Slow and steady feature analysis: higher order temporal coherence in video

Dinesh Jayaraman
UT Austin
dineshj@cs.utexas.edu

Kristen Grauman
UT Austin
grauman@cs.utexas.edu

Abstract

How can unlabeled video augment visual learning? Existing methods perform "slow" feature analysis, encouraging the representations of temporally close frames to exhibit only small differences. While this standard approach captures the fact that high-level visual signals change slowly over time, it fails to capture how *the visual content changes. We propose to generalize slow feature analysis to "steady" feature analysis. The key idea is to impose a prior that higher order derivatives in the learned feature space must be small. To this end, we train a convolutional neural network with a regularizer on tuples of sequential frames from unlabeled video. It encourages feature changes over time to be smooth, i.e., similar to the most recent changes. Using five diverse datasets, including unlabeled YouTube and KITTI videos, we demonstrate our method's impact on object, scene, and action recognition tasks. We further show that our features learned from unlabeled video can even surpass a standard heavily supervised pretraining approach.*

1. Introduction

Visual feature learning with deep neural networks has yielded dramatic gains for image recognition tasks in recent years [22, 37]. While the main techniques involved in these methods have been known for some time, a key factor in their recent success is the availability of large human-labeled image datasets like ImageNet [6]. Deep convolutional neural networks (CNNs) designed for image recognition typically have millions of parameters, necessitating notoriously large training databases to avoid overfitting.

Intuitively, however, visual learning should not be restricted to sets of category-labeled exemplars. Taking human learning as an obvious example, children build up visual representations through constant observation and action in the world. This hints that machine-learned representations would also be well served to exploit long-term *video* observations, even in the absence of deliberate labels. Indeed, researchers in cognitive science find that *temporal coherence* plays an important role in visual learning. For example, altering the natural temporal contiguity of visual stimuli hinders translation invariance in the inferior temporal cortex [26], and functions learned to preserve temporal coherence share behaviors observed in complex cells of the primary visual cortex [4].

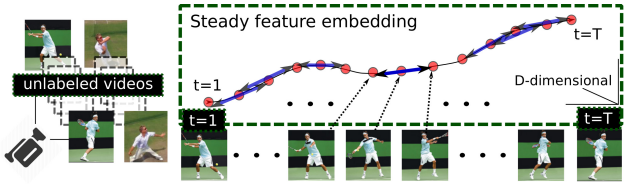

Figure 1: From unlabeled videos, we learn "steady features" that exhibit consistent feature transitions among sequential frames.

Our goal is to exploit unlabeled video, as might be obtained freely from the web, to improve visual feature learning. In particular, we are interested in improving learned image representations for visual recognition tasks.

Prior work leveraging video for feature learning focuses on the concept of *slow feature analysis* (SFA). First formally proposed in [42], SFA exploits temporal coherence in video as "free" supervision to learn image representations invariant to small transformations. In particular, SFA encourages the following property: in a learned feature space, temporally nearby frames should lie close to each other, *i.e.* for a learned representation $\mathbf{z}$ and adjacent video frames $\boldsymbol{a}$ and $\boldsymbol{b}$, one would like $\mathbf{z}(\boldsymbol{a}) \approx \mathbf{z}(\boldsymbol{b})$. The rationale behind SFA rests on a simple observation: high-level semantic visual concepts associated with video frames typically change only gradually as a function of the pixels that compose the frames. Thus, representations useful for recognizing high-level concepts are also likely to possess this property of "slowness". Another way to think about this is that scene changes between temporally nearby frames are usually small and represent label-preserving transformations. A slow representation will tolerate minor geometric or lighting changes, which is essential for high-level visual recognition tasks. The value of exploiting temporal coherence for recognition has been repeatedly verified in ongoing research, including via modern deep convolutional neural

network implementations [30, 3, 14, 46, 12, 41].

However, existing approaches require only that high-level visual signals change slowly over time. Crucially, they fail to capture *how* the visual content changes over time. In contrast, our idea is to incorporate the *steady visual dynamics* of the world, learned from video. For instance, if trained on videos of walking people, slow feature-based approaches would only require that images of people in nearby poses be mapped *close* to one another. In contrast, we aim to learn a feature space in which frames from a novel video of a walking person would follow a smooth, predictable trajectory. A learned *steady* representation capturing such dynamics would be influenced not only by object motions, but also other types of visual transformations. For instance, it would capture how colors of objects in the sunlight change over the course of a day, or how the views of a static scene change as a camera moves around it.

To this end, we propose *steady feature analysis*—a generalization of slow feature learning. The key idea is to impose higher order temporal constraints on the learned visual representation. Beyond encouraging temporal coherence *i.e.*, *small feature differences* between nearby frame pairs, we would like to encourage *consistent feature transitions* across sequential frames. In particular, to preserve second order slowness, we look at triplets of temporally close frames $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and encourage the learned representation to have $\mathbf{z}(\boldsymbol{b}) - \mathbf{z}(\boldsymbol{a}) \approx \mathbf{z}(\boldsymbol{c}) - \mathbf{z}(\boldsymbol{b})$. We develop a regularizer that uses contrastive loss over tuples of frames to achieve such mappings with CNNs. Whereas slow feature learning insists that the features not change too quickly, the proposed steady learning insists that—in whichever way the features are evolving—they *continue to evolve in that same way* in the immediate future. See Figure 1.

We hypothesize that higher-order temporal coherence could provide a valuable prior for recognition by embedding knowledge of the rich dynamics of the visual world into the feature space. We empirically verify this hypothesis using five datasets for a variety of recognition tasks, including object instance recognition, large-scale scene recognition, and action recognition from still images. In each case, by augmenting a small set of labeled exemplars with unlabeled video, the proposed method generalizes better than both a standard discriminative CNN as well as a CNN regularized with existing slow temporal coherence metrics [14, 30]. Our results reinforce that unsupervised feature learning from unconstrained video is an exciting direction, with promise to offset the large labeled data requirements of current state-of-the-art computer vision approaches by exploiting virtually unlimited unlabeled video.

