize the KL divergence via the following optimization that forces the network to keep only a sparse set of active units, while maintaining the same output distribution:

$$\begin{aligned} &\underset{M_l \in \{0,1\}^N}{\text{minimize}} \quad D_{KL}(y||\hat{y}_l) \\ &\text{subject to} \quad \|M_l\|_0 \leq s_l. \end{aligned} \tag{4.2}$$

Where M_l is a $2D$ binary mask that disables spatial units at the input to layer l where its elements are 0, and s_l is the desired sparsity over the N spatial units in layer l. For simplicity, we use the same fixed sparsity throughout all layers.

To perform the above optimization we use a greedy algorithm that traverses the network once from top to bottom and minimizes the objective with respect to the constraint at every layer. For each layer, we iterate over all spatial locations of its input feature map and output a binary mask M_l which removes all spatial units that are not necessary for computing the output distribution y. We jointly disable all features grouped at a single spatial location (all channels for a single location). Note that when M does not remove any spatial locations this objective is minimized but the sparsity constraint is violated. We therefore start from a full mask M of all 1's for each layer and remove (zero out) those spatial locations whose removal increases the value of the objective function as little as possible. This approach is similar to Orthogonal Matching Pursuit [130], although in that case the objective function is usually a Euclidean distance. For a detailed description, refer to Algorithm 1.

4.6.3 Gradient Approximation

The iterative greedy algorithm of removing one spatial location at a time at each layer is too slow to run in practice for lower-level layers of the CNN since it iterates over all spatial locations of the feature map for every spatial unit it disables. To make the optimization faster we first present an alternative interpretation of Algorithm 1 and then show how to approximate the expected change in loss for any unit using a single backward pass through the network.

At each step of Algorithm 1 we find a spatial single unit i, which when set to 0 increases the loss the least. This increase in loss can be measured as follows:

$$d_i = D_{KL}(y||\hat{y}_l{}') - D_{KL}(y||\hat{y}_l), \tag{4.3}$$

Algorithm 1 Greedy top-down selection of spatial units

for each layer l **do**
  Start from a mask M_l of all 1's
  while number of active spatial units $> s_l$ **do**
    for each spatial location i **do**
      Zero out i in layer l
      Run a forward pass from l, zeroing locations in higher layers h that were previously disabled
      Compute predicted output $\hat{y}_l$ and loss $D_{KL}(y||\hat{y}_l)$
    end for
    Zero out the spatial location with the smallest increase in the loss function: $D_{KL}(y||\hat{y}_l)$
  end while
end for

where $\hat{y}_l$ and $\hat{y}_l{}'$ are at a single spatial location i that is zeroed out in $\hat{y}_l{}'$. Equation 4.3 can be thought of as a finite-difference approximation with $d_i = z_{l,i}\frac{\partial}{\partial z_{l,i}} D_{KL}(y||\hat{y}_l)$ (though the difference here may be large), and can thus be approximated by the product of the gradient of the KL divergence objective function $\frac{\partial}{\partial z_{l,i}} D_{KL}(y||\hat{y}_l)$ and the value of the input activations $z_{l,i}$ of layer l. While this linear approximation is crude it works well in practice and only requires a single backward pass through the network.

Note that when the two distributions, y and $\hat{y}_l$, are equal the gradient of the objective is zero. In implementing this approximation we therefore reverse the direction of the optimization – we start from a mask M_l of all 0's and add the subset of spatial units that are necessary to maintain the output distribution.

4.6.4 Experimental Setup

We run our spatial unit selection algorithm on the dating classification network that we fine tuned from the ILSVRC-trained VGG [86] as in section 4.5. The VGG network consists of a deep stack of convolutional layers and two fully-connected layers at the top. While the algorithm runs out-of-the-box on VGG, the fully connected layers discard the spatial component of their input feature maps that was maintained throughout the convolutional stack. We therefore modify the network where, following Long et al. [131], we replace the fully-connected layers with convolutional ones creating a fully convolutional version of VGG. Unlike Long et al., we use 1×1 convolutions to replace all upper layers, reducing the parameters of the model as well as the

Simonyan et al. [107] Ours

Figure 4.13: Discriminative regions for a 1940 portrait found by each method, overlaid on the mean training image.

receptive field size of each unit. This allows us to treat each image-pixel as an independent predictor for image-class c. Since we do not have pixel-level ground truth annotations for the image-level dating classification task, we take the final image-level date prediction to be the average over all spatial predictions. In our experiments we use a fixed sparsity $s_l = 20\%$ for all layers of the network.

4.6.5 Quantitative Evaluation

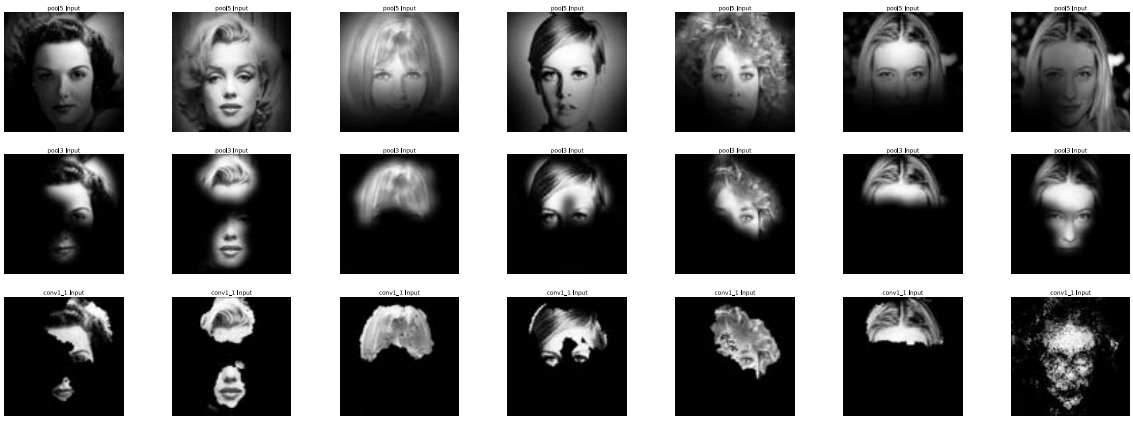
Unfortunately, network visualization papers have historically only provided qualitative evaluations of their results. However, we provide one quantitative measure of the discriminativeness of the discovered regions. We do this by testing how a pre-trained network could predict the year label of Yearbook images **only** from the discovered elements. To this end, we use a network that has been fine-tuned on the original training data to classify the pixel-level discriminative regions for different methods. For each test instance, we start with the training-set mean image and add the color values of the discovered regions (see Figure 4.13). Table 4.2 shows the accuracy of our approach compared with [107] on the resulting images. As expected, our method achieves a higher classification accuracy since it retains more discriminative elements.

4.6.6 Qualitative Evaluation

The results of applying our spatial-unit selection algorithm are shown in Figure 4.14 and compared to the results of the Simonyan et al. [107] method. Our algorithm extracts image parts that are meaningful for dating such as 40's and 50's red lipstick, 60's flat bangs, 80's curls and 90's hair partings. In comparison, the

Method	Accuracy	Avg L1 Error	Med L1 Error
thresholded scoremap	0.034	16.71	12.0
Simonyan et al. [107]	0.017	24.0	20.0
ours	**0.033**	**18.1**	**11.0**

Table 4.2: Classification accuracy and errors on visual elements.



(a) 1945 (b) 1953 (c) 1966 (d) 1966 (e) 1989 (f) 1999 (g) [107]

Figure 4.14: (a)-(f) Results on celebrity portraits from different eras. (g) In comparison, [107] tends to focus on the middle of the object, the nose and forehead. While the unit selection process is a hard-selection, we shade the receptive field of each unit in the pooling layers using a tent filter for displaying purposes.

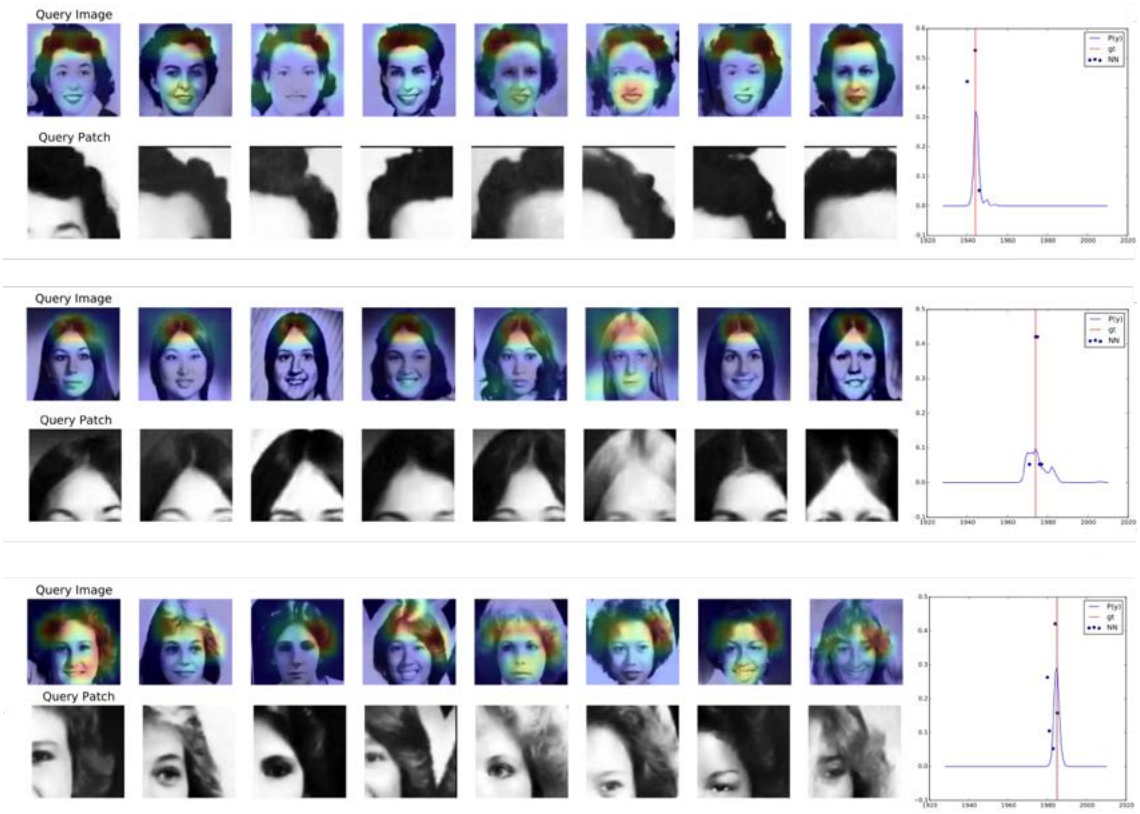


Figure 4.15: The nearest neighbors of the most important patch in a query image (left column) are similar both visually and temporally. Compare the query image prediction distribution (blue line), the ground truth year (red) and the distribution of the ground truth years of the nearest neighbors (blue dots).

Simonyan et al. method tends to pick out the center of the object, here the forehead and nose of the depicted person, which is less relevant for predicting the era of the photograph. The images used here are all correctly predicted images from the *unseen* set of female celebrity portraits.

While the selected spatial regions in Figure 4.14 seem reasonable, we further ensure that the most important visual elements selected are consistent across images of the same era. For the most important pixels of a given query image, we look for nearest neighbors in the learned feature space by picking one $pool_5$ spatial feature (all 512 channels) as the feature for each image and normalized correlation as the distance metric. In Figure 4.15 we visualize for each image a heat map of the spatial locations selected by our algorithm as most useful for the classification task and the most important patch for a query image (left column). We then display its nearest

neighbors to the right, and compare the prediction distribution for the query image (blue line) with the ground truth year (red) and the distribution of the ground truth years of the nearest neighbors (blue dots). We note that the nearest neighbors are similar to the query patch both visually and temporally. For example, the curls are the element that signifies the 40's; the parting of the hair is unique to the 70's and the feathered hair and dark eye shadow are discriminative of the 80's. Referring back to Figure 4.9, we have verified that we can localize the visual elements that resulted in these full image decade clusters.

4.7 Discussion

In this chapter, we presented a large-scale historical image dataset of yearbook portraits, which we have made publicly available. These provide us with a unique opportunity to observe how fashions and habits change over time in a restricted, fixed visual framework. We demonstrated the use of various techniques for mining visual patterns and trends in the data that significantly decrease the time and effort needed to arrive at the type of conclusions often researched in the humanities. We showed how deep learning techniques can leverage the time-specific visual information in a single facial image to date portraits with great accuracy. Moreover, we presented a technique to visualize which parts of the image are used in dating the portraits thus finding the discriminative visual elements of each time period.

Much remains to be done in the application of machine learning techniques to visual historical datasets, and in particular the one at hand. For example, historical yearbook portraits can be used to discover the cycle-length of fashion fads and can be used as a basis of data-driven style transfer algorithms. Ultimately, we believe that data-driven methods applied to large-scale historical image datasets can radically change the methodologies in which visual cultural artifacts are employed for humanities research.

Chapter 5

Learning to Disentangle the Time-Varying Illumination from the Permanent Geometry in Urban Scenes

Figure 5.1: **We learn to disentangle temporally-varying scene factors from permanent ones.** We can manipulate the learned factors to relight scenes, e.g., by editing sun position and sky conditions. While we train our model on panoramas of NYC *(top)*, it generalizes at test time to images of other cities such as Paris *(bottom)*.

We propose a learning-based framework for disentangling outdoor scenes into temporally-varying illumination and permanent scene factors. Inspired by the classic

intrinsic image decomposition, our learning signal builds upon two insights: 1) combining the disentangled factors should reconstruct the original image, and 2) the permanent factors should stay constant across multiple temporal samples of the same scene. To facilitate training, we assemble a city-scale dataset of outdoor timelapse imagery from Google Street View, where the same locations are captured repeatedly through time. This data represents an unprecedented scale of spatio-temporal outdoor imagery. We show that our learned disentangled factors can be used to manipulate novel images in realistic ways, such as changing lighting effects and scene geometry.[1]

5.1 Motivation

> *"The city of Sophronia is made up of two half-cities... One of the half-cities is permanent, the other is temporary."*
>
> — Italo Calvino, *Invisible Cities*

Imagine taking an image from every possible location on Earth at every possible time instant throughout history. Adelson and Bergen called this hypothetical construct the *plenoptic function* [1]. In practice, of course, it would be impossible to capture or store such a massive dataset. Yet, the data must also be highly redundant and compressible. There will be many images of the same view with slightly different illumination, many images capturing different places under the same conditions, etc. In other words, each image within this hypothetical dataset should have a low intrinsic dimensionality. Rather than store all pixels, we could instead store a small number of intrinsic, disentangled factors representing scene geometry, illumination conditions, etc.—if only we knew what those parameters were and how to reconstruct an image from them.

In this chapter, we ask whether we can learn such a lower-dimensional representation from a sparse sampling of the plenoptic function on the scale of an entire city. Until recently, large-scale visual data that varies both in space and, separately, in time was difficult to obtain. Fortunately, there have been systematic efforts to capture the world through projects like Google Street View (GSV). While GSV is known for its worldwide coverage, it has also accumulated many samples of the world

[1] This work was first published as *Learning to Factorize and Relight a City* in ECCV, 2020 [132].

over time, powering features like Street View Time Machine (GSV-TM). However, GSV-TM still represents an extremely sparse sampling of the plenoptic function.

We use GSV to learn to factor a city's worth of outdoor panoramas into a single low-dimensional representation. In particular, we organize a large set of historical GSV panoramas of New York City into *assembled timelapses* at 100,000 fixed locations captured over time. These enable us to train an unsupervised model to disentangle two latent factors: illumination factors that vary over time, and geometric scene properties that are more permanent.

