Unsupervised Learning of Spatiotemporal Features by Video Completion

Adithya Reddy Nallabolu

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master's of Science

in

Computer Engineering

Kevin B Kochersberger, Chair

Jia-Bin Huang, Co-chair

Harpreet Singh Dhillon

July 10, 2017

Blacksburg, Virginia

Keywords: Representation Learning, Supervised, Unsupervised

Unsupervised Learning of Spatiotemporal Features by Video Completion

Adithya Reddy Nallabolu

(ABSTRACT)

In this work, we present an unsupervised representation learning approach for learning rich spatiotemporal features from videos without the supervision from semantic labels. We propose to learn the spatiotemporal features by training a 3D convolutional neural network (CNN) using video completion as a surrogate task. Using a large collection of unlabeled videos, we train the CNN to predict the missing pixels of a spatiotemporal hole given the remaining parts of the video through minimizing per-pixel reconstruction loss. To achieve good reconstruction results using color videos, the CNN needs to have a certain level of understanding of the scene dynamics and predict plausible, temporally coherent contents. We further explore to jointly reconstruct both color frames and flow fields. By exploiting the statistical temporal structure of images, we show that the learned representations capture meaningful spatiotemporal structures from raw videos. We validate the effectiveness of our approach for CNN pre-training on action recognition and action similarity labeling problems. Our quantitative results demonstrate that our method compares favorably against learning without external data and existing unsupervised learning approaches.

Unsupervised Learning of Spatiotemporal Features by Video Completion

Adithya Reddy Nallabolu

(GENERAL AUDIENCE ABSTRACT)

The current supervised representation learning methods leverage large datasets of millions of labeled examples to learn semantically meaningful visual representations. Thousands of boring human hours are spent on manually labeling these datasets. But, do we need semantically labeled images to learn good visual representation ? Humans learn visual representations using little or no semantic supervision but the existing approaches are mostly supervised.

In this work, we propose an unsupervised visual representation learning algorithm to learn useful spatiotemporal features by formulating a video completion problem. To predict the missing pixels of the video, the model needs to have a high-level semantic understanding and motion patterns of people and objects. We demonstrate that video completion task effectively learns semantically meaningful spatiotemporal features from raw natural videos without semantic labels. The learned representation provide a good network weight initialization for applications with few training examples. We show significant performance gain over training the model from scratch and demonstrate improved performance in action recognition and action similarity labeling tasks when compared with competitive unsupervised learning algorithms.

To my beloved family,

Mom, Dad and Brother,

Thanks for endless love, support and sacrfices.

Acknowledgments

First and foremost, I would like to thank my two graduate advisors, Prof.Jia-Bin Huang and Prof. Kevin Kochersberger, who have always shown faith in me and consistently guided me all throughout graduate school. They have always been extremely supportive, accommodating and open to ideas throughout my research while constantly driving me in the right direction when necessary. I would like to specially thank Prof.Jia-Bin for spending many hours discussing ideas and giving me the freedom to explore many research topics. I express my gratitude towards Prof. Kevin Kochersberger for funding my research during my masters.

I'd like to thank Prof. Dhillon for serving as my graduate thesis committee member and providing insightful and valuable comments that guide me to improve the thesis. I'd also like to thank Larry from Dupont and Dr. Bird for the valuable discussions.

I thoroughly enjoyed my time being part of Vision & Learning and Unmanned Aerial Systems Lab's at Virginia Tech. I would like to thank my labmates, including Jin-woo, John Peterson, Karim, Sneha, Sanket, Koel for all of their helpful comments and suggestions.

I would thank my friends and cousins: Dinesh, Prashanth, Shwethank, Srilekhya, Linga, Bindu, Lavanya, Abhishek for the distractions and hilarious conversations over the last couple years.

Last but not least, I express my sincere gratitude towards my family: my dearest dad, mom and brother for their unconditioned love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Conventional machine learning algorithms often have difficulty in processing high-dimensional raw natural signals in the form of pixels or text characters. Feature representation extraction — representing raw data in terms of a fixed dimensional vector — is a key to achieving the generalizable performance of machine learning systems. A significant effort in applying machine learning techniques lies in designing preprocessing pipelines that extract and organize discriminative information from the raw data. Figure 1.1 shows the pipeline of conventional machine learning algorithms for image, speech, and text modalities.

For the past few decades, computer vision research mostly focused on the design of hand-crafted features such as Spatial Invariant Feature Transform (SIFT) [37] , Histogram of Oriented Gradients (HOG) [5], spin images [23], or textons [24]. These features capture low-level gradient details such as edges, corners, color features in the images, ignoring the high-level visual semantics (e.g., object parts, scene layout). Figure 1.3 provides examples of the feature representation extraction for SIFT and HOG features. Though these features achieved state of the art performance during their time, the overall performance is limited.

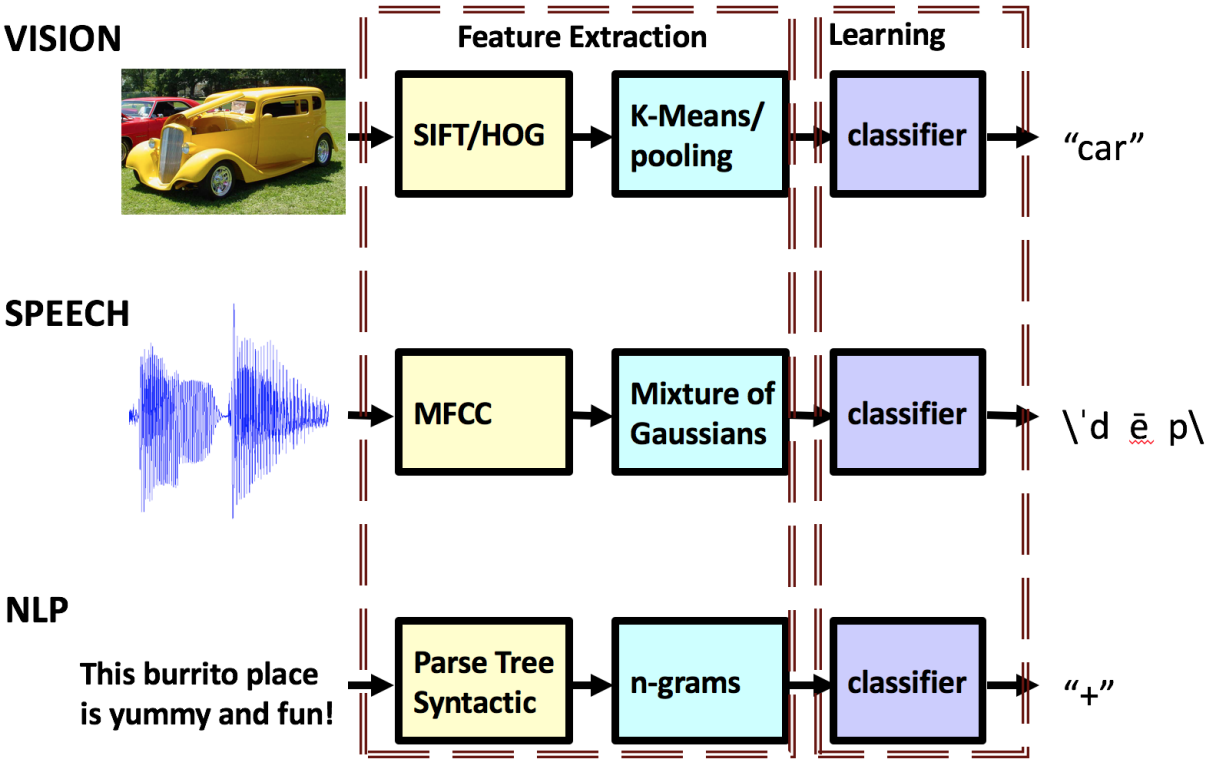


Figure 1.1: **Conventional machine learning systems.** Conventional machine learning systems are implemented in two stages. The first stage is feature representation extraction, where a fixed dimensional features are extracted from raw high dimensional signals like image, voice and text. The second stage is training a classification model using the extracted features.

However, the last four years have seen the resurgence of learning visual representations directly from pixels themselves using the deep learning or Convolutional Neural Networks (CNNs). The CNNs are a class of multi-level neural networks composed of a series of trainable feature transform followed by a classifier. By training the CNNs end to end using back propagation and stochastic gradient descent, the network learns a hierarchy of abstract features at multiple levels. The CNN's have demonstrated state-of-the-art performance on a wide variety of vision tasks [30] such as image classification, object recognition, segmentation. Figure 1.2 shows a CNN model capable of learning a hierarchy of multi-level features directly from pixels.

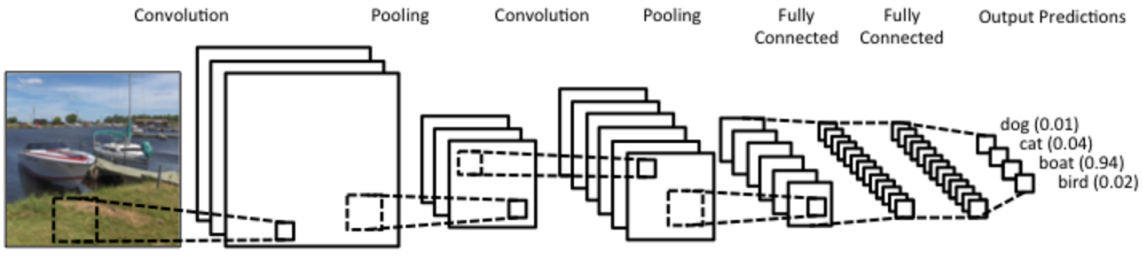


Figure 1.2: **Convolutional Neural Network (CNN's)** The CNN's are sequential multi-level neural networks with a few special layers such as pooling, convolution, fully-connected etc. The CNN's are trained end to end using back propagation and stochastic gradient descent. Reprinted from "Understanding Convolutional Neural Networks for NLP", In wildml by Danny Britz, Retrieved November 7, 2015, from http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. Copyright 2015 by Danny Britz. Reprinted with permission.

1.1 Motivation

At the heart of CNNs is a completely supervised learning paradigm. Often millions of examples are labeled using Mechanical Turk to create tens of millions of training instances. Mechanical Turk (MTurk) is a crowd-sourcing Internet marketplace enabling researchers to collect semantic labels from human workers. The cost of collecting large-scale labeled data (i.e. hundreds of billions of images) for supervised learning, however, is substantial as it requires tens of thousands of manual hours. The task of labeling is also boring as these tasks are typically dull and repetitive. Sometimes there is a good chance that multiple workers disagree and have different answers for the same question. This pushes the need to re-validate the obtained labels based on the worker's disagreement taking additional time for collecting labels. There is also evidence which showed that many of the workers did not enjoy the work and only completed the task for monetary compensation.

In this thesis, we ask if the fully supervised learning setting is necessary for training these CNNs? Do we really need tens of millions of manually labeled images to learn good visual representations? Or, can we learn good visual representations using few labeled examples? The humans or other

biological organisms learn visual representations with little or no semantic supervision. However, the current computational approaches still remain entirely supervised. It is thus of great interests to develop unsupervised learning algorithms that can learn semantically meaningful and generalizable features from raw data without semantic labels.

1.2 Background

There have been extensive research efforts to develop unsupervised algorithms for learning semantically meaningful representations from raw data. Clustering, dimensionality reduction and maximum likelihood density estimation are some of the early work in unsupervised learning. The primary goals of these algorithms are on data compression and are not focused towards learning discriminative features.

Recently there has been a surge of interest in the unsupervised learning approaches , which exploit freely available cues in the visual data as supervisory signals to learn discriminative features. Examples of supervisory signals include spatial contexts within images [7, 46, 49], image matching [34, 62], and ego-motion [1, 21] are exploited to create a pre-text task to learn powerful visual representations.

In addition to using image data, several recent work leverage the temporal coherence and redundancy in videos for representation learning, *e.g.*, order verification [42], frame prediction [57, 40], video generation [61], and atomic 3D flow prediction [38]. These tasks exploit the freely available temporal context in the videos.

While existing unsupervised feature learning approaches have demonstrated good performance for pre-training CNNs, most of the approaches use either the spatial or the temporal contexts of visual data *separately*. As a result, the learned representation may not be effective in extracting informa-

tive spatiotemporal features from videos. Deep three-dimensional CNN (C3D) [59] provides an efficient and simple way to capture the scene dynamics and semantics. However, training such a network requires large-scale manually annotated video examples (*e.g.*, Sports-1M [25]). To avoid training a C3D model from scratch, the initial network weights of C3D can be transferred from a 2D spatial CNN pre-trained on ImageNet [39]. However, it remains an open question how we can learn semantically meaningful features from unlabeled videos.

1.3 Our Contributions

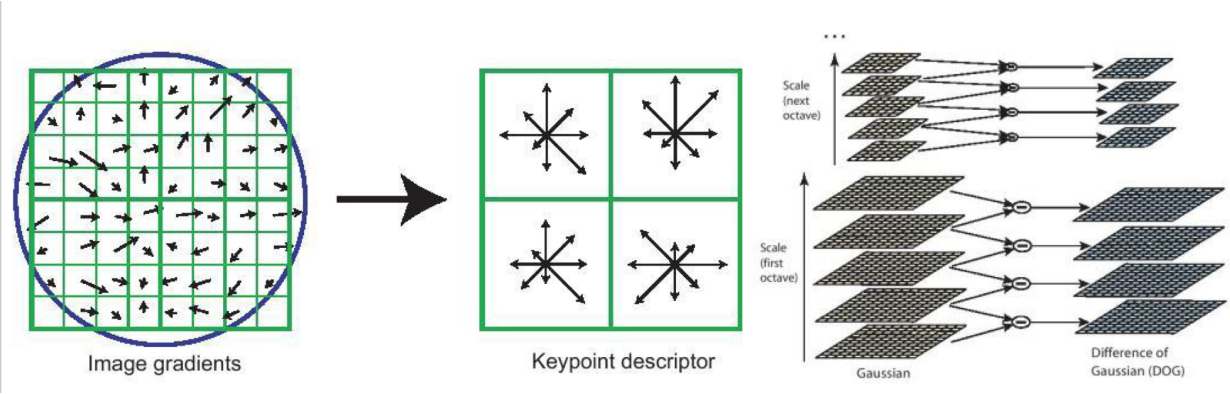
In this work, we propose an unsupervised representation learning algorithm to learn useful spatiotemporal features. We formulate a feature learning task as a video completion problem. Given an input video, we mark the center region of frames as missing pixels (*i.e.*, assigning the pixel values to 1) across all the frames in the video sequence (as shown in Fig. 1.4). We then train a 3D CNN model to predict the missing pixel values conditioned on the known contents of the video using pixelwise reconstruction loss. Here, as the missing pixels form a space-time cuboid, the CNN needs to consider simultaneously *both* spatial and temporal contexts to understand and predict the motion in order to fill in plausible and temporally coherent contents. In Fig. 1.4, we show that the learned model is capable of producing plausible missing contents.

We validate the proposed spatiotemporal feature learning task (*i.e.* video completion) by jointly taking both color and flow videos into account. The color and the flow representations are complementary. To predict the missing color pixels, the model needs to have a high-level understanding of motion patterns of people and objects. Similarly, to predict the missing flow fields, the model needs to reason the about the semantics from color videos. To exploit the complementary information in color and flow modalities, we formulate a joint flow-color context encoder for simultaneously learning video representations using both color and flow.