2. Related Work

To build a robust object recognition system, the image representation must incorporate some degree of *invariance* to changes in pose, illumination, and appearance. While invariance can be manually crafted, such as with spatial pooling operations or gradient descriptors, it may also be learned. One approach often taken in the convolutional neural network (CNN) literature is to pad the training data by systematically perturbing raw images with label-preserving transformations (e.g., translation, scaling, intensity scaling, etc.) [36, 38, 8]. A good representation will ensure that the jittered versions originating from the same content all map close by in the learned feature space.

In a similar spirit, unlabeled video is an appealing resource for recovering invariance. The simple fact that things typically cannot change too quickly from frame to frame makes it possible to harvest sets of sequential images whose learned representations ought not to differ substantially. Slow feature analysis (SFA) [42, 16] leverages this notion to learn features from temporally adjacent video frames.

Recent work uses CNNs to explore the power of learning slow features, also referred to as "temporally coherent" features [30, 3, 46, 12, 41]. The existing methods either produce a holistic image embedding [30, 3, 12, 14], or else track local patches to learn a localized representation [46, 47, 41]. Most methods exploit the learned features for object recognition [30, 46, 3, 41], while others employ them for dimensionality reduction [14] or video frame retrieval [12]. In [30], a standard deep CNN architecture is augmented with a temporal coherence regularizer, then trained using video of objects on clean backgrounds rotating on a turntable. The method of [3] builds on this concept, proposing the use of decorrelation to avoid trivial solutions to the slow feature criterion, with applications to handwritten digit classification. The authors of [12] propose injecting an auto-encoder loss and explore training with unlabeled YouTube video. Building on SFA subspace ideas [42], researchers have also examined slow features for action recognition [45], facial expression analysis [44], future prediction [39], and temporal segmentation [31, 27].

Related to all the above methods, we aim to learn features from unlabeled video. However, whereas all the past work aims to preserve feature *slowness*, our idea is to preserve higher order feature *steadiness*. Our learning objective is the first to move beyond adjacent frame neighborhoods, requiring not only that sequential features change gradually, but also that they change in a similar manner in adjacent time intervals.

Another class of methods learns *transformations* [29, 28, 33]. Whereas the above feature learning methods (and ours) train with unlabeled video spanning various unspecified transformations, these methods instead train with pairs of images for which the transformation is known and/or consistent. Then, given a novel input, the model can be used to predict its transformed output. Rather than use learned transformations for extrapolation like these approaches, our

goal is to exploit transformation patterns in unlabeled video to learn features that are useful for recognition.

Aside from inferring the transformation that implicitly separates a pair of training instances, another possibility is to explicitly predict the transformation parameters. Recent work considers how the camera's ego-motion (e.g., as obtained from inertial sensors, GPS) can be exploited as supervision during CNN training [17, 2]. These methods also lack the higher-order relationships we propose. Furthermore, they require training data annotated with camera/ego-pose parameters, which prevents them from learning with "in the wild" videos (like YouTube) for which the camera was not instrumented with external sensors to record motor changes. In contrast, our method is free to exploit arbitrary unlabeled video data.

Several recent papers [5, 40, 13] have trained unsupervised image representations targeting specific narrow tasks. [5] learn efficient generative codes to synthesize images, while [40] learn features to predict pixel-level optical flow maps for video frames. Contemporary with an earlier version of our work [18], [13] proposed to learn features that vary linearly in time, for the specific task of extrapolating future video frames given a pair of past frames. They report qualitative results for toy video frame synthesis. While our formulation also encourages collinearity in the feature space, our aim is to learn generally useful features from real videos without supervision, and we report results on natural image scene, object, and action recognition tasks.

3. Approach

Given auxiliary raw unlabeled video, we wish to learn an embedding amenable to a supervised classification task. We pose this as a feature learning problem in a convolutional neural network, where the hidden layers of the network are tuned not only with the backpropagation gradients from a classification loss, but also with gradients computed from the unlabeled video that exploit its temporal steadiness.

3.1. Notation and framework overview

A *supervised training dataset* $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$ provides target class labels $\boldsymbol{y}_i \in \mathcal{Y} = [1, 2, .., C]$ for images $\boldsymbol{x}_i \in \mathcal{X}$ (represented in pixel space). The *unsupervised training dataset* $\mathcal{U} = \{\boldsymbol{x}_t\}$ consists of ordered video frames, where $\boldsymbol{x}_t$ is the video frame at time instant t.[1]

Importantly, we do *not* assume that the video $\mathcal{U}$ necessarily stems from the same categories or even the same domain as images in $\mathcal{S}$. For example, in results we will demonstrate cases where $\mathcal{S}$ and $\mathcal{U}$ consist of natural scene images and autonomous vehicle video, respectively; or Web photos of human actions and YouTube video spanning dozens of distinct activities. The idea is that training with diverse unlabeled video should allow the learner to recover fundamental cues about how objects move, how scenes evolve over time, how occlusions occur, how illumination varies, etc., independent of their specific semantic content.

The full image-pixels-to-class label classifier we learn will have the compositional form $\hat{y}_{\boldsymbol{\theta},W} = f_W \circ \mathbf{z}_{\boldsymbol{\theta}}(.)$, where $\mathbf{z}_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{R}^D$ is a D-dimensional feature map operating on images in the pixel space, and $f_W : \mathcal{R}^D \rightarrow \mathcal{Y}$ takes as input the feature map $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$, and outputs the class estimate. We learn a linear classifier f_W represented by a $C \times D$ weight matrix W with rows $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_C$. At test time, a novel image is classified as $\hat{y}_{\boldsymbol{\theta},W} = \arg\max_i \boldsymbol{w}_i^T \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x})$.