Once we learn a disentangled set of latent factors, we can synthesize missing data in our incomplete sampling of the plenoptic function by simply swapping or modifying the underlying factors. As illustrated in Fig. 5.1, our learned factorization can generate synthetic images of the same scene with completely novel illumination. Our disentangled factors are flexible enough to relight test scenes from a single panorama and can even be applied to entirely new cities like Paris.

5.2 Background

Intrinsic Images. Decomposing images into their underlying components is a well-studied problem [133]. For instance, the classic intrinsic images problem describes images as a combination of *reflectance* (i.e., scene albedo), and *shading* (effects induced by lighting) [134]. This problem is underconstrained as there are an infinite number of possible solutions for a single image. However, the regularities in natural scenes and lighting conditions allow for priors on the decomposition. While such priors can be manually crafted [135], many recent methods attempt to learn priors from data, using full supervision from synthetic data [136], sparse supervision from human annotations [137, 138], or self-supervision from synthetic models [139]. Yet another kind of supervision comes from *timelapse videos* [140], which feature image sequences with constant reflectance but varying illumination. Such work hearkens back to classic work on deriving intrinsic images from image stacks [141], and is an inspiration for our work. However, while intrinsic image methods allow for editing reflectance or shading for a specific image, they use high-dimensional *pixel-level* descriptions of lighting that are not transferable across scenes. In our case, our model learns an illumination descriptor that can be meaningfully transferred from one image to another, e.g., to relight an image with an illumination from a completely different scene. Such "mix-and-match" capabilities are beyond the power of standard intrinsic images.

Inverse Graphics. An alternative way to factor visual appearance is via 3D reconstruction of the scene into underlying physical components like 3D shape,

materials, and lighting. Such methods have been successful in several specific domains, including faces [142], single objects [143, 144], or indoor scenes trained from synthetic data [145, 146]. 3D reconstruction has also been used explicitly as a preprocess to aid in modeling visual appearance [147–150]. Most relevant to us are Martin-Brualla *et al.* [147], who organized millions of internet photos into a dense 3D and temporal reconstructions, and Meshry *et al.* [150], who employed a dense 3D reconstruction with a neural rendering pipeline to synthesize scene appearances. However, explicit 3D reconstruction methods require hundreds of images to create a 3D model and cannot generalize to novel test-time scenes. In contrast, we choose to handle geometry implicitly—allowing us to holistically learn to disentangle factors across many scenes composed of a few images each, and then generalize to novel settings, even single images.

Some recent inverse graphics methods learn to infer shape, appearance, and materials for new outdoor scenes, not just scenes observed during training. Yu and Smith train on multi-view stereo data using a physics-based inverse graphics model, and can infer explicit scene properties for novel test images, enabling relighting tasks [151]. Our work achieves a similar capability, but relies on a more implicit representation of geometry and illumination that can be learned solely from timelapse data, without requiring depth or surface normals during training.

Timelapse and Webcam Data. Timelapses are a popular source of data for capturing time-related effects. Applications include intrinsic images [140, 141], scene-specific factorizations via physical shading models [152], illuminant transfer [153], analysis of worldwide temporal variations [154], motion denoising [155], learning temporal object transformations [156], and weather attribute manipulation [157]. However, prior work is limited by the variety and size of available data. The largest existing set of standard webcam data is the AMOS dataset of Jacobs *et al.* [154], which archived 29,445 webcams and 95 million images. BigTime [140] uses a much smaller set of 6,500 images from 195 timelapse sequences. Both datasets sample *time* much more densely than *space*. In contrast, we leverage the vast amounts of data from Google Street View to create *assembled timelapses* of the same location captured at different times, across a large number of locations. This allows us to collect an order of magnitude more data than previously published [154]. We additionally note that data collection from Street View scales more easily than [154] which requires crawling the internet for webcam streams.

Learning from Street View. Google Street View (GSV), a large dataset of images sampling much of the world's streets, represents a compelling source of data for computer vision research. Researchers have utilized Google Street View images to learn about visual elements [102] or historical architectural styles [96]

Figure 5.2: **GSV-TM Data.** *Left:* A Manhattan intersection. *Center:* Multiple Google Street View panoramic captures of this intersection forms an *assembled timelapse* stack. *Right:* The train and test split over the greater NYC area. Training stacks are drawn from the blue region, and test stacks from the yellow region.

specific to certain cities like Paris, to predict non-visual city attributes [158–160], for localization [161], or to understand the relationship between satellite imagery and street-level views [162]. In our work we use historical GSV Time Machine imagery to observe how the world changes over time by assembling timelapses for a large number of locations. Such a large, comprehensive dataset is key to our unsupervised approach for learning to factor illumination from scene geometry.

5.3 Google Street View Time Machine Data

Google Street View (GSV) hosts an amazing quantity of panoramas capturing street scenes worldwide. Because GSV repeatedly captures many places over time, it can be treated as a sparse, imperfectly aligned, and irregularly-sampled collection of timelapse videos. These historical images are saved as part of the GSV-Time Machine (GSV-TM), which we mine to collect our dataset.

We focus on New York City, due to the richness of NYC scenes and the relative wealth of data. To assemble timelapses, we collect panoramas within NYC along with their timestamps and camera poses in a geographic coordinate system [163]. We greedily cluster nearby panoramas into sets of eight, which we refer to as *stacks*. The region we use and an example stack are shown in Fig. 5.2.

From the area shown in Fig. 5.2 (right) we collect ∼100K assembled timelapse stacks for training (comprised of 800K individual panoramas stitched from 10 million captures) and 16K test stacks. We crop the sky and ground regions such that our final panoramas are 960×320. These sRGB panoramas can optionally be gamma-corrected before further processing.

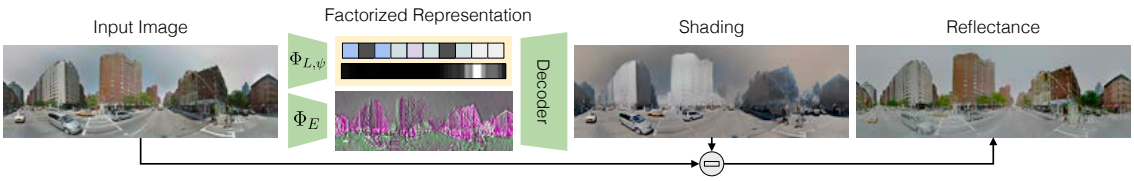


Figure 5.3: **Disentangling a single image.** At test time, we *encode* a single image into disentangled time-varying and permanent factors. We train with the constraint that shading and reflectance images can be *decoded* from this learned factored representation.

5.4 Method

Our goal is to discover a low-dimensional representation of the world where temporally varying effects, such as different illumination conditions, are disentangled from permanent objects, such as buildings and roads.

One form of disentanglement is *intrinsic images*, a per-pixel decomposition into reflectance and shading images. However, such a disentangled representation is very low-level—a particular shading image cannot be used to relight a different scene. Instead, we seek to encode an image into higher-level latent factors capturing scene and illumination properties described above, as illustrated in Fig. 5.3. How can we find such a factorization? Our insight is that we should still be able to *decode* intrinsic images from our factored representation, as illustrated on the right side of Fig. 5.3. The decoded reflectance and shading images should recombine to form the original image, providing us with an autoencoder-style method for learning our high-level factorization [139]. However, such an image reconstruction framework alone would provide a very weak supervision signal. Our second insight is to learn from huge numbers of *timelapse stacks* mined from GSV-TM. Within such stacks, we assume the scene factors to be constant. This insight is inspired by the work of Li and Snavely, who learn intrinsic images from timelapse videos [140]. In our case we learn a high-level factorization that enables more powerful capabilities.

5.4.1 Encoder-Decoder Architecture

Fig. 5.3 shows our encoder-decoder architecture with its learnt factored representation. Given an image, our encoders produce latent factors, capturing various temporal and permanent effects, that can be decoded to a log-shading intrinsic image. We use the intrinsic images equation ($\log(\text{Reflectance}) = \log(\text{Image}) - \log(\text{Shading})$) to compute a reflectance image by subtracting the temporally varying effects, represented

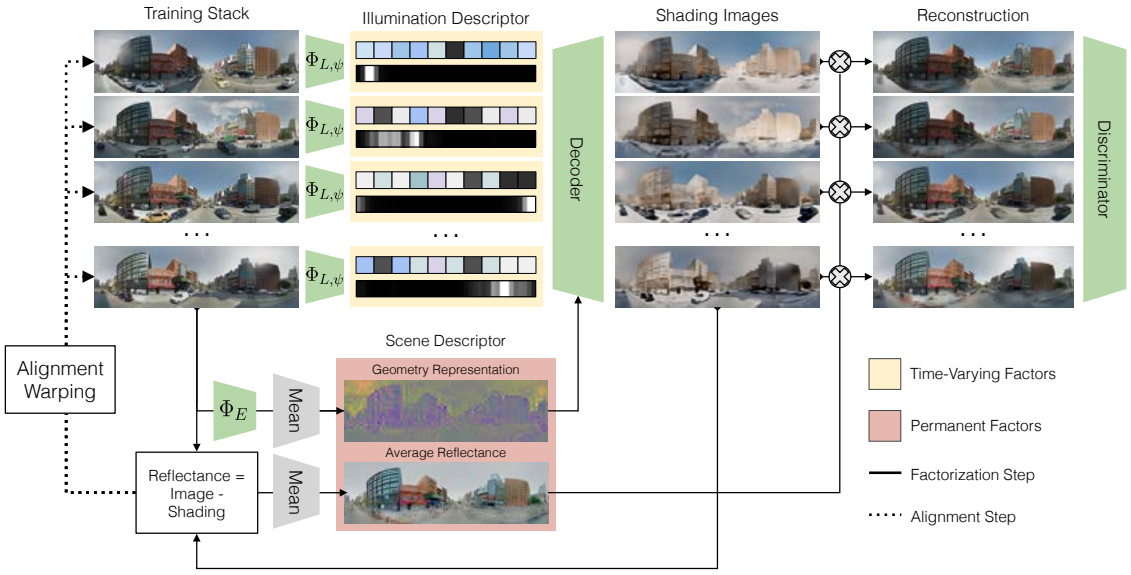


Figure 5.4: **Training with timelapses.** We train encoders to disentangle an assembled timelapse stack into two factors: *illumination descriptors* that capture the time-varying aspects of each image, and a single *scene descriptor* that captures the permanent elements of the entire timelapse stack, such as the scene geometry. We train a generator to transform the disentangled factors into shading and reflectance images from which we can reconstruct the original images. As indicated by the dotted pathways, we also simultaneously solve for the alignment of the individual frames in the input timelapse.

by the shading image, from the original image.

Our model's latent factors are organized into two sets of descriptors, as shown in Figure 5.4: an *illumination descriptor* represents temporally varying aspects of the scene and a *scene descriptor* represents the permanent aspects.

Illumination Descriptor: Our illumination descriptor captures the factors of the world that encode temporal variation like lighting. This descriptor is comprised of two disentangled sub-factors:

The *lighting context* $L \in \mathbb{R}^{32}$ is a global latent feature that captures the overall ambient illumination properties, such as atmospheric conditions and cloud cover. Our lighting context encoder Φ_L encodes an image to this embedding.

The *sun azimuth angle*, φ is an explicit factor representing the horizontal position of the sun in a given panorama. We model sun azimuth explicitly because, unlike illumination patterns, variations in sun azimuth have a simple geometric meaning,

with a value in the range $[-\pi, \pi]$. Despite this simple parameterization, the effect of sun azimuth on a rendered scene is highly complex. Therefore an explicit azimuth factor allows our model to combine the factor's underlying mathematical simplicity with a network's ability to model complex behaviors.

Rather than regress to a scalar angle, we instead represent φ internally as a discretized distribution over sun angle (with $k = 40$ bins). Inspired by prior work on illumination estimation [164], our azimuth encoder Φ_φ is a horizontally fully-convolutional network that takes as input a panorama, and produces a 40-way softmax distribution φ, where each bin corresponds to the probability that the sun azimuth is located in the bin's corresponding angular range. Note that given this discrete distribution over angles, we can differentiably compute a single scalar angle as the (circular) expectation of the distribution, $\bar{\varphi}$. This predicted scalar sun angle is used by our decoder for normalizing sun position.

Scene Descriptor: Our scene descriptor captures the permanent structure of the world that is invariant to the temporally varying effects described above. We also divide this descriptor into two disentangled sub-factors:

The *geometry representation* is a spatial map of learned features that captures scene properties (e.g. surface normals and material properties) that are independent of illumination, but nonetheless are important to determining the rendering of a shading images. The fully convolutional encoder Φ_E outputs $E \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 16}$ where H and W are the resolution of a panorama.

The *reflectance image* is an RGB estimate of the underlying scene albedo. In contrast to the shading image, we chose to not use an encoder-decoder to compute reflectance for two reasons: (1) neural networks can have difficulties preserving high-frequency textures that are important for visual quality and (2) it suffices to predict only one intrinsic image component because its complement component has a closed form solution based on the intrinsic images equation.

Decoder: Given a set of learned factors (sun azimuth angle $\bar{\varphi}$, lighting context L, and geometry factor E), our decoder G is trained to generate an outdoor shading image. To facilitate training of G, one insight is that it is easier to learn to synthesize shading images with a fixed sun azimuth angle than with all possible angles. Further, we can normalize a panorama by its predicted sun azimuth angle by simply rotating it by the negative of that angle (i.e., circular horizontal translation). Hence, our decoder operates as follows: (1) use the predicted sun azimuth angle $\bar{\varphi}$ to rotate the geometry factor image E to a fixed sun angle, (2) decode the sun-normalized geometry image with lighting context L to a shading image, and (3) rotate the result back to the original coordinate frame.

We use the Spatial Adaptive Instance Normalization (SPADE) generator of Park *et*

al. [165] to model the complex interactions between geometry and illumination in our decoder G. The SPADE generator takes the lighting context L as the network's noise input. We apply the insights from above and rotate the geometry representation E by $-\bar{\varphi}$ before using it as the SPADE conditioning.

While some prior works model shading with a grayscale image, such a model cannot capture real-world, colored illumination. Inspired by Sunkavalli *et al.* [152], we augment our decoder's gray-scale shading predictions with a bi-color assumption by additionally predicting two global color illuminants c_1 and c_2, corresponding to sunlight and skylight, and a per-pixel mixing weight M that models how much each pixel is illuminated by the sun or sky.

5.4.2 Training

Learning to factor single images without *any* supervision is challenging—there is simply not enough information in a single image to disentangle scene factors from illumination factors. However, a GSV-TM stack depicts the same underlying permanent scene under diverse temporally varying illuminations, providing a useful training signal. Our training procedure, shown in Fig. 5.4, learns to disentangle factors *within* a stack by separating the permanent geometry of the scene shared by all images in the stack from the varying lighting. The trained model can be applied to a single image at test time.

Given a timelapse stack, we run our encoder on individual frames to get a stack of encoded geometry representations and illumination descriptors. Because we assume the stack's geometry to be constant across time, we average the encoded geometry maps over the stack, resulting in a single shared geometry map, $\bar{E}$. From this shared geometry map, and the per-image illumination factors, our decoder produces a stack of shading and reflectance image pairs. As with geometry, we wish the scene's albedo to be constant across time. Accordingly, we impose a reflectance consistency loss $\mathcal{L}_{\mathsf{RC}}$ that computes the L_1 distance between pairs of reflectance images from different frames. This loss encourages the encoder-decoder network to remove temporal variation from the encoded permanent factors such that the reflectances are constant across a stack.