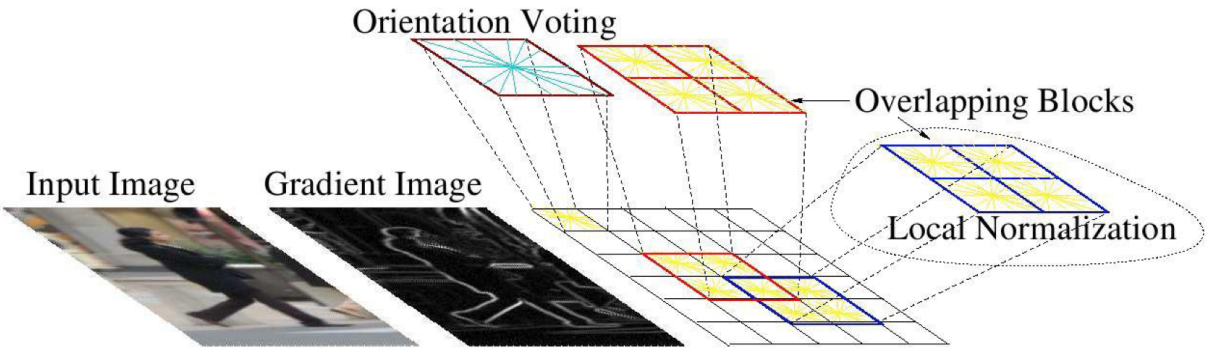
We validate the effectiveness of the learned spatiotemporal features on action recognition and action similarity labeling tasks. Using our method a CNN pre-training method, we show that our method compares favorably against existing unsupervised feature learning techniques.

We make the following three contributions in this work:

- We propose an unsupervised representation learning approach for learning semantically meaningful spatiotemporal features from raw natural videos without semantic labels. We demonstrate the importance of considering both spatial and temporal contexts.
- We formulate a joint feature learning task using color and flow video and demonstrate its advantages over using individual modality.
- We show that learned representation from our method can provide good network weight initialization for applications with few training examples. We achieve significant performance gain over training the model from scratch and demonstrate improved performance in action recognition and action similarity labeling tasks when compared with competitive unsupervised learning algorithms.



(a) SIFT



(b) HoG

Figure 1.3: **Conventional features.** Conventional features in vision capture low-level details like edges, corners or color features in the images. (a) SIFT (Spatial Invariant Feature Transform). Adapted from opencv tutorials by Doxygen. Retrieved December 18, 2015, from http://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html. Copyright 2015 by Doxygen. Reprinted with permission. (b) HoG (Histogram of Oriented Gradients). Adapted from "Unsupervised learning of human action categories using spatial-temporal words" by Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei, 2008, International journal of computer vision, 79(3):299318. Copyright 2008 by Springer. Adapted with permission.

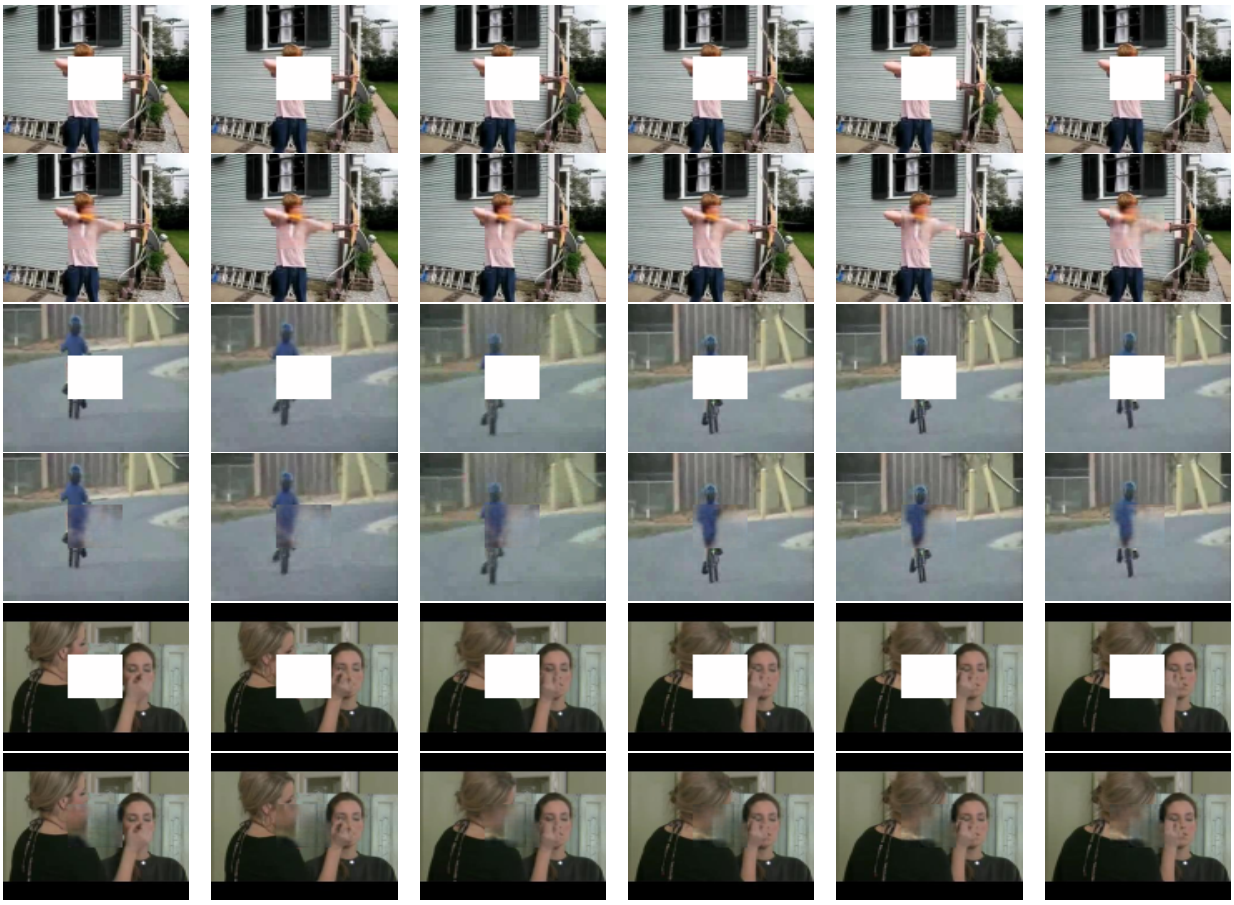


Figure 1.4: **Unsupervised spatiotemporal Feature Learning**. Given an video with a masked cuboid volume as input (odd rows), we train the model to predict the masked cuboid volume of the video as output (even rows). While the training process does not involve semantic labels, predicting these missing pixels requires a certain level of understanding of the spatiotemporal dynamics. In this work, we leverage video completion as a surrogate task to learn powerful spatiotemproal features in an unsupervised manner.

Chapter 2

Related Work

2.1 Feature learning for videos

The success of 2D CNN models for image recognition leads to the quest of developing a powerful representation for videos. There are three main model architectures with different temporal connectivity pattern for modeling the motion information presented in videos.

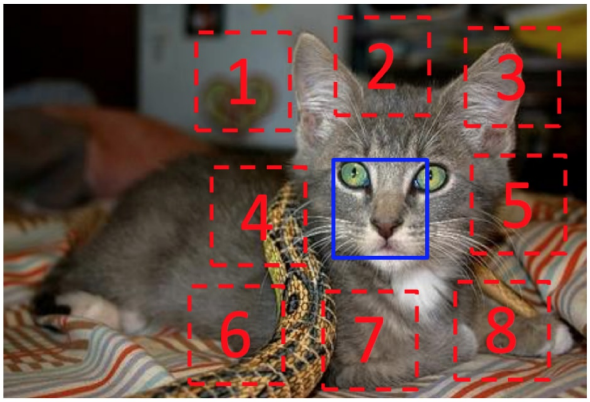
First, using a 2D CNN model to extract appearance features in each frame and then use Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) to capture the temporal variations [57]. Second, 3D CNN architectures [59, 25] directly encode motion information using 3D convolution filters. Third, the 3D CNN and LSTM can be combined to capture both local motion and global temporal changes [65, 66]. In this paper, we aim to learn spatiotemporal features with a 3D CNN architecture due to its simplicity. The 3D CNN architecture also are more suitable for spatiotemporal feature learning as they demonstrate superior performance in action recognition over other model architectures.

2.2 Unsupervised representation learning

While CNN based model has shown record performance on recognition tasks, training such a model requires millions of labeled examples. There has been a recent surge of interest in unsupervised learning. The visual world is very diverse, yet highly structured, and humans have an uncanny ability to make sense of this structure. Inspired from humans, several recent unsupervised learning works exploit the visual structure that are readily available within visual data to use them as intrinsic reward signals to learn generalizable visual features. Unsupervised representation learning can also be formulated as learning an embedding (i.e. a feature vector for each image or video). Where the images or videos are separated by their similarity in semantic content in the embedding space.

From unlabeled images One line of work focuses on generative models for learning representations in a reconstruction based encoder-decoder framework [47, 8, 60, 52, 18, 2]. These reconstruction-based algorithms struggle with low- level phenomena, like stochastic textures, making it hard to even measure whether a model is generating well.

Spatial context is freely available in the natural images and can be exploited to learn meaningful representations for images. A similar convention (of exploiting context) exists in the text domain, where skip-gram [41] models have been shown to generate powerful word representations. Doersch *et al*. [7] trains a CNN model to predict the relative spatial position between two image patches as a pre-text task to learn semantically meaningful and generalizable image representations. This straightforward context prediction task is extended to solving jigsaw puzzles [46]. Pathak *et al*. [49] trains a generative CNN model to generate the missing contents of an arbitrary image region conditioned on its surroundings. Doing well on these pre-text tasks requires the trained model to understand the semantics of object parts, objects, and scenes in the real world. Figure 2.1 shows



(a) Context prediction

(b) Pixelwise context prediction

Figure 2.1: **Spatial context as supervisory signal** many contemporary unsupervised learning algorithms explore spatial context as a supervisory signal to learn rich visual features. (a) Context prediction. Doersch *et al*. [7] trains a CNN to predict the relative spatial position between two image patches.Adapted from "Unsupervised visual representation learning by context prediction" by Carl Doersch, Abhinav Gupta, and Alexei A Efros, 2015, ICCV. Copyrights 2015 by Carl Doersch. Adpated with permission. (b) Pixelwise context prediction. Pathak *et al*. [49] trains a generative CNN model to generate the missing contents of an arbitrary image region conditioned on its surroundings. Adapted from "Context encoders: Feature learning by inpainting" by Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros, 2016, CVPR. Copyrights 2016 by Deepak Pathak. Adapted with permission.

two models which exploit spatial context a supervisory signal to learn visual features.

Visual semantics can be learned by training the CNN to predict the color histogram [33] from the input gray scale image. This task is extended to predict the cross channel in Zhang *et al*. [67]. The trained model has to interpret the semantic composition of the scene (what is in the image) as well as localize objects (where things are) in order to predict cross channel data.

Ambient sounds [48] also provide a supervisory signal (as sound captured is correlated to objects and scenes) for learning visual representations. Predicting the success of robotic tasks [51] like grasping, pushing can also be used as pre-text task to learn visual representations. Training a CNN

model to predict random Noise As Targets (NAT) [3] solves the standard unsupervised learning issues of trivial solutions and collapsing of features.

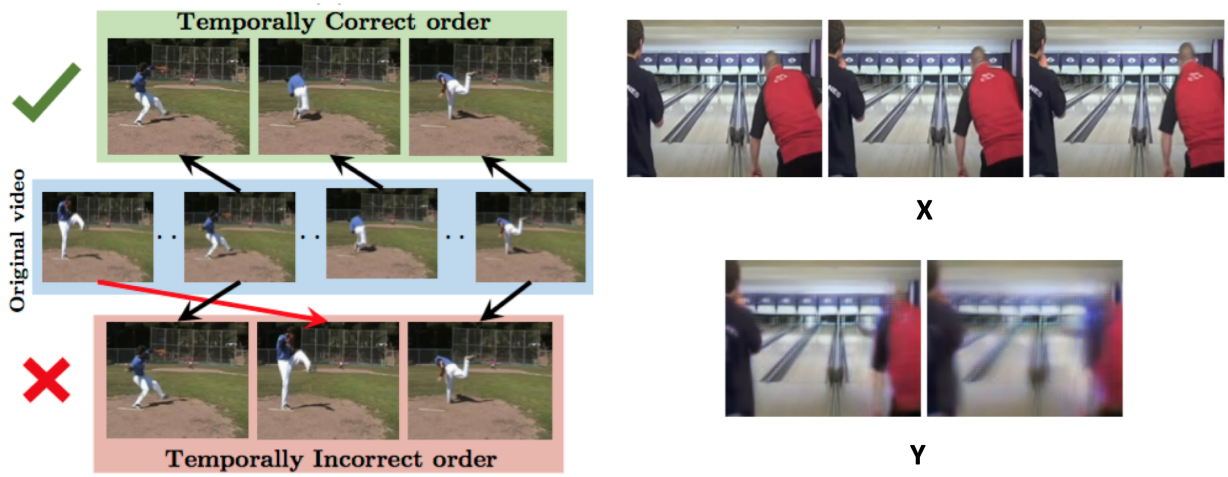
From unlabeled videos Compared to image data, video data provides richer supervisory signals for learning effective representations with the additional time dimension. Videos have temporal context (*i.e*. the motion pattern of objects and people) in addition to the spatial context in the images.

For examples, the rich visual representation can be learned from imposing temporal smoothness constraints [9, 64, 68, 22, 58]. These constraints capture the fact that high-level visual signals in a video change slowly over time. Some works [1, 21] use ego-motion constraints from video to further constrain the learning. Jayaraman *et al*. [21] show how they can learn equivariant transforms from such constraints.

Visual tracking [62] can be exploited for capturing the appearance variation of objects and enforce that two patches connected by a track should have similar visual representation in deep feature space since they probably belong to the same object. A sequence order verification [42] of frames in a video help in learning powerful temporal features. Figure 2.2 shows two example models which exploit temporal context a supervisory signal to learn visual features.

Several works exploit the temporal dimension in the video and propose several pre-text tasks like frame interpolation [35], future frame prediction [57, 40, 36] and learn video features cheaply without any processing. Recent work also proposes to predict future atomic motions [38] using pre-computed discrete flows as a pre-text task. These video representation capture long-term motion dependencies and spatial-temporal relations.

Our work shares the same goal with existing work on unsupervised representation learning from video. We propose to learn spatiotemporal features by data-driven video completion. The work



(a) Shuffle & learn

(b) Pixelwise future prediction

Figure 2.2: **Temporal context as supervisory signal** many contemporary unsupervised learning algorithms explore temporal context as a supervisory signal to learn rich visual features. (a) Shuffle & learn. Misra *et al*. [42] trains a CNN to verify a sequence of frames are in right order or not. Adapted from "Shuffle and learn: unsupervised learning using temporal order verification" by Ishan Misra, C Lawrence Zitnick, and Martial Hebert, 2016, ECCV. Copyrights 2016 by Ishan Misra. Adapted with permission. (b) Pixelwise future prediction. Srivastava *et al*. [57] trains a generative model to predict future frames conditioned on the input frame sequence. Adapted from "Unsupervised learning of video representations using lstms" by Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov, 2015, ICML. Copyrights 2015 by Nitish Srivastava. Adapted with permission.

most related to our method is that of Pathak *et al*. [49] for feature learning using image completion. We also exploit the similar type of context encoder for learning the features.

Our method differs in the following two aspects. First, instead of learning *image-based* features using a 2D CNN, we focus on learning *spatiotemporal* features for videos. We show that joint modeling of spatial and temporal contexts is critical for learning effective spatiotemporal features. Second, we propose a joint flow-color context encoder that allows us to simultaneously learn features for flow and color videos. We demonstrate that leveraging the complementary nature of flow and color leads to improved performance over learning from individual modality.