To learn the classifier $\hat{y}_{\boldsymbol{\theta},W}$, we optimize an objective function of the form:

$$(\boldsymbol{\theta}^*, W^*) = \arg\min_{\boldsymbol{\theta},W} L_s(\boldsymbol{\theta}, W, \mathcal{S}) + \lambda L_u(\boldsymbol{\theta}, \mathcal{U}), \quad (1)$$

where $L_s(.)$ represents the supervised classification loss, $L_u(.)$ represents an unsupervised regularization loss term, and λ is the regularization hyperparameter. The parameter vector $\boldsymbol{\theta}$ is common to both losses because they are both computed on the learned feature space $\mathbf{z}_{\boldsymbol{\theta}}(.)$. The supervised loss is a softmax loss:

$$L_s(\boldsymbol{\theta}, W, \mathcal{S}) = -\frac{1}{N_s} \sum_{i=1}^{N_s} \log(\sigma_{y_i}(W \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i))), \quad (2)$$

where $\sigma_{y_i}(.)$ is the softmax probability of the correct class and N_s is the number of labeled training instances in $\mathcal{S}$.

In the following, we first discuss how the unsupervised regularization loss $L_u(.)$ may be constructed to exploit temporal smoothness in video (Sec 3.2). Then we generalize this to exploit temporal steadiness and other higher order coherence (Sec 3.3). Sec 3.4 then shows how a neural network corresponding to $\hat{y}_{\boldsymbol{\theta},W}$ may be trained to minimize Eq (1) above.

3.2. Review: First-order temporal coherence

As discussed above, slow feature analysis (SFA) [42] seeks to learn image features that vary slowly over the frames of a video, with the aim of learning useful invariances. This idea of exploiting "slowness" or "temporal coherence" for feature learning has been explored in the context of neural networks [30, 14, 3, 46, 12]. We briefly review that underlying objective before introducing the proposed higher order generalization of temporal coherence.

A temporal neighbor pair dataset $\mathcal{U}_2$ is first constructed from the unlabeled video $\mathcal{U}$, as follows:

$$\mathcal{U}_2 = \{\langle (j,k), p_{jk} \rangle : \boldsymbol{x}_j, \boldsymbol{x}_k \in \mathcal{U} \text{ and } p_{jk} = \mathbb{1}(0 \leq j - k \leq T)\}, \quad (3)$$

[1] For notational simplicity, we will describe our method assuming that the unsupervised training data is drawn from a single continuous video, but it is seamless to train instead with a batch of unlabeled video clips.

where T is the temporal neighborhood size, and the subscript 2 signifies that the set consists of *pairs*. $\mathcal{U}_2$ indexes image pairs with neighbor-or-not binary annotations p_{jk}, automatically extracted from the video. We discuss the setting of T in results. In general, one wants the time window spanned by T to include motions that are small enough to be label-preserving, so that correct invariances are learned; in practice this is typically on the order of a second or less.

With this dataset, the SFA property translates as $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \approx \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_k), \forall p_{jk} = 1$. A simple formulation of this as an unsupervised regularizing loss would be as follows:

$$R'_2(\boldsymbol{\theta}, \mathcal{U}) = \sum_{(j,k)\in\mathcal{N}} d(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)), \tag{4}$$

where $d(.,.)$ is a distance measure (e.g., ℓ_1 in [30] and ℓ_2 in [14]), and $\mathcal{N} \subset \mathcal{U}_2$ denotes the subset of "positive" neighboring frame pairs *i.e.* those for which $p_{jk} = 1$. This loss by itself admits problematic minimizers such as $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in \mathcal{X}$, which corresponds to $R'_2 = 0$. Such solutions may be avoided by a *contrastive* [14] version of the loss function that also exploits "negative" (non-neighbor) pairs:

$$\begin{aligned} R_2(\boldsymbol{\theta}, \mathcal{U}) &= \sum_{(j,k)\in\mathcal{U}_2} D_\delta(\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_j), \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_k), p_{jk}) \\ &= \sum_{(j,k)\in\mathcal{U}_2} p_{jk}\, d(\mathbf{z}_{\boldsymbol{\theta}j}, \mathbf{z}_{\boldsymbol{\theta}k}) + \overline{p_{jk}}\, \max(\delta - d(\mathbf{z}_{\boldsymbol{\theta}j}, \mathbf{z}_{\boldsymbol{\theta}k}), 0), \end{aligned} \tag{5}$$

where $\mathbf{z}_{\boldsymbol{\theta}i}$ denotes $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ and $\overline{p} = 1 - p$. As shown above, the contrastive loss $D_\delta(\boldsymbol{a}, \boldsymbol{b}, p)$ penalizes distance between $\boldsymbol{a}$ and $\boldsymbol{b}$ when the pair are neighbors ($p = 1$), and encourages distance between them when they are not ($p = 0$), up to a margin δ.

3.3. Higher-order temporal coherence

The slow feature formulation of Eq (5) encourages feature maps that produce small first-order temporal derivatives in the learned feature space: $d\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)/dt \approx 0$. This first-order temporal coherence is restricted to learning to ignore small jitters in the visual signal.

Our idea is to model higher order temporal coherence in the unlabeled video, so that the features can further capture rich structure in *how* the visual content changes over time. In the general case, this means we want a regularizer that encourages higher order derivatives to be small: $d^n\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)/dt^n \approx 0, \forall n = 1, 2, ..N$. Accordingly, we need to generalize from pairs of temporally close frames to tuples of frames.

In this work, we focus specifically on learning *steady* features—the second-order case, which can be encoded with triplets of frames, as we will see next. In a nutshell, whereas slow learning insists that the features not change too quickly, steady learning insists that feature *changes* in the immediate future remain similar to those in the recent past.