As demonstrated in the right half of Fig. 5.4, we average the stack's reflectance images across frames to get the stack's shared reflectance. The shared reflectance is recomposited with the shading image of each frame in the stack to reconstruct the original pixels of each input frame. These reconstructions are used to drive the learning process via image synthesis losses.

Figure 5.5: **Alignment results.** We show stack averages, cropped for emphasis, before and after our alignment process. Aligning the estimated permanent reflectances rather than the input images results in good alignment and therefore crisp stack averages.

5.4.3 Stack alignment

Unlike traditional webcam data, our assembled GSV-TM timelapses do not come from stationary cameras. While each stack consists of nearby panoramas, they are not perfectly co-located and aligned. As shown in Fig. 5.5, the average of the stack reveals visible misalignment artifacts resulting from this parallax.

We could use 3D reconstruction methods as the basis for image alignment, but opted for a simpler 2D approach inspired by image congealing [166], and compute 2D warps that best align the images in each stack. Given a raw stack of imperfectly aligned images, we define Θ, an 8×32 grid of per-image control points initialized as the identity warp. The control points define a 2D spline used to differentiably warp each image within a stack to align with the rest.

To find the control points that best align images within a stack, we run gradient descent to minimize pixel alignment error. While one could use original image pixels to measure misalignment, we found that photometric differences across the stack due to varying lighting conditions led to poor alignments. Instead, we compute error on estimated *reflectance* images by reusing our previously defined reflectance consistency loss, $\mathcal{L}_{\mathsf{RC}}$, to update alignment parameters. This approach is indicated by the dotted pathway in Fig. 5.4. By jointly minimizing alignment and intrinsic image decomposition, we create a positive feedback loop—as timelapse alignment improves, factorization becomes easier and vice versa.

5.4.4 Losses

Our losses are optimized over alignment parameters Θ, factorization encoders Φ_L Φ_φ, and Φ_E, and decoder G. We train a multi-scale patch discriminator [43,84] D to ensure that the stack reconstructions with shared reflectances look realistic.

Our primary loss for learning the disentanglement is the reflectance consistency loss $\mathcal{L}_{\mathsf{RC}}$ described in Sec. 5.4.2. We include standard image generation losses on the reconstructed stack to ensure high quality synthesis results: a perceptual loss $\mathcal{L}_{\mathsf{VGG}}$ [85], an adversarial loss $\mathcal{L}_{\mathsf{GAN}}$ [167], and a feature matching loss $\mathcal{L}_{\mathsf{FM}}$ [84]. Finally, because intrinsic images have a fundamental color ambiguity, we also include a white light penalty, $\mathcal{L}_{\mathsf{WL}}$ that biases our encoder-decoder towards white-balanced reflectance outputs. Our overall objective function is:

$$\min_{\Theta} \max_{D} \min_{G,\Phi_L,\Phi_\varphi,\Phi_E} \mathcal{L}_{\mathsf{RC}} + \mathcal{L}_{\mathsf{Gen}} + \mathcal{L}_{\mathsf{GAN}} \tag{5.1}$$

where $\mathcal{L}_{\mathsf{Gen}}$ is a weighted sum of $\mathcal{L}_{\mathsf{FM}}, \mathcal{L}_{\mathsf{WL}}, \mathcal{L}_{\mathsf{VGG}}$ that measures the generative quality of the reconstructed images.

5.5 Experiments

We evaluate our factorization method in two ways: 1) we compare to intrinsic image decomposition baselines in the single-scene setting, and 2) we apply our method to the task of transferring illumination descriptors across different scenes, a new capability enabled by our disentanglement. In both cases, we measure success by the quality of reconstructed images derived from swapping their disentangled factors with ones borrowed from other images as in [138].

Data. At test time, our network can take as input either an assembled timelapse stack or a single panorama. In order to align test-time stacks like those shown in Fig. 5.5, we estimate spline parameters by computing a gradient for alignment only, while keeping the weights of the factorization part of the network frozen. Below, we present results for stack as well as single-image inputs.

In particular, we show single-image test-time results on GSV imagery from cities never seen during training, such as Paris, as well as images from the Outdoor Laval HDR dataset [164]. This dataset contains HDR panoramas of outdoor scenes that are tonemapped to sRGB to match GSV. We use this data to compare to existing sRGB intrinsic image methods and to test generalization from GSV to a different domain of panoramas.

Baselines. Given the novelty of our problem, we perform model ablations to measure the individual benefits of various components. All ablated models are trained with

the same losses and number of iterations as our full method. We report results on the following ablations:

- **Mono-color shading**: We ablate the bi-color shading by training our model with a mono-color assumption similar to that of Li and Snavely [140].
- **w/o alignment training**: Trained without the alignment feedback loop.
- **w/ unaligned test stacks**: Uses unaligned test stacks to measure the effect of ablating alignment at training (above) vs. at both training and test time.
- **w/o azimuth encoder**: Our model trained without an azimuth encoder nor normalizing for sun position.

Additionally, we consider the following baselines:

- **Pixel nearest neighbor**: Given a target image, we find the pixel-wise nearest neighbor in its aligned stack and report the error resulting from using that image as our synthesized result.
- **Weiss's MLE Intrinsics** [141]: use handcrafted priors on gradients extracted from image sequences.
- **Zhou *et al.*** [138]: learn to mimic human judgments of relative reflectance.
- **Li and Snavely's BigTime** [140] learn shading priors from image sequences.

5.5.1 Within-Scene Decomposition

Intrinsic image methods aim to decompose an image into shading and reflectance. The quality of a decomposition is measured by its ability to separate illumination effects, like cast shadows, from permanent properties such as albedo. In Fig. 5.6, we show reflectance and shading computed from a single image using our method and the two deep learning baselines. Both BigTime and Zhou *et al.* fail to remove cast shadows, as seen by residual shadows encoded in their reflectance. Unlike Zhou *et al.*, our method produces shading images that are piecewise smooth, as expected for planar surfaces like building facades. BigTime struggles in outdoor settings because their single global illuminant cannot predict multiple illumination colors. Finally, both baselines incorrectly encode blue sky pixels as reflectance despite the fact that sky color is a temporal property. To further illustrate the advantages of our method over these baselines, Fig. 5.7 shows the results of relighting pairs of images of the

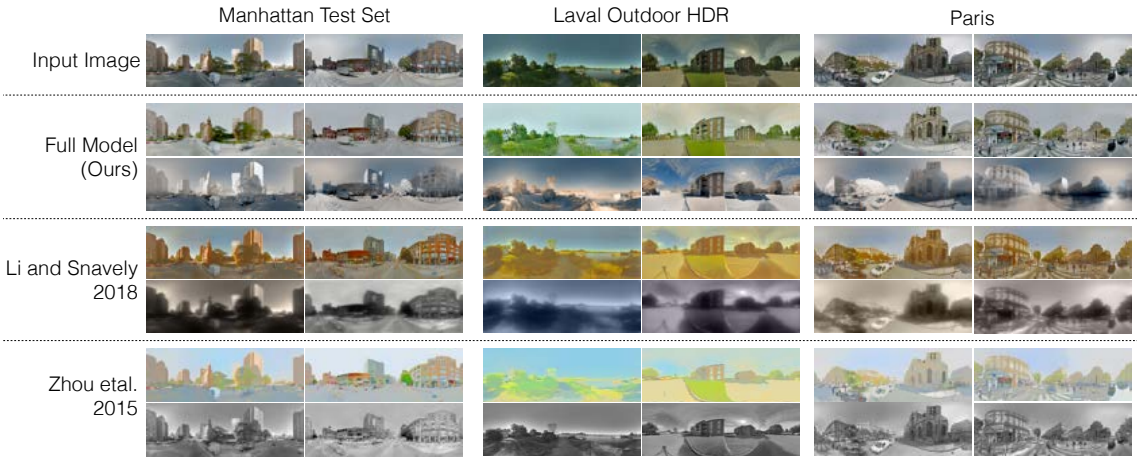


Figure 5.6: **Qualitative results on an intrinsic image decomposition task.** We compare single-image decompositions of our method with Li and Snavely [140] and Zhou *et al.* [138]. Compared to the baselines, our reflectance images do not have residual shadows. Our method, trained on NYC, generalizes at test-time to Laval Outdoor HDR Panoramas [164] as well as to GSV imagery from Paris.

same scene by swapping reflectances within the pair. Unlike the baselines, our clean reflectance image allows us to relight the scene successfully.

Scene consistency verification. Since MLE Intrinsics [141] only works on time-lapse stacks of single scenes, we devise a way to quantitatively compare to their method. We split our aligned test stacks to two smaller substacks of 4 images each. For each substack, each method predicts a single reflectance image and four shading images. Since both substacks capture the same underlying scene, the predicted reflectances should be consistent across the two. As in the case of single images (Fig. 5.6), we can test the consistency of the predicted reflectance for the depicted scene by swapping the predicted reflectance images between the two substacks and reconstructing the four input images in each substack from their shading and *swapped* reflectance images. We refer to this experiment as *scene consistency verification* because the reconstruction error is minimized when the predicted reflectances are identical for the two substacks.

We report the mean squared reconstruction error (MSE) between the input stack and the swap reconstructions in Table 5.1. Our method outperforms the three baselines at image reconstruction in this setting. We speculate that prior methods are hindered by their reliance on hand-defined shading priors and limited training data. In contrast, our massive dataset provides enough supervision for learning a good decomposition without shading priors. Interestingly, ablating the azimuth

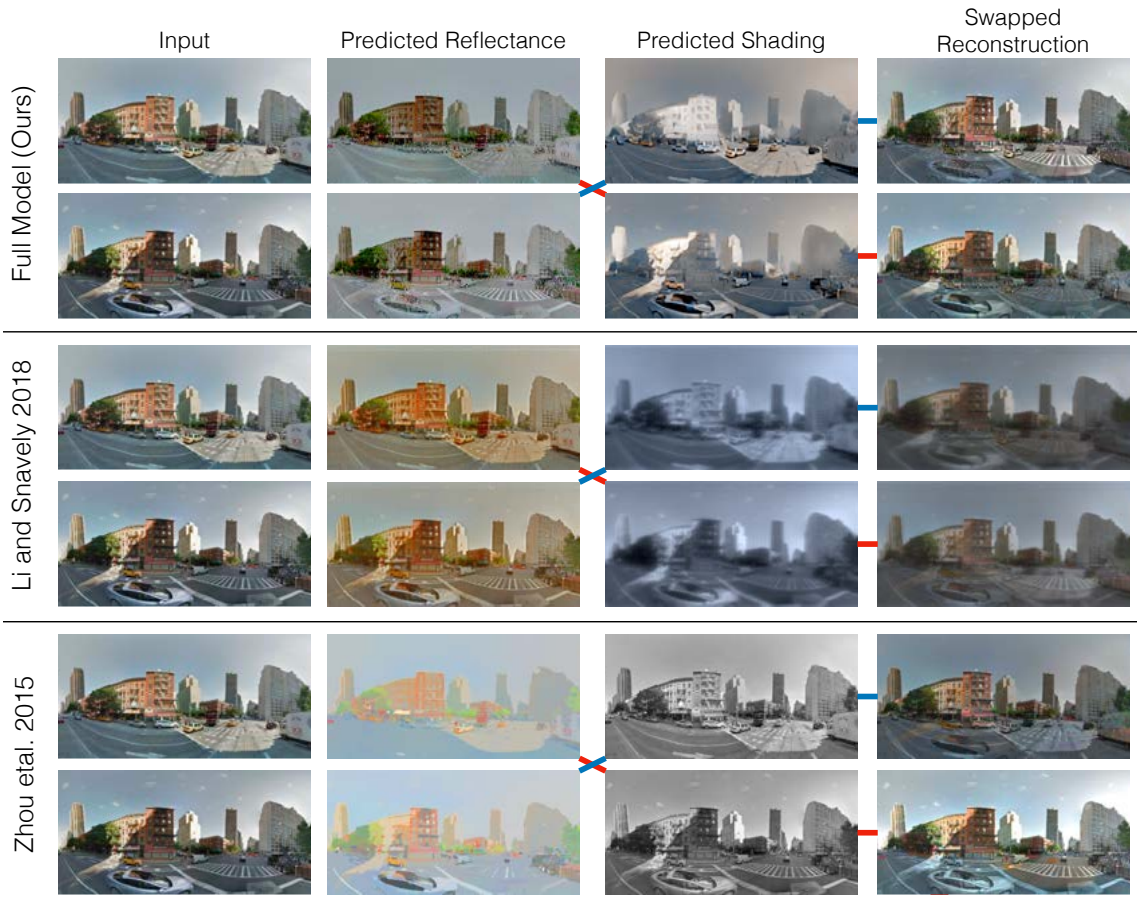


Figure 5.7: **Transferring illumination within a scene.** Given a pair of images of the same scene under different illuminations (*left*), we disentangle the permanent and varying factors and decode their reflectance and shading (*middle*). To test the permanency of the estimated reflectance for the depicted scene, we swap reflectances within the pair and combine them with the estimated shading to reconstruct the original images (*right*). Red and blue paths connect the components used to reconstruct each image. Our method produces a reflectance, clean of any lighting, which can be safely swapped between captures of the same scene and still result in good reconstructions.

encoder does not degrade performance on this task, suggesting that a simpler setup is sufficient for within-scene illumination transfer.

Model	Consistency	Completion
Full model (ours)	**0.071**	**0.196**
Mono-color shading	0.077	0.215
w/o alignment training	0.082	**0.201**
w/ unaligned test stacks	0.090	0.210
w/o azimuth encoder	**0.072**	0.240
Pixel nearest neighbors	0.274	0.278
MLE Intrinsic [141]	0.114	—
BigTime [140]	0.180	—
Zhou *et al.* [138]	0.217	—

Table 5.1: **Relighting results.** We define two image reconstruction tasks for evaluation. *Scene consistency verification* evaluates whether the estimated reflectance is consistent across multiple captures of a single scene. *Space-time completion* evaluates the ability to transfer illumination across different scenes. We report MSE reconstruction error. Lower is better.

5.5.2 Cross-Scene Factorization

Unlike intrinsic images methods, our factorization allows us to transfer illumination descriptors *across* scenes. Using our disentangled factors, we can synthesize a given scene under completely new lighting conditions, borrowed from a *different* location. For the purpose of evaluating the success of this cross-scene relighting process, we devise a way to compare the novel synthesis to ground truth. Namely, because illumination changes relatively slowly, we assume that images captured within 5 minutes across the city have the same illumination descriptor. Hence, we can relight a given scene, A, captured at time T_1 using illumination descriptors transferred from a different location, B, captured at time T_2. We then compare the resulting synthetic image of scene A at time T_2 to ground truth captures of scene A captured at a time close to T_2.

We name this task *space-time matrix-completion*. A row in the matrix represents a unique point in "space" and a column represents a unique point in "time". A single panorama represents an entry in this matrix at the row corresponding to its depicted scene and column corresponding to its capture time. We can withhold entries in the matrix and reconstruct them by combining a scene descriptor derived from images in the same row, with an illumination descriptor extracted from a different scene from the same column. Table 5.1 shows the reconstruction MSE for each ablation between held-out and reconstructed views. Our full model and the *w/o alignment training* ablation show significant improvements over other ablations.

While alignment training does not significantly affect the performance of our model on this task (*w/o alignment training*), its performance degrades significantly on unaligned stacks (*w/ unaligned test stacks*). This indicates that alignment may be optional during training but is crucial for reconstruction. Additionally, unlike with the substack swap task, explicitly representating sun azimuth improves transferability of lighting descriptors across scenes.