2.3 Image and video generation

Deep generative models have recently attracted considerable attention [54, 11, 27, 43]. Recent efforts include the use of Laplacian model [6], adversarial (GAN) loss function [52, 11], and video generation [61]. Our work can also be viewed as a deep video generation model *conditioned* on the surrounding contents. As a proof of concept, we use the standard pixelwise reconstruction loss to train the model. We believe that incorporating advanced techniques in deep generative models (*e.g.*, augmenting the reconstruction loss with the adversarial loss as used in [49]) may further improve the quality of the learned spatiotemporal features thanks to the ability to handle multiple modes in the output video space.

2.4 Video completion

Video completion aims at filling spatiotemporal hole with plausible contents. State-of-the-art algorithms use patch-based optimization [19, 45, 63] or segmentation-based methods [12, 14] to synthesize the missing regions by *transferring* patch/segments from known regions. These video completion algorithms synthesize the missing (target) regions by sampling spatio-temporal patches/segments from the known (source) regions or by solving spatio-temporal shift-maps using graph cuts In contrast, our network is trained to *predict* the pixel values of missing regions using the latent features extracted from the encoder.

Chapter 3

Video Completion

In this section, we describe the proposed algorithm for learning spatiotemporal features. We start with presenting the proposed network architecture for training the pixel prediction network in Section 3.1. In Section 3.2, we describe the video completion task. We explore three different types of training data to train the proposed network: (1) spatial context only, (2) temporal context only, and (3) spatiotemporal context. These variants allow us to study the relative importance of incorporating spatial and temporal surrounding contexts for the reconstruction task.

3.1 Network architecture

Fig. 3.1 shows the proposed architecture for reconstructing missing spatiotemporal contents. We use the conventional deep autoencoders as our network. The overall architecture consists of an encoder and a decoder. The encoder takes an input video with missing pixels located at the centered region and punching through the entire video sequence. Here, the purpose of the encoder is to extract a latent spatiotemporal feature representation. We then use two fully-connected layers to transform the features so that it captures the entire space-time contents of the video. The decoder

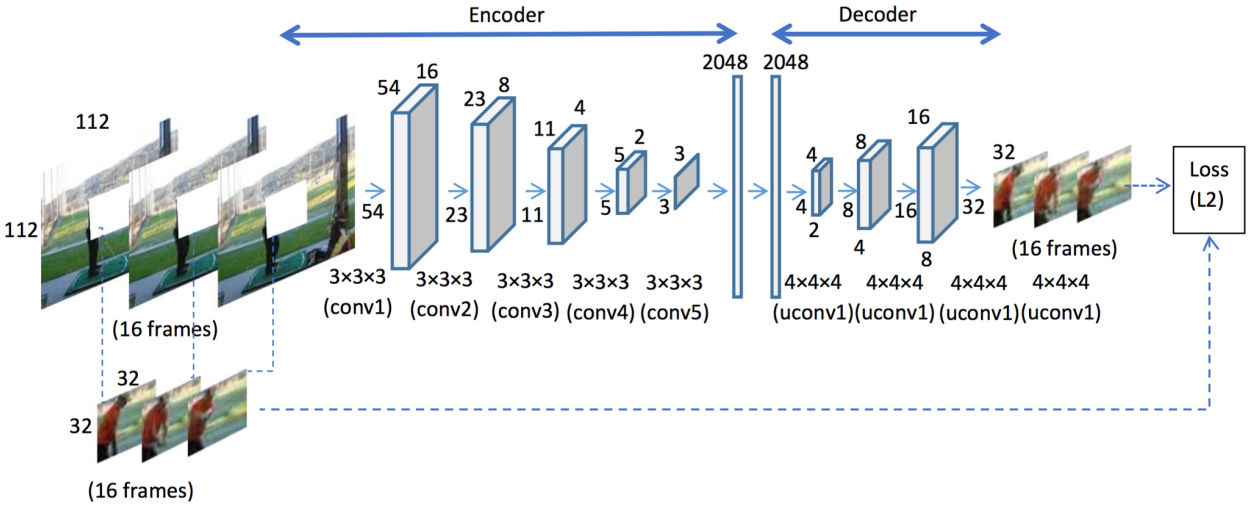


Figure 3.1: **Overview of the video completion architecture.** A video with masked cuboid volume is given as input to the spatiotemporal model. The encoder of the model encodes the masked video into a latent representation space as explained in Section 3.1.1 . The decoder produces the masked video from latent representation

then takes the transformed feature representation (after the fully-connected layers) and predict the missing spatiotemporal hole.

During the encoding process, the spatial and temporal resolution are progressively reduced due to the pooling operations. In the decoder, we use a cascade of up-convolutional layers (also known as transposed convolution) to recover the spatial and temporal resolution.

3.1.1 Encoder

We adopt the C3D architecture [59] as our encoder. The encoder takes an input video of 16 frames of size 112×112 pixels ($3 \times 16 \times 112 \times 112$ dimensions) and produces a 2048-dimensional latent feature vector as its output. The encoder consists of five convolution layers followed by two fully connected layers as shown in Figure 3.1. All convolution layers are 3D volumetric convolutions followed by volumetric pooling and ReLU layers. The convolution layers contain 64, 128, 256,

256, 256 filters in each layer, respectively. All the convolution filters are of size $3 \times 3 \times 3$ with stride one. As reported in [59], using small and simple $3 \times 3 \times 3$ convolutional filters achieves better performance than other kernel sizes. All pooling layers are max pooling with a kernel size of $2 \times 2 \times 2$ except the first pooling layer. The first pooling layer has a kernel size of $1 \times 2 \times 2$. This prevents the encoder losing valuable temporal information in the first layer.

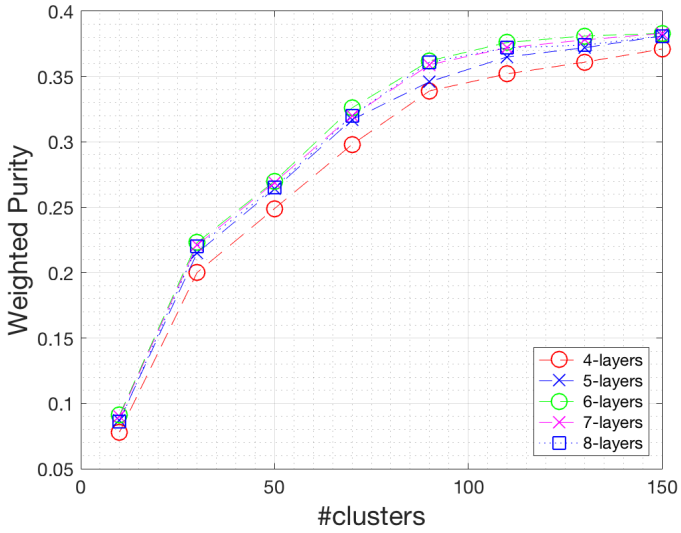


Figure 3.2: **Weighted purity by varying the number of conv layers of encoder**. Quantitative comparison of our learned features in the weighted purity of clusters versus the number of clusters on the HMDB-51 dataset.

Ablation study on the encoder architecture We studied the network architecture design by varying the number of layers in the encoder (from 4 to 8 layers). Using the purity of the learned unsupervised representations, we quantify the performance of different architecture. We have used purity of the learned unsupervised representations as a quantitative metric to determine the best architecture for the C3D classification model. Figure 3.2 shows the quantitative performance comparison of unsupervised spatiotemporal features by changing the encoder layers. The best purity score of 0.37 is achieved for the encoder with 6 layers with 100 clusters.

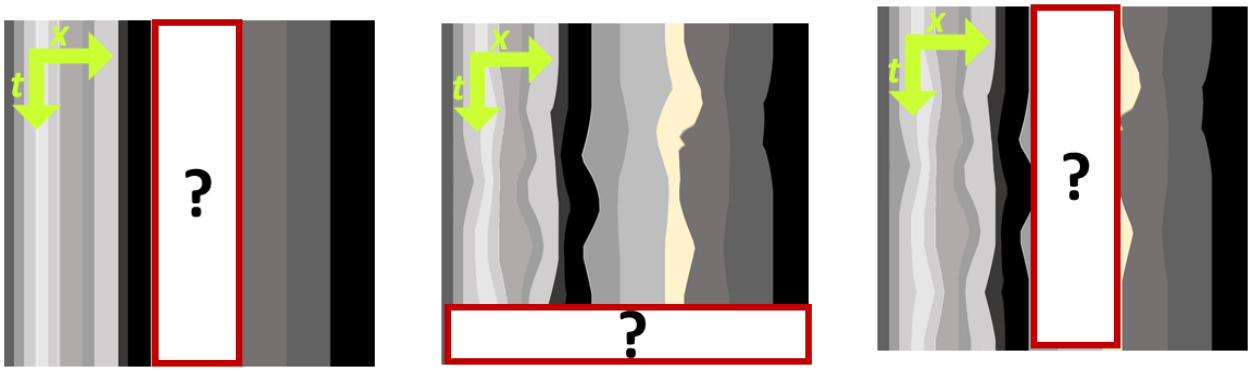
3.1.2 Decoder

The decoder takes the extracted 2048-dimensional feature vector and use a series of four up-convolutional layers (followed by a tanh normalization layer and ReLU layer) to upsample the temporal resolution to 16 frames and the spatial resolution of 32×32. The tanh normalization layer is required to produce normalized missing visual contents as output. Each convolutional layer uses a filter of size $4 \times 4 \times 4$ (except for the first layer which uses the kernel of size $2 \times 4 \times 4$).

3.1.3 Loss function

We impose standard pixelwise reconstruction loss ℓ_1-norm on the network output using the ground truth RGB pixel values (the hold-out center region in the original video). We note that there could be many possible ways to generate plausible missing regions that are consistent with the rest of the input video (*i.e.*, the output space may have multiple modes). Minimizing the reconstruction loss (ℓ_1-norm) may fail to the multiple modes of the solution space and lead to blurry results. The reconstruction quality can be further improved by adding adversarial loss (as done in [49]). Our main goal, however, is to learn effective spatiotemporal features rather than high-quality video completion.

L2 vs L1 loss The compare the quantitative performance of unsupervised feature learning using pixelwise ℓ_2-norm and ℓ_1-norm reconstruction loss, we initialize the weights of action recognition model with the weights learned using either ℓ_2-norm and ℓ_1-norm. We then finetune the action recognition model on UCF-101 training data for 90K iterations with a learning of 3×10^{-3}. The performance of action recognition model with ℓ_1-norm is better than ℓ_2-norm by 0.2%. But, the qualitative video completion results produced from training using ℓ_1-norm are slightly sharper than ℓ_2-norm.



(a) Spatial context only (b) Temporal context only (c) Spatial + temporal context

Figure 3.3: **Spatial, temporal, and spatiotemporal.** X-T slice visualization of training data collection for unsupervised representation learning using spatial, temporal and spatiotemporal context. (a) For spatial context only, we replicate a single frame across the temporal dimension. (b) For temporal context only, we hold-out only the last frame from a video. (c) For spatiotemporal context, we hold-out a spatiotemporal volume.

3.2 Learning by video completion

We now introduce the training data generation for learning effective spatiotemporal features. We also explore two other variants of training data to evaluate the importance of jointly considering spatiotemporal context. Specifically, we generate training data so that either only spatial context or only temporal context is used. By training a video completion network using these three sets of data, we can evaluate the relative importance of exploiting spatial and temporal contexts.

Spatial context only: We use an image with a held out center patch as input. We duplicate the image across the temporal channel. The model predicting the missing image patch can only rely on the available spatial context.

Temporal context only: We take a sequence of frames and train the model to predict one future frame from the past observation.

We use the same network to train all the variants of the training data. The main differences lie in the way to collect training data and the uconv parameters of the decoder. For the spatial model, we duplicate a single frame 16 times to generate the input. Similar to the spatiotemporal model, we use 16 consecutive frames as input to the temporal model. We modify the uconv parameters of the decoder in the video completion model to predict the desired output in spatial and temporal models. The spatial model generates the missing image patch as output. The temporal model generates the next frame with a spatial resolution of 112×112 as output. Figure 3.3 illustrates the difference between the spatial, temporal and spatiotemporal tasks using spatiotemporal slices.

Chapter 4

Joint Flow-Color Context Encoder

A video can be decomposed into static and dynamic components. The static part, in the form of individual frame appearance, carries information about scenes and objects depicted in the video. The dynamic part, in the form of motion across the frames *i.e*. optical flow, conveys the movement of the observer (the camera) and the objects.

4.1 Optical flow

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene. The optical flow methods compute the pixelwise motion between a pair image frames.

Optical flow can be seen as a set of displacement vector fields d_t between the pairs of consecutive frames t and $t+1$. The displacement vector at the point (u,v) in frame t is $d_t(u,v)$, which moves the point to the corresponding point in the following frame $t+1$. The horizontal and vertical components of the displacement vector field are d_t^x and d_t^y, which can be represented as two

separate image channels. We also compute a total distance component from horizontal and vertical displacement components $d_t\sqrt[2]{x^2+y^2}$ as a third channel.

4.2 Joint spatiotemporal feature learning

In this section, we describe the proposed algorithm for learning of joint spatiotemporal features using color and flow videos. Figure 4.1 illustrates the overall architecture design. In the above Section 3.2, we formulated video completion task as predicting the missing pixel values conditioned on the known contents of the video. The proposed feature learning task is generic and can be applied to any volumetric data. The color and the flow representations are complementary. To predict the missing color pixels, the model needs to have a high-level understanding of motion patterns of people and objects. Similarly, to predict the missing flow fields, the model needs to reason the about the semantics from color videos. To exploit the complementary information in color and flow modalities, we formulate a joint flow-color context encoder for simultaneously learning video representations using both color and flow.

Here, we validate our proposed feature learning task on color (color model) and flow (flow model) modalities of videos. The *(*color model) is trained on the video completion task to predict the missing color pixel values conditioned on the known contents of the color video. Similarly, the *(*flow model) is trained to predict the missing flow vector fields. We also propose a *(*joint flow-color context encoder) to simultaneously learn spatiotemporal features using color and flow videos as shown in Figure 4.1. we discuss the network architecture of our proposed spatiotemporal feature learning model in Section 4.3.

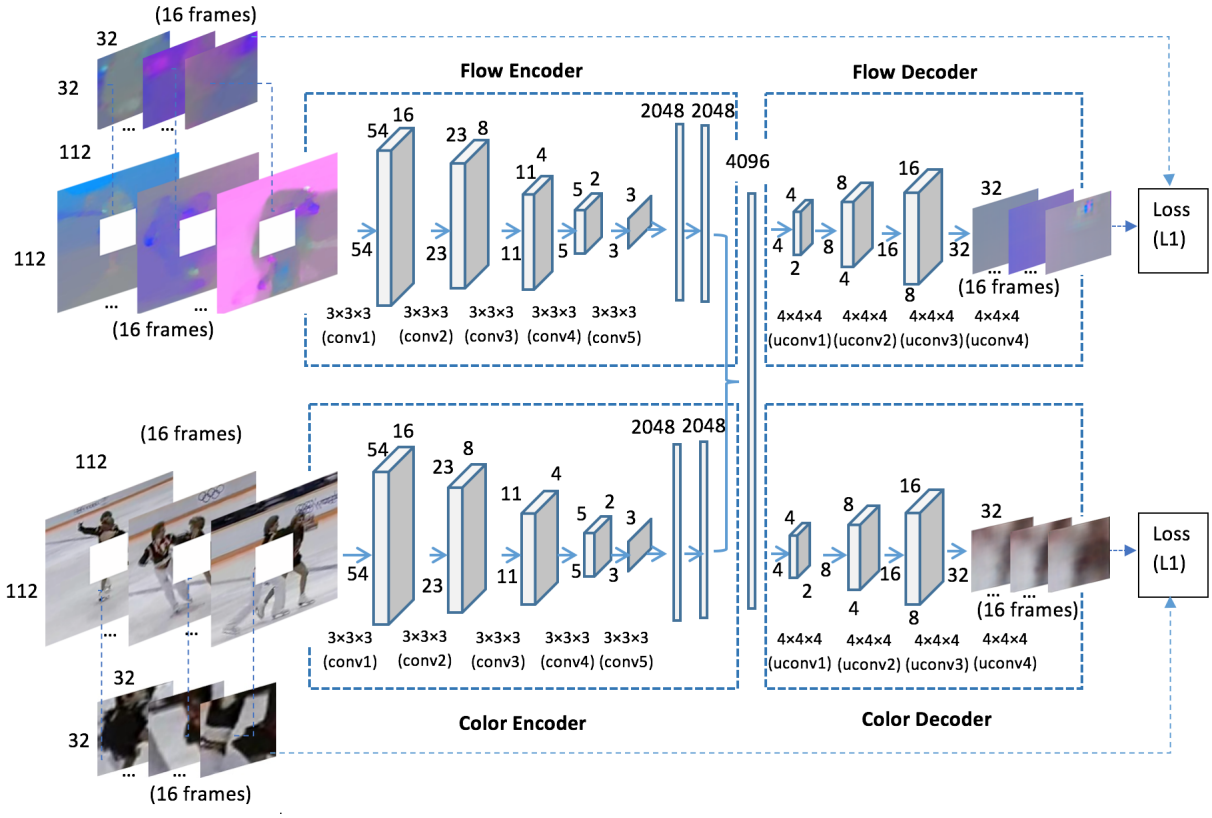


Figure 4.1: **Overview of the proposed joint flow–color context encoder.** For each given sample sequence of size (112×112×16) , our model takes the color video and the corresponding flow video with held-out regions (a centered masked cuboid of size 32×32×16) as inputs. The network architecture consists of two encoders: one for color video and one for flow. The role of two encoders is to map the input flow and color videos to latent feature representations. The two latent features are then concatenated to form a joint flow–color feature. We then use two decoders to convert this joint feature vector to predict the corresponding masked flow and color regions. While there two modality (color and motion), we are able to training this joint flow–color context encoder in an end-to-end fashion without the need of stage-wise optimization.