First, we create a triplet dataset $\mathcal{U}_3$ from the unlabeled video $\mathcal{U}$ as:

$$\begin{aligned} \mathcal{U}_3 = \{\langle (l,m,n), p_{lmn}\rangle : \boldsymbol{x}_l, \boldsymbol{x}_m, \boldsymbol{x}_n \in \mathcal{U} \text{ and} \\ p_{lmn} = \mathbb{1}(0 \le m - l = n - m \le T)\}. \end{aligned} \tag{6}$$

$\mathcal{U}_3$ indexes image triplets with binary annotations indicating whether they are in-sequence, evenly spaced frames in the video, within a temporal neighborhood T. In practice, we select "negatives" ($p_{lmn} = 0$) from triplets where $m - l \le T$ but $n - m \ge 2T$ to provide a buffer and avoid noisy negatives.

We construct our steady feature analysis regularizer using these triplets, as follows:

$$R_3(\boldsymbol{\theta}, \mathcal{U}) = \sum_{(l,m,n)\in\mathcal{U}_3} D_\delta(\mathbf{z}_{\boldsymbol{\theta}l} - \mathbf{z}_{\boldsymbol{\theta}m},\ \mathbf{z}_{\boldsymbol{\theta}m} - \mathbf{z}_{\boldsymbol{\theta}n},\ p_{lmn}), \tag{7}$$

where $\mathbf{z}_{\boldsymbol{\theta}l}$ is again shorthand for $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_l)$ and D_δ refers to the contrastive loss defined above. For positive triplets—meaning those occurring in sequence and within a temporal neighborhood—the above loss penalizes distance between the adjacent pairwise feature *difference* vectors. For negative triplets, it *encourages* this distance, up to a maximum margin distance δ. Effectively, R_3 encourages the feature representations of positive triplets to be collinear *i.e.* $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_l) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_m) \approx \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_m) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$. See Figure 1.

Our final optimization objective combines the first and second order losses (Eq (5) and (7)) into the unsupervised regularization term:

$$L_u(\boldsymbol{\theta}, \mathcal{U}) = R_2(\boldsymbol{\theta}, \mathcal{U}) + \lambda' R_3(\boldsymbol{\theta}, \mathcal{U}), \tag{8}$$

where λ' controls the relative impact of the two terms. Recall this regularizer accompanies the classification loss in the main objective of Eq (1).

Beyond second-order coherence: The proposed framework generalizes naturally to the n-th order, by defining R_n analogously to Eq (7) using a contrastive loss over $(n-1)$-th order discrete derivatives, computed over recursive differences on n-tuples. While in principle higher n would more thoroughly exploit patterns in video, there are potential practical drawbacks. As n grows, the number of samples $|\mathcal{U}_n|$ would likely need to also grow to cover the space of n-frame motion patterns, requiring more training time, compute power, and memory. Besides, discrete n-th derivatives computed over large n-frame time windows may grow less reliable, assuming steadiness degrades over longer temporal windows in typical visual phenomena. Given these considerations, we focus on second-order steadiness combined with slowness, and find that slow and steady does indeed win the race (Sec 4). The empirical question of applying $n > 2$ is left for future work.

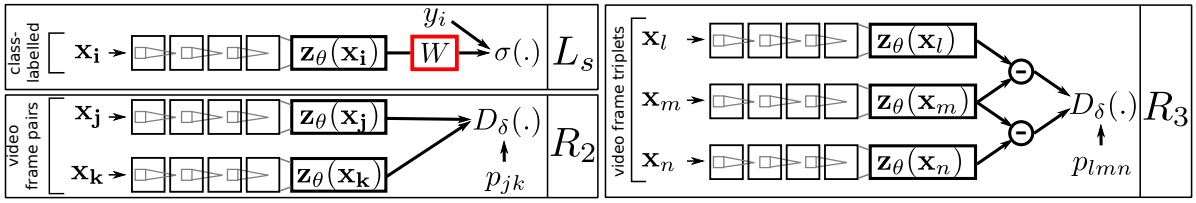

Figure 2: "Siamese" network configuration (shared weights for the $\mathbf{z}_{\boldsymbol{\theta}}$ layer stacks) with portions corresponding to the 3 terms L_s, R_2 and R_3 in our objective. R_2 and R_3 compose the unsupervised loss L_u in Eq (1). L_s is the supervised loss for recognition in static images.

Equivariance-inducing property of $R_3(\boldsymbol{\theta}, \mathcal{U})$: While first-order coherence encourages invariance, the proposed second-order coherence may be seen as encouraging the more general property of *equivariance*. $\mathbf{z}(.)$ is equivariant to an image transformation g if there exists some "simple" function $\mathbf{f}_g : \mathcal{R}^D \rightarrow \mathcal{R}^D$ such that $\mathbf{z}(g\boldsymbol{x}) \approx \mathbf{f}_g(\mathbf{z}(\boldsymbol{x}))$. Equivariance has been found to be useful for visual representations [15, 35, 25, 17]. To see how feature steadiness is related to equivariance, consider a video with frames $\boldsymbol{x}_t, 1 \leq t \leq T$. Given a small temporal neighborhood Δt, frames $\boldsymbol{x}_{t+\Delta t}$ and $\boldsymbol{x}_t$ must be related by a *small* transformation g (small because of *first* order temporal coherence assumption) *i.e.* $\boldsymbol{x}_{t+\Delta t} = g\boldsymbol{x}_t$. Assuming *second* order coherence of video, this transformation g itself remains approximately constant in a small temporal neighborhood, so that, in particular, $\boldsymbol{x}_{t+2\Delta t} \approx g\boldsymbol{x}_{t+\Delta t}$.

Now, for equivariant features $\mathbf{z}(.)$, by the definition of equivariance and the observations above, $\mathbf{z}(\boldsymbol{x}_{t+2\Delta t}) \approx \mathbf{f}_g(\mathbf{z}(\boldsymbol{x}_{t+\Delta t})) \approx \mathbf{f}_g \circ \mathbf{f}_g(\mathbf{z}(\boldsymbol{x}_t))$. Further, given that g is a small transformation, $\mathbf{f}_g$ is well-approximated in a small neighborhood by its first order Taylor approximation, so that: (1) $\mathbf{z}(\boldsymbol{x}_{t+\Delta t}) \approx \mathbf{z}(\boldsymbol{x}_t) + \mathbf{c}(t)$, and (2) $\mathbf{z}(\boldsymbol{x}_{t+2\Delta t}) \approx \mathbf{z}(\boldsymbol{x}_t) + 2\mathbf{c}(t)$. In other words, under the realistic assumption that natural videos evolve smoothly, within small temporal neighborhoods, feature equivariance is equivalent to the second order temporal coherence formulated in Eq (7), with l, m, n set to $t, t+\Delta t, t+2\Delta t$ respectively. This connection between equivariance and the second order temporal coherence induced by R_3 helps motivate why we can expect our feature learning scheme to benefit recognition.