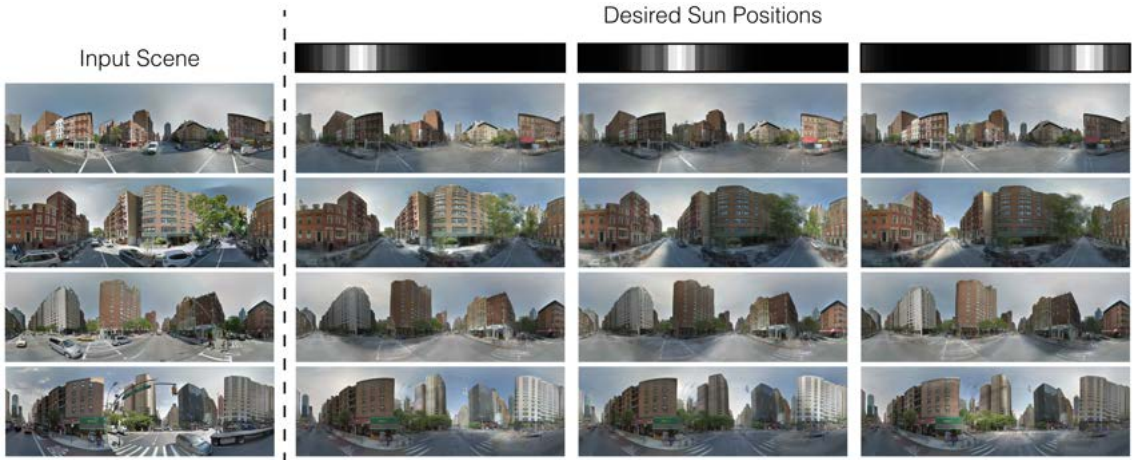


Figure 5.8: **Manipulating sun position.** We can specify the sun position for an input scene and relight it realistically.

5.6 Applications

We now present applications where we synthetically modify a panorama. These applications are uniquely enabled by our intrinsic factorization that disentangles time-varying effects from the permanent scene properties.

Changing sun position. Our model disentangles sun azimuth angle from scene and lighting context factors. Once a scene is factorized, we can visualize what a scene looks like when the sun angle is changed. Fig. 5.8 shows examples of test scenes synthesized with new sun azimuth angles. Note that cast shadows and illumination on building faces change realistically with the rotation.

Relighting a *novel* scene. Our lighting context encodes the stylistic quality of illumination. As shown in Fig. 5.9, we can transfer the whole illumination descriptor, including sun azimuth, from one panorama to another with a new scene geometry.

Editing scene geometry. While shading and azimuth capture the essence of time, the scene descriptor encodes structures. By copy-pasting regions of the scene descriptors, we can transplant the buildings into new panoramas and relight them to match the scene.

5.7 Discussion

We proposed a novel source of large-scale timelapse data from historical Street View data, and a learning-based method for factorizing temporal and permanent

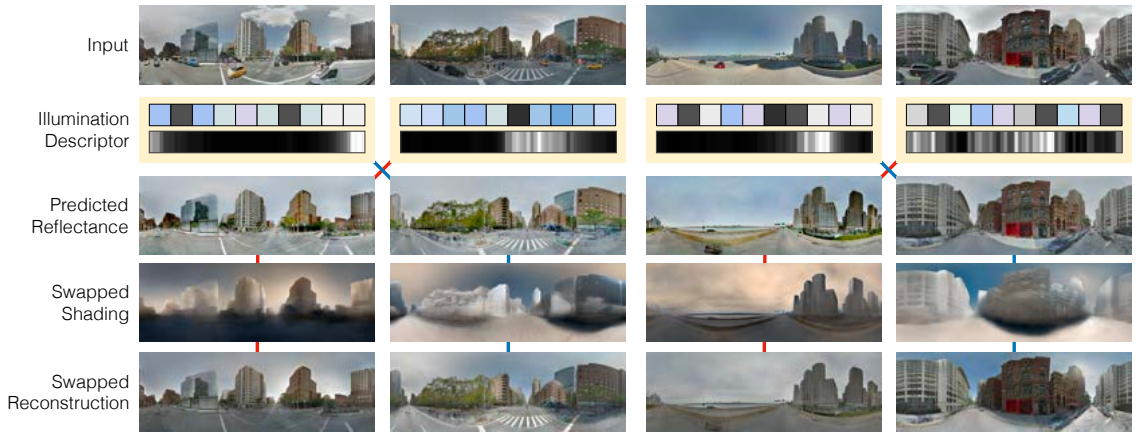


Figure 5.9: **Changing sky illumination.** We can relight *novel* scenes by transferring the disentangled time-varying factors from one scene to another. Here we swap the illumination descriptors of a pair of input scenes to visualize what each scene might look like under a new illumination. The red and blue paths indicate the the components used to reconstruct each relit scene.

variations across imagery covering an entire city. Our learned factorization outperforms state-of-the-art intrinsic images methods, and enables cross-scene style transfer via manipulating our learned factors.

Our method has a few limitations. First, the scene descriptor learns to encode transient objects like cars. While moving objects are temporal effects, the network chooses to encode them in the scene descriptor, resulting in wispy cars appearing in the generator output. Second, high-frequency details such as cast shadows from tree branches are difficult to synthesize. Third, when the alignment module fails, the shared reflectance of a stack will appear blurry. Finally, when our permanence assumptions fail to hold—for instance when buildings are repainted or rebuilt—our assumption that the scene descriptor is constant across time is violated.

Despite these limitations, our work points towards a new approach to modeling and synthesizing the space of outdoor scenes, wherein we can learn to separate factors that persist at different time scales. An intriguing direction for future work is to expand to a richer range of timescales, for instance modeling transient effects, effects with annual cycles, like seasons, or long-term changes like weathering.

Part III

How Should We Test For Rich Perception?

Chapter 6

Abstract Art as a Perceptual Testbed

Although the human visual system is surprisingly robust to extreme distortion when recognizing objects, most evaluations of computer object detection methods focus only on robustness to natural form deformations such as people's pose changes. To determine whether algorithms truly mirror the flexibility of human vision, they must be compared against human vision at its limits. For example, in Cubist abstract art, painted objects are distorted by object fragmentation and part-reorganization, to the point that human vision often fails to recognize them. In this chapter, we evaluate existing object detection methods on these abstract renditions of objects, comparing human annotators to four state-of-the-art object detectors on a corpus of Picasso paintings. Our results demonstrate that while human perception significantly outperforms current methods, human perception and part-based models exhibit a similarly graceful degradation in object detection performance as the objects become increasingly abstract and fragmented, corroborating the theory of part-based object representation in the brain.[1]

6.1 Motivation

The human visual system is amazingly robust to abstractness of object representation. While we can recognize people in natural, realistic images such as camera snapshots, we are also easily able to identify human figures rendered in numerous forms of artistic depiction such as oil paintings, cartoons and line drawings, despite the fact that they frequently have little in common with a real human being in terms of texture, color and form. At the extreme, abstract artists intentionally push the

[1]This work was first published as *Detecting People in Cubist Art* in Visart Workshop on Computer Vision for Art Analysis, ECCV 2014 [168].

envelope of human vision to the point at which objects are completely unrecognizable, yet we still find such distorted shapes reminiscent of the human form. Computer vision detection algorithms can also recognize objects outside the realm of natural images, though their models of the visual world may not always align with the human one [169]. Since algorithmic object detection in computer vision is usually compared against humans using natural images, such differences are seldom apparent. However, if we seek to design algorithms that achieve the power and flexibility of the human visual system by mimicking it, we should attempt to align the visual models of the detectors we train with the human model. Moreover, we must also evaluate their correspondence on images that stretch the limits of human vision.

In this chapter, we employ the abstract depiction of objects in the Cubist paintings of Picasso as one such test corpus. Cubist paintings depict radically fragmented objects as if they were seen from many viewpoints at once, breaking them into medium sized cube-like parts that appear out of their natural ordering and do not conform to the rules of perspective [170]. Because the abstract objects present in Cubist art are not normally seen in nature, the human visual system must strain to recognize them. Since humans are usually still able to identify the depicted object, Cubism shows us that human perception does not rely on exact geometry and is tolerant to a rearrangement of mid-level object parts. However, findings from neuroscience show that this ability degrades as images become more scrambled or abstract [171] [172]. We use the fragmentation and reordering of parts in Cubist paintings as an example of extreme conditions for the human visual system. Our claim is that if a method mimics human perceptual performance well, then it should behave similarly in these (as well as other) extreme conditions. We therefore aim to test whether there are existing detection methods that behave like human vision under these conditions. This stands in stark contrast to the common practice of evaluating computer vision models on datasets consisting only of camera snapshots and represents a novel contribution, as there is little research into how current systems perform on novel input.

We choose to focus on part-based detection methods that permit the rearrangement of medium-complexity parts, as they have been proven to do well at representing naturally occurring form deformations [173], and evaluate them in comparison to both human participants and object-level detection methods. Moreover, in order to chart the performance of the methods as human vision approaches its limit, we ask participants to divide the paintings into subsets according to the level of their abstractness and compare the performance of the humans and detection methods on each subset. Our results show that (1) existing part-based methods are relatively successful at detecting people even in abstract images, (2) that there is a natural correspondence between user ratings of image abstraction in the Cubist sense and

part-based method performance, and (3) that these properties are not nearly as evident in non-part-based methods. By demonstrating that part-based methods mimic human performance, we both show that these methods are valuable for object recognition in non-traditional settings, and corroborate the theory of part-based object representation in the brain.

6.2 Background

Since in most tasks the human visual system serves as an upper bound benchmark for computer vision, some studies focus on characterizing its capabilities at the limit. For instance, Sinha and Torralba examined face detection capabilities in low resolution, contrast negated and inverted images [174]. In other cases, computational models are used to test the validity of theories from neuroscience [175]. We take inspiration from these studies and evaluate human object detection in man-made art in order to provide a less restrictive benchmark of robustness to form abstraction and deformation than natural images. By using this benchmark we hope to discover parallels between the characteristics of human and algorithmic object detection.

From research in neuroscience, we know that the human visual system can detect and recognize objects even when they are manipulated in various ways [176]. For instance, humans are able to recognize inverted objects, although their performance is degraded, especially when the objects are faces or words [174, 177]. Similar results were obtained when comparing scrambled images to non-scrambled ones, leading to a theory of object-fragments rather than whole-object representations in the brain [178, 179]. This theory is strengthened by recordings from neurons in the macaque middle face patch that indicate both part-based and holistic face detection strategies [180]. Thus, although humans are capable of recognizing images distorted by scrambling, they are less adept at doing so. By analogy, we might expect methods trained on natural images to suffer a similar degradation in the face of a reorganization of object parts.

Object detection is one of the prominent unsolved problems in computer vision. Traditionally, object detection methods were holistic and template-based [116], but recent successful detection methods such as Poselets [181, 182] and deformable part-based models [173] have focused on identifying mid-level parts in an appropriate configuration to indicate the presence of objects. Other part-based methods discover mid-level discriminative patches in an unsupervised way [102, 111], use visual features of intermediate complexity for classification [175], or rely on distinctive sparse fragments for recognition [183]. Finally, a model inspired by Cubism itself that assembles intermediate-level parts even more loosely has shown success in detecting

objects [184]. Another approach to detection that has recently shown remarkable detection results is based on convolutional neural networks [185, 186]. We discuss the methods that we have chosen to benchmark in Section 6.4.

6.3 Cubist 'fragments of perception'

While there are many kinds of visual deformation to which human vision is robust, we choose to focus on the object fragmentation and part-reorganization exhibited by Cubist paintings as it has an appealing correlation with the strengths of part-based detection methods. Cubism is an art movement that peaked in the early 20th century with the work of artists such as Picasso and Braque. Cubist painters moved away from the two-dimensional representation of perspective characteristic of realism [170]. Instead, they strove to capture the perceptual experience of viewing a three dimensional object from a close distance where in order to perceive an object, the viewer is forced to observe it part by part from different directions. Cubist painters collapsed these 'fragments of perception' onto one two-dimensional plane, distorting perspective so that the whole object can be viewed simultaneously. Despite the abstraction of form, the original object is often readily detectable by the viewer as the parts chosen to represent it are naturalistic enough and discriminative enough to allow for the recognition of the object as a whole. However, this becomes harder with the degree of departure from reality [172].

The fact that humans can detect non-figurative objects in Cubist paintings without prior training makes these paintings well-suited to benchmark robustness to abstractions of form in detection methods trained on natural images. In order to provide intuition that part-based models will be able to perform well on this task, we provide some initial evidence that computer vision methods can successfully identify key parts in the Cubist depictions of objects. We train an unsupervised discovery method of mid-level discriminative patches on the PASCAL 2010 "person" class images [102, 111, 187], and compare the part-detector activations on natural training images and Cubist paintings by Picasso in Figures 6.1 and 6.2. Despite the difference in low-level image statistics, the detectors are able to discover the patches that discriminate people from non-people in both image domains. In the rest of the chapter, we build on these results to test whether part-based object models can use the detected parts in order to recognize the depicted objects as a whole.

[2]For each detector, the 10 most confident activations are presented by decreasing confidence, excluding lower confidence duplicates of 50% overlap or more. Detectors are sorted by the average score of their 20 top activations, excluding detectors where over 1/4 of activations are duplicates of activations from higher rated detectors.

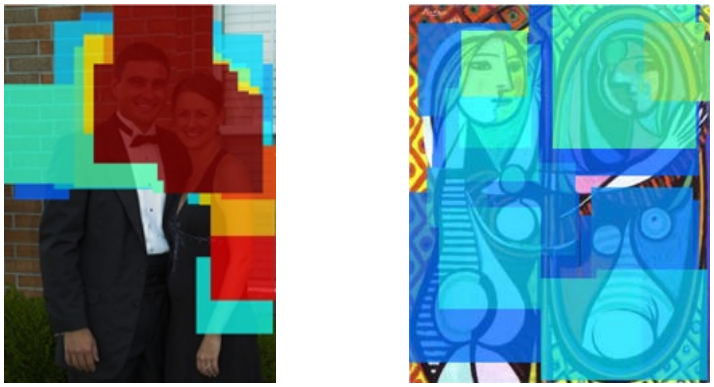


Figure 6.1: **Discriminative patches activations.**Heat maps showing the discriminative patches activations on a natural training image (Left) and "Girl Before a Mirror 1932", a Picasso Cubist painting (Right). The color palette correlates with confidence score and ranges from blue (lowest) to red (highest). In both cases the most discriminative patches for class person are parts of faces and upper bodies, suggesting that computer vision methods are able to identify the key parts of human figures even when they are split into 'fragments of perception' in Cubist paintings.

6.4 Object Detection Methods in Comparison

We chose four person detectors that represent the range of available approaches to test whether state-of-the-art detection methods mimic the human visual system at its limits. These are presented in the time ordering in which they were proposed: one holistic template-based method, one part-based model where the parts are learned automatically, one part-based model where the parts are learned from human annotations, and the most recent deep learning method. Here we discuss the details of each one.

Dalal and Triggs: Object appearance in images can be characterized by histograms of orientations of local edge gradients binned over a dense image grid (HOG) [116]. The Dalal and Triggs (D&T) method trains an object-level HOG template for detection using bounding box annotations. Since the features are binned, the detector is robust only to image deformation within the bins.

Deformable Part Models: A holistic HOG template cannot recognize objects in the face of non-rigid deformations such as varied pose, which result in a rearrangement of the limbs versus the torso. Therefore, the deformable part-based models detection method (DPM) represents objects as collections of parts that can be arranged with respect to the root part [173]. In practice, the model trained on natural images often learns sub-part models (such as half a face).