4.3 Joint Network architecture

The architecture of a video completion model contains an encoder followed by a decoder. The encoder takes an input video with missing pixels located at the centered region and punching through the entire video sequence. Here, the purpose of the encoder is to extract a latent spatiotemporal feature representation. We then use two fully-connected layers to transform the features so that it captures the entire space-time contents of the video. The decoder then takes the transformed feature representation (after the fully-connected layers) and predict the missing values in the spatiotemporal hole. During the encoding process, the spatial and temporal resolution are progressively reduced due to the pooling operations. In the decoder, we use a cascade of up-convolutional layers (also known as transposed convolution) to recover the spatial and temporal resolution.

We use identical architectures of the video completion network for training on color and flow videos. The architecture of the *joint model* contains two parallel encoders (to encode color & flow videos) and a concatenation layer followed by two parallel decoders (to decode the missing color and flow values in the hole). Figure 4.1 shows the proposed network architecture for simultaneously reconstructing the missing spatiotemporal contents from input color and flow videos. While we have separate color and flow encoders/decoders, the network is trained in an end-to-end fashion.

Chapter 5

Experimental Results

5.1 Implementation details

We implement all models in torch, a popular open source deep learning framework . The network initializations play a significant role in the convergence of a neural network to the optimal weights. Xavier [10] and Kaiming [17] are some popular initializations strategies which are shown to yield good convergence speeds for training a deep network. But, we initialize the network weights from a standard normal distribution with variance 0.01 to compare our performance with other baseline approaches. We use a learning rate of 3×10^{-3} and reduce it 10 times for every 40K iterations. The network is trained with a batch size of 30. We use ADAM [26], a first-order gradient-based optimization of the stochastic gradient descent.

We use videos from UCF-101 dataset [56] for our unsupervised pre-training task. UCF-101 is an action recognition data set of realistic action videos, collected from YouTube, having 101 action categories. This data set is an extension of UCF-50 data set which has 50 action categories.

We only use videos from the *train* set of UCF-101 dataset to compare our unsupervised pre-training

with existing unsupervised approaches [42]. The train set contains over 9.5K videos . From each video, we sample 5 to 6 video clips of 16 frames. We resize the spatial resolution of the videos to 128×172. We then randomly crop the video to obtain a spatial resolution of 112×112. We also randomly flip the training data horizontally with a probability of 0.5.

We use flow videos for training our unsupervised joint pre-training task. The flow field is computed with [4] and then transformed into a flow image by scaling and shifting x and y flow values to a range of $\left[128, +128\right]$. We add a third channel for the flow image with the flow magnitude. We use precomputed optical flow frames from [55] to reduce the processing time during training. We sample the flow frames with RGB frames for training the joint flow-color context encoder. Training one one epoch with 4200 batches of the training data takes an average of 15 hours on NVIDIA Tesla K80 GPU. All the source code and pre-trained models will be made publicly available.

5.2 Action recognition

Recognition of human actions in videos is a challenging task which has received a significant amount of attention in the research community Compared to still image classification, the temporal component of videos provides an additional (and important) clue for recognition, as a number of actions can be reliably recognized based on the motion information. Additionally, video provides natural data augmentation (jittering) for single image (video frame) classification Background clutter, fast irregular motion, occlusion, viewpoint changes pose significant challenges in solving action recognition.

Video recognition research has been largely driven by the advances in image recognition methods, which were often adapted and extended to deal with video data. A large family of video action recognition methods is based on shallow high-dimensional encodings of local spatio-temporal features. For instance, the algorithm of [32] consists in detecting sparse spatio-temporal interest

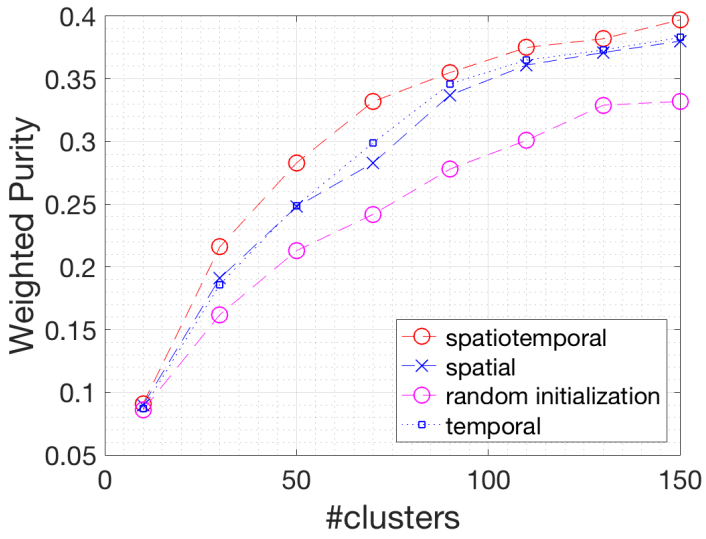


Figure 5.1: **Comparisons of spatial, temporal, and spatiotemporal context.** Quantitative comparison of our learned features in the weighted purity of clusters versus the number of clusters on the HMDB-51 dataset. Features which approach one at a faster rate have better performance. Here we compare features learned from spatial, temporal, and spatiotemporal context as well as random initialization. The results show that leveraging spatiotemporal context produces higher purity measure compared to the random initialization and other alternatives.

points, which are then described using local spatio-temporal features: Histogram of Oriented Gradients (HOG) [5] and Histogram of Optical Flow (HOF). The features are then encoded into the Bag Of Features (BoF) representation, which is pooled over several spatio-temporal grids (similarly to spa- tial pyramid pooling) and combined with an SVM classifier. Further research has shown that dense sampling of local features outperforms sparse interest points.

5.2.1 Datasets.

We use two publicly available benchmark datasets UCF-101 [56] and HMDB-51 [31] to evaluate the performance on action recognition using our CNN pre-training network.

UCF-101 UCF-101 dataset consists of $13,320$ videos with 101 action labels comprising of human-human or human-object interactions. With 13320 videos from 101 action categories, UCF-101 gives the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc., it is the most challenging data set to date. As most of the available action recognition data sets are not realistic and are staged by actors, UCF-101 aims to encourage further research into action recognition by learning and exploring new realistic action categories. The videos in 101 action categories are grouped into 25 groups, where each group can consist of $4-7$ videos of an action. The videos from the same group may share some common features, such as similar background, similar viewpoint, etc.

HMDB-51 HMDB-51 dataset contains 3 splits for train and test set, each with about 3.4K videos for training and 1.4K videos for testing. There are in total 51 action categories.HMDB51 video sequences are also extracted from commercial movies as well as YouTube. The dataset represents a fine multifariousness of light conditions, situations and surroundings in which the action can appear, captured with different camera types and recording techniques such as points of view. The point of view is another criterion of subdivision the HMDB supports. For an all-around coverage the perspectives frontal, lateral (right and left) and backwards view of motions are distinguishable.

5.2.2 Comparison with existing unsupervised learning methods.

We compare the performance of the learned spatiotemporal features using color videos with the existing unsupervised methods. We initialize the weights of our action recognition model with the weights of the color encoder of the pre-trained joint model. We then finetune the action recognition model on UCF-101 training data for 90K iterations with a learning of 3×10^{-3}. The learning is reduced by 10 times after every 40K iterations. On HMDB-51, we only finetune the network for

Table 5.1: **Quantitative comparison on action recognition datasets.** Action recognition performance (in terms of classification accuracy %) on the split-1 of the UCF-101 and HMDB-51 datasets. The first block shows two models pre-trained trained with large-scale semantic labels. The second block compares the performance of unsupervised CNN pre-training algorithms. All the models use the available training data in UCF-101 or HMDB-51 to finetune the networks for action recognition. For fair comparison, here "Ours" indicates our color encoder.

Methods	UCF-101	HMDB-51
Simonyan *et al*. [55]	65.4	46.6
Supervised C3D	85.2	-
Wang *et al*. [62]	40.7	15.6
Mobahi *et al*. [44]	45.4	15.9
Hadsell *et al*. [15]	45.7	16.3
Misra *et al*. [42]	50.9	19.8
Vondrick *et al*. [61]	52.1	-
Random init	39.8	18.4
Ours	**53.8**	**26.7**

50K iterations with a learning of 3×10^{-3}. The learning rate is reduced by 10 times after every 25K iterations.

In Tab. 5.1, we evaluate and compare the action recognition accuracy with other unsupervised learning models [62, 44, 15, 42, 61]. We also show results from models that are pre-trained with large-scale semantic labels, *e.g*., simonyan *et al*. [55] and supervised C3D [59]. We note that all the models are fine-tuned using either UCF-101 or HMDB-51 training videos. Our model contains 8,414,528 parameters which is 8 times less compared to the parameters in the Misra *et al*. [42]. Our approach achieves 14% and 8.3% improvement over the randomly initialized weights on UCF-101 and HMDB-51 datasets, respectively.

The results show that our model (initialized with the proposed unsupervised learning network) compares favorably against existing unsupervised models on both UCF-101 and HMDB-51 datasets. Even though using 8 times lesser number of free parameters, Our model performs existing works by approax 3 % on UCF-101 and 6% on HMDB-51 datasets.

5.3 Joint vs separate pre-training?

We validate spatiotemporal feature learning using color and flow videos. In this section, we validate the advantages of joint flow-color training over learning from individual modalities. We have two separate networks trained using color or flow videos: (1) color model (predicting the missing color pixel values from the input color video) (2) flow model (predicting the missing flow pixel values from the input flow video) and a (3) joint flow-color model (simultaneously predicting the missing color and flow values from the input color and flow videos).

To compare the joint and separate pre-training in color modality, we initialize our action recognition model with the pre-trained weights of the (i) encoder of color model and (ii) the color encoder of the joint model. (See the first block of Tab. 5.2.) Similarly, for the flow modality, we initialize our action recognition model with the pre-trained weights of the (i) encoder of the flow model and (ii) the flow encoder of the joint model. (See the second block of Tab. 5.2.) We then fine-tune all the models on UCF-101 and HMDB-51 training data for 90K and 50K iterations, respectively with a learning rate of 3×10^{-3}. The learning rate is reduced by 10 times after every 40K and 25K iterations on UCF-101 and HMDB-51 dataset. In Tab. 5.2, we compare the performance of our joint vs separate pre-training using color and flow videos on UCF-101 and HMDB-51 datasets.

The results show that the joint pre-training consistently outperforms separate pre-training in either flow and color modality across the two datasets. We also evaluate the combined performance of our spatiotemporal features using both color and flow modality. We initialize a two stream model [55] with the pre-trained weights of two encoders of our joint feature learning model. The class prediction is determined by averaging the predictions of the color and flow stream The combined pre-training achieves 11.6% and 6.5% percent improvement over randomly initialized weights on UCF-101 and HMDB-51 datasets, respectively.

Table 5.2: **Ablation analysis of our joint flow-color model.** Action recognition performance on the split-1 of UCF-101 and HMDB-51 datasets. The first and second blocks compares the performance of joint vs separate pre-training in color and flow modality. The third block shows the combined performance of our spatiotemporal pre-training in color and flow modality using a two stream action recognition model.

Modality	Pre-training model	UCF-101	HMDB-51
Color	Random init	39.8	18.4
	Color	52.7	25.7
	Joint	53.8	26.7
Flow	Random init	65.5	39
	Flow	76.5	45.2
	Joint	77.1	46.0
Two stream	Random init	66.2	40.4
	Joint	77.8	46.5

5.4 Unsupervised pre-training or random initialization?

The initialization of the network plays an important role to generalize with very limited training data. We evaluate the recognition accuracy on the HMDB-51 dataset by initializing our action recognition model using the weights of (i) pretrained encoder in the color model (trained to predict the missing color pixel values from the given color video) and (ii) randomly initialization. We show in Figure 5.2 the accuracy on the HMDB-51 dataset using both initialization approach. We train the network fractional training data (20%, 40%, 60%, 80% and 100%). The quantitative results show that the weights learned from the unsupervised model can quickly adapt to the new dataset. Using the proposed unsupervised CNN pre-training, we can outperform the randomly initialized C3D trained with the complete set of data with less than 20% of the training data. The results demonstrate the importance of CNN pre-training for applications where the training examples are sparse. Figure 5.2 shows the testing accuracy under different numbers of epochs on the HMDB-51 dataset. Compared with the model using random initialization, the pretrained weights of the color model converges to a better solution at a faster rate.

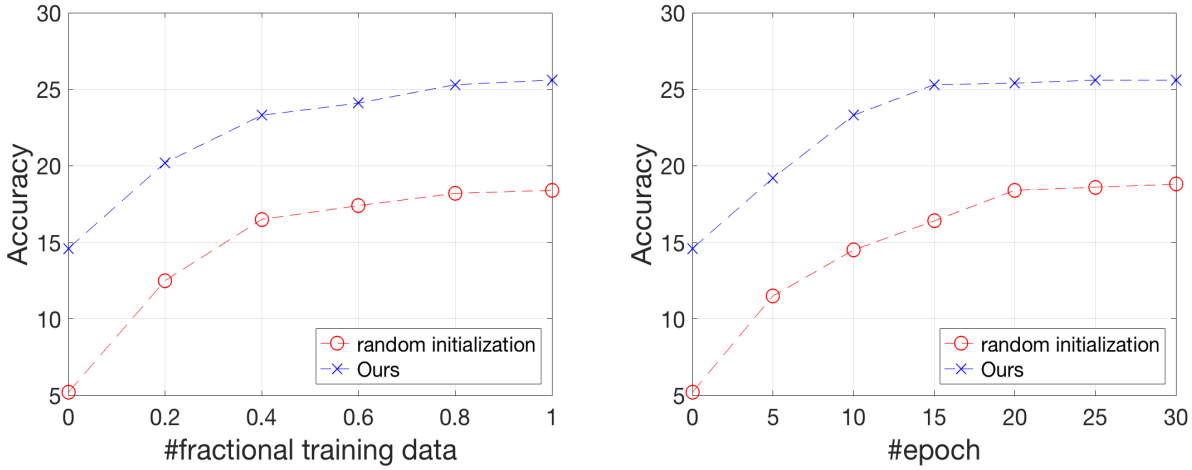


Figure 5.2: **The effect of the number of training samples and epochs.** Left: Performance comparison of learning using different amounts of training examples (20%, 40%, 60%, 80% and 100%) on HMDB-51 dataset using either random initialization or the pre-trained color model using our unsupervised learning approach. Right: Performance comparison of the our pre-trained color model and random initialization under the number of epochs on the HMDB-51 dataset.