3.4. Neural networks for the feature maps

We use a convolutional neural network (CNN) architecture to represent the feature mapping function $\mathbf{z}_{\boldsymbol{\theta}}(.)$. The parameter vector $\boldsymbol{\theta}$ represents the CNN's learned layer weight matrices. See Sec 4.1 and Supp for architecture choices.

To optimize Eq (1) with the regularizer in Eq (8), we employ standard mini-batch stochastic gradient descent (as implemented in [19]) in a "Siamese" setup, with 6 replicas of the stack $\mathbf{z}_{\boldsymbol{\theta}}(.)$, as shown in Fig 2, 1 stack for L_s (input: supervised training samples $\boldsymbol{x}_i$), 2 for R_2 (input: temporal neighbor pairs $(\boldsymbol{x}_j, \boldsymbol{x}_k)$) and 3 for R_3 (input: triplets $(\boldsymbol{x}_l, \boldsymbol{x}_m, \boldsymbol{x}_n)$). The shared layers are initialized to the same random values and modified by the same gradients (sum of the gradients of the 3 terms) in each training iteration, so they remain identical throughout. See Supp for details.

4. Experiments

We test our approach using five challenging public datasets for three tasks—object, scene, and action recognition—spanning 432 categories. We also analyze its ability to learn higher order temporal coherence with a sequence completion task.

4.1. Experimental setup

Our three recognition tasks (specified by the names of the unsupervised and supervised datasets as $\mathcal{U} \rightarrow \mathcal{S}$) are NORB→NORB object recognition, KITTI→SUN scene recognition and HMDB→PASCAL-10 single-image action recognition. Table 1 (left) summarizes key dataset statistics.

Supervised datasets $\mathcal{S}$: (1) **NORB** [24] has 972 images each of 25 toys against clean backgrounds captured over a grid of camera elevations and azimuths. (2) **SUN** [43] contains Web images of 397 scene categories. (3) **PASCAL-10** [9] is a still-image human action recognition dataset with 10 categories. For all three datasets, we use few labeled training images (see Table 1), since unsupervised regularization schemes should have most impact when labeled data is scarce [17, 30]. This is an important scenario, given the "long tail" of categories lacking ample labeled exemplars.

Unsupervised datasets $\mathcal{U}$: (1) **NORB** consists of pose-registered turntable images (not video), but it is straightforward to generate the pairs and triplets for $\mathcal{U}_2$ and $\mathcal{U}_3$ assuming smooth motions in the annotated pose space. We mine these pairs and triplets from among the 648 images per class that are not used for testing. (2) **KITTI** [10] has videos captured from a car-mounted camera in a variety of locations around the city of Karlsruhe. Scenes are largely static except for traffic, but there is large and systematic camera motion. (3) **HMDB** [23] contains 6849 short Web and movie video clips containing 51 diverse actions. We select 1000 clips at random. While some videos include camera motion (*e.g.* to follow an athlete running), most have stationary cameras and small human pose-change motions. The time window T is a hyperparameter of both our method as well

Task	Img/frame dims	#Classes	Recog. Task	#Train	#Test	Unsup. Input Type	#Pairs (1:3)	#Triplets (1:1)
NORB→NORB	96×96×1	25	object	150	8100	pose-reg. images	50,000	75,000
KITTI→SUN	32×32×1	397	scene	2382	7940	car-mounted video	100,000	100,000
HMDB→PASCAL-10	32×32×3	10	action	50	2000	web video	100,000	100,000

Datasets→	NORB	KITTI	HMDB
SFA-1 [30]	0.95	31.04	2.70
SFA-2 [14]	0.91	8.39	2.27
SSFA (ours)	**0.53**	**7.79**	**1.78**

Table 1: Left: Statistics for the unsupervised and supervised datasets ($\mathcal{U} \rightarrow \mathcal{S}$) used in the recognition tasks (positive to negative ratios for pairs and triplets indicated in headers). **Right:** Sequence completion normalized correct candidate rank η. Lower is better. (See Sec 4.2.)

as existing SFA methods. We fix $T = 2$ and $T = 0.5$ seconds for KITTI and HMDB, respectively, based on cross-validation for best performance by the SFA baselines.

Baselines: We compare our slow-and-steady feature analysis approach (SSFA) to four methods, including two key existing methods for learning from unlabeled video. The three unsupervised baselines are: (1) UNREG: An unregularized network trained only on the supervised training samples $\mathcal{S}$. (2) SFA-1: An SFA approach proposed in [30] that uses ℓ_1 for $d(.)$ in Eq 5. (3) SFA-2: Another SFA variant [14] that sets the distance function $d(.)$ to the ℓ_2 distance in Eq 5. The SFA methods train with the unlabeled pairs, while SSFA trains with both the pairs and triplets.

These comparisons are most crucial to gauge the impact of the proposed approach versus the state of the art for feature learning with unlabeled video. However, we are also interested to what extent learning from unlabeled video can even start to compete with methods learned from heavily labeled data (which costs substantial human effort). Thus, we also compare against a *supervised* pretraining and finetuning approach denoted SUP-FT (details in Sec 4.3).