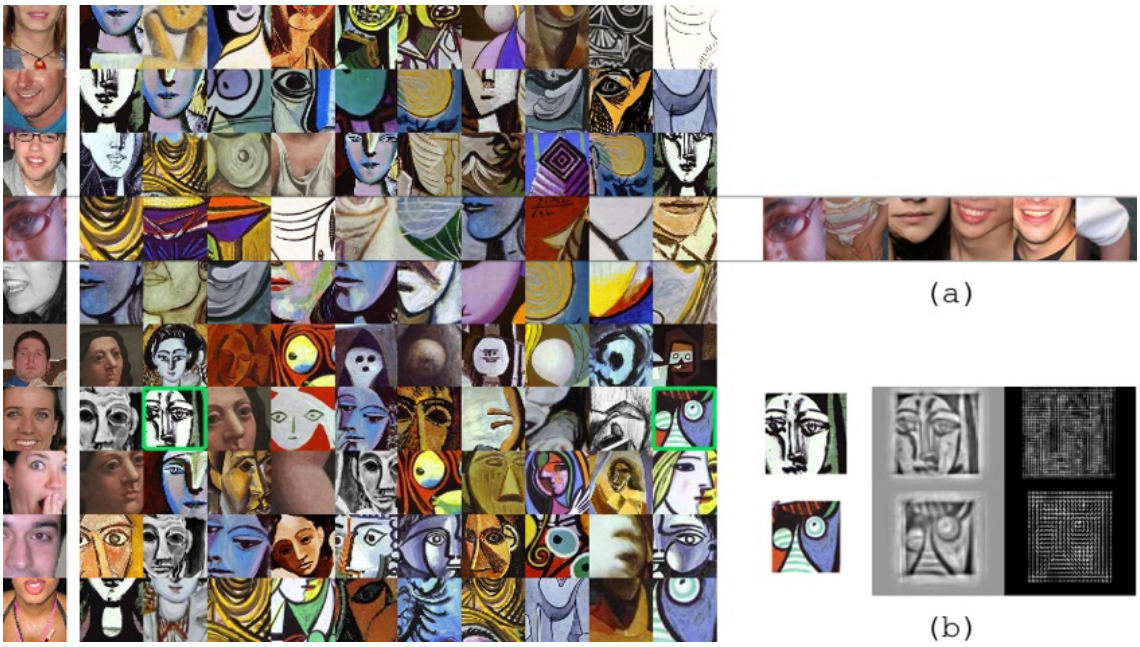


Figure 6.2: **Discriminative patches in Cubist paintings.** Discriminative patch detectors trained on natural images are able to detect the parts that characterize person figures in Cubist paintings. (Left) Each row displays the top ten discriminative patches activations[2]. The leftmost column shows the top activation on the training data. Most of the detectors find corresponding face parts in the natural and painting images, although many are false positives. The fourth patch-detector from the top detects patches with little visual consistency on the paintings as well as the training data (a). Some false positive activations (b)(Bottom) seem more similar to true positives (b)(Top) in Hoggles space [188] (b)(Middle) than in HOG (b)(Right) or the original RGB (Left, marked in green) spaces.

Poselets: Poselets is a similar HOG-based part model that considers extra human supervision during training [181, 182]. Here, parts are not discovered but learned from body-part annotations. Poselets do not necessarily correspond to anatomical body parts like limbs or torsos as these are often not the most salient features for visual recognition. In fact, a highly discriminative Poselet for person detection corresponds to "half of a frontal face and a left shoulder" [181].

R-CNN: R-CNN replaces the earlier rigid HOG features with features learned by a deep convolutional neural network [185, 189]. While R-CNN does not have an explicit representation of parts, it is trained under a detection objective to be invariant to deformations of objects by using a large amount of data. Deep methods outperform previous algorithms by a large margin on natural data. Here we test this state-of-the-art method on abstract paintings.

6.5 Experimental Setup

Since we are interested in comparing the above methods to human vision on data that approaches the limit of perception, we conduct two kinds of experiments. First, we study the human and algorithm performance on person detection over our full corpus of Cubist paintings. Second, we examine the degradation in performance of human perception and detectors as the paintings become more abstract. We conduct all comparisons using the PASCAL VOC evaluation mechanism, in which true positives are selected based on a 50% overlap between detection and ground truth bounding box [187].

6.5.1 Picasso Dataset

In the experiments described below we used as our test data a set of 218 Picasso paintings that have titles indicating that they depict people. These ranged from figurative portraits to abstract depictions of person figures as a collection of distorted parts. The set of paintings we used is highly biased in comparison to PASCAL person class images [187]. Given the nature of the art form, Cubist paintings usually depict people in full frontal or portrait views where most of the canvas area is devoted to the torso of a person. This results in higher average precision scores as a random detection that contains over 50% of the image would count as a true positive. This issue exists in any PASCAL VOC evaluation, but it is especially pronounced in this case.

6.5.2 Human Perception Study Setup

We conducted two experiments as part of our perception study. First, we recorded human detections of person figures in Cubist paintings. Second, we asked participants to bucket the paintings by their degree of abstraction compared to photorealistic depictions of people. For each painting, raters were asked to pick a classification on a 5-point Likert scale, where 1 corresponded to "*The figures in this painting are very lifelike*" and 5 corresponded to "*The figures in this painting are not at all lifelike*".

Participants

We recruited eighteen participants to partake in our perception study. Sixteen participants were undergraduate students at our institution, one was a graduate student and one a software engineering professional. Seventeen participants were male and one was female.

Mechanism

Participants completed the study on their personal laptops using an online graphical annotation tool we wrote for the purpose. Each participant spent an hour on the study and received a compensation of $15. Each participant annotated 146 randomly chosen paintings out of the total 218, so that every painting was annotated by 14 - 15 unique participants.

6.5.3 Detector Study Setup

We compare the human recognition performance we measured during the perception study to four object detection methods. We train all methods using the PASCAL 2010 "person" class training and validation images [187]. We train the methods using natural images so that they do not enjoy an advantage over humans by training on the paintings. However, some research suggests that human recognition in Cubist paintings does improve with repeated exposures [171]. We set all parameters in all four methods to the same settings used in the original papers, except for the Poselets detection score which we set to 0.2 based on cross validation on the training data.

6.5.4 Ground Truth

Because Picasso did not explicitly label the human figures in his paintings, there is no clear cut gold standard for human figure annotations in our image corpus. As a result, we rely on our human participants to form a ground truth annotation

set. We do so by capturing the average rater annotation as follows. Since each painting might have more than one human figure, we use k-means clustering to group annotations by the human figure they correspond to. For each cluster, we obtain a ground truth bounding box by taking the median of each corner of the bounding boxes in that cluster along every dimension. This yields one ground truth bounding box per human figure per image, which we can now use to evaluate both human and detector annotations. For each human rater we withhold her annotations and construct a modified leave-one-out ground truth from the annotations of all other raters. When evaluating detectors, however, we include annotations from all human raters in the ground truth.

It is worth noting that since humans themselves are error-prone (especially when recognizing objects in more abstract paintings), our ground truth cannot be a perfect oracle. Rather, all evaluation is comparing performance to the average, imperfect human. From this perspective, our evaluation of human raters can be seen as a measure of their agreement, and our evaluation of detectors can be seen as a measure of similarity to the average human.

6.5.5 Evaluating Humans and Detectors

Unlike detectors, humans provide only one annotation per figure without confidence scores, so we cannot compute average precision. Our primary metric for humans is the F-measure (F1 score), which is the harmonic mean of precision and recall. In order to combine the F-measures of all participants, we consider both (qualitatively) the shape of the distribution of the scores, and (quantitatively) the mean F-measure.

To compare detectors with humans, we pick the point on the methods' precision-recall curve that optimizes F-measure, and return the precision, recall, and F-measure computed at this point. This is generous to the detectors but captures their performance if they were well tuned for this task.

6.6 Detection Performance on the Picasso Dataset

In the first part of the comparison we evaluate the performance of the humans and the four methods at detecting human figures in Picasso paintings.

6.6.1 Human Performance on the Picasso Dataset

First, we evaluate our human participants against the leave-one-out ground truth to determine how effective people are at recognizing human figures in Cubist art.

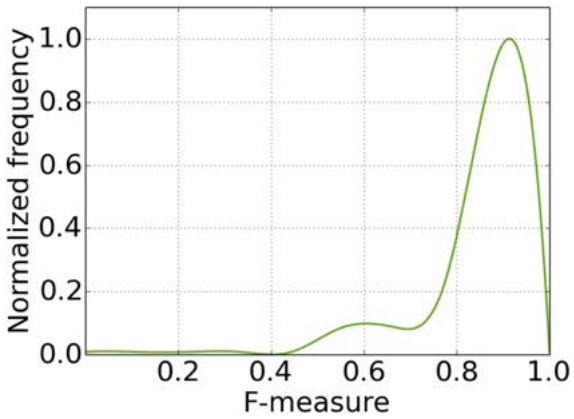


Figure 6.3: **Human F-measure recognition scores.**Frequency distribution of human F-measure recognition scores in all 218 Cubist paintings against the leave-one-out ground truth bounding boxes. Due to the small number of participants, the curve has been smoothed for clarity.

Figure 6.3 displays the distribution of human F-measures for this task. Qualitatively, we see that humans perform quite well, as the distribution has its peak around 0.9, and there is little variance among the scores. It is worth noting the bump in the distribution around 0.6—there were a few raters whose annotations were significantly different from the ground truth annotation due to either a failure to recognize the images, or a misunderstanding of the annotation interface and instructions. Quantitatively, the first row of Table 6.1 shows the mean human precision, recall, and F-measure. These numbers confirm our impressions of the distribution—humans tend to agree with each other on the location of person figures in the paintings.

6.6.2 Detector Performance on the Picasso Dataset

Qualitative Results

Part-based models trained purely on photographs performed surprisingly well when tested against the Picasso data. As can be seen in Figures 6.4 and Figure 6.5, Poselets and DPMs successfully produce bounding boxes for figures in the paintings, though they have their fair share of false positives and misses. The non-part-based methods do not perform as well, as is evident in Figure 6.6 where each row displays the top ten detections of a method over the entire painting dataset, sorted by each method's confidence score.

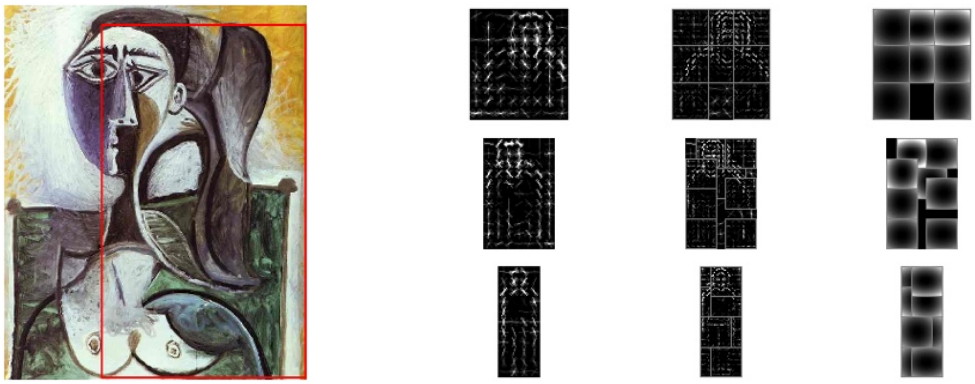


Figure 6.4: **DPM can detect split faces.**(Left) DPM detects one of the two viewpoints of the split face, resulting in a shifted localization of the person as a whole. (Right) The DPM model trained on natural images learns sub-face parts in three different scales. This provides insight as to why DPM is able to detect a split-face patch resulting in the bounding box detection on the left.

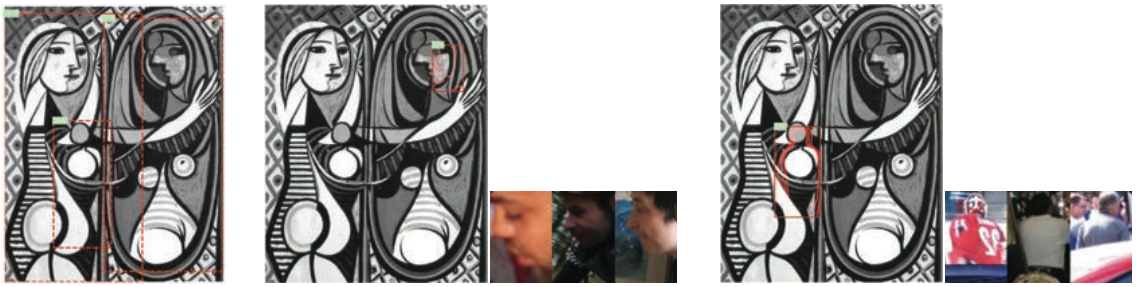


Figure 6.5: **The Poselets method is able to find person-parts in Cubist paintings and use them to detect person figures as a whole.** (a) Bounding box detections in Picasso's "Girl Before a Mirror 1932" with a false positive detection in the center. (b) A true positive Poselet activation on the painting (Left) together with the corresponding activation on the training data (Right). In both image domains the Poselet detects a downward-angled face in profile. (c)(Left) The false positive activation that results in the incorrect bounding box detection in (a). Here the Poselet falsely detects an away-facing person in the painting (c)(Right).

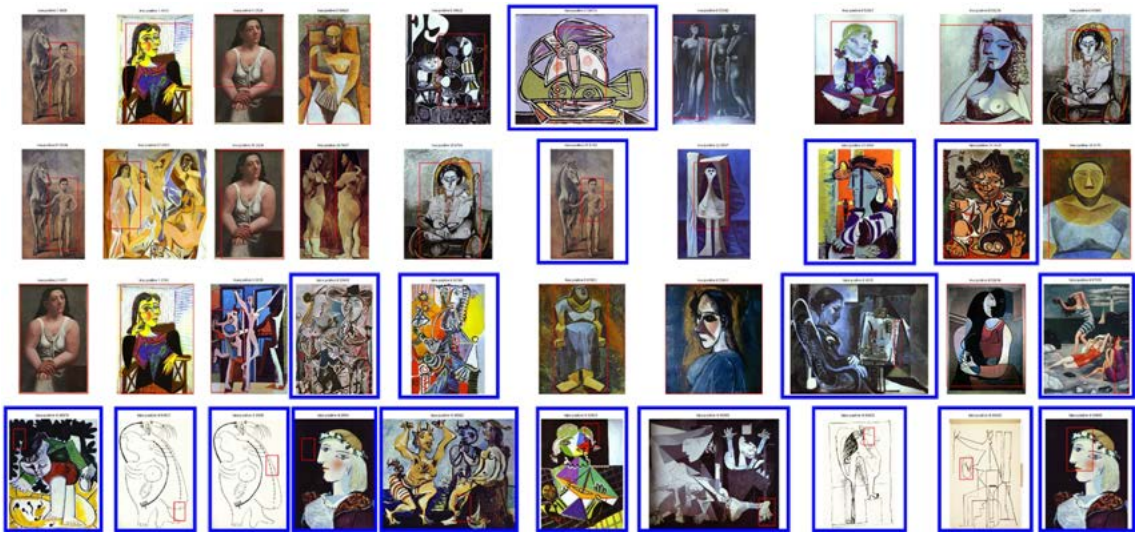


Figure 6.6: **Top ten detections for each method according to confidence from left to right.** First row: DPM. Second row: Poselets. Third row: R-CNN. Fourth row: D&T. False positives are marked in blue. Qualitatively, it is evident that part-based models outperform the other approaches.