5.5 Spatial, temporal, or spatiotemporal?

We evaluate the learned features with the three different sets of training data: (i) using spatial context only (image inpainting), (ii) using temporal context only (video frame prediction), and (iii) using spatiotemporal context (video completion). We initialize the action recognition model using the weights of encoder of the pre-trained with spatial, temporal and spatiotemporal training data. We finetune these networks using the training data from UCF-101 and HMDB-51 datasets. Tab. 5.3 shows the performance comparisons of these models pretrained with spatial, temporal and spatiotemporal training data. We also use the weighted purity to measure the quality of the features from the unsupervised pre-training. Figure 5.1 shows the purity of clusters versus the number of clusters on the HMDB-51 dataset. As training with spatial context prediction only learns the appearance of *still* images, the learned representation may have difficulty in recognizing the scene dynamics. We demonstrate that incorporating both spatial and temporal contexts does help improve the generalization ability.

Table 5.3: **Performance comparisons of using spatial, temporal, spatiotemporal context**. Performance comparison of the unsupervised pre-training using spatial, temporal, spatiotemporal training data on the split-1 of HMDB-51 and UCF-101 datasets

Unsupervised models	UCF-101	HMDB-51
Random Init	39.8	18.4
Spatial	48.2	23.1
Temporal	49.4	23.4
spatiotemporal	52.7	25.7

5.6 Action similarity labeling

The Action Similarity Labeling (ASLAN) [29] task is to decide if two videos present the same action or not, following training with same and not-same labeled video pairs. Actions included in the test sets are not available at training. This means that there is no opportunity during training to build action models for actions present in the testing.

Dataset. The ASLAN set contains 3697 action samples from 1571 unique YouTube videos divided into 432 non-trivial action categories. An ”action sample” is defined as a sub-sequence of a shot presenting a detected action, that is, a consecutive set of frames taken by the same camera presenting one action. The action samples have been manually labeled with the name of the activities carried out in each of them. 316 of the categories contain more than one sample. The dataset contains ten splits, each split containing about 300 positive pairs and 300 negative pairs.

Comparison with existing approaches. The standard evaluation protocol uses a ten-fold cross-validation. There are 116 actions with only *one* training video. Also, the test set contains actions that are *not* present in the training set. We initialize the action similarity model using the pretrained weights of color encoder from the *joint model*. We finetune the model for 50K iterations with a learning of 3×10^{-3} using the training data. The learning rate is reduced by 10 times after every 25K iterations. We randomly sample 16 consecutive frames from each of the action video. We

Table 5.4: **Quantitatvie evaluation on action similarity.** Performance comparison of our unsupervised pre-training with other state-of-the-art approaches for action similarity on the ASLAN dataset. In this table, Acc and AUC stand for Accuracy and Area Under the Curve of the ROC, respectively. STIP, MIP and MBH are abbreviations for Space-Time Interest Points, Motion Interchange Patterns and Motion Boundary Histogram.

Method	Features	Model	Acc	AUC
Kliper *et al*. [29]	STIP	linear	60.9	65.3
Kliper *et al*. [28]	MIP	metric	65.5	71.9
Hanani *et al*. [16]	MIP+STIP+MBH	metric	66.1	73.2
Peng *et al*. [50]	iDT+FV	metric	68.7	75.4
ImageNet [30]	linear	linear	67.5	73.8
Supervised C3D [59]	C3D	linear	78.3	86.5
Randomly Init	C3D	linear	61.7	69.4
Ours	C3D	linear	67.2	73.4

extract features at conv5, fc6, fc7 layers of the finetuned model. The features are then normalized to have zero mean and unit variance. We compute similarity between two video features using 12 distances provided in Kliper *et al*. [29]. We compute the similarity measure between each pair of videos using 12 distances provided in Kliper *et al*. [29] on the three extracted features. In total, we have a 36 (3×12) dimensional feature vector describing similarity measure between each pair of videos. We use a linear SVM over the extracted features for classification.

Tab. 5.4 shows the comparison of our unsupervised approach with other existing supervised approaches. Our unsupervised pre-training achieve competitive performance with other approaches based on space-time interest points [29], complex features encodings like Fisher vectors (FV), and Vector of Locally Aggregated Descriptors (VLAD), and the supervised C3D features [59].

5.7 Egocentric object recognition

The Egocentric Vision Dataset [53] contains 42 everyday objects that vary in size, shape, color and textures. The video sequences are shot for each object under different illuminations and back-

Table 5.5: **Quantitative evaluation on egocentric object detection.** Performance comparison (in terms of detection accuracy %) of our unsupervised pre-training with other state-of-the-art approaches for object detection on the egocentric objects dataset.

Methods	Accuracy
Ren *et al*. [53]	12
Imagenet [30]	25.7
Supervised C3D [59]	22.3
Randomly initialized	11.7
Ours	18.4

grounds. We evaluate the performance of our unsupervised finetuned C3D features and compare with the existing supervised approaches.

We apply two types of features learned in supervised settings: 1) the C3D [59] features trained on large-scale Sports-1M and UCF-101 datasets, 2) the AlexNet [30] features trained on ImageNet. We also report results from Ren *et al*. [53] that uses RBF-kernel on SIFT-RANSAC feature matching. In our approach, we finetune our unsupervised C3D features on the egocentric objects dataset for 10 epochs with a learning rate of $1\exp^{-6}$. We reduce the learning rate 5 times for every 2 epochs. We train the the network with a batch size of 16. We use ADAM, a first-order gradient-based optimization of the stochastic gradient descent, for training the network. We than extract features from our finetuned model and apply a linear SVM for object recognition. While our model is not trained on external large-scale dataset with semantic label, we are able to decent performance compared to the fully supervised models. We achieve 18.4% accuracy which is 3.9% lower than the supervised C3D model [59]. Our unsupervised initialization performs 6.7% better than the randomly initialized model. This proves the superiority of our unsupervised initialization.

Table 5.5 shows the performance comparison of our unsupervised finetuned model with the existing completely supervised models.

5.8 Qualitative results

The by-product of the unsupervised feature learning process produces a video completion model. Figure 5.3 shows sample qualitative results on video completion. While the reconstructed regions are not yet photo-realistic and seamless, our model produces reasonable reconstruction. Figure 5.4 shows the qualitative comparison between the results by our joint model and the color model. Joint pre-training produces improved video completion results compared to pre-training using only color videos.

Failure modes. Figure 5.5 shows sample complete results with inconsistent or blurry reconstruction. We attribute the failure here to the inability to capture high-level semantics from limited observation (*e.g.*, occluded face and body).

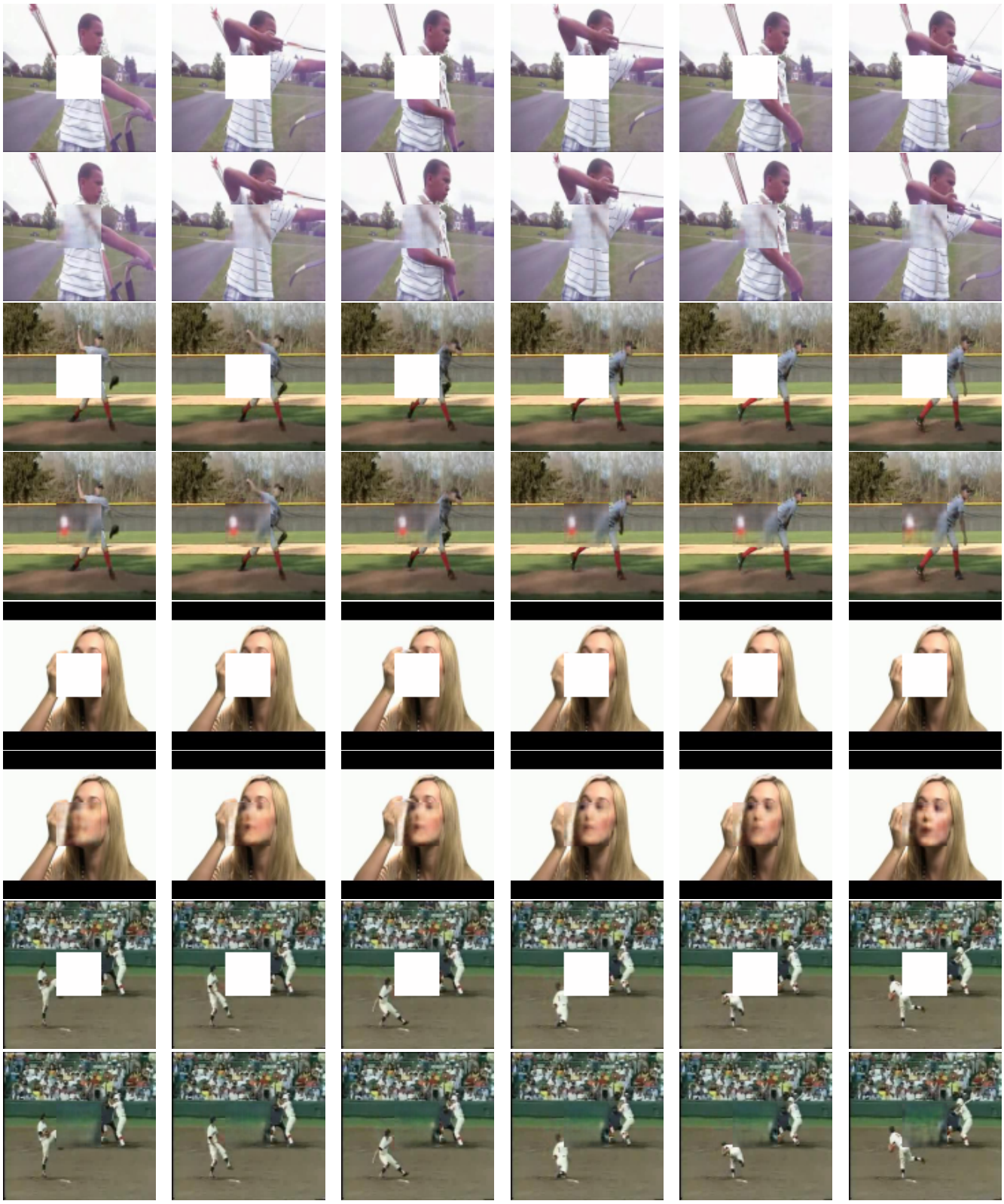


Figure 5.3: **Qualitative results.** Sample results of video completion on UCF-101 testing videos. While the completion quality by our network is not yet photo-realistic, the completion results are plausible and capture the spatiotemporal dynamics of the scene.

Figure 5.4: **Joint vs. separate.** Qualitative results of our joint video completion model (row 2) with color only model (row 3). Joint model provides improved reconstruction quality over color only model.

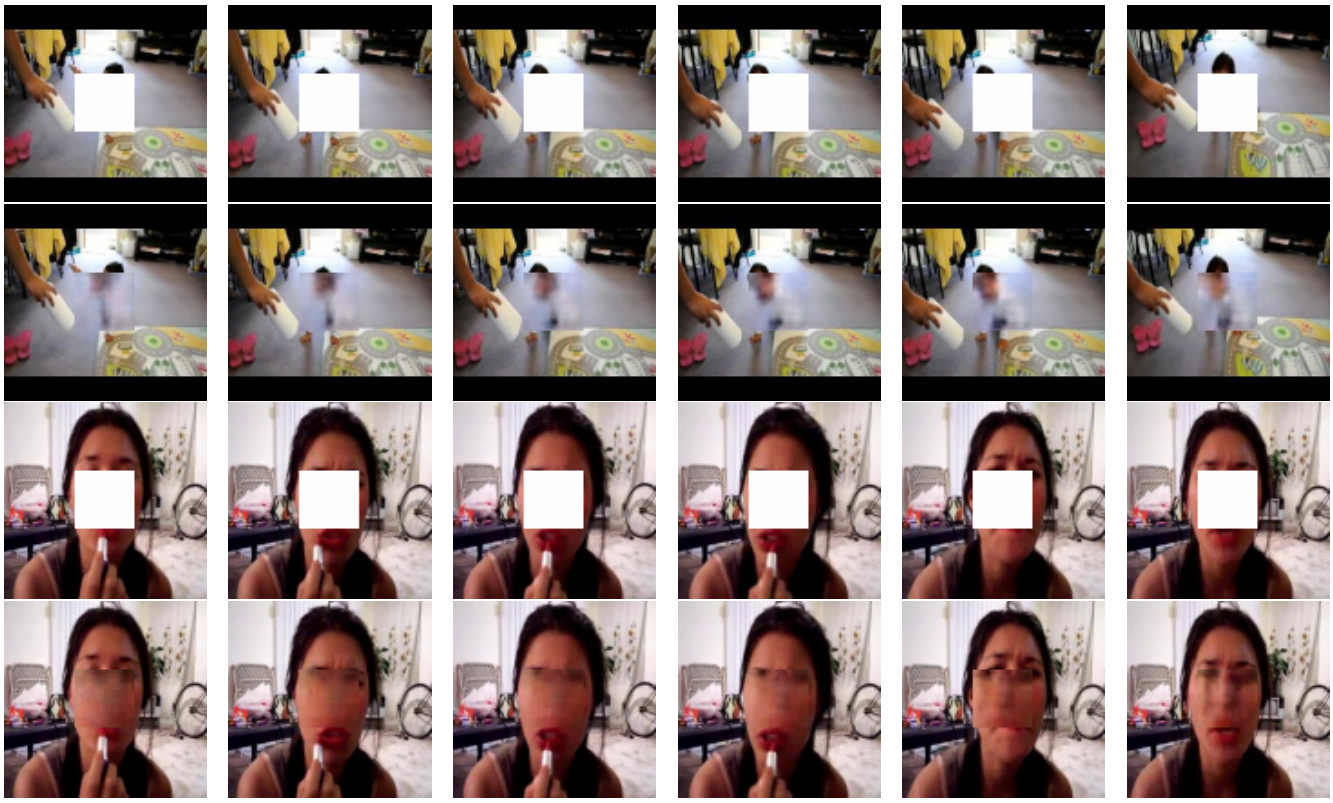


Figure 5.5: **Failure cases.** Our model sometimes produces temporally inconsistent outputs or blurry reconstruction when the surrounding background do not provide sufficient information.

Chapter 6

Discussions

We presented an unsupervised learning framework to learn meaningful spatiotemporal features using video completion. The proposed spatiotemporal task highlights the importance of exploiting both spatial and temporal context in videos. We show that the unsupervised learning framework is applicable for simultaneously learning features for flow and color videos. We demonstrate that the learned features can be used a pre-trained CNN model for action recognition. The quantitative results show that our method compares favorably against competitive unsupervised feature learning algorithms.

There are several interesting future directions. For example, augmenting the loss function with adversarial loss may further improve the quality of the features. The ability to learn meaningful 3D features from volumetric data may be helpful in tackling problems in other domains where the training samples are extremely sparse, *e.g.*, medical imaging.

Bibliography

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015.

[2] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1), 2009.

[3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. 2017.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[7] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ECCV*, 2016.

[9] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[12] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *ECCV*, 2012.

[13] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *ECCV*, 2012.

[14] Miguel Granados, James Tompkin, K Kim, Oliver Grau, Jan Kautz, and Christian Theobalt. How not to be seenobject removal from videos of crowded scenes. In *Computer Graphics Forum*, 2012.