Network architectures: For the NORB→NORB task, we use a fully connected network architecture: input → 25 hidden units → ReLU nonlinearity → D=25 features. For the other two tasks, we resize images to 32×32 to allow fast and thorough experimentation with standard CNN architectures known to work well with tiny images [1], producing D=64-dimensional features. Recognition tasks on 32×32 images are much harder than with full-sized images, so these are highly challenging tasks. All networks are optimized with Nesterov-accelerated stochastic gradient descent until validation classification loss converges or begins to increase. Optimization hyperparameters are selected greedily through cross-validation in the following order: base learning rate, λ and λ' (starting from λ=λ'=0). The relative scales of the margin parameters δ of the contrastive loss $D_\delta(.)$ in Eq (5) and Eq (7) are validated per dataset. See Supp for more details on the 32×32 architecture, data pre-processing and optimization.

4.2. Quantifying steadiness

First we use a sequence completion task to analyze how well the desired steadiness property is induced in the learned features. We compose a set of sequential triplets from the pool of test images, formed similarly to the positives in Eq (6). At test time, given the first two images of each triplet, the task is to predict what the third looks like.

We apply our SSFA to infer the missing triplet item as follows. Recall that our formulation encourages sequential triplets to be collinear in the feature space. As a result, given $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_1)$ and $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_2)$, we can extrapolate $\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_3)$ as $\tilde{\mathbf{z}}_{\boldsymbol{\theta}}(\boldsymbol{x}_3) = 2\mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_2) - \mathbf{z}_{\boldsymbol{\theta}}(\boldsymbol{x}_1)$. To backproject to the image space, we identify an image closest to $\tilde{\mathbf{z}}_{\boldsymbol{\theta}}(\boldsymbol{x}_3)$ in feature space. Specifically, we take a large pool $\mathcal{C}$ of candidate images, map them all to their features via $\mathbf{z}_{\boldsymbol{\theta}}$, and rank them in increasing order of distance from $\tilde{\mathbf{z}}_{\boldsymbol{\theta}}(\boldsymbol{x}_3)$. The rank r of the correct candidate $\boldsymbol{x}_3$ is now a measure of sequence completion performance. See Supp for details.

Tab 1 (right) reports the mean percentile rank $\eta = \mathbb{E}[r/|\mathcal{C}|] \times 100$ over all query pairs. Lower η is better. Clearly, our SSFA regularization induces steadiness in the feature space, reducing η nearly by half compared to baseline regularizers on NORB and by large margins on HMDB too. Our regularizer R_3 is closely matched to this task, so these gains are expected. Note however that these gains are reported after training to minimize the *joint* objective, which includes L_s and R_2, apart from R_3, and with regularization weights tuned for *recognition* tasks.

Fig 3 shows sequence completion examples from all 3 video datasets. Particularly impressive results are the third NORB example (where despite a difficult viewpoint, the sequence is completed correctly by the top-ranked candidate), and the third HMDB example, where a highly dynamic baseball pitch sequence is correctly completed by the third ranked image. The top-ranked candidate for this example illustrates a common failure mode—the second image of the query pair is itself picked to complete the sequence. This may reflect the fact that HMDB sequences in particular exhibit very little motion (camera motions rare, mostly small object motions). Usually, as in the third KITTI example, even the top-ranked candidates other than the ground truth frame are highly plausible completions.

4.3. Recognition results

Unlabeled video as a prior for supervised recognition: Now we report results on the 3 unsupervised-to-supervised recognition tasks. Table 2 shows the results. Our SSFA method comprehensively outperforms not only the purely supervised UNREG baseline, but also the popular SFA-1 and

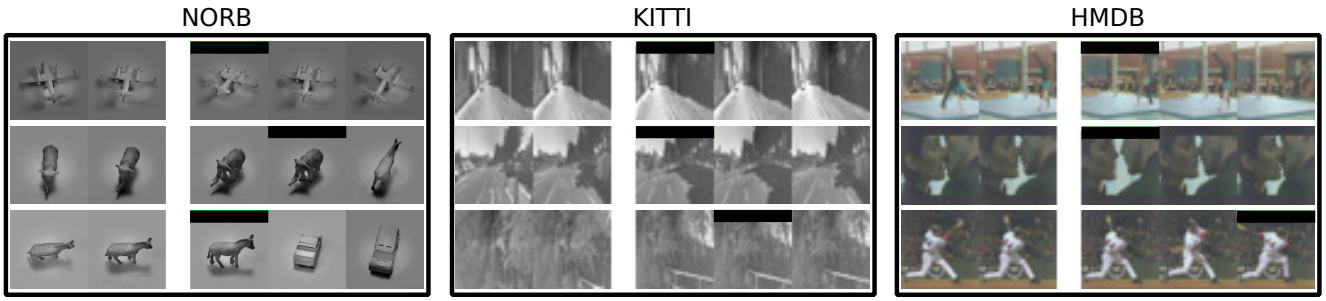

Figure 3: Sequence completion examples from all three video datasets. In each instance, a query pair is presented on the left, and the top three completion candidates as ranked by our method are presented on the right. Ground truth frames are marked with black highlights.

Task type→	Objects	Scenes		Actions
Datasets→	NORB→NORB	KITTI→SUN		HMDB→PASCAL-10
Methods↓	[25 cls]	[397 cls]	[397 cls, top-10]	[10 cls]
random	4.00	0.25	2.52	10.00
UNREG	24.64±0.85	0.70±0.12	6.10±0.67	15.34±0.28
SFA-1 [30]	37.57±0.85	1.21±0.14	8.24±0.25	19.26±0.45
SFA-2 [14]	39.23±0.94	1.02±0.12	6.78±0.32	19.04±0.24
SSFA (ours)	**42.83±0.33**	**1.65±0.04**	**9.19±0.10**	**20.95±0.13**

Table 2: Recognition results (mean ± standard error of accuracy % over 5 repetitions) (Sec 4.3). Our method outperforms both existing slow feature/temporal coherence methods and the unregularized baseline substantially, across three distinct recognition tasks.