Quantitative Results

Figure 6.7 displays the precision-recall curves for each method when evaluated on all paintings. There is a clear ordering in accuracy: DPM performs the best by far, Poselets and RCNN come next, and Dalal and Triggs, though characterized by high recall, has extremely low precision, leading to terrible performance. In general, all of the methods achieve recall of up to 0.8, which implies that they are capable of recognizing most human figures in the image. However, DPM is the only method which can maintain a reasonable precision as recall increases, which explains its significantly greater AP. The table in Table 6.1 confirms these insights. DPM's performance on this task is quite encouraging, as its AP (0.38) is not too far off from its performance on photorealistic PASCAL 2010 photographs (0.41 without context rescoring) [189]. In practice this a favorable comparison as the PASCAL dataset, unlike ours, includes many images that do not contain people.

6.6.3 Comparing Human and Detector Performance

As is clear from the graphical and tabular data in Figure 6.7 and Table 6.1, a pair of humans are much better than a human and a detector at reaching an agreement about where the person figures are in Picasso paintings. The green dot in the upper

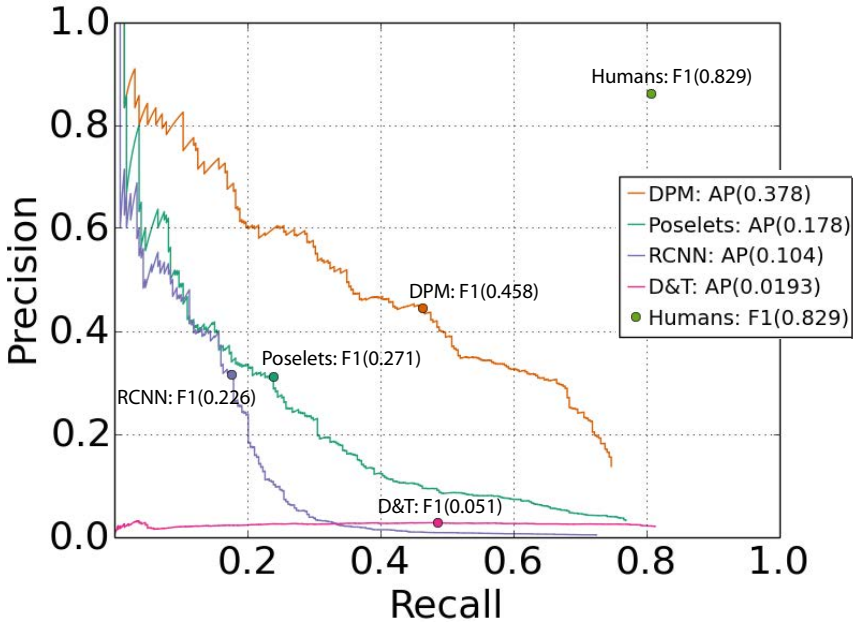


Figure 6.7: **Performance comparison via precision-recall curves.** For detectors, precision, recall, and F-measure are the maxima over the entire precision-recall curve. For humans, these numbers are averages across the raters. While DPM outperforms other methods, none of the methods reach human performance.

Annotator	Precision	Recall	F-measure	AP
Human	0.804	0.860	0.829	N/A
DPM	0.444	0.464	0.458	0.378
Poselets	0.311	0.240	0.271	0.178
RCNN	0.315	0.177	0.226	0.104
D&T	0.027	0.486	0.051	0.019

Table 6.1: **Performance comparison via tabular data.** For detectors, precision, recall, and F-measure are the maxima over the entire precision-recall curve. For humans, these numbers are averages across the raters. While DPM outperforms other methods, none of the methods reach human performance.

right corner of the graph shows human precision and recall to be far higher than the orange dot of DPM, the highest-performing method.

6.6.4 Discussion of Performance on the Picasso Dataset

The comparison between the human participants and four methods on the task of detecting person figures in Picasso paintings demonstrates clearly that humans are highly skilled at this task, and that detectors are much less effective but can still achieve results within an order of magnitude of their detection performance on natural images. Among algorithms, part-based object detection methods perform better than object-level methods on images containing form abstraction. DPM and Poselets, our two part-based methods, demonstrate the best performance on the object detection task. This is likely due to the fact that part-based methods are able to recognize medium-level parts that remain intact even in Cubist paintings where standard human body parts are highly fragmented and rearranged. We emphasize the part-based approach here, as we have no reason to believe that the HOG features used by both DPM and Poselets carry any advantage over other image features organized in a part-based model.

Given its success in object detection on natural images, it is interesting that R-CNN does not perform well on this task. One reason for this could be that R-CNN is not a part-based model, however this is only partly true because the convolutional filters can be thought of as parts, and the max pooling as performing deformations. A second factor might be the fact that R-CNN over fits to the natural visual world and fails at adapting to the domain of paintings. There has been little research into how CNN-based networks perform on distorted images, but an initial investigation suggests that tiny changes to an image may cause drastic changes to their output [190].

6.7 Performance Degradation with Increased Abstraction

In the second part of the comparison we study the degradation of performance of humans and methods with increased painting abstractness.

6.7.1 Classifying Images by Degree of Abstraction

In our user study (described in Section 6.5.2), each rater labeled images on a scale from 1 (*The figures in this painting are very lifelike*) to 5 (*The figures in this*

painting are not at all lifelike). By taking the rounded average of user labels for each image, we divide the images into five 'buckets of abstraction' in order to evaluate object detection performance as the abstraction of form increases. An example of a painting with an average rating of 1 is Picasso's "Seated Woman 1921", and an example of a painting with an average rating of 5 is Picasso's "Nude and Still Life 1931" (copyrighted paintings not reproduced). The number of images in each bucket is shown in Figure 6.8(Top Left).

6.7.2 Human Performance Degradation

Figure 6.8(Top Right) demonstrates the impact of abstraction of form on human object detection performance. As the images become more abstract, the distribution of human F-measures shifts clearly to the left. This indicates that human performance worsens on more scrambled human figures, which is consistent with previous results [174, 177]. As noted in Section 6.6, there are a few annotators with low F-measures in each of the curves, which implies that these raters' errors are independent of image abstraction and are most likely due to a failure to follow the instructions rather than an inability to recognize objects.

6.7.3 Detector Performance Degradation

Qualitative Results

In Figure 6.9 we compare the top detections per method on paintings from bucket 2 versus bucket 5. The top discriminative-patches activations on these two buckets (Right) help visualize the difficulty in detecting meaningful mid-level parts as the paintings become more abstract.

Quantitative Results

Figure 6.8(Bottom Right) shows the precision recall curves for the DPM method with varying abstraction buckets. As with human performance, increasing the image abstraction of form causes a pronounced decrease in performance. The overlap between the curves for buckets 1 and 2 may be due to variance as a consequence of the low number of images in those two buckets.

6.7.4 Comparing Human and Detector Degradation

Figure 6.8(Bottom Left) compares the performance change of humans and detectors with varying abstraction buckets. As can be seen, the performance of all

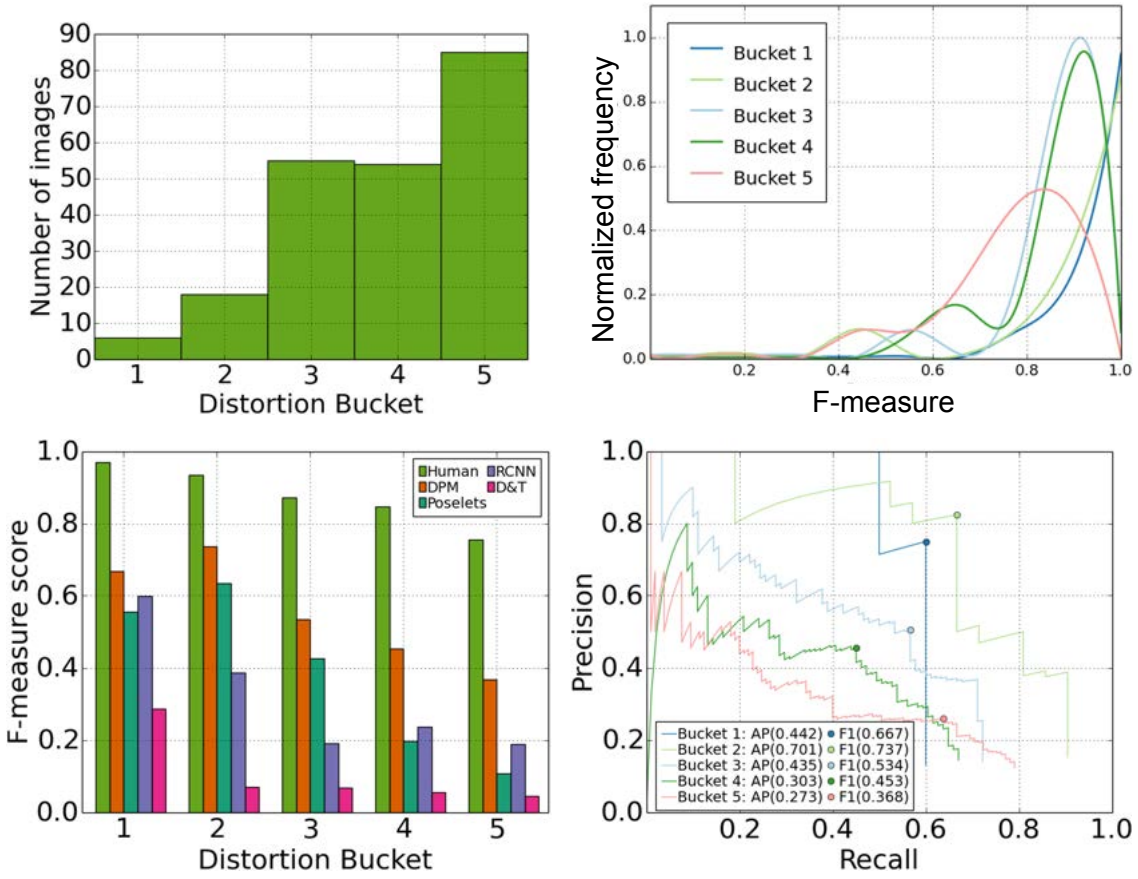


Figure 6.8: **Impact of increasing form abstraction on object detection performance.** Bucket 1 contains images that appear most natural, and bucket 5 contains images that appear most abstract. (Top Left) A histogram showing the number of images per bucket. (Top Right) Human F-measure distributions by abstraction bucket. (Bottom Right) DPM precision-recall curves by bucket. (Bottom Left) Comparing humans and methods. Part-based models show a similar degradation behavior to human performance as the images become more abstract.

detectors degrades in a similar pattern to human performance as images become increasingly abstract, but the part-based methods follow the human pattern most closely. This matches our intuition about the similarities between part-based object detection and the human visual system. In contrast, the template-based Dalal and Triggs method abruptly breaks down after bucket 1.

6.7.5 Discussion of Degradation with Increased Abstraction

As we have demonstrated, part-based models for object detection show a smooth degradation in precision and recall as the images become more abstract. This is consistent with results from neuroscience, which indicate that humans are capable of detecting objects cut into parts, but that their ability degrades significantly when the parts are scrambled. The correspondence between human and computational method performance on this task suggests that a part-based object representation might be a good approximation for the mechanisms of object detection in the human brain. The ability to model these mechanisms computationally further corroborates the neuroscience theory of part-based object detection strategies. This is encouraging, even though current methods cannot yet perform at the level of human vision.

We note that the correspondence between part-based models and human perception is a somewhat surprising one. At their core, the methods we used are based on HOG features that we expected to be highly dependent on image statistics. It was pleasantly surprising to observe the correlation between these methods, trained on natural images, and human perception on Cubist paintings with completely different statistics. We believe that better performance could be achieved using part-based models that rely on higher level features than HOG.

6.8 Discussion

Since computer vision aims to attain not only the performance of human vision but also its flexibility and robustness, we should characterize our algorithms' performance on novel and extreme inputs. In this chapter, we have argued that object detection under abstraction of object form is an example of a challenging perception task that existing image benchmarks do not properly evaluate. We have proposed Cubist paintings as an additional corpus for object detection, as they contain rearranged object parts that are nevertheless recognizable to the human visual system as whole objects. Using this dataset, our evaluation comparing human performance to that of various object detection methods demonstrates that part-based models are a step in the right direction for modeling human robustness to part-rearrangement, since their

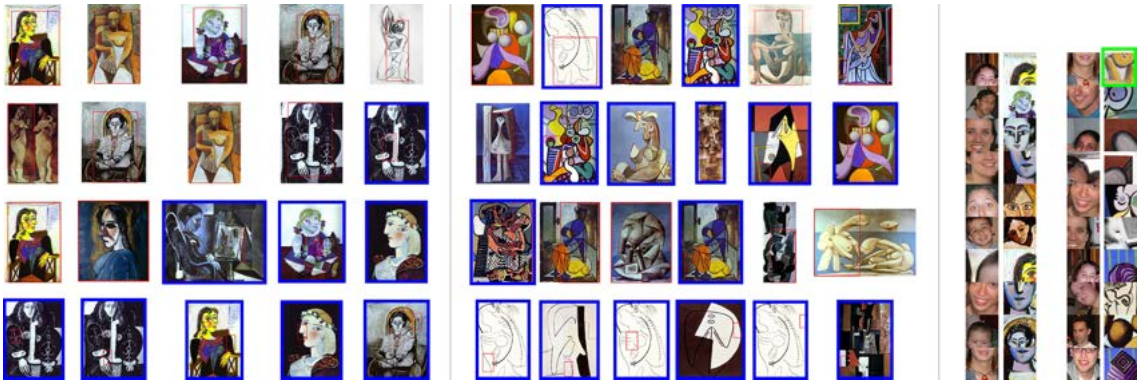


Figure 6.9: **The degradation in performance with image difficulty.** Top five detections per method (rows correspond to: DPM, Poselets, R-CNN, D&T) on images from bucket 2 (Left) and bucket 5 (Middle). False positives are marked in blue. This comparison shows that all detection methods perform worse on more abstract images. (Right) Degradation in performance is also evident in the detection of isolated parts. Top 10 discriminative patches activations for images of bucket 2 and bucket 5, with corresponding activations on PASCAL images. All activations on bucket 2 are true positives compared to only one from bucket 5 (in green).

performance degrades comparably to humans as the abstractness in images increases. By showing that these models can be trained on photographic data yet still perform on abstract data we demonstrate that they are less over-fit to the natural world than template-based and deep models.

Part-reorganization in Cubism is one example that pushes the envelope of human perception, but there are other artistic movements with characteristic abstractions, such as the use of blurring in Impressionism, that would provide rich grounds for study. Future work in the design of computer vision methods should be cognizant of the limitations of traditional camera snapshot datasets and look for complementary resources when evaluating computational methods. Ultimately, a plethora of such resources should not only be used for testing, but must be combined into rich training datasets when designing methods that truly mimic the wide range of human perception.

Chapter 7

Conclusions and Discussion

This dissertation demonstrated that endowing machines with rich perception is indeed possible. It explored several tasks that require modeling complex visual details and proposed weakly-supervised and unsupervised methods to address them. A central theme was the use of video and image synthesis for the evaluation of the proposed methods, as producing rich content requires the perception and modeling of rich detail. The dissertation then suggested an alternative method for testing for rich perception using abstract art as a testbed.

Part I considered modeling visual details of motion and appearance directly from video data. Chapter 2 described an approach to model the multi-modal relationships between speech and gesture. Chapter 3 continued to discuss learning appearance details from video data of a person in motion. Here, video synthesis together with reconstruction loss was used as a way to test whether the method learned a good model of the target person.