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[16] Yair Hanani, Noga Levy, and Lior Wolf. Evaluating new variants of motion interchange patterns. In *CVPR*, 2013.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[18] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[19] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM TOG (Proc. SIGGRAPH)*, 35(6):196, 2016.

[20] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM TOG (Proc. SIGGRAPH)*, 35(6):196, 2016.

[21] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015.

[22] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. 2016.

[23] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21:433–449, 1999.

[24] Bela Julesz et al. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981.

[25] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[26] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[28] Orit Kliper-Gross, Yaron Gurovich, Tal Hassner, and Lior Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[29] Orit Kliper-Gross, Tal Hassner, and Lior Wolf. The action similarity labeling challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):615–621, 2012.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[31] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[32] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[33] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. 2016.

[34] Dong Li, Wei-Chih Hung, Jia-Bin Huang, Shengjin Wang, Narendra Ahuja, and Ming-Hsuan Yang. Unsupervised visual representation learning by graph-based consistent constraints. In *ECCV*, 2016.

[35] Gucan Long, Laurent Kneip, Jose M Alvarez, and Hongdong Li. Learning image matching by simply watching video. In *ECCV*, 2016.

[36] William Lotter, Gabriel Kreiman, and David Cox. Unsupervised learning of visual structure using predictive generative networks. In *ICLR*, 2016.

[37] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[38] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. *CVPR*, 2017.

[39] Elman Mansimov, Nitish Srivastava, and Ruslan Salakhutdinov. Initialization strategies of spatio-temporal convolutional neural networks. *arXiv preprint arXiv:1503.07274*, 2015.

[40] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2015.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[42] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.

[43] Volodymyr Mnih, Joshua M Susskind, Geoffrey E Hinton, et al. Modeling natural images using gated mrfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2206–2222, 2013.

[44] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, 2009.

[45] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[46] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. 2016.

[47] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[48] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. 2016.

[49] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[50] Xiaojiang Peng, Yu Qiao, Qiang Peng, and Qionghua Wang. Large margin dimensionality reduction for action similarity labeling. *IEEE Signal Processing Letters*, 21(8):1022–1025, 2014.

[51] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. 2016.

[52] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICCV*, 2015.

[53] Xiaofeng Ren and Matthai Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *CVPR*, 2009.

[54] Salah Rifai, Yoshua Bengio, Aaron Courville, Pascal Vincent, and Mehdi Mirza. Disentangling factors of variation for facial expression recognition. In *ECCV*, 2012.

[55] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[56] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. 2012.

[57] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[58] David Stavens and Sebastian Thrun. Unsupervised learning of invariant features using video. In *CVPR*, 2010.

[59] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[60] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[61] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[62] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[63] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *TPAMI*, 29(3):463–476, 2007.

[64] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

[65] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *ICME*, 2015.

[66] Yuancheng Ye and Yingli Tian. Embedding sequential information into spatiotemporal features for action recognition. In *CVPR*, 2016.

[67] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. 2017.

[68] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *NIPS*, 2012.

Appendix A

Additional qualitative video completion results

Video completion aims at filling spatiotemporal hole with plausible contents. Here, we show additional qualitative video completion results from our unsupervised spatiotemporal model. Note that the goal of our work is *not* trying to outperform non-parametric patch-based optimization algorithms [20, 45] or segmentation-based methods [13, 14] for video completion. Instead, we aim to exploit the task of predicting spatiotemporal to help train the model for extracting meaningful spatiotemporal features.

We show our sample qualitative video completion results in Figures A.1, A.2, A.3 & A.4. Figure A.5 & A.6 highlights several failure completion results which are inconsistent or blurry. We compare our video completion results with Newson *et al*. [45] in Figures A.7, A.8, A.9, A.10 & A.11. Our model often produces blurry, but semantically meaningful completion results compared to Newson *et al*. [45]. We refer the readers to the accompanying supplementary video for the video results.

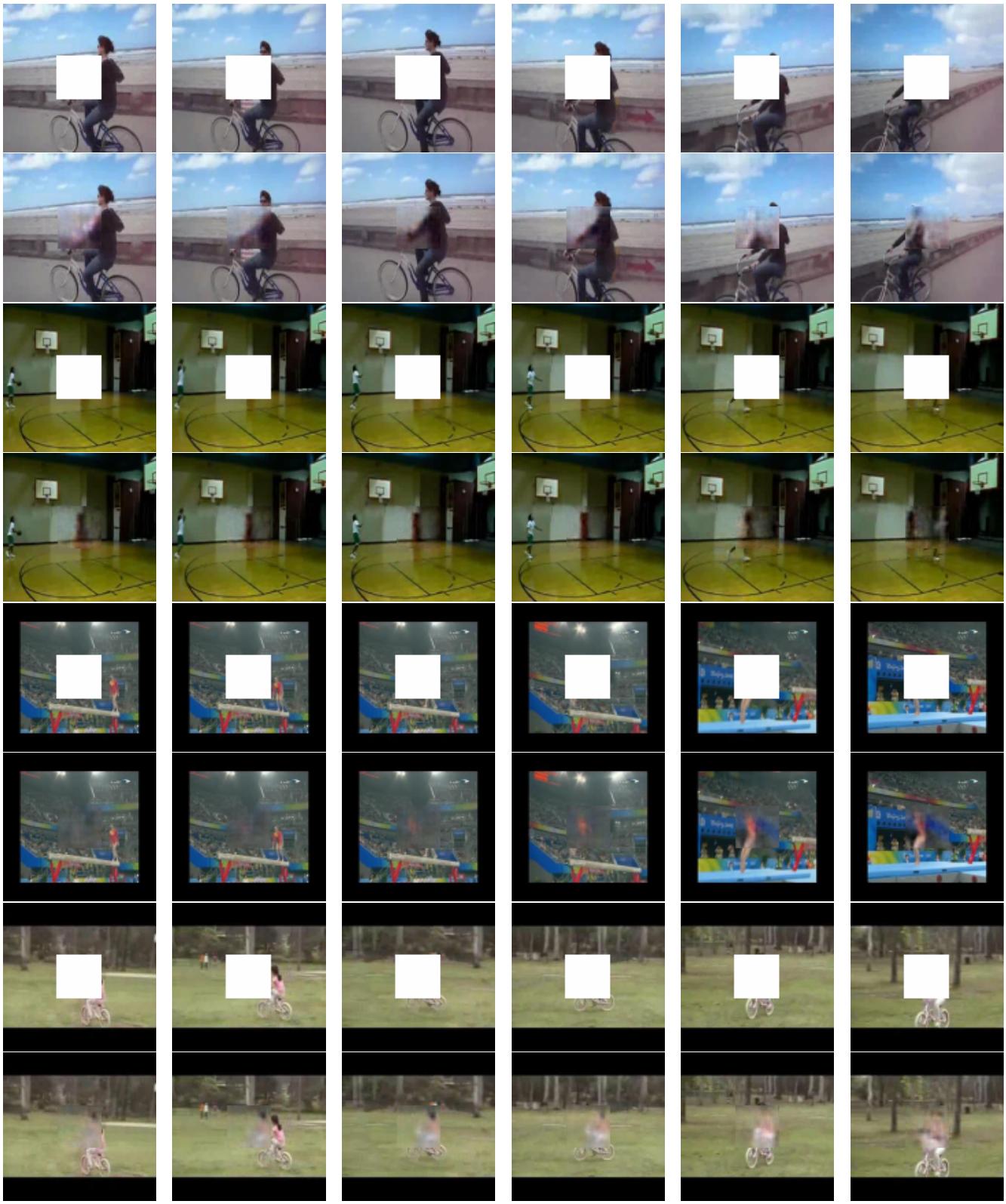


Figure A.1: **Qualitative results.** Sample results of our video completion model.

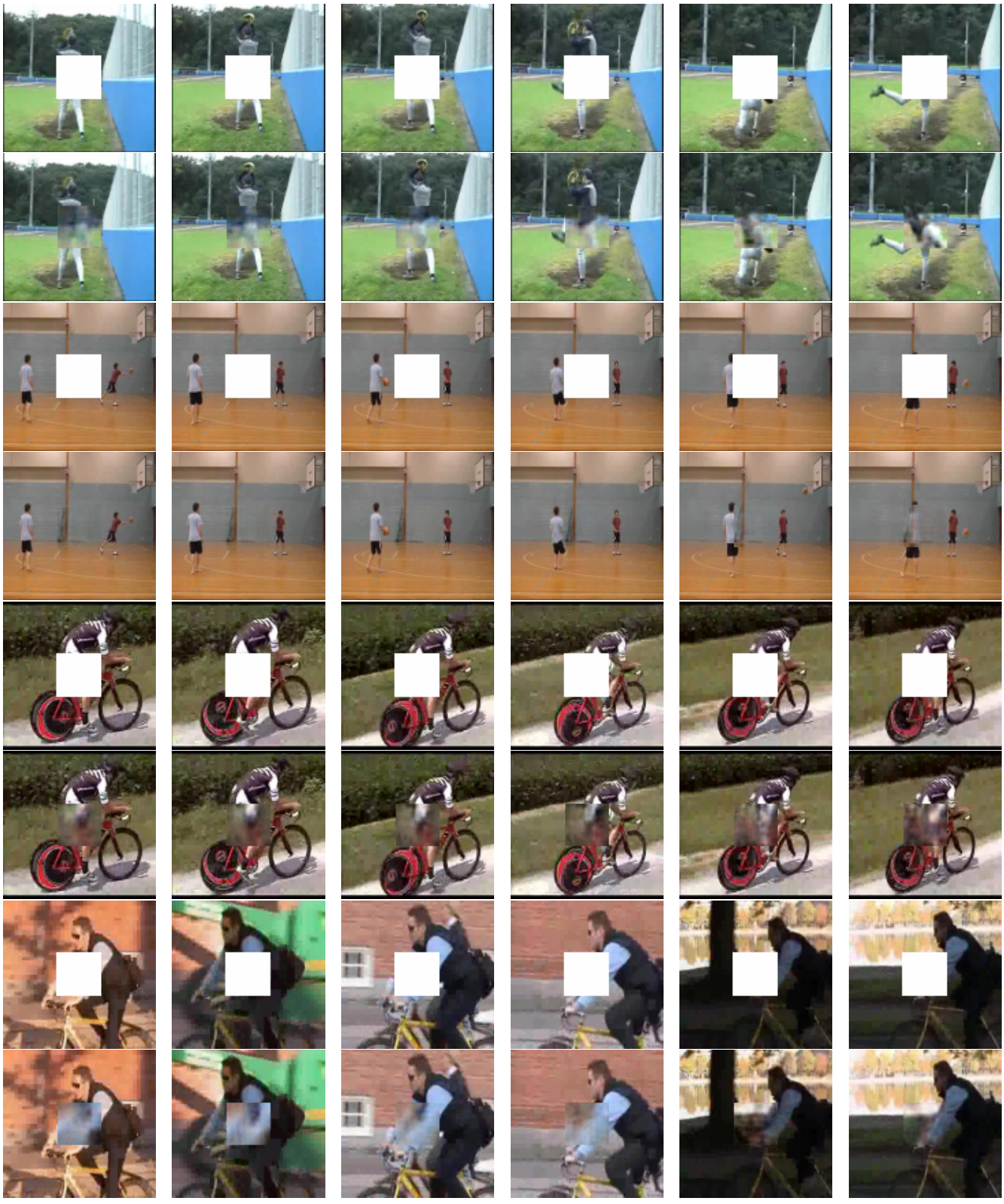


Figure A.2: **Qualitative results.** Additional sample results of our video completion model.

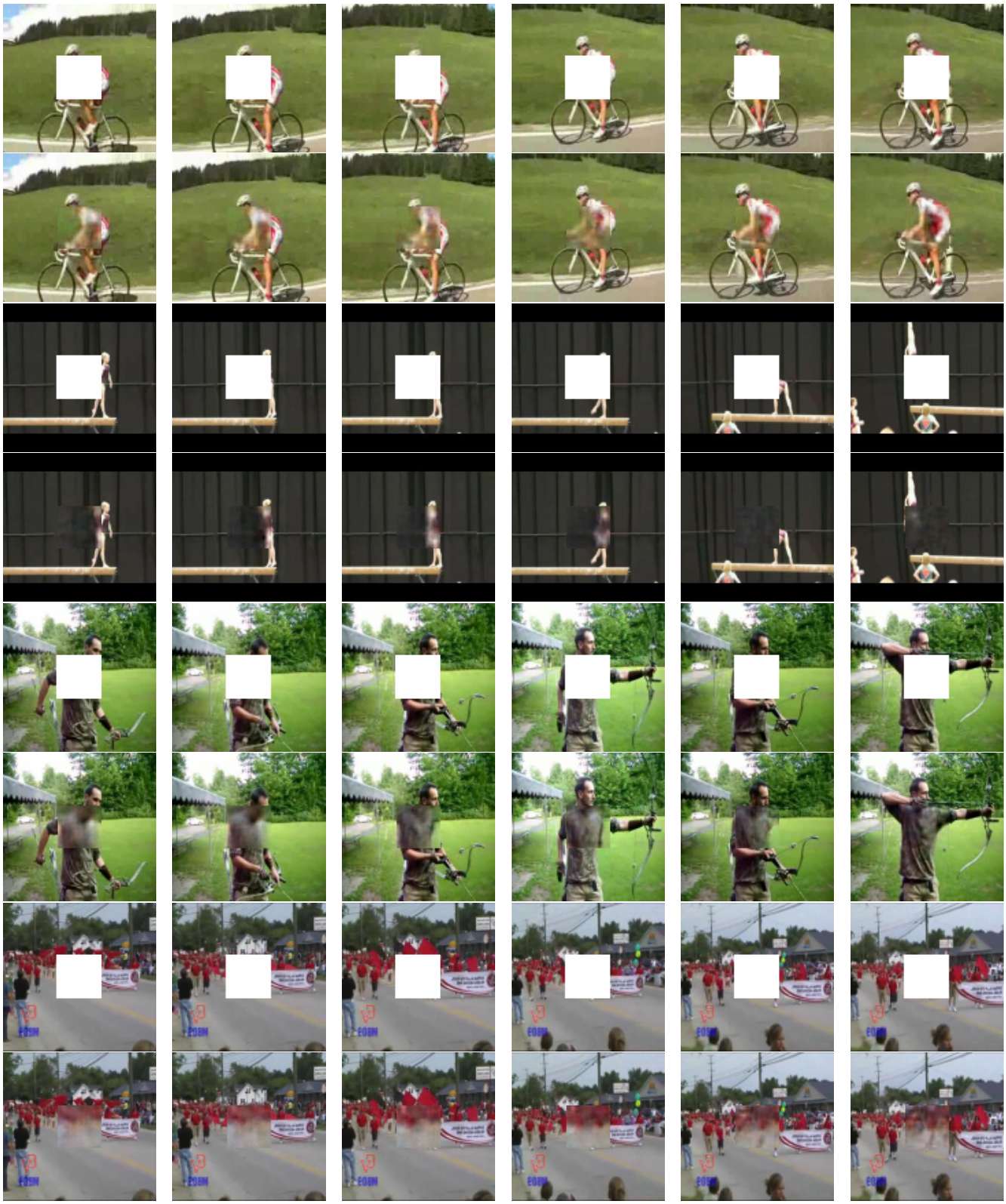


Figure A.3: **Qualitative results.** S Additional sample results of our video completion model.