SFA-2 slow feature learning approaches, beating the best baseline for each task by 9%, 36% and 9% respectively. The results on KITTI→SUN and HMDB→PASCAL-10 are particularly impressive because the unsupervised and supervised dataset domains are mismatched. All KITTI data comes from a single car-mounted road-facing camera driving through the streets of one city, whereas SUN images are downloaded from the Web, captured by different cameras from diverse viewpoints, and cover 397 scene categories mostly unrelated to roads. PASCAL-10 images are bounding-box-cropped and therefore centered on single persons, while HMDB videos, which are mainly clips from movies and Web videos, often feature multiple people, are not as tightly focused on the person performing the action, and are of low quality, sometimes with overlaid text *etc*.

Aside from the diversity of tasks (object, scene, and action recognition), our unsupervised datasets also exhibit diverse types of motion. NORB is generated from planned, discrete camera manipulations around a central object of interest. The KITTI camera moves through a real largely static landscape in smooth motions on roads at varying speeds. HMDB videos on the other hand are usually captured from stationary cameras with a mix of large and small foreground and background object motions. Even the dynamic camera videos in HMDB are sometimes captured from hand-held devices leading to jerky motions, where our temporal steadiness assumptions might be stressed.

Pairing unsupervised and supervised datasets: Thus far, our pairings of unsupervised and supervised datasets reflect our attempt to learn from video that *a priori* seems related to the ultimate recognition task, *e.g.* HMDB human action videos are paired with PASCAL-10 Action still images. However, as discussed above, the domains are only roughly aligned. Curious about the impact of the choice of unlabeled video data, we next try swapping out HMDB for KITTI in the PASCAL action recognition task. On this new KITTI→PASCAL task, we still easily outperform our nearest baseline, although our gain drops by ≈ 0.9% (SFA-2:19.06% vs. our SSFA:20.01%). Despite the fact that the human motion dynamics of HMDB ostensibly match the action recognition task better than the egomotion dynamics of KITTI (where barely any people are visible), we maintain our advantage over the purely slow methods. This indicates that there is reasonable flexibility in the choice of unlabeled videos fed to SSFA.

Increasing supervised training sets: Thus far, we have kept labeled sets small to simulate the "long tail" of categories with scarce training samples where priors like ours and the baselines' have most impact. In a preliminary study for larger training pools, we now increase SUN training set sizes from 6 to 20 samples per class for KITTI→SUN. Our method retains a 20% gain over existing slow methods (SSFA: 3.24% vs SFA-2: 2.65%). This suggests our approach is valuable even with larger supervised training sets.

Varying unsupervised training set size: To observe the effect of unsupervised training set size, we now restrict SSFA to use varying-sized subsets of unlabeled video on the HMDB→PASCAL-10 task. Performance scales roughly log-linearly with the duration of video observed,[2] suggesting that even larger gains may be achieved simply by training SSFA with more freely available unlabeled video.

Purely unsupervised feature learning: We now evaluate the usefulness of features trained to optimize the unsupervised SSFA loss L_u (Eq (8)) alone. Features trained on HMDB are evaluated at various stages of training, on

[2] At 3, 12.5, 25, and 100% resply. of the full unlabeled dataset (≈32k frames), performance is 18.06, 19.74, 20.36, and 20.95% (see Supp)

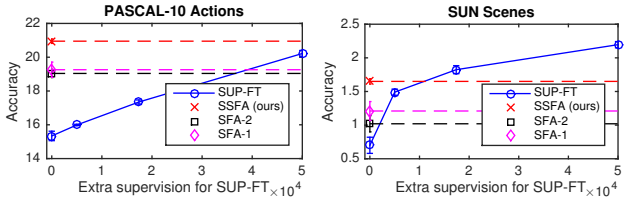

Figure 4: Comparison to CIFAR-100 supervised pretraining SUP-FT, at various supervised training set sizes. Flat dashed lines reflect that our method (and SFA) always use zero additional labels.

the task of k-nearest neighbor classification on PASCAL-10 (k =5, and 100 training images per action). Starting at $\approx$ 17.8% classification accuracy for randomly initialized networks, unsupervised SSFA training steadily improves the discriminative ability of features to 19.62, 20.32 and 22.14% after 1, 2 and 3 passes respectively over training data (see Supp). This shows that SSFA can train useful image representations even without jointly optimizing a supervised objective.

Comparison to supervised pretraining and finetuning: Recently, a two-stage supervised pretraining and finetuning strategy (SUP-FT) has emerged as the leading approach to solve visual recognition problems with limited training data where high-capacity models like deep neural networks may not be directly learned [11, 7, 32, 20]. In the first stage ("supervised pretraining"), a neural network "NET1" is first trained on a related problem for which large training datasets *are* available. In a second stage ("finetuning"), the weights from NET1 are used to initialize a second network ("NET2") with similar architecture. NET2 is then trained on the target task, using reduced learning rates to minimally modify the features learned in NET1.

In principle, completely unsupervised feature learning approaches like ours have important advantages over the SUP-FT paradigm. In particular, (1) they can leverage essentially infinite unlabeled data without requiring expensive human labeling effort thus potentially allowing the learning of higher capacity models and (2) they do not require the existence of large *"related"* supervised datasets from which features may be meaningfully transferred to the target task. While the pursuit of these advantages continues to drive vigorous research, unsupervised feature learning methods still underperform supervised pretraining for image classification tasks, where great effort has gone into curating large labeled databases, e.g., ImageNet [6], CIFAR [21].

As a final experiment, we examine how the proposed unsupervised feature learning idea competes with the popular supervised pretraining model. To this end, we adopt the CIFAR-100 dataset consisting of 100 diverse object categories as a basis for supervised pretraining.[3] The new baseline SUP-FT trains NET1 on CIFAR (see Supp), then finetunes NET2 for either PASCAL-10 action or SUN scene recognition tasks using the exact same (few) labeled instances given to our method. In parallel, our method "pretrains" only via the SSFA regularizer learned with unlabeled HMDB / KITTI video respectively for the two tasks. Our method uses *zero* labeled CIFAR data.