Part II explored learning temporal visual patterns, a task that is easy for humans but not yet for computer vision systems. Chapter 4 described a unique historical dataset of a century of high school yearbook portraits and data-driven methods to discover temporal patterns in this data. Chapter 5 proposed an unsupervised method to learn to disentangle what visually changes from what stays permanent over time. Image synthesis was used here as a way to test whether the method learned a good disentanglement into semantically-meaningful factors.

Finally, Chapter 6 in Part III proposed one possible direction for testing whether our systems align with human perception, by using abstract art as a testbed for generalization.

While these tasks and approaches are quite different from each other in some respects, what they have in common is that they do not simply reduce to problems where the state of the art supervised methods in computer vision such as classification,

detection or segmentation can be cleanly applied. Solving problems that require rich perception involves both new techniques and applying existing technology in novel ways. This dissertation took steps in both these directions. Moreover, as suggested in Part III, perhaps advancing in these areas would be well served by the development of similarly novel and creative benchmarks for future algorithms.

There are several potential directions ripe for exploration that build on the insights gained by this dissertation. For example, the synthesis tools we have used for translating motion to appearance in Chapter 3, could be applied to decoding human neural visual representations to visualize what an fMRI subject is looking at or imagining at the time of activation. The main challenge here is that unlike typical deep learning datasets, there is very little training data due to the limitations of recording from human subjects in fMRI. One would, therefore, have to come up with ways to inject priors about natural images into the decoding process to fill in the gaps and hallucinate the rich visual details of the stimuli.

Another direction is social communication. Chapter 2 narrows in on the communicative gestures produced during single-person monologues. But most social interaction involves pairs or groups of people. How can we learn about and replicate the ability to participate in social interactions? Can the behavior, speech, motion of one person predict that of another? Can we learn to insert AI agents into natural human social settings? These are all questions left for future work. Such abilities can also allow us to provide tools to better understand humans in social dynamics. For example, abnormalities in the minute synchronous motions that happen during dyadic interactions may provide early markers of autism or depression.

Moreover, in this setting, we can explore some of the ideas that have been around in the developmental and neuroscience literature, such as the synchronization that happens during dyadic interactions. To start working toward modeling these interactions from a computational standpoint, computer scientists would benefit from working closely with psychologists and neuroscientists. I believe such future efforts will give us intuition on exactly what these rich representations beyond human-annotated labels should be.

Appendix A

Dissertation in the Time of Corona

A dissertation which primarily concerns itself with visual subtleties and historical imagery cannot be complete without some visual evidence of the current historical times. Here we are, alone together and together alone. Hiding in our dens and caverns. Covering our breathing holes and sanitizing our surfaces. Finding spaces to work and spaces to be in the nooks and crannies of our protective walls—in closets, in stairwells, in hallways, and garages.

Figure A.1: Next to the car.

I am writing this dissertation amid our domestic routine which governs our current lives like a desert storm. Two children. One and four naps, respectively, but never at the same time. Three meals and one snack (did we always eat so much? We can no longer seem to remember). One walk. Two baths. Laundry. Cooking. Cleaning. Sleep—for those of us who need not work at night. And within this routine joy resides and wonder flowers of growing bodies and brains.

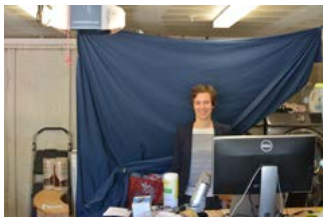


Figure A.2: And the washing machine.

And once in a while, we emerge out of domesticity into our other spaces. Into these nooks and crannies of work, where there are screens, keyboards, cameras, and microphones. Where we need, for example, to don a suit jacket matched with pajama pants to defend our dissertation or give a job talk in front of no one, talking to ourselves while being watched by tens of disembodied people. And yet, here we are, next to the car (Figure A.1) and the washing machine (Figure A.2), behind all the boxes we failed to return before the lockdown, silent reminders of the time we used to have and the spaces we used to allow

ourselves to carelessly occupy, and all the mail, still waiting to be opened after its scheduled quarantine days (Figure A.3).

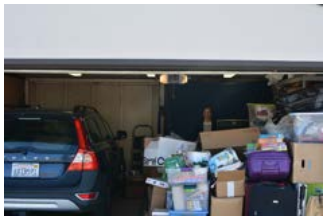


Figure A.3: Behind the boxes.

And yet, they do come, alone together and together alone. Committees meet, listen, and approve over remote video conferencing dotted with a chocolate truffle and a cup of Peet's (Figure A.4). Faraway family members enjoy, in the darkness of night, a unique opportunity to join in a ceremony that would have been inaccessible to them in any normal times. Others who have left before us and spread around this vast country (Figure A.5) can take advantage of the situation and come to a party—each with their time zone and glass of wine. Trying to overcome the awkwardness of disembodied Charades for a brief opportunity to feel together again. Thank you all so much for coming (Figure A.6), and so long.

Figure A.4: Myself, on a scolarly fake background, with my distinguished dissertation committee over zoom: Professors Alexei A. Efros, Alison Gopnik and Jitendra Malik.

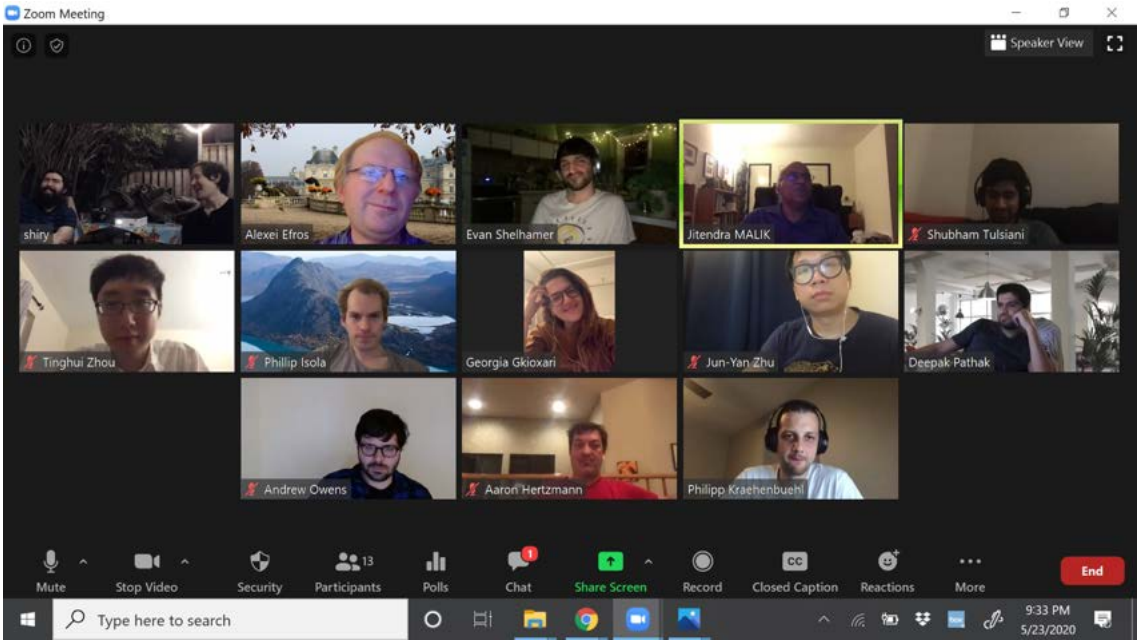


Figure A.5: Dissertation party.

Figure A.6: *"Thank you so much for coming."*
Cartoon by David Sipress. The New Yorker. Published in the print edition of the June 8 & 15, 2020, issue.

Bibliography

[1] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, pp. 3–20, MIT Press, 1991.

[2] L. Von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum, "Improving accessibility of the web with a computer game," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 79–82, ACM, 2006.

[3] L. Von Ahn, S. Ginosar, M. Kedia, and M. Blum, "Improving image search with phetch," in *Acoustics, speech and signal processing, 2007. icassp 2007. ieee international conference on*, vol. 4, pp. IV–1209, IEEE, 2007.

[4] B. Sayim and P. Cavanagh, "What line drawings reveal about the visual brain," *Frontiers in Human Neuroscience*, vol. 5, no. 118, 2011.

[5] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2019.

[6] S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, and J. Malik, "Learning individual styles of conversational gesture," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] D. McNeill, *Hand and Mind: What Gestures Reveal about Thought*. Chicago: University of Chicago Press, 1992.

[8] J. Cassell, D. McNeill, and K.-E. McCullough, "Speech-gesture mismatches: Evidence for one underlying representation of linguistic and nonlinguistic information," *Pragmatics and Cognition*, vol. 7, no. 1, pp. 1–34, 1999.

[9] k. Kendon, *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

[10] W. C. So, S. Kita, and S. Goldin-Meadow, "Using the hands to identify who does what to whom: Gesture and speech go hand-in-hand," *Cognitive Science*, vol. 33, pp. 115–125, Feb. 2009.

[11] J. P. de Ruiter, A. Bangerter, and P. Dings, "The interplay between gesture and speech in the production of referring expressions: Investigating the tradeoff hypothesis," *Topics in Cognitive Science*, vol. 4, pp. 232–248, Mar. 2012.

[12] B. Butterworth and U. Hadar, "Gesture, speech, and computational stages: A reply to McNeill," *Psychological Review*, vol. 96, pp. 168–74, Feb. 1989.

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

[14] P. Wagner, Z. Malisz, and S. Kopp, "Gesture and speech in interaction: An overview," *Speech Communication*, vol. 57, pp. 209 – 232, 2014.

[15] P. Buehler, A. Zisserman, and M. Everingham, "Learning sign language by watching tv (using weakly aligned subtitles)," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 2961–2968, IEEE, 2009.

[16] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Asian Conference on Computer Vision*, pp. 538–552, Springer, 2014.

[17] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 3793–3802, IEEE, 2016.

[18] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2018.

[19] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," in *Workshop on Automatic Face and Gesture Recognition*, IEEE, June 1995.

[20] T. J. Darrell, I. A. Essa, and A. P. Pentland, "Task-specific gesture analysis in real-time using interpolated views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1236–1242, Dec. 1996.

[21] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, IEEE, 2007.

[22] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari, "Multimodal human discourse: gesture and speech," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 9, no. 3, pp. 171–193, 2002.

[23] J. Cassell, J. Sullivan, E. Churchill, and S. Prevost, *Embodied conversational agents*. MIT press, 2000.

[24] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone, "Animated conversation: Rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents," in *Computer Graphics and Interactive Techniques*, ACM Trans. Graphics (SIGGRAPH), pp. 413–420, ACM, 1994.

[25] S. Kopp, B. Krenn, S. Marsella, A. N. Marshall, C. Pelachaud, H. Pirker, K. R. Thórisson, and H. Vilhjálmsson, "Towards a common framework for multimodal generation: The behavior markup language," in *International workshop on intelligent virtual agents*, pp. 205–217, Springer, 2006.

[26] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore, "Beat: the behavior expression animation toolkit," in *Life-Like Characters*, pp. 163–185, Springer, 2004.

[27] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel, "Gesture modeling and animation based on a probabilistic re-creation of speaker style," *ACM Transactions on Graphics*, vol. 27, pp. 5:1–5:24, Mar. 2008.

[28] S. Levine, C. Theobalt, and V. Koltun, "Real-time prosody-driven synthesis of body language," in *ACM Transactions on Graphics*, vol. 28, p. 172, ACM, 2009.

[29] S. Levine, P. Krähenbühl, S. Thrun, and V. Koltun, "Gesture controllers," in *ACM Transactions on Graphics*, vol. 29, p. 124, ACM, 2010.

[30] S. Marsella, Y. Xu, M. Lhommet, A. Feng, S. Scherer, and A. Shapiro, "Virtual character performance from speech," in *Symposium on Computer Animation*, SCA, pp. 25–35, ACM, 2013.

[31] N. Sadoughi and C. Busso, "Retrieving target gestures toward speech driven animation with meaningful behaviors," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pp. 115–122, ACM, 2015.

[32] C.-C. Chiu and S. Marsella, "How to train your avatar: A data driven approach to gesture generation," in *International Workshop on Intelligent Virtual Agents*, pp. 127–140, Springer, 2011.

[33] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann, "Smartbody: Behavior realization for embodied conversational agents," in *International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 1, pp. 151–158, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[34] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch, "All Together Now: Introducing the Virtual Human Toolkit," in *13th International Conference on Intelligent Virtual Agents*, (Edinburgh, UK), Aug. 2013.

[35] C. Bregler, M. Covell, and M. Slaney, "Video rewrite: Driving visual speech with audio," in *Computer Graphics and Interactive Techniques*, ACM Trans. Graphics (SIGGRAPH), pp. 353–360, ACM, 1997.

[36] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Neural Information Processing Systems*, 2016.

[37] T. R. Langlois and D. L. James, "Inverse-foley animation: Synchronizing rigid-body motions to sound," *ACM Transactions on Graphics*, vol. 33, pp. 41:1–41:11, July 2014.

[38] E. Shlizerman, L. Dery, H. Schoen, and I. Kemelmacher-Shlizerman, "Audio to body dynamics," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2018.

[39] J. S. Chung, A. Jamaludin, and A. Zisserman, "You said that?," in *British Machine Vision Conference*, 2017.

[40] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Transactions on Graphics*, vol. 36, pp. 95:1–95:13, July 2017.

[41] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[42] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351 of *LNCS*, pp. 234–241, Springer, 2015.

[43] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[44] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing*, 2017.

[45] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 776–780, Mar. 2017.

[46] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *Trans. Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2878–2890, Dec. 2013.

[47] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *IEEE International Conference on Computer Vision*, (Nice, France), pp. 726–733, 2003.

[50] G. Mori, A. Berg, A. Efros, A. Eden, and J. Malik, "Video based motion synthesis by splicing and morphing," Tech. Rep. UCB//CSD-04-1337, University of California, Berkeley, June 2004.

[51] M. Gleicher, "Retargetting motion to new characters," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 33–42, ACM, 1998.

[52] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 39–48, ACM Press/Addison-Wesley Publishing Co., 1999.

[53] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen, "Real-time motion retargeting to highly varied user-created morphologies," in *ACM Trans. Graphics*, vol. 27, p. 27, ACM, 2008.

[54] G. K. Cheung, S. Baker, J. Hodgins, and T. Kanade, "Markerless human motion transfer," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pp. 373–378, IEEE, 2004.

[55] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt, "Video-based characters: creating new human performances from a multi-view video database," in *ACM Trans. Graphics*, vol. 30, p. 32, ACM, 2011.

[56] D. Casas, M. Volino, J. Collomosse, and A. Hilton, "4D Video Textures for Interactive Character Appearance," *Computer Graphics Forum (Proceedings of EUROGRAPHICS)*, vol. 33, no. 2, pp. 371–380, 2014.

[57] H. Kim, P. Carrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graphics*, vol. 37, no. 4, p. 163, 2018.

[58] R. Martin-Brualla, R. Pandey, S. Yang, P. Pidlypenskyi, J. Taylor, J. Valentin, S. Khamis, P. Davidson, A. Tkach, P. Lincoln, A. Kowdle, C. Rhemann, D. B. Goldman, C. Keskin, S. Seitz, S. Izadi, and S. Fanello, "Lookingood: Enhancing performance capture with real-time neural re-rendering," *ACM Trans. Graph.*, vol. 37, pp. 255:1–255:14, Dec. 2018.

[59] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[60] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[61] W. J. Baddar, G. Gu, S. Lee, and Y. M. Ro, "Dynamics transfer gan: Generating video by transferring arbitrary temporal dynamics from a source video to a single target image," *arXiv preprint arXiv:1712.03534*, 2017.