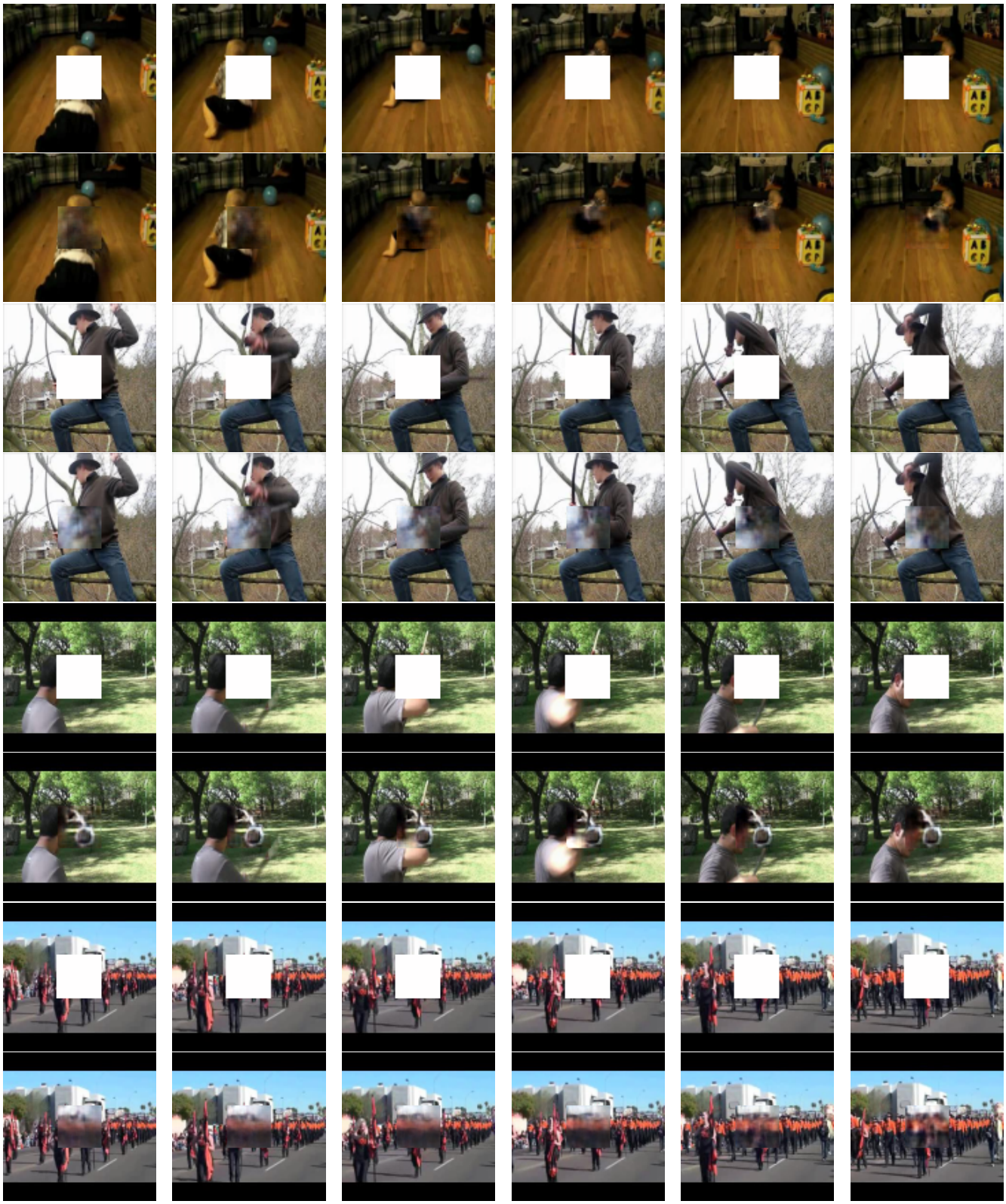


Figure A.4: **Qualitative results.** Additional sample results of our video completion model.

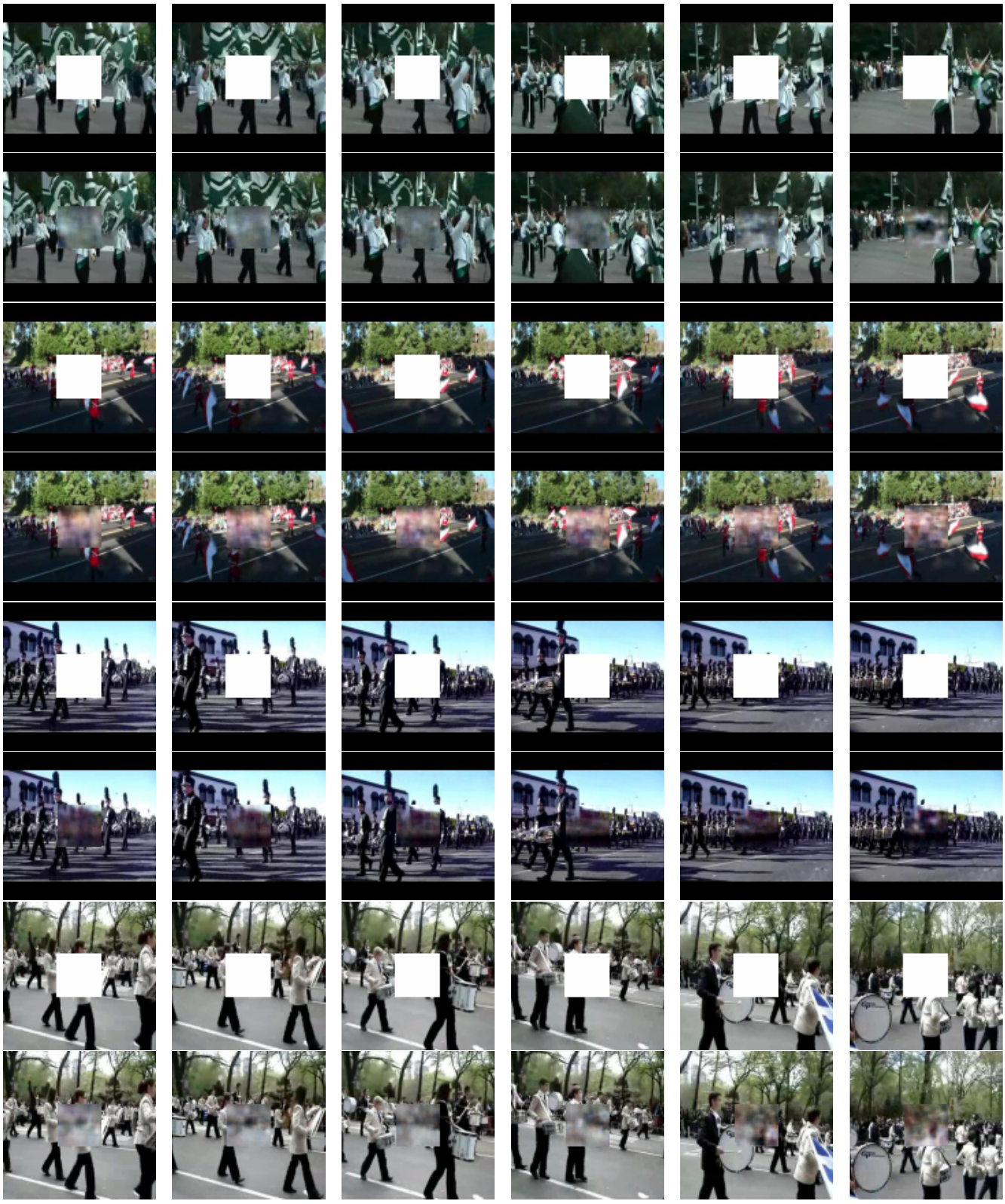


Figure A.5: **Blurry & inconsistent.** Some blurry and inconsistent qualitative results produced by our video completion model.

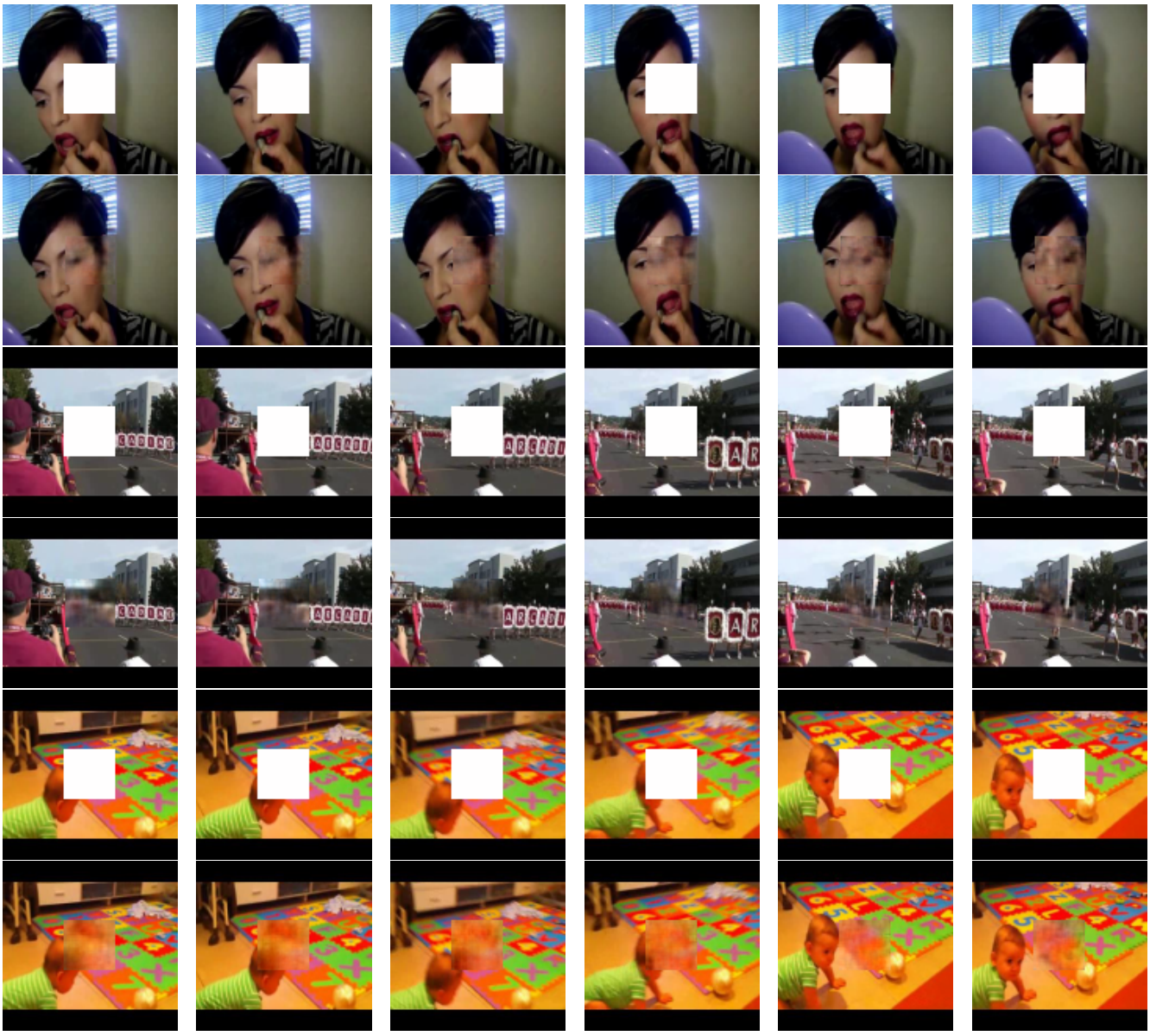


Figure A.6: **Blurry & inconsistent.** Additional blurry and inconsistent qualitative results produced by our video completion model.

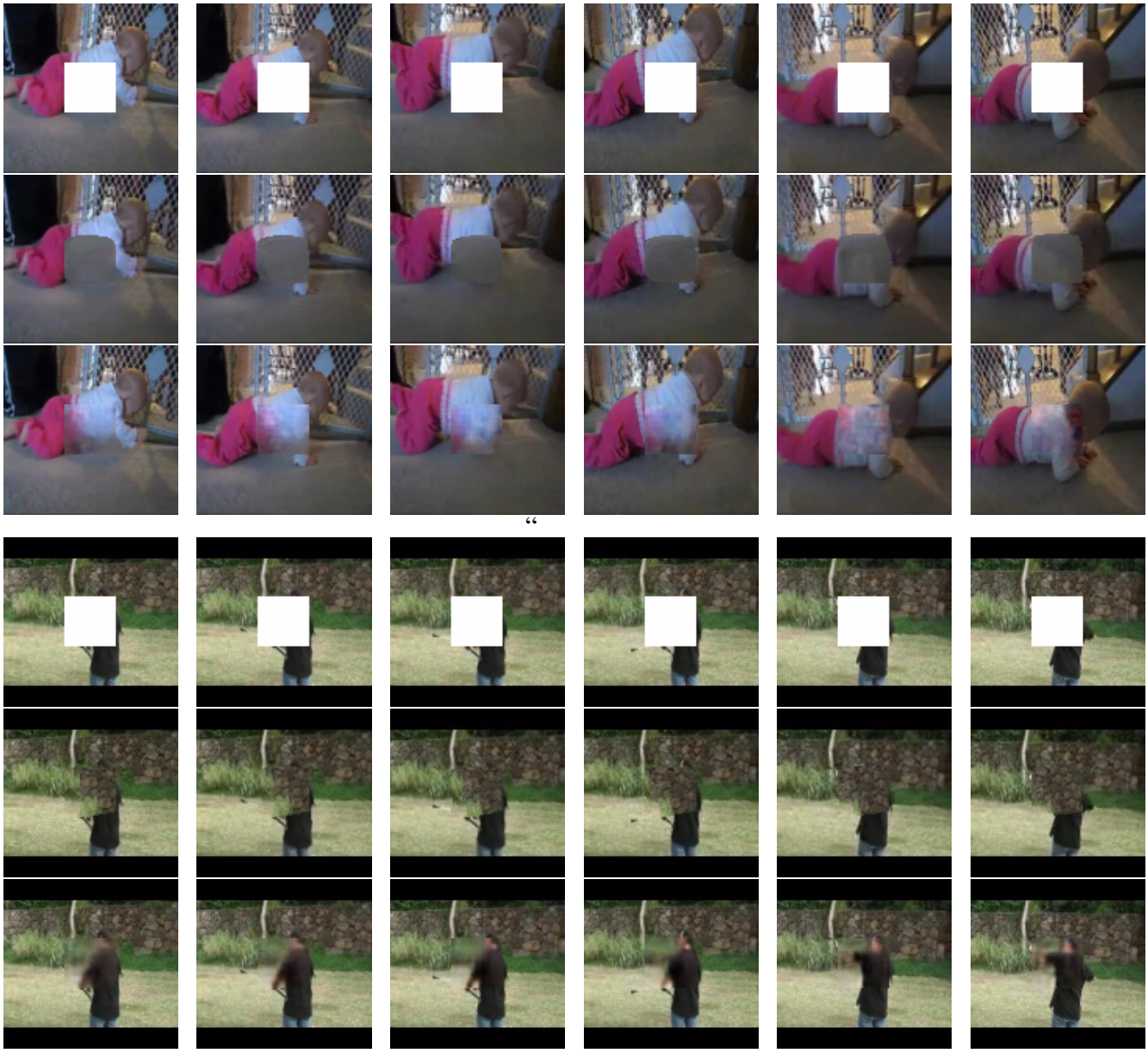


Figure A.7: **Ours vs Newson *et al*.** Qualitative results comparing our video completion results with Newton*et al*. (a non-parametric patch based optimization model. First row of every video is given as input to both the models. The second row is the output of video completion from Newson *et al*. The third rows is the output from our video completion model.