Fig 4 shows the results. On PASCAL-10 action recognition (left), our method significantly outperforms SUP-FT pretrained with all 50,000 images of CIFAR-100! Gathering image labels from the crowd for large multi-way problems can take on average 1 minute per image [34], meaning we are getting better results while also saving $\sim$ 830 hours of human effort. On SUN scene recognition (right), SSFA outperforms SUP-FT with 5K labels and remains competitive even when the supervised method has a 17,500 label advantage. However, SUP-FT-50K's advantage on the SUN task is more noticeable; its gain is similar to our gain over the best slow-feature method.

The upward trend in accuracy for SUP-FT with more CIFAR-100 labeled data indicates that it successfully transfers generic recognition cues to the new tasks. On the other hand, the fact that it fares worse on PASCAL actions than SUN scenes reinforces that *supervised* transfer depends on having large curated datasets in a *strongly related* domain. In contrast, our approach successfully "transfers" what it learns from purely unlabeled video. In short, our method can achieve better results with substantially less supervision. More generally, we view it as an exciting step towards unlabeled video bridging the gap between unsupervised and supervised pretraining for visual recognition.

5. Conclusion

We formulated an unsupervised feature learning approach that exploits higher order temporal coherence in unlabeled video, and demonstrated its powerful impact for several recognition tasks. Despite over 15 years of research surrounding slow feature analysis (SFA), its variants and applications, to the best of our knowledge, we are the first to identify that SFA is only the first order approximation of a more general temporal coherence idea. This basic observation leads to our intuitive approach that can be easily plugged into applications where first order temporal coherence has already been found useful [30, 3, 46, 12, 41, 14, 45, 44, 31, 27]. To our knowledge, ours are the first results where unsupervised learning from video actually surpasses the accuracy of today's favored approach, heavily supervised pretraining.

Acknowledgements: We thank Texas Advanced Computing Center for their generous support. This work was supported in part by ONR YIP N00014-15-1-2291.

[3] We choose CIFAR-100 for its compatibility with the 32 $\times$ 32 images used throughout our results, which let us leverage standard CNN architectures known to work well with tiny images [1].

References

[1] Cuda-convnet. https://code.google.com/p/cuda-convnet/. 6, 8

[2] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. *ICCV*, 2015. 3

[3] J. Bergstra and Y. Bengio. Slow, decorrelated features for pretraining complex cell-like networks. In *NIPS*, 2009. 2, 3, 8

[4] P. Berkes and L. Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of vision*, 5(6), 2005. 1

[5] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *ICLR*, 2016. 3

[6] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 8

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 8

[8] A. Dosovitskiy, J.T. Springenberg, M. Riedmiller, and T. Brox. Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. *NIPS*, 2014. 2

[9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 5

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. *CVPR*, 2012. 5

[11] R. Girshick, J. Donahue, T. Darrell, and J.Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 8

[12] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised Learning of Spatiotemporally Coherent Metrics. *ICCV*, 2014. 2, 3, 8

[13] Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to linearize under uncertainty. In *NIPS*, 2015. 3

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. *CVPR*, 2006. 2, 3, 4, 6, 7, 8

[15] G. Hinton, A. Krizhevsky, and S.D. Wang. Transforming Auto-Encoders. *ICANN*, 2011. 5

[16] J. Hurri and A. Hyvarinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3), 2003. 2

[17] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. *ICCV*, 2015. 3, 5

[18] D. Jayaraman and K. Grauman. Slow and Steady Feature Analysis: Higher Order Temporal Coherence in Video. *CoRR*, abs/1506.04714, June 2015. 3

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, Sergio S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv*, 2014. 5

[20] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. *BMVC*, 2014. 8

[21] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. 8

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. *ICCV*, 2011. 5

[24] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. *CVPR*, 2004. 5

[25] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *CVPR*, 2015. 5

[26] N. Li and J. DiCarlo. Unsupervised natural experience rapidly alters invariant object representation in visual cortex. *Science*, 321, 2008. 1

[27] S. Liwicki, S. Zafeiriou, and M. Pantic. Incremental slow feature analysis with indefinite kernel for online temporal video segmentation. In *ACCV*, 2012. 2, 8

[28] R. Memisevic. Learning to relate images. *PAMI*, 2013. 2

[29] V. Michalski, R. Memisevic, and K. Konda. Modeling Deep Temporal Dependencies with Recurrent Grammar Cells. *NIPS*, 2014. 2

[30] H. Mobahi, R. Collobert, and J. Weston. Deep Learning from Temporal Coherence in Video. *ICML*, 2009. 2, 3, 4, 5, 6, 7, 8

[31] F. Nater, H. Grabner, and L. Van Gool. Temporal relations in videos for unsupervised activity analysis. In *BMVC*, 2011. 2, 8

[32] M. Oquab, L. Bottou, I. Laptev, and J.Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *CVPR*, 2014. 8

[33] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv*, 2014. 2

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 8

[35] U. Schmidt and S. Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. *CVPR*, 2012. 5

[36] P. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *ICDAR*, 2003. 2

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 1

[38] P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. *ICML*, 2008. 2

[39] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, abs/1504.08023, 2015. 2

[40] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015. 3

[41] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *ICCV*, 2015. 2, 8

[42] L. Wiskott and T. J. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural computation*, 2002. 1, 2, 3

[43] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. *CVPR*, 2010. 5

[44] L. Zafeiriou, M. Nicolaou, S. Zafeiriou, S. Nikitidis, and M. Pantic. Learning slow features for behaviour analysis. In *ICCV*, 2013. 2, 8

[45] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *PAMI*, 2012. 2, 8

[46] W. Zou, S. Zhu, K. Yu, and A. Ng. Deep learning of invariant features via simulated fixations in video. *NIPS*, 2012. 2, 3, 8

[47] Will Y Zou, Andrew Y Ng, and Kai Yu. Unsupervised learning of visual invariance with temporal coherence. In *NIPS Workshop on Deep Learning*, 2011. 2