[62] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[63] R. de Bem, A. Ghosh, T. Ajanthan, O. Miksik, N. Siddharth, and P. Torr, "A semi-supervised deep generative model for human body analysis," in *Proc. European Conf. on Computer Vision (ECCV)*, pp. 0–0, 2018.

[64] R. De Bem, A. Ghosh, A. Boukhayma, T. Ajanthan, N. Siddharth, and P. Torr, "A conditional deep generative model of people in natural images," in *Proc. Winter Conf. on Computer Vision (WACV)*, pp. 1449–1458, IEEE, 2019.

[65] D. Joo, D. Kim, and J. Kim, "Generating a fusion image: One's identity and another's shape," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 1635–1643, 2018.

[66] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A generative model of people in clothing," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 853–862, 2017.

[67] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Neural Information Processing Systems*, pp. 406–416, 2017.

[68] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 99–108, 2018.

[69] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable gans for pose-based human image generation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[70] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," *arXiv preprint arXiv:1704.05831*, 2017.

[71] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 8857–8866, 2018.

[72] M. Zanfir, A.-I. Popa, A. Zanfir, and C. Sminchisescu, "Human appearance transfer," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 5391–5399, 2018.

[73] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *International Conference on Computer Vision*, 2017.

[74] K. Aberman, M. Shi, J. Liao, D. Liscbinski, B. Chen, and D. Cohen-Or, "Deep video-based performance cloning," in *Computer Graphics Forum*, vol. 38, pp. 219–233, Wiley Online Library, 2019.

[75] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-gan: Unsupervised video retargeting," in *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

[76] L. Liu, W. Xu, M. Zollhoefer, H. Kim, F. Bernard, M. Habermann, W. Wang, and C. Theobalt, "Neural rendering and reenactment of human actor videos," *ACM Transactions on Graphics 2019 (TOG)*, 2019.

[77] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Neural Information Processing Systems*, 2018.

[78] I. K. Rıza Alp Güler, Natalia Neverova, "Densepose: Dense human pose estimation in the wild," *arXiv*, 2018.

[79] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Neural Information Processing Systems*, pp. 469–477, 2016.

[80] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Neural Information Processing Systems*, pp. 700–708, 2017.

[81] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2017.

[82] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the*

34th International Conference on Machine Learning-Volume 70, pp. 1857–1865, JMLR. org, 2017.

[83] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. Int. Conf. on Computer Vision (ICCV)*, vol. 1, p. 3, 2017.

[84] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[85] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. European Conf. on Computer Vision (ECCV)*, 2016.

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations*, 2015.

[87] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[88] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[89] K. McDonald, "Dance x Machine Learning: First Steps." https://medium.com/@kcimc/discrete-figures-7d9e9c275c47, 2019. [Online; accessed 21-March-2019].

[90] Xpire, "Using AI to make NBA players dance." https://tinyurl.com/y3bdj5p5, 2019. [Online; accessed 21-March-2019].

[91] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894, JMLR. org, 2017.

[92] S. Ginosar, K. Rakelly, S. M. Sachs, B. Yin, C. Lee, P. Krahenbuhl, and A. A. Efros, "A century of portraits: A visual historical record of american high school yearbooks," *IEEE Transactions on Computational Imaging*, vol. 3, pp. 421–431, Sept 2017.

[93] "Flappers flaunt fads in footwear," *The New York Times*, p. 34, Sunday, January 29, 1922.

[94] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Holberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden, "Quantitative analysis of culture using millions of digitized books," *Science*, vol. 331, no. 6014, pp. 176–182, 2010.

[95] Y. J. Lee, A. A. Efros, and M. Hebert, "Style-aware mid-level representation for discovering visual connections in space and time," in *ICCV*, pp. 1857–1864, 2013.

[96] S. Lee, N. Maisonneuve, D. Crandall, A. Efros, and J. Sivic, "Linking past to present: Discovering style in two centuries of architecture," in *IEEE International Conference on Computational Photography (ICCP)*, 2015.

[97] F. Palermo, J. Hays, and A. A. Efros, "Dating historical color images," in *Proc. European Conf. on Computer Vision (ECCV)*, pp. 499–512, 2012.

[98] B. Fernando, D. Muselet, R. Khan, and T. Tuytelaars, "Color features for dating historical color images," in *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, pp. 2589–2593, 2014.

[99] T. Salem, S. Workman, M. Zhai, and N. Jacobs, "Analyzing human appearance as a cue for dating images," in *Proc. Winter Conf. on Computer Vision (WACV)*, 2016.

[100] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Hipster wars: Discovering elements of fashion styles," in *Proc. European Conf. on Computer Vision (ECCV)*, pp. 472–488, 2014.

[101] S. C. Hidayati, K.-L. Hua, W.-H. Cheng, and S.-W. Sun, "What are the fashion trends in new york?," in *Proceedings of the ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), pp. 197–200, ACM, 2014.

[102] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes Paris look like Paris?," *ACM Trans. Graphics (SIGGRAPH)*, vol. 31, no. 4, pp. 101:1–101:9, 2012.

[103] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet

Large Scale Visual Recognition Challenge," *Int. J. of Computer Vision*, pp. 1–42, April 2015.

[104] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Tech. Rep. 1341, University of Montreal, June 2009. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada.

[105] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," in *Int. Conf. on Learning Representations*, 2015.

[106] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014.

[107] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at ICLR*, 2014.

[108] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[109] A. Dosovitskiy and T. Brox, "Inverting convolutional networks with convolutional networks," *CoRR*, vol. abs/1506.02753, 2015.

[110] K. Matzen and N. Snavely, "Bubblenet: Foveated imaging for visual discovery," in *Proc. International Conf. on Computer Vision*, 2015.

[111] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. European Conf. on Computer Vision (ECCV)*, 2012.

[112] C. Gibson, "Population of the 100 largest cities and other urban places in the united states: 1790 to 1990." https://www.census.gov/population/www/documentation/twps0027/twps0027.html. Accessed: 2014-12-11.

[113] C. Goldin, "America's graduation from high school: The evolution and spread of secondary schooling in the twentieth century," *Journal of Economic History*, 1998.

[114] C. Goldin and L. F. Katz, "The race between education and technology: The evolution of u.s. educational wage differentials, 1890 to 2005," Working Paper 12984, National Bureau of Economic Research, March 2007.

[115] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2013.

[116] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005.

[117] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Proc. European Conf. on Computer Vision (ECCV)*, 2012.

[118] F. Galton, "Composite portraits made by combining those of many different persons into a single figure," *Nature*, vol. 18, no. 447, pp. 97–100, 1878.

[119] C. Kotchemidova, "Why we say "cheese": Producing the smile in snapshot photography," *Critical Studies in Media Communication*, vol. 22, no. 1, pp. 2–25, 2005.

[120] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard, "BP4D-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database," *Image and Vision Computing*, vol. 32, no. 10, pp. 692–706, 2014.

[121] J. M. Girard, *Automatic Detection and Intensity Estimation of Spontaneous Smiles*. PhD thesis, University of Pittsburgh, 2014.

[122] M. Lafrance, M. A. Hecht, and E. L. Paluck, "The contingent smile: A meta-analysis of sex differences in smiling," *Psychological Bulletin*, pp. 305–334, 2003.

[123] J. M. Ragan, "Gender displays in portrait photographs," *Sex Roles*, vol. 8, no. 1, pp. 33–43, 1982.

[124] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2009.

[125] C. Rousset and P.-Y. Coulon, "Frequential and color analysis for hair mask segmentation," in *International Conference on Image Processing (ICIP)* (I. S. P. Society, ed.), no. CFP08CIP-CDR, (San Diego, United States), p. 2276, Oct. 2008.

[126] P. Garland, "Is the afro on its way out?," *Ebony*, pp. 128–136, February 1973.

[127] V. Sherrow, *Encyclopedia of Hair: A Cultural History*. Greenwood Press, 2006.

[128] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Neural Information Processing Systems*, 2013.

[129] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[130] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, pp. 40–44, 1993.

[131] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR)*, Nov. 2015.

[132] A. Liu, S. Ginosar, T. Zhou, A. A. Efros, and N. Snavely, "Learning to factorize and relight a city," in *Proc. European Conf. on Computer Vision (ECCV)*, 2020.

[133] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," *Computer Vision Systems*, 1978.

[134] E. H. Adelson and A. P. Pentland, "The perception of shading and reflectance," in *Perception As Bayesian Inference* (D. C. Knill and W. Richards, eds.), ch. The Perception of Shading and Reflectance, pp. 409–423, New York, NY, USA: Cambridge University Press, 1996.

[135] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *Trans. Pattern Analysis and Machine Intelligence*, 2015.

[136] Z. Li and N. Snavely, "CGIntrinsics: Better intrinsic image decomposition through physically-based rendering," in *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

[137] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Trans. Graphics (SIGGRAPH)*, vol. 33, pp. 159:1–159:12, July 2014.

[138] T. Zhou, P. Krähenbühl, and A. A. Efros, "Learning data-driven reflectance priors for intrinsic image decomposition," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2015.

[139] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. Tenenbaum, "Self-supervised intrinsic image decomposition," in *Neural Information Processing Systems*, pp. 5936–5946, Curran Associates, Inc., 2017.

[140] Z. Li and N. Snavely, "Learning intrinsic image decomposition from watching the world," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[141] Y. Weiss, "Deriving intrinsic images from image sequences," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2001.

[142] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning shape, refectance and illuminance of faces in the wild," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[143] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. European Conf. on Computer Vision (ECCV)*, 2018.

[144] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. B. Tenenbaum, and W. T. Freeman, "Visual object networks: Image generation with disentangled 3D representations," in *Neural Information Processing Systems*, 2018.

[145] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz, "Neural inverse rendering of an indoor scene from a single image," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2019.

[146] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[147] R. Martin-Brualla, D. Gallup, and S. M. Seitz, "Time-lapse mining from internet photos," *ACM Trans. Graphics (SIGGRAPH)*, vol. 34, pp. 62:1–62:8, July 2015.

[148] P.-Y. Laffont and J.-C. Bazin, "Intrinsic decomposition of image sequences from local temporal variations," in *Proc. Int. Conf. on Computer Vision (ICCV)*, December 2015.

[149] J. Philip, M. Gharbi, T. Zhou, A. A. Efros, and G. Drettakis, "Multi-view relighting using a geometry-aware network," *ACM Trans. Graphics (SIGGRAPH)*, vol. 38, 07 2019.

[150] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, "Neural rerendering in the wild," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[151] Y. Yu and W. A. Smith, "Inverserendernet: Learning single image inverse rendering," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[152] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz, "Factored time-lapse video," in *ACM Trans. Graphics (SIGGRAPH)*, SIGGRAPH '07, (New York, NY, USA), ACM, 2007.

[153] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Webcam clip art: Appearance and illuminant transfer from time-lapse sequences," *ACM Trans. Graphics (SIGGRAPH)*, vol. 28, December 2009.

[154] N. Jacobs, N. Roman, and R. Pless, "Consistent temporal variations in many outdoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–6, June 2007.

[155] M. Rubinstein, C. Liu, P. Sand, F. Durand, and W. T. Freeman, "Motion denoising with application to time-lapse photography," *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 313–320, June 2011.

[156] Y. Zhou and T. L. Berg, "Learning temporal transformations from time-lapse videos," in *Proc. European Conf. on Computer Vision (ECCV)*, 2016.

[157] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Trans. Graphics (SIGGRAPH)*, vol. 33, no. 4, 2014.

[158] N. Naik, J. Philipoom, R. Raskar, and C. Hidalgo, "Streetscore – predicting the perceived safety of one million streetscapes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 793–799, June 2014.

[159] S. M. Arietta, A. A. Efros, R. Ramamoorthi, and M. Agrawala, "City forensics: Using visual elements to predict non-visual city attributes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 2624–2633, Dec 2014.

[160] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, and L. Fei-Fei, "Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states," *Proceedings of the National Academy of Sciences*, vol. 114, no. 50, pp. 13108–13113, 2017.

[161] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[162] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Proc. European Conf. on Computer Vision (ECCV)*, 2016.

[163] J. R. Bryan Klingner, David Martin, "Street view motion-from-structure-from-motion," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2013.

[164] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde, "Deep outdoor illumination estimation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[165] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[166] G. B. Huang, V. Jain, and E. Learned-Miller, "Unsupervised joint alignment of complex images," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2007.

[167] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Int. Conf. on Learning Representations*, 2019.

[168] S. Ginosar, D. Haas, T. Brown, and J. Malik, "Detecting people in cubist art," in *Proc. European Conf. on Computer Vision (ECCV) Workshops*, pp. 101–116, Springer International Publishing, 2014.

[169] E. Hsiao and A. A. Efros, "DPM superhuman." http://www.cs.cmu.edu/~efros/courses/LBMV09/presentations/latent_presentation.pdf slides 43-51.

[170] P. M. Laporte, "Cubism and science," *The Journal of Aesthetics and Art Criticism*, vol. 7, no. 3, pp. pp. 243–256, 1949.

[171] M. Wiesmann and A. Ishai, "Training facilitates object recognition in cubist paintings," *Frontiers in Human Neuroscience*, vol. 4, p. 11, 2010.

[172] A. Ishai, S. L. Fairhall, and R. Pepperell, "Perception, memory and aesthetics of indeterminate art," *Brain Research Bulletin*, vol. 73, no. 4â€“6, pp. 319 – 324, 2007.

[173] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, 2010.

[174] P. Sinha and A. Torralba, "Detecting faces in impoverished images," *Journal of Vision*, vol. 2, November 2002.

[175] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification.," *Nature Neuroscience*, vol. 5, pp. 682–687, July 2002.

[176] M. B. Lewis and A. J. Edmonds, "Face detection: Mapping human performance," *Perception*, vol. 32, no. 8, pp. 903–920, 2003.

[177] D. Y. Tsao and M. S. Livingstone, "Mechanisms of face perception.," *Annual Review of Neuroscience*, vol. 31, pp. 411–437, 2008.

[178] K. Grill-Spector, T. Kushnir, and T. Hendler, "A sequence of object-processing stages revealed by fMRI in the human occipital lobe," *Human Brain Mapping*, vol. 6, no. 4, pp. 316–328, 1998.

[179] R. Vogels, "Effect of image scrambling on inferior temporal cortical responses.," *Neuroreport*, vol. 10, pp. 1811–1816, June 1999.

[180] W. A. Freiwald, D. Y. Tsao, and M. S. Livingstone, "A Face Feature Space in the Macaque Temporal Lobe.," *Nature Neuroscience*, vol. 12, pp. 1187–1196, Sept. 2009.

[181] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *Proc. Int. Conf. on Computer Vision (ICCV)*, pp. 1365–1372, 2009.

[182] L. D. Bourdev, S. Maji, T. Brox, and J. Malik, "Detecting People Using Mutually Consistent Poselet Activations.," in *Proc. European Conf. on Computer Vision (ECCV)*, pp. 168–181, 2010.

[183] A. Akselrod-Ballin and S. Ullman, "Distinctive and compact features," *Image and Vision Computing*, vol. 26, pp. 1269–1276, September 2008.

[184] R. C. Nelson and A. Selinger, "A Cubist Approach to Object Recognition.," in *Proc. Int. Conf. on Computer Vision (ICCV)*, pp. 614–621, 1998.

[185] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[186] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, CBLS, 2014.

[187] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results." http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[188] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing Object Detection Features," in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2013.

[189] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5." http://people.cs.uchicago.edu/~rbg/latent-release5/.

[190] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.