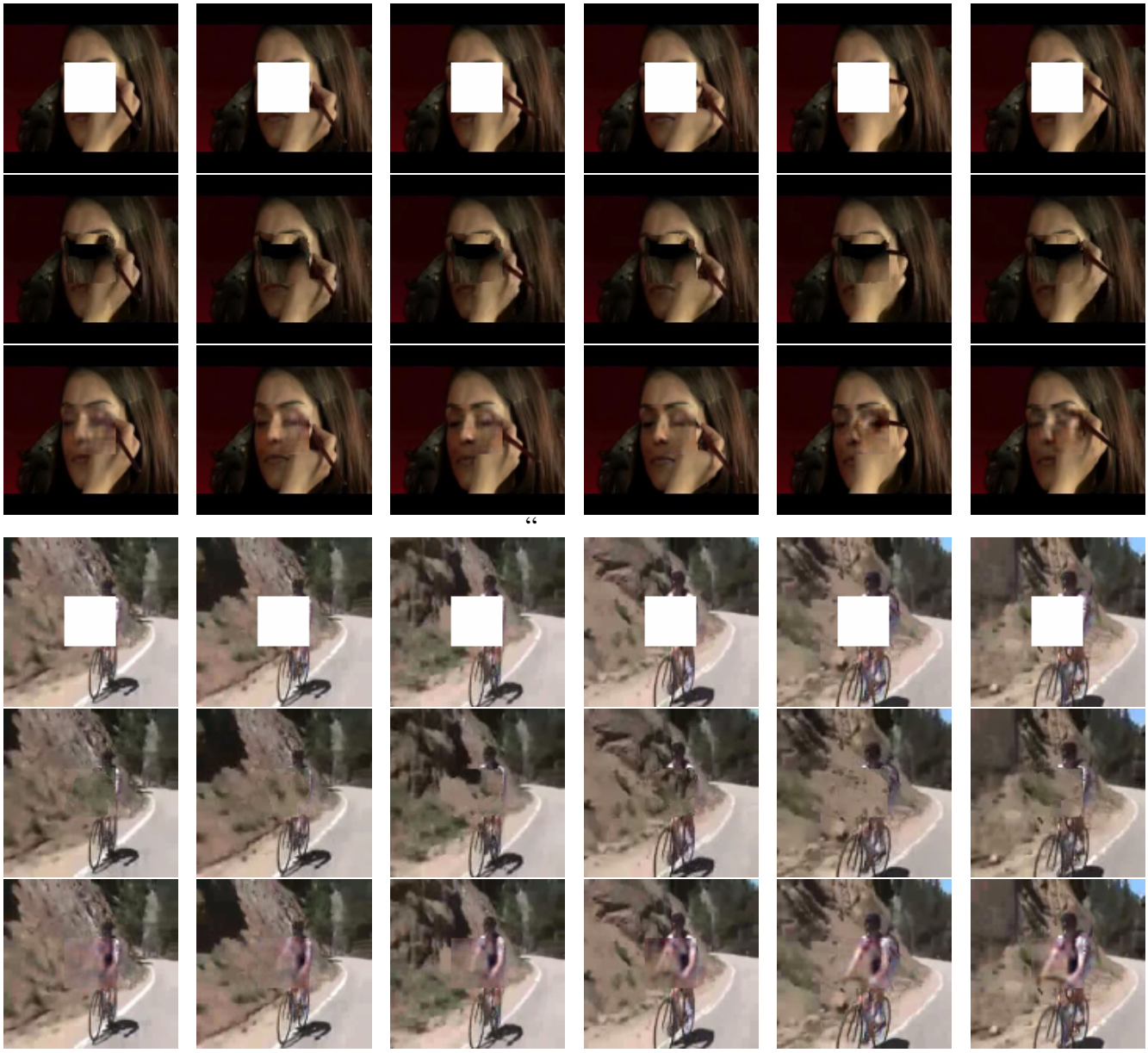


Figure A.8: **Ours vs Newson *et al.*** Additional qualitative results comparing our video completion results with Newton*et al*. (a non-parametric patch based optimization model. First row of every video is given as input to both the models. The second row is the output of video completion from Newson *et al*. The third rows is the output from our video completion model.

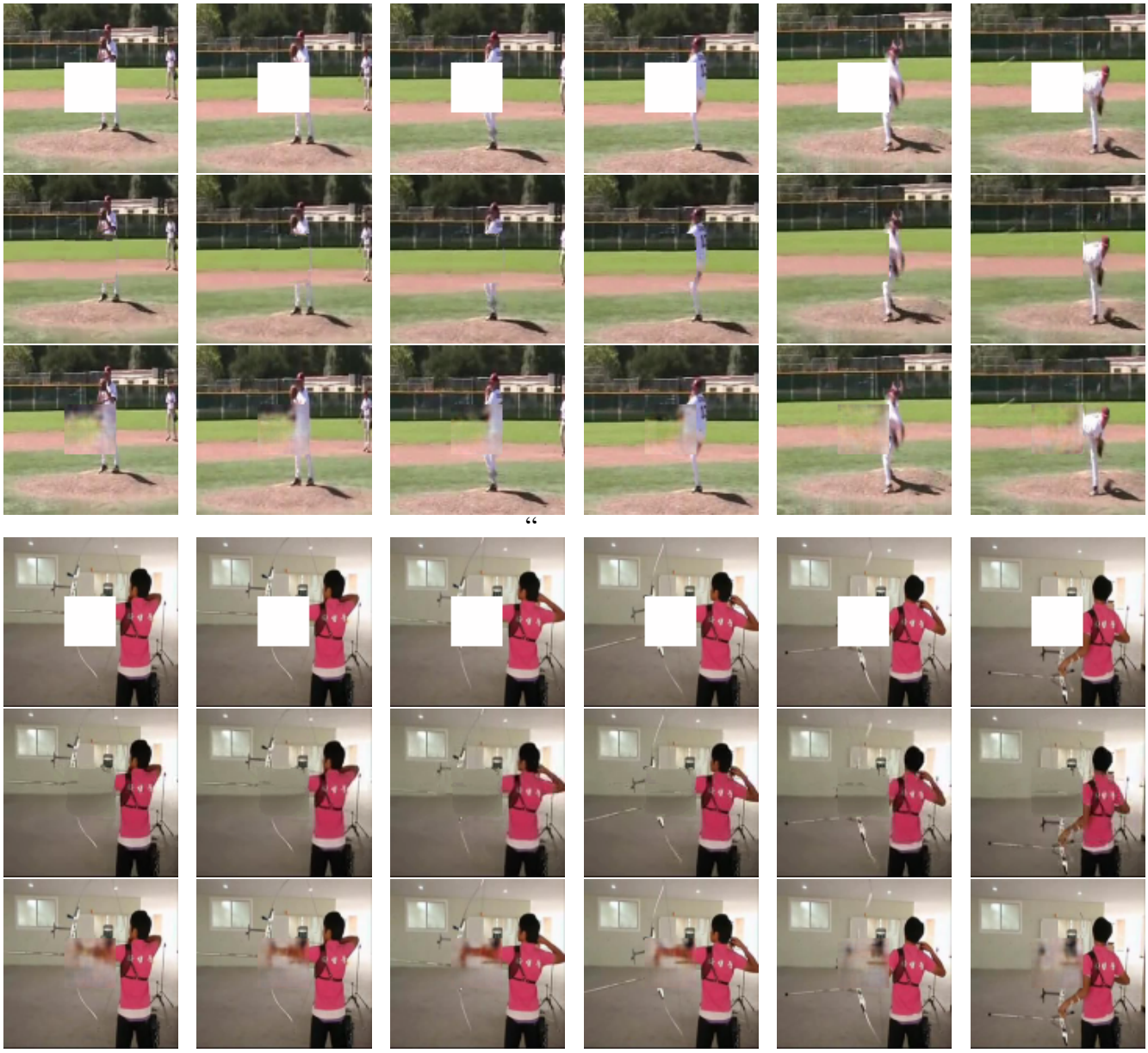


Figure A.9: **Ours vs Newson *et al*.** Additional qualitative results comparing our video completion results with Newton*et al*. (a non-parametric patch based optimization model. First row of every video is given as input to both the models. The second row is the output of video completion from Newson *et al*. The third rows is the output from our video completion model.

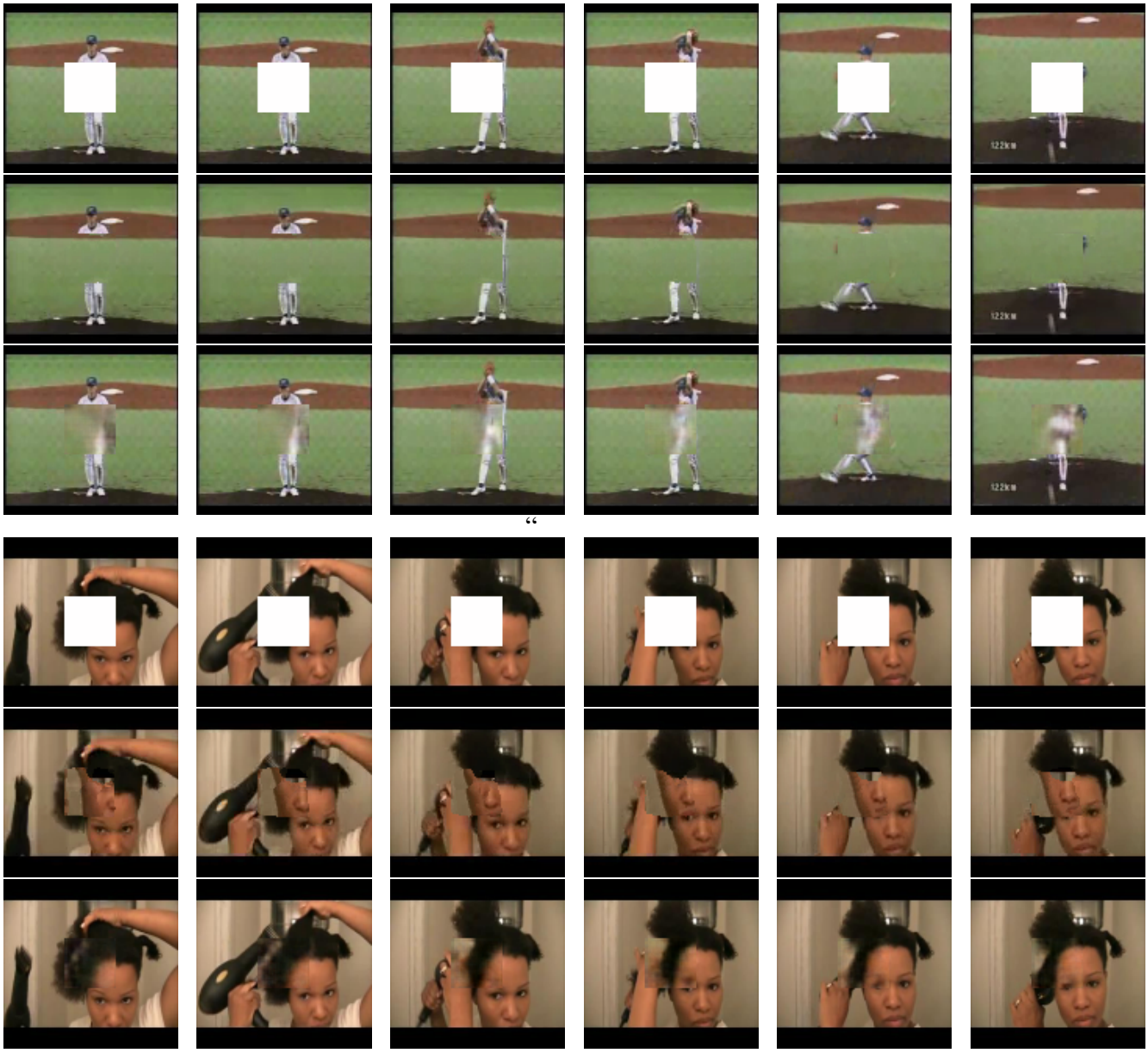


Figure A.10: **Ours vs Newson *et al*.** Additional qualitative results comparing our video completion results with Newton*et al*. (a non-parametric patch based optimization model. First row of every video is given as input to both the models. The second row is the output of video completion from Newson *et al*. The third rows is the output from our video completion model.

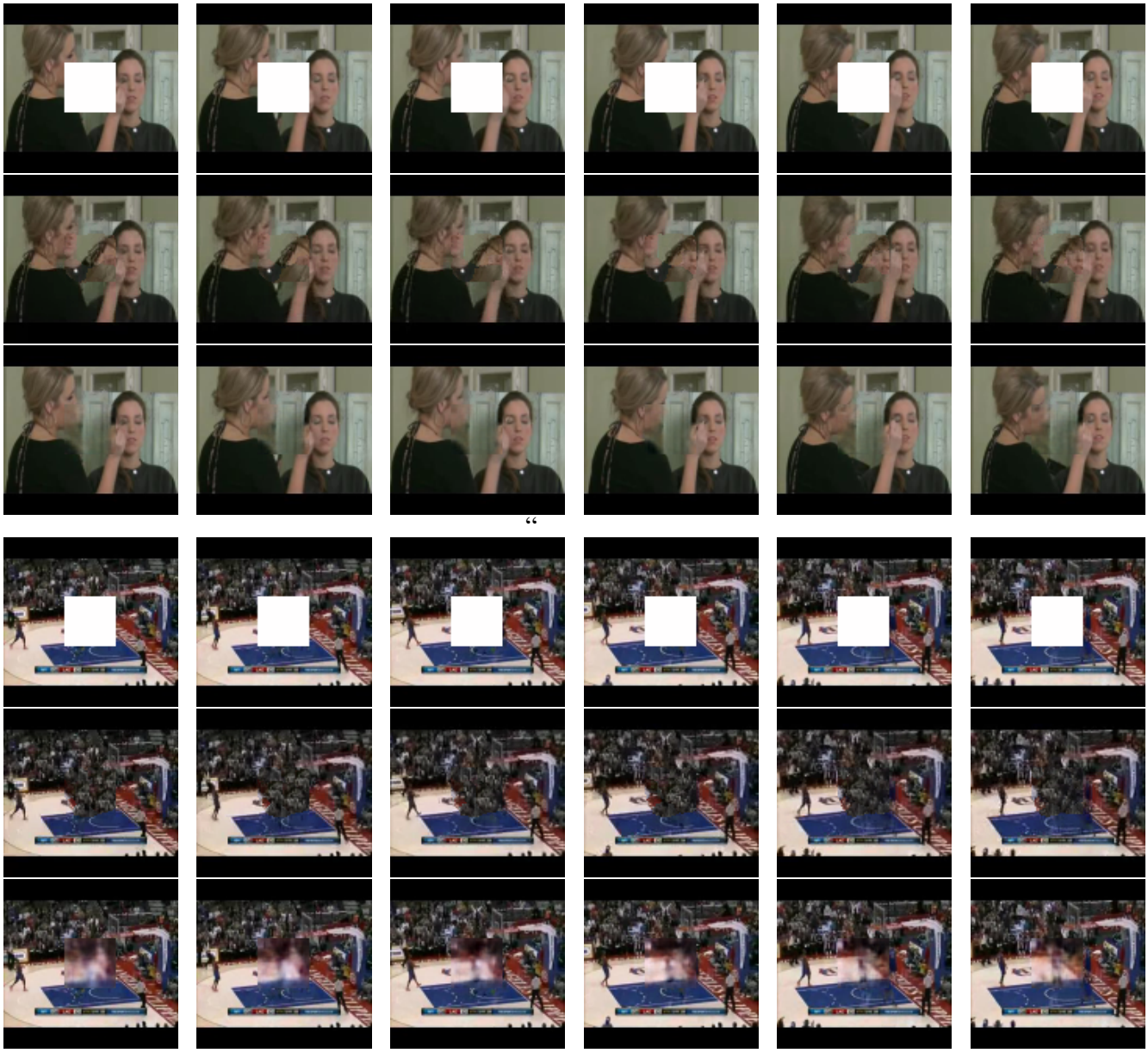


Figure A.11: **Ours vs Newson *et al*.** Additional qualitative results comparing our video completion results with Newton*et al*. (a non-parametric patch based optimization model. First row of every video is given as input to both the models. The second row is the output of video completion from Newson *et al*. The third rows is the output from our video completion model.